



系统管理指南：设备和文件系统



Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

文件号码 819-7062-10
2006 年 9 月

版权所有 2006 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. 保留所有权利。

对于本档中介绍的产品，Sun Microsystems, Inc. 对其所涉及的技术拥有相关的知识产权。需特别指出的是（但不局限于此），这些知识产权可能包含一项或多项美国专利，或在美国和其他国家/地区申请的待批专利。

美国政府权利—商业软件。政府用户应遵循 Sun Microsystems, Inc. 的标准许可协议，以及 FAR（Federal Acquisition Regulations，即“联邦政府采购法规”）的适用条款及其补充条款。

本发行版可能包含由第三方开发的内容。

本产品的某些部分可能是从 Berkeley BSD 系统衍生出来的，并获得了加利福尼亚大学的许可。UNIX 是 X/Open Company, Ltd. 在美国和其他国家/地区独家许可的注册商标。

Sun、Sun Microsystems、Sun 徽标、Solaris 徽标、Java 咖啡杯徽标、docs.sun.com、Java 和 Solaris 是 Sun Microsystems, Inc. 在美国和其他国家/地区的商标或注册商标。所有 SPARC 商标的使用均已获得许可，它们是 SPARC International, Inc. 在美国和其他国家/地区的商标或注册商标。标有 SPARC 商标的产品均基于由 Sun Microsystems, Inc. 开发的体系结构。FireWire 和 FireWire 徽标是 Apple Computer, Inc. 的商标，其使用已获得许可。X/Open 是 X/Open Company, Ltd. 的注册商标。DLT 已申请为 Quantum Corporation 在美国和其他国家/地区的商标。

OPEN LOOK 和 SunTM 图形用户界面是 Sun Microsystems, Inc. 为其用户和许可证持有者开发的。Sun 感谢 Xerox 在研究和开发可视或图形用户界面的概念方面为计算机行业所做的开拓性贡献。Sun 已从 Xerox 获得了对 Xerox 图形用户界面的非独占性许可证，该许可证还适用于实现 OPEN LOOK GUI 和在其他方面遵守 Sun 书面许可协议的 Sun 许可证持有者。

本出版物所介绍的产品以及所包含的信息受美国出口控制法制约，并应遵守其他国家/地区的进出口法律。严禁将本产品直接或间接地用于核设施、导弹、生化武器或海上核设施，也不能直接或间接地出口给核设施、导弹、生化武器或海上核设施的最终用户。严禁出口或转口到美国禁运的国家/地区以及美国禁止出口清单中所包含的实体，包括但不限于被禁止的个人以及特别指定的国家/地区的公民。

本文档按“原样”提供，对于所有明示或默示的条件、陈述和担保，包括对适销性、适用性或非侵权性的默示保证，均不承担任何责任，除非此免责声明的适用范围在法律上无效。

目录

前言	21
1 管理可移除介质（概述）	25
Solaris 10 1/06 发行版中可移除介质方面的新增功能	25
vold 由服务管理工具 (Service Management Facility, SMF) 进行管理	25
对卷管理 (vold) 的改进	27
Solaris 10 发行版中可移除介质方面的新增功能	27
DVD+RW 和 DVD-RW 支持	27
新增的 cdrw 选项	28
列出可移除介质设备	28
播放音乐 CD	29
有关管理可移除介质任务的参考信息	29
可移除介质的功能和优点	29
手动挂载和自动挂载的比较	30
使用卷管理可以执行的操作	30
2 管理可移除介质（任务）	33
管理可移除介质（任务列表）	33
格式化软盘	34
可移除介质硬件注意事项	34
▼ 如何装入可移除介质	35
▼ 如何格式化软盘 (rmformat)	36
▼ 如何在可移除介质上创建文件系统	37
▼ 如何检查可移除介质上的文件系统	39
▼ 如何修复可移除介质上的坏块	40
将读/写保护和口令保护应用于可移除介质	40
▼ 如何在可移除介质上启用或禁用写保护	41

▼ 如何在可移除介质上启用或禁用读/写保护以及设置口令	41
3 访问可移除介质（任务）	43
访问可移除介质（任务列表）	43
访问可移除介质	44
使用可移除介质名称	44
访问可移除介质数据的指导原则	44
▼ 如何添加新的可移除介质驱动器	45
▼ 如何停止和启动卷管理 (vold)	45
▼ 如何访问可移除介质上的信息	46
▼ 如何从可移除介质复制信息	47
▼ 如何确定可移除介质是否仍在使用的	48
▼ 如何弹出可移除介质	49
在远程系统上访问可移除介质（任务列表）	49
▼ 如何使本地介质可供其他系统使用	49
▼ 如何访问远程系统上的可移除介质	54
4 写入 CD 和 DVD（任务）	57
使用音频 CD 以及数据 CD 和 DVD	57
CD/DVD 介质常用术语	58
写入数据 CD 和 DVD 及音频 CD	59
通过 RBAC 限制用户对可移除介质的访问	59
▼ 如何通过 RBAC 限制用户对可移除介质的访问	60
▼ 如何标识 CD 或 DVD 写入者	60
▼ 如何检查 CD 或 DVD 介质	61
创建数据 CD 或 DVD	62
▼ 如何为数据 CD 或 DVD 创建 ISO 9660 文件系统	62
▼ 如何创建多会话数据 CD	63
创建音频 CD	66
▼ 如何创建音频 CD	66
▼ 如何提取 CD 上的声道	67
▼ 如何复制 CD	68
▼ 如何删除 CD-RW 介质	69

5 管理设备（概述/任务）	71
Solaris 10 6/06 发行版在设备管理方面的新增功能	71
x86: 支持 PCI Express (PCIe)	71
USB 和 1394 (FireWire) 支持增强功能	72
Solaris 10 发行版在设备管理方面的新增功能	73
USB 设备增强功能	73
x86 系统的 1394 (FireWire) 和海量存储支持	73
设备文件系统 (devfs)	73
光纤通道设备的电源管理	74
有关设备管理任务的参考信息	74
关于设备驱动程序	75
设备的自动配置	75
自动配置的功能和优点	76
不受支持的设备所需的内容	76
显示设备配置信息	77
driver not attached 消息	77
▼ 如何显示系统配置信息	77
向系统添加外围设备	84
▼ 如何添加外围设备	84
▼ 如何添加设备驱动程序	85
 6 动态配置设备（任务）	87
动态重新配置和热插拔	87
附加点	88
拆离 PCI 或 PCIe 适配卡	90
连接 PCI 或 PCIe 适配卡	90
使用 cfgadm 命令执行 SCSI 热插拔（任务列表）	90
使用 cfgadm 命令执行 SCSI 热插拔	91
▼ 如何显示有关 SCSI 设备的信息	91
▼ 如何取消配置 SCSI 控制器	92
▼ 如何配置 SCSI 控制器	93
▼ 如何配置 SCSI 设备	94
▼ 如何断开 SCSI 控制器连接	95
▼ SPARC: 如何连接 SCSI 控制器	96
▼ SPARC: 如何将 SCSI 设备添加到 SCSI 总线中	97

▼ SPARC: 如何更换 SCSI 控制器上的相同设备	98
▼ SPARC: 如何移除 SCSI 设备	100
对 SCSI 配置问题进行疑难解答	101
▼ 如何解析失败的 SCSI 取消配置操作	103
使用 <code>cfgadm</code> 命令执行 PCI 或 PCIe 热插拔 (任务列表)	103
使用 <code>cfgadm</code> 命令执行 PCI 或 PCIe 热插拔	104
PCIe LED 指示灯行为	104
▼ 如何显示 PCI 插槽配置信息	104
▼ 如何移除 PCI 适配卡	106
▼ 如何添加 PCI 适配卡	108
对 PCI 配置问题进行疑难解答	110
重新配置调整管理器 (Reconfiguration Coordination Manager, RCM) 脚本概述	111
什么是 RCM 脚本?	111
RCM 脚本的功能	112
RCM 脚本流程的工作原理	112
RCM 脚本任务	113
应用程序开发者 RCM 脚本 (任务列表)	113
系统管理员 RCM 脚本 (任务列表)	113
命名 RCM 脚本	114
安装或删除 RCM 脚本	114
▼ 如何安装 RCM 脚本	115
▼ 如何删除 RCM 脚本	115
▼ 如何测试 RCM 脚本	115
磁带备份 RCM 脚本示例	116
7 使用 USB 设备 (概述)	123
Solaris 10 6/06 发行版在 USB 设备方面的新增功能	123
Solaris 10 1/06 发行版在 USB 设备方面的新增功能	124
x86: 在基于 GRUB 的引导中支持 USB CD 和 DVD	124
USB 虚拟键盘和鼠标支持	124
<code>vold</code> 用于识别热插拔 USB 设备	125
Solaris 10 发行版在 USB 设备方面的新增功能	125
Solaris 对 USB 设备的支持	125
USB 设备概述	126
常用的 USB 首字母缩略词	127

USB 总线说明	127
关于 Solaris OS 中的 USB	131
USB 2.0 功能	131
USB 键盘和鼠标设备	134
USB 主机控制器和集线器	134
SPARC: USB 电源管理	135
USB 电缆连接原则	135
8 使用 USB 设备（任务）	137
在 Solaris OS 中管理 USB 设备（指南）	137
使用 USB 海量存储设备（任务列表）	138
使用 USB 海量存储设备	139
使用 USB 软盘设备	140
使用不兼容的 USB 海量存储设备	141
▼ 如何使用不兼容的 USB 海量存储设备	141
热插拔 USB 海量存储设备	142
▼ 如何在不运行 <code>vold</code> 的情况下添加 USB 海量存储设备	143
▼ 如何添加 USB 相机	143
▼ 如何在不运行 <code>vold</code> 的情况下移除 USB 海量存储设备	145
在运行 <code>vold</code> 的情况下准备使用 USB 海量存储设备	145
▼ 如何在不运行 <code>vold</code> 的情况下准备使用 USB 海量存储设备	146
▼ 如何显示 USB 设备信息	147
▼ 如何在不运行 <code>vold</code> 的情况下在 USB 海量存储设备中创建文件系统	148
▼ 如何在不运行 <code>vold</code> 的情况下在 USB 海量存储设备中修改分区和创建 PCFS 文件系 统	151
▼ 如何在不运行 <code>vold</code> 的情况下在 USB 海量存储设备中创建 Solaris 分区和修改片 ...	157
▼ 如何在不运行 <code>vold</code> 的情况下挂载或取消挂载 USB 海量存储设备	160
USB 海量存储设备的疑难解答提示	161
禁用特定的 USB 驱动程序	162
▼ 如何禁用特定的 USB 驱动程序	162
▼ 如何删除未使用的 USB 设备链路	162
使用 USB 音频设备（任务列表）	163
使用 USB 音频设备	163
热插拔多个 USB 音频设备	164
▼ 如何添加 USB 音频设备	165

▼ 如何确定系统的主音频设备	165
▼ 如何更改主 USB 音频设备	167
USB 音频设备问题疑难解答	167
使用 <code>cfgadm</code> 命令热插拔 USB 设备（任务列表）	167
使用 <code>cfgadm</code> 命令热插拔 USB 设备	168
▼ 如何显示 USB 总线信息 (<code>cfgadm</code>)	169
▼ 如何取消配置 USB 设备	170
▼ 如何配置 USB 设备	171
▼ 如何以逻辑方式断开 USB 设备连接	171
▼ 如何以逻辑方式连接 USB 设备	172
▼ 如何以逻辑方式断开 USB 设备子树连接	172
▼ 如何重置 USB 设备	173
▼ 如何更改多重配置 USB 设备的缺省配置	173
9 使用 InfiniBand 设备（概述/任务）	175
InfiniBand 设备概述	175
动态重新配置 IB 设备（任务列表）	176
动态重新配置 IB 设备 (<code>cfgadm</code>)	178
▼ 如何显示 IB 设备信息	178
▼ 如何取消配置 IOC 设备	181
▼ 如何配置 IOC 设备	181
▼ 如何取消配置 IB 端口、HCA_SVC 或 VPPA 设备	182
▼ 如何配置 IB 端口、HCA_SVC 或 VPPA 设备	182
▼ 如何取消配置 IB 伪设备	183
▼ 如何配置 IB 伪设备	183
▼ 如何显示 HCA 的内核 IB 客户机	184
▼ 如何取消配置与 HCA 连接的 IB 设备	185
配置 IB HCA	186
▼ 如何更新 IB <code>p_key</code> 表	186
▼ 如何显示 IB 通信服务	186
▼ 如何添加 VPPA 通信服务	187
▼ 如何删除现有的 IB 端口、HCA_SVC 或 VPPA 通信服务	188
▼ 如何更新 IOC 配置	188
将 <code>uDAPL</code> 应用程序接口用于 InfiniBand 设备	189
▼ 如何启用 <code>uDAPL</code>	190

更新 DAT 静态注册表	190
▼ 如何更新 DAT 静态注册表	191
▼ 如何在 DAT 静态注册表中注册	191
▼ 如何从 DAT 静态注册表中注销	191
10 访问设备（概述）	193
访问设备	193
如何创建设备信息	193
如何管理设备	193
设备命名约定	194
逻辑磁盘设备名称	194
指定磁盘子目录	195
直接控制器和面向总线的控制器	196
x86: 带有直接控制器的磁盘	196
带有面向总线的控制器的磁盘	196
逻辑磁带设备名称	197
逻辑可移除介质设备名称	197
11 管理磁盘（概述）	199
Solaris 10 6/06 发行版在磁盘管理方面新增功能	199
iSCSI 启动器支持增强功能	199
Solaris 10 1/06 发行版在磁盘管理方面新增功能	200
x86: GRUB 引导环境中的磁盘管理	200
Solaris iSCSI 启动器支持	201
支持容量超过 2 TB 的 SCSI 磁盘	201
Solaris 10 发行版在磁盘管理方面新增功能	201
带有 EFI 磁盘标号的多 TB 磁盘支持	202
用于 SPARC 和 x86 系统的通用 SCSI 驱动程序	206
新的 fdisk 分区标识符	206
有关磁盘管理任务的参考信息	206
磁盘管理概述	207
磁盘术语	207
关于磁盘片	207
format 实用程序	211
关于磁盘标号	214

分区表术语	214
显示分区表信息	214
对磁盘分区	217
使用浮动片 (free hog slice)	217
12 管理磁盘 (任务)	219
管理磁盘 (任务列表)	219
确定系统中的磁盘	220
▼ 如何确定系统中的磁盘	220
格式化磁盘	222
▼ 如何确定磁盘是否已格式化	223
▼ 如何格式化磁盘	224
显示磁盘片	227
▼ 如何显示磁盘片信息	227
创建和检查磁盘标号	229
▼ 如何标记磁盘	230
▼ 如何检查磁盘标号	233
恢复损坏的磁盘标号	235
▼ 如何恢复损坏的磁盘标号	235
添加第三方磁盘	238
创建 format.dat 项	238
▼ 如何创建 format.dat 项	239
自动配置 SCSI 磁盘驱动器	239
▼ 如何自动配置 SCSI 驱动器	240
修复有缺陷的扇区	241
▼ 如何使用表面分析确定有缺陷的扇区	242
▼ 如何修复有缺陷的扇区	243
管理磁盘的提示和技巧	244
调试 format 会话	244
使用 prtvtoc 和 fmthard 命令标记多个磁盘	246
13 SPARC : 添加磁盘 (任务)	247
SPARC: 添加系统磁盘或辅助磁盘 (任务列表)	247
SPARC: 添加系统磁盘或辅助磁盘	248
▼ SPARC: 如何连接系统磁盘并进行引导	248

▼ SPARC: 如何连接辅助磁盘并进行引导	249
▼ SPARC: 如何创建磁盘片和标记磁盘	250
▼ SPARC: 如何创建 UFS 文件系统	258
▼ SPARC: 如何在系统磁盘上安装引导块	259
14 x86 : 添加磁盘 (任务)	261
x86: 添加系统磁盘或辅助磁盘 (任务列表)	261
x86: 添加系统磁盘或辅助磁盘	262
▼ x86: 如何连接系统磁盘	262
▼ 如何更改 Solaris fdisk 标识符	263
▼ x86: 如何连接辅助磁盘并进行引导	265
x86: fdisk 分区创建准则	266
▼ x86: 如何创建 Solaris fdisk 分区	267
▼ x86: 如何创建磁盘片和标记磁盘	276
▼ x86: 如何创建文件系统	277
▼ x86: 如何在系统磁盘上安装引导块	278
15 配置 Solaris iSCSI 启动器 (任务)	281
iSCSI 技术 (概述)	281
iSCSI 软件和硬件要求	282
设置 Solaris iSCSI 启动器 (任务列表)	283
配置 Solaris iSCSI 启动器	283
▼ 如何准备 Solaris iSCSI 配置	285
在基于 iSCSI 的存储网络中配置验证	285
▼ 如何配置对 iSCSI 配置的 CHAP 验证	285
使用第三方 Radius 服务器简化 iSCSI 配置中的 CHAP 管理	287
▼ 如何为 iSCSI 配置配置 RADIUS	287
▼ 如何配置 iSCSI 目标搜索	288
▼ 如何删除搜索到的 iSCSI 目标	289
访问 iSCSI 磁盘	290
▼ 监视 iSCSI 配置	291
修改 iSCSI 启动器和目标参数	294
▼ 如何修改 iSCSI 启动器和目标参数	295
▼ 如何为目标启用多个 iSCSI 会话	299
iSCSI 配置问题的疑难解答	301

从本地系统到 iSCSI 目标无连接	302
▼ 如何解决 iSCSI 连接问题	302
iSCSI 设备或磁盘在本地系统上不可用	303
▼ 如何解决 iSCSI 设备或磁盘的不可用性问题	303
在使用 iSNS 搜索方法时使用 LUN 屏蔽	304
一般 iSCSI 错误消息	304
16 format 实用程序 (参考)	309
有关使用 format 实用程序的建议和要求	309
format 菜单和命令说明	310
partition 菜单	312
x86: fdisk 菜单	313
analyze 菜单	314
defect 菜单	316
format.dat 文件	317
format.dat 文件的内容	318
format.dat 文件的语法	318
format.dat 文件中的关键字	318
分区表 (format.dat)	321
为 format 实用程序指定备用数据文件	322
format 命令的输入规则	322
为 format 命令指定数字	322
为 format 命令指定块编号	323
指定 format 命令名称	323
为 format 命令指定磁盘名称	324
获取有关 format 实用程序的帮助	325
17 管理文件系统 (概述)	327
Solaris 10 6/06 发行版中文件系统方面的新增功能	327
ZFS 文件系统	327
UFS 文件系统实用程序 (fsck、mkfs 和 newfs) 的增强功能	328
Solaris 10 发行版中文件系统方面的新增功能	338
缺省情况下启用 UFS 日志记录	339
NFS 版本 4	339
64 位: 对多 TB UFS 文件系统的支持	340

libc_hwcap	343
有关文件系统管理任务的参考信息	343
文件系统概述	343
文件系统的类型	344
用于文件系统管理的命令	349
文件系统命令如何确定文件系统的类型	349
通用和特定的文件系统命令的手册页	350
缺省的 Solaris 文件系统	350
UFS 文件系统	351
规划 UFS 文件系统	352
UFS 日志记录	352
UFS 快照	353
UFS 直接输入/输出 (Input/Output, I/O)	353
挂载和取消挂载文件系统	353
已挂载文件系统表	355
虚拟文件系统表	356
NFS 环境	357
自动挂载或 AutoFS	358
确定文件系统的类型	358
如何确定文件系统的类型	359
18 创建 UFS、TMPFS 和 LOFS 文件系统（任务）	361
创建 UFS 文件系统	361
▼ 如何创建 UFS 文件系统	362
▼ 如何创建多 TB UFS 文件系统	363
▼ 如何扩展多 TB UFS 文件系统	365
▼ 如何将 UFS 文件系统扩展为多 TB UFS 文件系统	367
对多 TB UFS 文件系统问题进行疑难解答	369
创建临时文件系统 (Temporary File System, TMPFS)	370
▼ 如何创建和挂载 TMPFS 文件系统	370
创建和挂载回送文件系统 (Loopback File System, LOFS)	371
▼ 如何创建和挂载 LOFS 文件系统	371
19 挂载和取消挂载文件系统（任务）	373
挂载文件系统概述	373

用于挂载和取消挂载文件系统的命令	374
常用挂载选项	375
/etc/vfstab 文件的字段说明	376
挂载文件系统	377
如何确定挂载的文件系统	377
▼ 如何向 /etc/vfstab 文件添加项	378
▼ 如何挂载文件系统 (/etc/vfstab 文件)	380
▼ 如何挂载 UFS 文件系统 (mount 命令)	382
▼ 如何挂载不包含大文件的 UFS 文件系统 (mount 命令)	382
▼ 如何挂载 NFS 文件系统 (mount 命令)	384
▼ x86: 如何从硬盘挂载 PCFS (DOS) 文件系统 (mount 命令)	384
取消挂载文件系统	385
取消挂载文件系统的先决条件	386
如何验证文件系统是否已取消挂载	386
▼ 如何停止访问文件系统的所有进程	386
▼ 如何取消挂载文件系统	387
20 使用 CacheFS 文件系统 (任务)	389
使用 CacheFS 文件系统的高级视图 (任务列表)	389
CacheFS 文件系统概述	390
CacheFS 文件系统的工作原理	390
CacheFS 文件系统的结构和行为	391
创建并挂载 CacheFS 文件系统 (任务列表)	392
▼ 如何创建高速缓存	393
在高速缓存中挂载文件系统	393
▼ 如何挂载 CacheFS 文件系统 (mount)	394
▼ 如何挂载 CacheFS 文件系统 (/etc/vfstab)	396
▼ 如何挂载 CacheFS 文件系统 (AutoFS)	397
维护 CacheFS 文件系统 (任务列表)	397
维护 CacheFS 文件系统	398
修改 CacheFS 文件系统	398
▼ 如何显示有关 CacheFS 文件系统的信息	399
CacheFS 文件系统的一致性检查	400
▼ 如何根据需要指定高速缓存一致性检查	401
▼ 如何删除 CacheFS 文件系统	401

▼ 如何检查 CacheFS 文件系统的完整性	403
压缩已高速缓存的文件系统（任务列表）	403
压缩 CacheFS 文件系统	404
▼ 如何在高速缓存中压缩文件	405
▼ 如何显示已压缩文件的信息	405
使用压缩列表	407
▼ 如何创建压缩列表	407
▼ 如何使用压缩列表在高速缓存中压缩文件	408
从高速缓存中解压缩文件或压缩列表	409
▼ 如何从高速缓存中解压缩文件或压缩列表	409
对 cachefspack 错误进行疑难解答	410
收集 CacheFS 统计信息（任务列表）	413
收集 CacheFS 统计信息	414
▼ 如何设置 CacheFS 日志记录	415
▼ 如何查找 CacheFS 日志文件	415
如何停止 CacheFS 日志记录	416
▼ 如何查看工作集（高速缓存）大小	416
查看 CacheFS 统计信息	418
▼ 如何查看 CacheFS 统计信息	418
21 配置其他交换空间（任务）	421
关于交换空间	421
交换空间与虚拟内存	421
交换空间与 TMPFS 文件系统	422
将交换空间作为转储设备	422
交换空间和动态重新配置	423
如何了解是否需要更多交换空间？	423
与交换相关的错误消息	423
与 TMPFS 相关的错误消息	424
如何分配交换空间	424
交换区域和 /etc/vfstab 文件	424
规划交换空间	425
监视交换资源	425
添加更多交换空间	426
创建交换文件	427

▼ 如何创建交换文件并使其可用	427
删除不用的交换文件	428
▼ 如何删除不需要的交换空间	429
22 检查 UFS 文件系统一致性（任务）	431
文件系统一致性	432
如何记录文件系统的状态	432
fsck 命令检查和尝试修复的内容	433
为什么可能出现 UFS 文件系统的不一致性问题	433
接受一致性检查的 UFS 组件	433
fsck 摘要消息	438
以交互方式检查和修复 UFS 文件系统	439
▼ 如何从备用引导设备检查根 (/)、/usr 或 /var 文件系统	440
▼ 如何检查其他文件系统（不是根 (/)、/usr 或 /var）	442
整理 UFS 文件系统	444
▼ 如何整理 UFS 文件系统	445
修复 fsck 命令无法修复的 UFS 文件系统。	445
恢复坏的超级块	446
▼ 如何恢复坏的超级块（仅限 Solaris 10 6/06 发行版）	446
▼ 如何恢复坏的超级块（Solaris 8、9 和 10 发行版）	451
fsck 命令的语法和选项	453
23 UFS 文件系统（参考）	455
UFS 文件系统的柱面组结构	455
引导块	455
超级块	456
Inode	456
数据块	457
空闲块	457
自定义 UFS 文件系统参数	458
逻辑块大小	458
段大小	459
最小空闲空间	459
旋转延迟	460
优化类型	460

Inode 数（文件）	460
UFS 文件和文件系统的最大大小	461
最大 UFS 子目录数	461
24 备份和恢复文件系统（概述）	463
有关备份和恢复任务的参考信息	463
备份和恢复文件系统介绍	464
备份文件系统的原因	464
应该备份的文件系统	465
选择备份类型	466
选择磁带设备	467
备份和恢复文件系统的高级视图（任务列表）	467
计划备份注意事项	468
备份计划指南	469
使用转储级别创建增量备份	470
样例备份计划	471
25 备份文件和文件系统（任务）	479
备份文件和文件系统（任务列表）	479
准备文件系统备份	480
▼ 如何查找文件系统名称	480
▼ 如何确定进行完整备份所需的磁带数目	481
备份文件系统	481
▼ 如何将文件系统备份到磁带	482
26 使用 UFS 快照（任务）	491
使用 UFS 快照（任务列表）	491
UFS 快照概述	492
使用 UFS 快照的原因	492
UFS 快照性能问题	493
创建和删除 UFS 快照	493
创建多 TB UFS 快照	494
▼ 如何创建 UFS 快照	494
▼ 如何显示 UFS 快照信息	496

删除 UFS 快照	497
▼ 如何删除 UFS 快照	497
备份 UFS 快照	498
▼ 如何创建 UFS 快照的完整备份 (ufsdump)	499
▼ 如何创建 UFS 快照的增量备份 (ufsdump)	500
▼ 如何备份 UFS 快照 (tar)	500
从 UFS 快照备份中恢复数据	501
27 恢复文件和文件系统（任务）	503
恢复文件和文件系统备份（任务列表）	503
准备恢复文件和文件系统	504
确定文件系统名称	504
确定所需的磁带设备类型	504
确定磁带设备名称	504
恢复文件和文件系统	505
▼ 如何确定要使用的磁带	505
▼ 如何以交互方式恢复文件	506
▼ 如何以非交互方式恢复特定文件	509
▼ 如何恢复完整的文件系统	512
▼ 如何恢复根 (/) 和 /usr 文件系统	516
28 UFS 备份和恢复命令（参考）	521
ufsdump 命令的工作原理	521
确定设备特征	521
检测介质结尾	521
使用 ufsdump 命令复制数据	522
/etc/dumpdates 文件的作用	522
备份设备 (dump-file) 参数	523
指定要备份的文件	524
指定磁带特征	525
ufsdump 命令的限制	525
指定 ufsdump 命令选项和参数	525
缺省的 ufsdump 选项	525
ufsdump 命令和安全问题	526
指定 ufsrestore 选项和参数	526

29 复制 UFS 文件和文件系统（任务）	527
用于复制文件系统的命令	527
在磁带之间复制文件系统	529
创建精确的文件系统副本	530
▼ 如何复制磁盘 (dd)	530
在文件系统之间复制目录 (cpio 命令)	533
▼ 如何在文件系统之间复制目录 (cpio)	534
将文件和文件系统复制到磁带	535
将文件复制到磁带 (tar 命令)	535
▼ 如何将文件复制到磁带 (tar)	535
▼ 如何列出磁带中的文件 (tar)	537
▼ 如何从磁带中恢复文件 (tar)	537
使用 pax 命令将文件复制到磁带	538
▼ 如何将文件复制到磁带 (pax)	538
使用 cpio 命令将文件复制到磁带	539
▼ 如何将目录中的所有文件复制到磁带 (cpio)	539
▼ 如何列出磁带中的文件 (cpio)	540
▼ 如何从磁带中恢复所有文件 (cpio)	541
▼ 如何从磁带中恢复特定文件 (cpio)	542
将文件复制到远程磁带设备	543
▼ 如何将文件复制到远程磁带设备 (tar 和 dd)	543
▼ 如何从远程磁带设备中提取文件	545
将文件和文件系统复制到软盘	546
将文件复制到软盘时应了解的信息	546
▼ 如何将文件复制到单张经过格式化的软盘 (tar)	546
▼ 如何列出软盘中的文件 (tar)	547
▼ 如何从软盘中恢复文件 (tar)	548
将文件归档到多张软盘	549
30 管理磁带机（任务）	551
选择要使用的介质	551
备份设备名称	552
指定磁带机的反绕选项	553
指定磁带机的不同密度	553
显示磁带机状态	554

- ▼ 如何显示磁带机状态 554
- 处理盒式磁带 555
 - 重新张紧盒式磁带 555
 - 反绕盒式磁带 556
- 磁带机维护和介质处理指南 556

- 索引 557

前言

《系统管理指南：设备和文件系统》是介绍 Solaris™ 系统管理信息重要内容的一套书中的一本。该指南包含基于 SPARC® 和基于 x86 的系统的信息。

本书假设您已经完成以下任务：

- 安装了 SunOS 5.10 操作系统
- 设置了任何计划使用的网络软件

SunOS 5.10 发行版属于 Solaris 产品系列，它还包含许多功能，其中包括 GNOME 桌面环境。SunOS 5.10 操作系统与 AT&T 的 System V, Release 4 操作系统兼容。

对于 Solaris 10 发行版，系统管理员感兴趣的新功能已在相应各章的名为“...的新增功能”的各节中介绍。

注 - 此 Solaris 发行版支持使用以下 SPARC 和 x86 系列处理器体系结构的系统：UltraSPARC®、SPARC64、AMD64、Pentium 和 Xeon EM64T。支持的系统可以在 <http://www.sun.com/bigadmin/hcl> 上的《Solaris 10 Hardware Compatibility List》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”指出了有关 AMD64 或 EM64T 系统的特定 64 位信息。
- “32 位 x86”指出了有关基于 x86 的系统的特定 32 位信息。

若想了解本发行版支持哪些系统，请参见《Solaris 10 Hardware Compatibility List》。

注 - Sun 对本文档中提到的第三方 Web 站点的可用性不承担任何责任。对于此类站点或资源中的（或通过它们获得的）任何内容、广告、产品或其他资料，Sun 并不表示认可，也不承担任何责任。对于因使用或依靠此类站点或资源中的（或通过它们获得的）任何内容、产品或服务而造成的或连带产生的实际或名义损坏或损失，Sun 概不负责，也不承担任何责任。

目标读者

本书适用于所有负责管理一个或多个运行 Solaris 10 发行版的系统的人员。要使用本书，您应当具备 1 到 2 年的 UNIX® 系统管理经验。参加 UNIX 系统管理培训课程可能会对您有所帮助。

系统管理卷的结构

以下是系统管理指南卷所包含主题列表。

书名	主题
《系统管理指南：基本管理》	用户帐户和组、服务器和客户机支持、关闭和引导系统、管理服务以及管理软件（软件包和修补程序）
《系统管理指南：高级管理》	打印服务、终端和调制解调器、系统资源（磁盘配额、记帐和 crontab）、系统进程以及 Solaris 软件问题疑难解答
《系统管理指南：设备和文件系统》	可移除介质、磁盘和设备、文件系统以及备份和恢复数据
《系统管理指南：IP 服务》	TCP/IP 网络管理、IPv4 和 IPv6 地址管理、DHCP（动态主机配置协议）、IPsec（Internet 协议安全）、IKE（Internet 密钥交换）、IP 过滤器、移动 IP、IP 网络多路径（IP network multipathing, IPMP）和 IPQoS（IP 服务质量）
《系统管理指南：命名和目录服务（DNS、NIS 和 LDAP）》	DNS、NIS 和 LDAP 名称和目录服务，包括从 NIS 到 LDAP 的转换和从 NIS+ 到 LDAP 的转换
《System Administration Guide: Naming and Directory Services (NIS+)》	NIS+ 名称和目录服务
《系统管理指南：网络服务》	Web 高速缓存服务器、与时间相关的服务、网络文件系统（NFS 和 Autofs）、邮件、SLP 和 PPP
《系统管理指南：安全性服务》	审计、设备管理、文件安全、BART（基本审计和报告工具）、Kerberos 服务、PAM（可插拔验证模块）、Solaris 加密框架、权限、RBAC（基于角色的访问控制）、SASL（简单身份验证和安全层）和 Solaris 安全 Shell
《系统管理指南：Solaris Containers—资源管理和 Solaris Zones》	资源管理主题项目和任务、扩展记帐、资源控制、公平共享调度程序（fair share scheduler, FSS）、使用资源上限设置守护进程（resource capping daemon, rcapd）的物理内存控制以及动态资源池；使用 Solaris Zones 软件分区技术的虚拟化
《Solaris ZFS 管理指南》	ZFS（Zettabyte 文件系统）存储工具以及文件系统的创建和管理、快照、克隆、备份、使用访问控制列表（Access Control List, ACL）保护 ZFS 文件、在安装区域的 Solaris 系统中使用 ZFS、仿真卷以及疑难解答和数据恢复

文档、支持和培训

Sun Web 站点提供有关以下附加资源的信息：

- 文档 (<http://www.sun.com/documentation/>)
- 支持 (<http://www.sun.com/support/>)
- 培训 (<http://www.sun.com/training/>)

印刷约定的含义

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>
AaBbCc123	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> <code>Password:</code>
<i>AaBbCc123</i>	要使用实名或值替换的命令行占位符	要删除文件，请键入 <code>rm filename</code> 。
<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。
新词术语强调	新词或术语以及要强调的词	请勿保存更改。
《书名》	书名	阅读《用户指南》的第 6 章。

命令中的 shell 提示符示例

下表列出了 C shell、Bourne shell 和 Korn shell 的缺省系统提示符和超级用户提示符。

表 P-2 Shell 提示符

Shell	提示符
C shell	<code>machine_name%</code>
C shell 超级用户	<code>machine_name#</code>
Bourne shell 和 Korn shell	<code>\$</code>

表 P-2 Shell 提示符 (续)

Shell	提示符
Bourne shell 和 Korn shell 超级用户	#

一般约定

请注意本书中使用的以下约定：

- 执行步骤或使用示例时，请确保完全按照括号中所示键入双引号 (")、左单引号 (') 和右单引号 (')。
- 在某些键盘上，回车键被标记为 Enter。
- 根路径通常包括 /sbin、/usr/sbin、/usr/bin 和 /etc 目录。因此，对于本书中的步骤，在显示这些目录中的命令时不带绝对路径名。对于那些使用其他不太常见目录中命令的步骤，在示例中会显示其绝对路径。
- 本书中的示例涉及的是基本 SunOS 软件安装，其中未安装二进制兼容性软件包，路径中也不带 /usr/ucb。



注意 - 如果搜索路径中包含 /usr/ucb，该目录应当始终位于搜索路径的末尾。除格式和选项之外，/usr/ucb 中类似 ps 或 df 的命令与 SunOS 命令相同。

管理可移除介质（概述）

本章提供有关在 Solaris OS 中管理可移除介质的一般原则。

以下是本章中概述信息的列表。

- 第 25 页中的 “Solaris 10 1/06 发行版中可移除介质方面的新增功能”
- 第 27 页中的 “Solaris 10 发行版中可移除介质方面的新增功能”
- 第 29 页中的 “有关管理可移除介质任务的参考信息”
- 第 29 页中的 “可移除介质的功能和优点”
- 第 30 页中的 “手动挂载和自动挂载的比较”
- 第 30 页中的 “使用卷管理可以执行的操作”

Solaris 10 1/06 发行版中可移除介质方面的新增功能

本节介绍 Solaris 10 1/06 发行版中可移除介质方面的新增功能。

vold 由服务管理工具 (Service Management Facility, SMF) 进行管理

Solaris 10 1/06：卷管理守护进程 vold 现在由服务管理工具 (Service Management Facility, SMF) 进行管理。这意味着，可以使用 `svcadm disable volfs` 命令禁用以下新增 volfs 服务（如果适用）：

```
# svcadm disable volfs
```

可以使用以下命令确定 volfs 服务的状态：

```
$ svcs volfs
```

```
STATE          STIME          FMRI
```

```
online          Sep_29   svc:/system/filesystem/volfs:default
```

有关更多信息，请参见 `smf(5)`。

可以使用 `svccfg` 命令显示和设置其他 `vold` 属性。例如，可以暂时启用 `vold` 日志记录功能以帮助对问题进行疑难解答。例如：

```
# svccfg
```

```
svc:> select system/filesystem/volfs
```

```
svc:/system/filesystem/volfs> setprop vold/log_debuglevel=3
```

```
svc:/system/filesystem/volfs> exit
```

```
# svcadm disable volfs
```

```
# svcadm enable volfs
```

还可以使用 `svccfg` 命令显示可设置 `vold` 属性的列表。

```
# svccfg
```

```
svc:> select volfs
```

```
svc:/system/filesystem/volfs> listprop vold/*
```

```
vold/config_file          astring
```

```
vold/log_debuglevel       count      3
```

```
vold/log_file             astring
```

```
vold/log_nfs_trace        boolean   false
```

```
vold/log_verbose          boolean   false
```

```
vold/root_dir             astring
```

```
vold/never_writeback_label boolean   false
```

```
svc:/system/filesystem/volfs> exit
```

有关这些属性的说明，请参见 `vold(1M)`。

对卷管理 (vold) 的改进

Solaris 10 1/06：改进了可移除介质管理。以前，对于不包含介质的可移除设备，`vold` 不创建设备链路。现在，对于不包含介质的设备，会正确创建设备链路，如下所示：

```
lrwxrwxrwx  1 root    root      28 Jun 13 13:09 /vol/dev/aliases/cdrom0 ->
/vol/dev/rdisk/c2t2d0/nomedia
```

现在，可以在运行 `vold` 时使用 `cdwr` 和 `rmformat` 命令列出不包含介质的设备。

通过对 `/etc/vold.conf` 文件中的 `support nomedia` 项进行如下更改，可以恢复到以前的 `vold` 行为：

```
support media
```

然后，重新启动 `vold`。有关更多信息，请参见 `vold.conf(4)`。

此外，`vold` 现在可识别热插拔。此改进意味着，如果插入可移除介质，则 `vold` 将自动检测并挂载该介质。应无需手动重新启动 `vold`，即可识别并挂载任何可移除介质设备中的文件系统。

如果使用的是传统的或非 USB 软盘设备，则可能需要先发出 `volcheck` 命令，`vold` 才能识别介质。

如果检测到介质，但由于某种原因未进行挂载，则需要运行以下命令：

```
# volrmount -i rmdisk0
```

在热移除可移除介质设备之前，请先弹出介质。例如：

```
# eject rmdisk0
```

Solaris 10 发行版中可移除介质方面的新增功能

以下几节介绍 Solaris 10 发行版中可移除介质方面的新增功能。

DVD+RW 和 DVD-RW 支持

在此 Solaris 发行版中，通过先使用 `mkisofs` 命令创建数据，可以使用 `cdwr` 命令在 DVD+RW 或 DVD-RW 驱动器上创建 DVD。但是，不能创建多会话数据 DVD。然后可以挂载这些光盘，并将其作为 HSFS 文件系统访问。

DVD+RW 和 DVD-RW 设备的定义如下：

- DVD+RW—数字视频光盘（可刻录/可重写）驱动器既可以写 DVD-R 光盘（可以在大多数 DVD 播放机和计算机驱动器上播放），也可以写 DVD-RW 可重写光盘
- DVD-RW—数字视频光盘（可重写）只能由 DVD-RW 驱动器读取

在本指南中，除非需要特定于 DVD+RW 设备或 DVD-RW 设备的信息，否则 DVD+RW 和 DVD-RW 设备统称为“DVD”设备。

写 DVD 时 `cdwr` 命令使用一次写入 (Disk-At-Once, DAO) 模式，该模式执行以下操作：

- 完成写操作后关闭介质
- 防止添加任何新的会话

使用 DVD+RW 设备时，请牢记以下要点：

- 不能清空（或擦除）DVD+RW 介质。
- 可通过在介质上写新映像来重复使用 DVD+RW 介质。`cdwr` 命令自动格式化并覆写现有介质。

有关将 USB 海量存储类兼容的 CD 或 DVD-RW 设备添加到系统的说明，请参见第 139 页中的“使用 USB 海量存储设备”。

新增的 `cdwr` 选项

在此 Solaris 发行版中添加了以下 `cdwr` 选项，以改进介质的管理：

- `-b fast` 选项用于快速擦除介质。此选项可在大约 30 秒内擦除介质，而非 10-15 分钟。使用此选项可擦除介质的 TOC。如果介质已损坏，则需要使用 `-b all` 清除整个介质。
- 使用 `-l` 选项可打开以前关闭的 CD-RW 介质。此选项擦除最后一个导出区，且允许您将更多会话添加到多会话 CD-RW。

列出可移除介质设备

可以使用 `rmformat` 命令列出系统上的可移除介质设备。使用 `rmformat -l` 选项和特定设备可提供有关该设备的详细信息，如 `vol` 使用的名称以及逻辑和物理设备名称。

例如：

```
# rmformat

Looking for devices...

1. Volmgt Node: /vol/dev/aliases/rmdisk1
```

```
Logical Node: /dev/rdisk/c5t0d0s2

Physical Node: /pci@1e,600000/usb@b/hub@2/storage@4/disk@0,0

Connected Device: TEAC      FD-05PUB      1026

Device Type: Floppy drive
```

有关更多信息，请参见 `rmformat(1)`。

播放音乐 CD

只需将音乐 CD 插入 CD-ROM 驱动器再启动 `gnome-cd` 应用程序，即可播放它。确保 `vold` 正在运行。有关更多信息，请参见 `gnome-cd.1`。

有关管理可移除介质任务的参考信息

请参见以下内容，以了解管理可移除介质的逐步说明。

可移除介质管理任务	更多信息
访问可移除介质	第 3 章
格式化可移除介质	第 2 章
写数据 CD 和 DVD 以及音乐 CD	第 4 章

可移除介质的功能和优点

此 Solaris 发行版为用户和软件开发者提供了用于处理可移除介质的标准接口。称为**卷管理**的此接口具有以下三个主要优点：

- 自动挂载可移除介质。有关手动挂载和自动挂载的比较，请参见下一节。
- 不必成为超级用户即可访问可移除介质。
- 允许您为网络中的其他系统提供对本地系统上任何可移除介质的自动访问。有关更多信息，请参见[第 3 章](#)。

手动挂载和自动挂载的比较

下表对手动挂载（不使用卷管理）和自动挂载（使用卷管理）可移除介质所涉及的步骤进行了比较。

表 1-1 手动挂载和自动挂载可移除介质的比较

步骤	手动挂载	自动挂载
1	插入介质。	插入介质。
2	成为超级用户。	对于软盘，使用 <code>volcheck</code> 命令。
3	确定介质设备的位置。	卷管理 (<code>vol</code>) 自动执行以前手动挂载和使用可移除介质所需的许多任务。
4	创建挂载点。	
5	确保当前不在挂载点目录中。	
6	挂载设备并使用适当的 <code>mount</code> 选项。	
7	退出超级用户帐户。	
8	使用介质上的文件。	使用介质上的文件。
9	成为超级用户。	
10	取消挂载介质设备。	
11	弹出介质。	弹出介质。
12	退出超级用户帐户。	

使用卷管理可以执行的操作

在本质上，使用卷管理可以访问可移除介质，就像手动挂载那样，但是前者更容易且不需要超级用户访问权限。要使可移除介质更易于使用，可以在容易记住的位置挂载可移除介质。

表 1-2 如何访问由卷管理 (`vol`) 所管理的可移除介质上的数据

访问	插入	文件位置
第一个软盘上的文件	软盘，然后在命令行上键入 <code>volcheck</code>	<code>/floppy</code>
可移除硬盘上的文件	可移除硬盘，然后在命令行上键入 <code>volcheck</code>	<code>/rmdisk/rmdisk0</code> 或 <code>/rmdisk/rmdisk1</code>

表 1-2 如何访问由卷管理 (vold) 所管理的可移除介质上的数据 (续)

访问	插入	文件位置
第一个 CD 上的文件	CD，然后等待几秒	/cdrom/volume-name
第一个 DVD 上的文件	DVD，然后等待几秒	/cdrom/volume-name
第一个 PCMCIA 上的文件	PCMCIA，然后等待几秒	/pcmem/pcmem0

如果系统具有多种类型的可移除设备，请参见下表以了解其访问点。

表 1-3 访问可移除介质的位置

介质设备	使用此路径访问文件系统	使用此路径访问原始数据
第一个软盘驱动器	/floppy/floppy0	/vol/dev/aliases/floppy0
第二个软盘驱动器	/floppy/floppy1	/vol/dev/aliases/floppy1
第一个 CD-ROM 驱动器	/cdrom/cdrom0	/vol/dev/aliases/cdrom0
第二个 CD-ROM 驱动器	/cdrom/cdrom1	/vol/dev/aliases/cdrom1
第一个或第二个可移除硬盘	/rmdisk/rmdisk0 或 /rmdisk/rmdisk1	/vol/dev/aliases/rmdisk0 或 /vol/dev/aliases/rmdisk1
第一个 PCMCIA 驱动器	/pcmem/pcmem0	/vol/dev/aliases/pcmem0

管理可移除介质（任务）

本章介绍在 Solaris OS 中如何从命令行管理可移除介质。

有关与管理可移除介质关联的过程信息，请参见第 33 页中的“管理可移除介质（任务列表）”。有关可移除介质的背景信息，请参见第 1 章。

管理可移除介质（任务列表）

以下任务列表介绍了有关管理可移除介质的任务。

任务	说明	参考
1. 装入介质。	将软盘插入驱动器，然后键入 <code>volcheck</code> 命令。	第 35 页中的“如何装入可移除介质”
2. （可选）格式化软盘。	格式化软盘。	第 36 页中的“如何格式化软盘 (<code>rmformat</code>)”
3. （可选）添加 UFS 或 PCFS 文件系统。	添加 UFS 或 PCFS 文件系统以使用该介质传输文件。	第 37 页中的“如何在可移除介质上创建文件系统”
4. （可选）检查介质。	验证介质上文件系统的完整性。	第 39 页中的“如何检查可移除介质上的文件系统”
5. （可选）修复介质上的坏块。	如有必要，请修复介质上的任何坏块。	第 40 页中的“如何修复可移除介质上的坏块”
6. （可选）应用读/写保护和口令保护。	如有必要，请在介质上应用读/写保护或口令保护。	第 41 页中的“如何在可移除介质上启用或禁用写保护”

格式化软盘

`rmformat` 命令是可以用来格式化和保护可重写软盘的实用程序。除非 `vold` 没有运行，否则此实用程序不需要超级用户权限。卷管理 (`vold`) 将自动挂载文件系统。因此，如果介质包含现有的文件系统，则可能必须取消挂载该介质，然后才能对其进行格式化。

`rmformat` 命令具有以下三个格式化选项：

- `quick`—此选项用于格式化软盘，但对介质上的某些磁道不进行检查或进行有限的检查。
- `long`—此选项用于完全格式化软盘。对于某些设备，使用此选项可能包括按驱动器检查整个介质。
- `force`—此选项用于完全格式化软盘，而无需用户确认。对于具有口令保护机制的软盘，此选项可在格式化之前清除口令。此功能在忘记口令时非常有用。在没有口令保护的软盘上，此选项可强制长格式。

可移除介质硬件注意事项

使用软盘和 PCMCIA 内存卡时，请牢记以下限制：

- SPARC 和 x86 UFS 格式是不同的。SPARC 使用小尾数法位编码，而 x86 使用大尾数法位编码。为 UFS 格式化的介质仅限于在对其进行格式化的硬件平台上使用。因此，在基于 SPARC 的平台上为 UFS 格式化的软盘不能用于 x86 平台上的 UFS。同样，在 x86 平台上为 UFS 格式化的软盘也不能在 SPARC 平台上使用。该限制同样适用于 PCMCIA 内存卡。
- 除了支持 SunOS 文件系统的结构外，SunOS™ 文件系统的完整格式还包含基本的“位”格式化。除了支持 MS-DOS 或 NEC-DOS 文件系统的结构外，DOS 文件系统的完整格式还包含基本的“位”格式化。为每种类型的文件系统准备介质所需的过程是不同的。因此，在格式化软盘或 PCMCIA 内存卡之前，请考虑要遵照哪个过程进行操作。有关更多信息，请参见第 33 页中的“管理可移除介质（任务列表）”。

软盘硬件注意事项

格式化软盘时，请牢记以下内容：

- 有关软盘名称的信息，请参见表 3-1。
- 对于未命名（即没有“标号”）的软盘，会为其指定缺省名称 `noname`。

Solaris 系统可以格式化以下类型的软盘：

- UFS
- MS-DOS 或 NEC-DOS (PCFS)
- UDFS

在 Solaris 系统（SPARC 或 x86）上，可以格式化以下密度的软盘。

软盘大小	软盘密度	容量
3.5”	高密度 (High density, HD)	1.44 MB
3.5”	双倍密度 (Double density, DD)	720 KB

缺省情况下，软盘驱动器会将软盘格式化为相同的密度。此缺省设置表示，除非另行指定，否则 1.44 MB 驱动器尝试将软盘格式化为 1.44 MB，而不管该软盘实际上是否为 1.44 MB 软盘。换句话说，可以将一张软盘格式化为其容量或更低容量，并且驱动器可以为软盘执行这一操作。

▼ 如何装入可移除介质

有关可移除介质硬件注意事项的信息，请参见第 34 页中的“可移除介质硬件注意事项”。

- 1 插入介质。
- 2 确保该介质已格式化。

如果无法确定，请插入介质并检查系统控制台窗口中的状态消息，如步骤 3 中所述。如果需要格式化介质，请转到第 36 页中的“如何格式化软盘 (rmformat)”。
- 3 （可选的）如果使用的是传统的非 USB 软盘设备，请通知卷管理。

```
$ volcheck -v
```

可能会显示两条状态消息：

media was found

卷管理检测到介质，并将尝试在表 3-1 所述的目录中挂载该介质。

如果介质已正确格式化，则控制台中不会出现错误消息。

如果介质未格式化，则仍会显示“media was found”消息。但是，在系统控制台窗口中会出现类似以下内容的错误消息：

```
fd0: unformatted diskette or no diskette in the drive

fd0: read failed (40 1 0)

fd0: bad format
```

必须格式化介质后，卷管理才能挂载该介质。有关更多信息，请参见第 2 章。

no media was found

卷管理未检测到介质。请确保介质已正确插入，然后再次运行 volcheck。如果不成功，请检查介质，该介质可能已损坏。另

外，也可以尝试手动挂载介质。

4 通过列出介质的内容，验证是否已挂载该介质。

例如，对软盘执行以下操作：

```
$ ls /floppy
```

```
floppy0 myfiles
```

提示 - `floppy0` 是指向软盘实际名称的符号链接。在本示例中为 `myfiles`。如果软盘没有名称但已正确格式化，则系统将其称为 `unnamed_floppy`。

如果 `/floppy` 目录为空，则表明软盘未挂载或者未正确格式化。要查明原因，请运行 `mount` 命令，然后查找以 `/floppy` 开头的行（通常位于列表结尾）：

```
/vol/dev/diskette0/ name 上的 /floppy/name
```

如果未出现该行，则表明未挂载软盘。请查看系统控制台窗口中的错误消息。

▼ 如何格式化软盘 (rmformat)

可以使用 `rmformat` 命令格式化软盘：缺省情况下，此命令会在介质上创建两个分区：分区 0 和分区 2（整个介质）。

1 验证卷管理是否正在运行。如果正在运行，则可以使用设备名称的较短昵称。

```
$ svcs volfs
```

```
STATE          STIME          FMRI
```

```
online          10:39:12 svc:/system/filesystem/volfs:default
```

有关启动 `vold` 的信息，请参见第 45 页中的“如何停止和启动卷管理 (`vold`)”。有关标识介质设备名称的信息，请参见第 44 页中的“使用可移除介质名称”。

2 格式化该软盘。

```
$ rmformat -F [ quick | long | force ] device-name
```

有关 `rmformat` 格式化选项的更多信息，请参见第 34 页中的“格式化软盘”。

如果 `rmformat` 输出指明存在坏块，请参见第 40 页中的“如何修复可移除介质上的坏块”。

3 （可选）使用 8 个字符的标号标记软盘。

```
$ rmformat -b label device-name
```

有关创建 DOS 标号的信息，请参见 `mkfs_pcfs(1M)`。

示例 2-1 格式化软盘

本示例说明如何格式化软盘。

```
$ rmformat -F quick /dev/rdiskette
```

```
Formatting will erase all the data on disk.
```

```
Do you want to continue? (y/n) y
```

```
.....
```

▼ 如何在可移除介质上创建文件系统

- 1 (可选的) 格式化该软盘。

```
$ rmformat -F quick device-name
```

- 2 (可选) 创建备用 Solaris 分区表。

```
$ rmformat -s slice-file device-name
```

将显示类似以下内容的片文件样例：

```
slices: 0 = 0, 30MB, "wm", "home" :
```

```
1 = 30MB, 51MB :
```

```
2 = 0, 94MB, "wm", "backup" :
```

```
6 = 81MB, 13MB
```

- 3 成为超级用户。
- 4 确定相应的文件系统类型，然后选择以下操作之一：

- 创建 UFS 文件系统。例如：

```
# newfs /vol/dev/aliases/floppy0
```

- 创建 PCFS 文件系统。例如：

```
# mkfs -F pcfs /dev/rdisk/c0t4d0s2:c
```

- 创建 UDFS 文件系统。例如：

```
# mkfs -F udfs /dev/rdisk/c0t1d0s1
```

示例 2-2 为 UFS 文件系统格式化软盘

以下示例说明如何格式化软盘并在该软盘上创建 UFS 文件系统。

```
$ rmformat -F quick /vol/dev/aliases/floppy0

Formatting will erase all the data on disk.

Do you want to continue? (y/n) y

$ su

# /usr/sbin/newfs /vol/dev/aliases/floppy0

newfs: construct a new file system /dev/rdiskette: (y/n)? y

/dev/rdiskette: 2880 sectors in 80 cylinders of 2 tracks, 18 sectors

1.4MB in 5 cyl groups (16 c/g, 0.28MB/g, 128 i/g)

super-block backups (for fsck -F ufs -o b=#) at:

32, 640, 1184, 1792, 2336,

#
```

示例 2-3 为 PCFS 文件系统格式化软盘

本示例说明如何使用备用 fdisk 分区创建 PCFS 文件系统。在这些示例中，vold 没有运行。

```
$ rmformat -F quick /dev/rdisk/c0t4d0s2:c

Formatting will erase all the data on disk.

Do you want to continue? (y/n) y

$ su

# fdisk /dev/rdisk/c0t4d0s2:c

# mkfs -F pcfs /dev/rdisk/c0t4d0s2:c

Construct a new FAT file system on /dev/rdisk/c0t4d0s2:c: (y/n)? y
```

```
#
```

本示例说明如何在不使用 `fdisk` 分区的情况下创建 PCFS 文件系统。

```
$ rmformat -F quick /dev/rdiskette
```

```
Formatting will erase all the data on disk.
```

```
Do you want to continue? (y/n) y
```

```
$ su
```

```
# mkfs -F pcfs -o nofdisk,size=2 /dev/rdiskette
```

```
Construct a new FAT file system on /dev/rdiskette: (y/n)? y
```

```
#
```

▼ 如何检查可移除介质上的文件系统

- 1 成为超级用户。
- 2 确定文件系统类型，然后选择以下操作之一：

- 检查 UFS 文件系统。

```
# fsck -F ufs device-name
```

- 检查 UDFS 文件系统。

```
# fsck -F udfs device-name
```

- 检查 PCFS 文件系统。

```
# fsck -F pcfs device-name
```

示例 2-4 检查可移除介质上的 PCFS 文件系统

以下示例说明如何检查介质上的 PCFS 文件系统的一致性。在本示例中，`vold` 没有运行。

```
# fsck -F pcfs /dev/rdisk/c0t4d0s2
```

```
** /dev/rdisk/c0t4d0s2
```

```
** Scanning file system meta-data

** Correcting any meta-data discrepancies

1457664 bytes.

0 bytes in bad sectors.

0 bytes in 0 directories.

0 bytes in 0 files.

1457664 bytes free.

512 bytes per allocation unit.

2847 total allocation units.

2847 available allocation units.

#
```

▼ 如何修复可移除介质上的坏块

如果驱动器支持坏块管理，则只能使用 `rmformat` 命令来验证、分析和修复在验证过程中找到的坏扇区。大多数软盘、PCMCIA 内存卡和 USB 内存条都不支持坏块管理。

如果驱动器支持坏块管理，则会尽最大努力纠正坏块。如果尽最大努力也无法纠正坏块，则会显示一条消息，指明修复介质失败。

1 修复可移除介质上的坏块。

```
$ rmformat -c block-numbers device-name
```

提供以前的 `rmformat` 会话中的十进制、八进制或十六进制格式的块编号。

2 验证介质。

```
$ rmformat -V read device-name
```

将读/写保护和口令保护应用于可移除介质

在支持此功能的可移除介质上，可以应用读保护或写保护，并设置口令。

▼ 如何在可移除介质上启用或禁用写保护

- 1 确定要启用还是禁用写保护，然后选择以下操作之一：

- 启用写保护。

```
$ rmformat -w enable device-name
```

- 禁用写保护。

```
$ rmformat -w disable device-name
```

- 2 验证是否已启用或禁用介质的写保护。

```
$ rmformat -p device-name
```

▼ 如何在可移除介质上启用或禁用读/写保护以及设置口令

对于支持此功能的可移除介质，可以应用最多为 32 个字符的口令。

如果尝试在不支持此功能的介质上应用口令，则会收到一条警告消息。

- 1 确定是要启用还是禁用读/写保护，并设置口令。选择以下操作之一：

- 启用读保护或写保护。

```
$ rmformat -W enable device-name
```

```
Please enter password (32 chars maximum): xxx
```

```
Please reenter password:
```

```
$ rmformat -R enable device-name
```

```
Please enter password (32 chars maximum): xxx
```

```
Please reenter password:
```

- 禁用读保护或写保护并删除口令。

```
$ rmformat -W disable device-name
```

```
Please enter password (32 chars maximum): xxx
```

```
$ rmformat -R disable device-name
```

```
Please enter password (32 chars maximum): xxx
```

2 验证是否已启用或禁用介质的读/写保护。

```
$ rmformat -p device-name
```

示例 2-5 启用或禁用读/写保护和口令保护

本示例说明如何在软盘上启用写保护并设置口令。

```
$ rmformat -W enable /vol/dev/aliases/floppy0
```

```
Please enter password (32 chars maximum): xxx
```

```
Please reenter password: xxx
```

本示例说明如何在软盘上禁用读保护并删除口令。

```
$ rmformat -R disable /vol/dev/aliases/floppy0
```

```
Please enter password (32 chars maximum): xxx
```

访问可移除介质（任务）

本章介绍在 Solaris OS 中如何从命令行访问可移除介质。

有关访问可移除介质过程的信息，请参见以下内容：

- 第 43 页中的“访问可移除介质（任务列表）”
- 第 49 页中的“在远程系统上访问可移除介质（任务列表）”

有关可移除介质的背景信息，请参见第 1 章。

访问可移除介质（任务列表）

以下任务列表介绍了有关访问可移除介质的任务。

任务	说明	参考
1.（可选）添加可移除介质驱动器。	如有必要，请将可移除介质驱动器添加到系统。	第 45 页中的“如何添加新的可移除介质驱动器”
2.（可选）确定是在运行还是不运行卷管理 (vold) 的情况下使用可移除介质。	缺省情况下，将运行卷管理 (vold)。确定是在运行还是不运行卷管理的情况下使用可移除介质。	第 45 页中的“如何停止和启动卷管理 (vold)”
3. 访问可移除介质。	在运行或不运行卷管理的情况下访问不同种类的可移除介质。	第 46 页中的“如何访问可移除介质上的信息”
4.（可选）复制文件或目录。	从介质复制文件或目录，类似于从文件系统中的其他任何位置进行复制。	第 47 页中的“如何从可移除介质复制信息”
5. 确定介质是否仍在使用。	弹出介质之前，确定其是否仍在使用。	第 48 页中的“如何确定可移除介质是否仍在使用”

任务	说明	参考
6. 弹出介质。	完成后，从驱动器中弹出介质。	第 49 页中的“如何弹出可移除介质”

访问可移除介质

无论是否使用卷管理，都可以访问有关可移除介质的信息。有关使用 GNOME 的文件管理器访问可移除介质相关信息的信息，请参见 GNOME 桌面文档。

卷管理 (vold) 可主动管理所有的可移除介质设备。因此，使用设备名称（如 `/dev/rdisk/cntndnsn` 或 `/dev/dsk/cn tndn sn`）访问可移除介质的任何尝试都将失败。

使用可移除介质名称

可以使用不同名称访问所有可移除介质。下表介绍了在使用或不使用卷管理的情况下可以访问的不同介质名称。

表 3-1 可移除介质名称

介质	卷管理设备名称	卷管理设备别名	设备名称
第一个软盘驱动器	/floppy	/vol/dev/aliases/floppy0	/dev/rdiskette
			/vol/dev/rdiskette0/
			volume-name
第一个、第二个、第三个 CD-ROM 或 DVD-ROM 驱动器	/cdrom0 /cdrom1 /cdrom2	/vol/dev/aliases/cdrom0 /vol/dev/aliases/cdrom1 /vol/dev/aliases/cdrom2	/vol/dev/rdisk/cnt n[dn] /
			volume-name
第一个、第二个、第三个 PCMCIA 驱动器	/pcmem/pcmem0 /pcmem/pcmem1 /pcmem/pcmem2	/vol/dev/aliases/pcmem0 /vol/dev/aliases/pcmem1 /vol/dev/aliases/pcmem2	/vol/dev/rdisk/cnt ndn /
			volume-name
USB 内存条	/rmdisk/noname	/vol/dev/aliases/rmdisk0	/vol/dev/dsk/cntndn/volume-name:c

访问可移除介质数据的指导原则

大多数 CD 和 DVD 都是按照可移植的 ISO 9660 标准格式化的。因此，卷管理可以挂载大多数 CD 和 DVD。但是，具有 UFS 文件系统的 CD 或 DVD 不能在体系结构之间移植。因此，它们必须在针对其设计的体系结构上使用。

例如，x86 平台无法识别 SPARC™ 平台上具有 UFS 文件系统的 CD 或 DVD。同样，SPARC 平台上的卷管理也无法挂载 x86 UFS CD。此限制通常也适用于软盘。但是，一些体系结构共享相同的位结构，因此有时特定于一种体系结构的 UFS 格式可被另一种体系结构识别。但是，UFS 文件系统结构的设计不能保证此兼容性。

要适应不同的格式，可将 CD 或 DVD 分成若干个片。这些片在作用上与硬盘上的分区类似。9660 部分可移植，但 UFS 部分特定于体系结构。如果在挂载 CD 或 DVD 时出现问题，尤其是安装 CD 或 DVD 时，应确保其 UFS 文件系统适合系统的体系结构。例如，可以检查 CD 或 DVD 上的标号。

▼ 如何添加新的可移除介质驱动器

通常，大多数新式总线类型支持热插拔。如果系统的总线类型支持热插拔，则可能只需执行下面的步骤 5。如果系统的总线类型不支持热插拔，则可能必须执行下面步骤 1-6 中介绍的任务。

- 创建 /reconfigure 文件。
- 重新引导系统，以便卷管理可识别新的介质驱动器。

有关热插拔设备的更多信息，请参见第 6 章。

- 1 成为超级用户。
- 2 创建 /reconfigure 文件。
touch /reconfigure
- 3 使系统转至运行级 0。
init 0
- 4 关闭系统电源。
- 5 连接新的介质驱动器。
有关特定说明，请参见硬件手册。
- 6 打开系统电源。
系统会自动进入多用户模式。

▼ 如何停止和启动卷管理 (vold)

有时，可能需要在不使用卷管理的情况下管理介质。本节介绍如何停止和重新启动卷管理。

1 确保未使用介质。

如果无法确定是否已找到介质的所有用户，请使用 `fuser` 命令。请参见第 48 页中的 “[如何确定可移除介质是否仍在使用](#)”。

2 成为超级用户。**3 选择以下操作之一：**

- 停止卷管理。

```
# svcadm disable volfs
```

- 启动卷管理。

```
# svcadm enable volfs
```

```
volume management starting.
```

▼ 如何访问可移除介质上的信息

1 插入介质。

几秒后将挂载介质。

2 （可选的）如果使用传统的非 USB 软盘设备，请检查驱动器中的介质。

```
% volcheck
```

3 列出介质的内容。

```
% ls /media
```

通过命令行界面使用适当的设备名称访问信息。有关设备名称的说明，请参见表 3-1。

示例 3-1 访问可移除介质上的信息

此示例说明如何访问软盘上的信息。

```
$ $ ls /floppy
```

```
myfile
```

此示例说明如何访问 USB 内存条上的信息。

```
$ ls /rmdisk
```

```
rmdisk0/          rmdisk1/
```

此示例说明如何访问 DVD 或 CD 上的信息。

```
$ ls /cdrom
```

```
cdrom0          sol_10_305_sparc
```

此示例说明如何查看 DVD 或 CD 上的符号链接。

```
$ ls -lL /cdrom/cdrom0
```

```
total 24
```

```
dr-xr-xr-x  2 root    sys          2048 Jan 23 03:24 s0
```

```
drwxr-xr-x 18 root    root          512 Jan 23 03:25 s1
```

```
drwxr-xr-x  2 root    root          512 Jan 23 03:27 s2
```

```
drwxr-xr-x  2 root    root          512 Jan 23 03:27 s3
```

```
drwxr-xr-x  2 root    root          512 Jan 23 03:27 s4
```

```
drwxr-xr-x  2 root    root          512 Jan 23 03:27 s5
```

此示例说明如何访问 PCMCIA 内存卡上的信息，如下所示：

```
$ ls /pcmem/pcmem0
```

```
pcmem0 myfiles
```

▼ 如何从可移除介质复制信息

与任何其他文件系统一样，可以访问可移除介质上的文件和目录。仅有的重要限制与拥有权和权限有关。

例如，如果您将文件从 CD 复制到文件系统中，则您即是属主。但是，您不会具有写入权限，因为 CD 上的文件从没有写入权限。您必须自行更改权限。

1 确保已挂载介质。

```
$ ls /media
```

ls 命令会显示已挂载介质的内容。如果未显示内容，请参见第 46 页中的“[如何访问可移除介质上的信息](#)”。

2 （可选）复制文件或目录。

例如，对于 DVD，可以执行以下操作：

```
$ cp /cdrom/sol_10_305_sparc/s0/Solaris_10/Tools/add_install_client .
```

```
$ ls -l
```

```
-rwxr-xr-x  1 pmorph      gelfs   66393 Jun 14 16:08 add_install_client
```

例如，对于 PCMCIA 内存卡，可以执行以下操作：

```
$ cp /pcmem/pcmem0/readme2.doc .
```

```
$ cp -r /pcmem/pcmem0/morefiles .
```

▼ 如何确定可移除介质是否仍在使用

- 1 成为超级用户。

- 2 确定正在访问介质的进程。

```
# fuser -u /media
```

-u 显示介质的用户。

有关更多信息，请参见 `fuser(1M)`。

- 3 （可选）中止正在访问介质的进程。

```
# fuser -u -k /media
```

-k 中止正在访问介质的进程。



注意 - 仅在紧急情况下才能中止正在访问介质的进程。

- 4 检验进程是否已中止。

```
# pgrep process-ID
```

示例 3-2 确定介质是否仍在使用

以下示例说明用户 `pmorph` 正在访问 `/cdrom/sol_10_305_sparc/s0/Solaris_10/Tools` 目录。

```
# fuser -u /cdrom/sol_10_305_sparc/s0/Solaris_10/Tools
```

```
/cdrom/sol_10_305_sparc/s0/Solaris_10/Tools: 13737c(pmorph) 4712c(pmorph)
```


▼ 如何弹出可移除介质

1 确保未使用介质。

请记住，如果 shell 或应用程序正在访问介质的任何文件或目录，则表明“正在使用”该介质。如果无法确定是否已找到 CD 的所有用户（例如，隐藏在桌面工具之后的 shell 可能正在访问 CD），请使用 `fuser` 命令。请参见第 48 页中的[“如何确定可移除介质是否仍在使用”](#)。

2 弹出介质。

`# eject media`

例如，对于 CD，可以执行以下操作：

`# eject cdrom`

例如，对于 USB 内存条，可以执行以下操作：

`# eject rmdisk0`

例如，对于 PCMCIA 内存卡，可以执行以下操作：

`# eject pcmem0`

在远程系统上访问可移除介质（任务列表）

以下任务列表介绍了在远程系统上访问可移除介质所需的任务。

任务	说明	参考
1. 使本地介质可供远程系统使用。	将系统配置为共享其介质驱动器，以便使这些驱动器中的任何介质可供其他系统使用。	第 49 页中的 “如何使本地介质可供其他系统使用”
2. 访问远程系统上的可移除介质。	在本地系统上访问远程介质。	第 46 页中的 “如何访问可移除介质上的信息”

▼ 如何使本地介质可供其他系统使用

可以将系统配置为共享其介质驱动器，以便使这些驱动器中的任何介质可供其他系统使用。音乐 CD 除外。共享介质驱动器后，其他系统只需挂载这些驱动器即可访问其包含的介质。有关说明，请参见第 54 页中的[“如何访问远程系统上的可移除介质”](#)。

1 成为超级用户。

2 创建要共享的伪目录。

```
# mkdir /dummy
```

dummy 挂载点可以是任何目录名，例如 */dummy*。此目录将不包含任何文件。其唯一的用途是“唤醒”NFS 守护进程，以便此进程通知共享的介质驱动器。

3 将以下项添加到 */etc/dfs/dfstab* 文件。

```
share -F nfs -o ro /dummy
```

启动 NFS 服务器服务时，该服务会遇到该项，“唤醒”NFS 守护进程，然后通知共享的介质驱动器。

4 确定 NFS 服务器服务是否正在运行。

```
# svcs *nfs*
```

如果 NFS 服务器服务正在运行，则 *svcs* 命令将返回以下输出：

```
online          14:28:43 svc:/network/nfs/server:default
```

5 确定 NFS 服务器的状态，然后选择以下操作之一：

- 如果 NFS 服务器服务正在运行，则转到[步骤 7](#)。
- 如果 NFS 服务器服务没有运行，则转到下一步。

6 启动 NFS 服务器服务。

```
# svcadm enable network/nfs/server
```

验证 NFS 守护进程是否正在运行。

例如：

```
# svcs -p svc:/network/nfs/server:default
```

STATE	STIME	FMRI
online	Aug_30	svc:/network/nfs/server:default
	Aug_30	319 mountd
	Aug_30	323 nfsd

7 弹出当前位于驱动器中的任何介质。

```
# eject media
```

8 指定对 */etc/rmmount.conf* 文件的超级用户写入权限。

```
# chmod 644 /etc/rmmount.conf
```

9 将以下行添加到 /etc/rmmount.conf 文件：

```
share media*
```

这些行共享装入系统 CD-ROM 驱动器中的任何介质。但是，可以将共享限制为特定的 CD 或 CD 系列，如 share(1M) 中所述。

10 删除 /etc/rmmount.conf 文件的写入权限。

```
# chmod 444 /etc/rmmount.conf
```

此步骤可将文件恢复为其缺省权限。

11 装入介质。

现在装入的介质和所有后续介质均可供其他系统使用。请记住，在检验此任务之前，应等待驱动器上的指示灯停止闪烁。

要访问介质，远程用户必须按照第 54 页中的“如何访问远程系统上的可移除介质”中的说明，按名称挂载介质。

12 检验介质是否确实可供其他系统使用。

如果介质可用，则会显示其共享配置，还将显示共享的 dummy 目录。

```
# share
```

```
-                /dummy  ro    ""
-                /cdrom/sol_10_305_sparc/s5  ro    ""
-                /cdrom/sol_10_305_sparc/s4  ro    ""
-                /cdrom/sol_10_305_sparc/s3  ro    ""
-                /cdrom/sol_10_305_sparc/s2  ro    ""
-                /cdrom/sol_10_305_sparc/s1  ro    ""
-                /cdrom/sol_10_305_sparc/s0  ro    ""
```

示例 3-3 使本地 CD 可供其他系统使用

以下示例说明如何使任何本地 CD 可供网络中的其他系统使用。

```
# mkdir /dummy
```

```
vi /etc/dfs/dfstab
```

(Add the following line:)

```
# share -F nfs -o ro /dummy

# svcs *nfs*

# svcadm enable network/nfs/server

# svcs -p svc:/network/nfs/server:default

# eject cdrom0

# chmod 644 /etc/rmmount.conf

# vi /etc/rmmount.conf

(Add the following line:)

share cdrom*

# chmod 444 /etc/rmmount.conf

(Load a CD.)

# share

-          /dummy    ro    ""

-          /cdrom/sol_10_305_sparc/s5    ro    ""

-          /cdrom/sol_10_305_sparc/s4    ro    ""

-          /cdrom/sol_10_305_sparc/s3    ro    ""

-          /cdrom/sol_10_305_sparc/s2    ro    ""

-          /cdrom/sol_10_305_sparc/s1    ro    ""

-          /cdrom/sol_10_305_sparc/s0    ro    ""
```

示例 3-4 使本地软盘可供其他系统使用

以下示例说明如何使任何本地软盘可供网络中的其他系统使用。

```
# mkdir /dummy

# vi /etc/dfs/dfstab

(Add the following line:)
```

```

share -F nfs -o ro /dummy

# svcs *nfs*

# svcadm enable network/nfs/server

# svcs -p svc:/network/nfs/server:default

# eject floppy0

# chmod 644 /etc/rmmount.conf

# vi /etc/rmmount.conf

(Add the following line:)

share floppy*

# chmod 444 /etc/rmmount.conf

(Load a diskette.)

# volcheck -v

media was found

# share

-                /dummy    ro    ""

-                /floppy/myfiles  rw    ""

```

示例 3-5 使本地 PCMCIA 内存卡可供其他系统使用

以下示例说明如何使任何本地 PCMCIA 内存卡可供网络中的其他系统使用。

```

# mkdir /dummy

# vi /etc/dfs/dfstab

(Add the following line:)

# svcs *nfs*

# share -F nfs -o ro /dummy

# svcadm enable network/nfs/server

```

```
# svcs -p svc:/network/nfs/server:default

# eject pcmem0

# chmod 644 /etc/rmmount.conf

# vi /etc/rmmount.conf

(Add the following line:)

share floppy*

svc:/network/nfs/server:default# chmod 444 /etc/rmmount.conf

(Load a PCMCIA memory card.)

# volcheck -v

media was found

# share

-          /dummy    ro    ""

-          /pcmem/myfiles  rw    ""
```

▼ 如何访问远程系统上的可移除介质

通过手动将介质挂载到文件系统中，可以访问远程系统上的该介质。另外，远程系统还必须已按照第 49 页中的“[如何使本地介质可供其他系统使用](#)”中的说明共享其介质。

- 1 选择一个现有目录用作挂载点。或者，创建一个挂载点。

```
$ mkdir /directory
```

其中 */directory* 是创建作为远程系统 CD 挂载点的目录的名称。

- 2 查找要挂载的介质的名称。

```
$ showmount -e system-name
```

- 3 以超级用户身份挂载介质。

```
# mount -F nfs -o ro system-name:/media/media-name local-mount-point
```

system-name: 将挂载其介质的系统的名称。

media-name 要挂载的介质的名称。

local-mount-point 将挂载远程介质的本地目录。

4 以超级用户身份注销。

5 检验是否已挂载介质。

```
$ ls /media
```

示例 3-6 在远程系统上访问 CD

以下示例说明如何从使用 AutoFS 的远程系统 starbug 自动访问名为 sol_10_305_sparc 的远程 DVD。

```
$ showmount -e starbug

export list for starbug:

/dummy                (everyone)

/cdrom/sol_10_305_sparc/s5 (everyone)

/cdrom/sol_10_305_sparc/s4 (everyone)

/cdrom/sol_10_305_sparc/s3 (everyone)

/cdrom/sol_10_305_sparc/s2 (everyone)

/cdrom/sol_10_305_sparc/s1 (everyone)

/cdrom/sol_10_305_sparc/s0 (everyone)

$ ls /net/starbug/cdrom/

sol_10_305_sparc
```

示例 3-7 在其他系统上访问软盘

以下示例说明如何从使用 AutoFS 的远程系统 mars 自动访问 myfiles。

```
$ showmount -e mars

$ cd /net/mars

$ ls /floppy

floppy0    myfiles
```

示例 3-8 在远程系统上访问 PCMCIA 内存卡

以下示例说明如何从使用 AutoFS 的远程系统 mars 自动访问名为 myfiles 的 PCMCIA 内存卡。

```
$ showmount -e mars
```

```
$ cd /net/mars
```

```
$ ls /pcmem
```

```
pcmem0      myfiles
```


写入 CD 和 DVD（任务）

本章提供有关使用 `cdwr` 命令写入和复制数据 CD 和 DVD 以及音频 CD 的逐步说明。

- 第 60 页中的 “如何通过 RBAC 限制用户对可移除介质的访问”
- 第 60 页中的 “如何标识 CD 或 DVD 写入者”
- 第 61 页中的 “如何检查 CD 或 DVD 介质”
- 第 62 页中的 “如何为数据 CD 或 DVD 创建 ISO 9660 文件系统”
- 第 63 页中的 “如何创建多会话数据 CD”
- 第 66 页中的 “如何创建音频 CD”
- 第 67 页中的 “如何提取 CD 上的声道”
- 第 68 页中的 “如何复制 CD”
- 第 69 页中的 “如何删除 CD-RW 介质”

使用音频 CD 以及数据 CD 和 DVD

有关 DVD 支持的新信息，请参见第 27 页中的 “Solaris 10 发行版中可移除介质方面的新增功能”。

可以使用 `cdwr` 命令在 CD-R、CD-RW、DVD-RW 或 DVD+RW 介质设备上以 ISO 9660 格式（带有 Rock Ridge 或 Joliet 扩展）向 CD 和 DVD 中写入文件系统。

可以使用 `cdwr` 命令执行以下任务：

- 创建数据 CD 和 DVD。
- 创建音频 CD。
- 从音频 CD 提取音频数据。
- 复制 CD 和 DVD。
- 删除 CD-RW 介质。

从以下发行版开始提供了 `cdwr` 命令：

- Solaris 8 操作环境 1/01 的 Software Supplement CD
- 从 Solaris 9 发行版开始的部分 Solaris™ 发行版

有关推荐的 CD-R 或 CD-RW 设备的信息，请转到：

http://www.sun.com/io_technologies/ihvindex.html

CD/DVD 介质常用术语

本节定义了与 CD/DVD 介质相关的常用术语。

术语	说明
CD-R	可以写入一次并且之后只能读取的 CD 读取介质。
CD-RW	可擦写的 CD 可重写介质。CD-RW 介质只能由 CD-RW 设备读取。
DVD-RW	数字视频光盘（可重写）只能由 DVD-RW 驱动器读取。
DVD+RW	数字视频光盘（可刻录/可重写）驱动器可以写入 DVD-R 光盘（可以在大多数 DVD 播放机和计算机驱动器上播放）和 DVD-RW 可重写光盘。
ISO 9660	<p>ISO 是 Industry Standards Organization（行业标准组织）的首字母缩略词，该组织负责制定计算机存储格式的标准。</p> <p>ISO 9660 文件系统是标准的 CD 或 DVD 文件系统，通过它可以在任何主要计算机平台上读取同一 CD 或 DVD。该标准于 1988 年发布，是由名为 High Sierra（以内华达州的 High Sierra 旅馆命名）的行业工作组编写的。几乎所有配有 CD 或 DVD 驱动器的计算机都可以从 ISO 9660 文件系统读取文件。</p>
Joliet extensions（Joliet 扩展）	添加 Windows 文件系统信息。
Rock Ridge extensions（Rock Ridge 扩展）	<p>添加 UNIX 文件系统信息。（Rock Ridge 以电影 Blazing Saddles 中的城镇命名。）</p> <p>注 - 这些扩展不互斥。可以同时指定 <code>mkisofs -R</code> 和 <code>-j</code> 选项以便与这两种系统兼容。（有关详细信息，请参见 <code>mkisofs(1M)</code>。）</p>
MMC-compliant recorder（符合 MMC 标准的刻录机）	Multi Media Command（多媒体命令）的首字母缩略词，表示这些刻录机符合通用命令集。如果程序可以写入到一个符合 MMC 标准的刻录机，则应该可以写入到其他所有刻录机。
Red Book CDDA（红皮书 CDDA）	Compact Disc Digital Audio（光盘数字音频）的首字母缩略词，一种用于在光盘上存储数字音频的行业标准方法，也因术语“红皮书”格式而得名。正式的行业规范要求以 44.1 千赫 (kHz) 的抽样率在 16 位立体声中抽样一个或多个音频文件。

写入 CD 介质时常用的术语如下：

术语	说明
blanking（消隐）	从 CD-RW 介质中删除数据的过程。
mkisofs	用于在 CD 上创建 ISO 文件系统的命令。
session（会话）	具有引入和引出信息的完整磁道。
track（磁轨）	完整的数据或音频单元。

写入数据 CD 和 DVD 及音频 CD

写入 CD 或 DVD 的过程不能中断，并且需要连续不断的数据流。请考虑使用 `cdrw -S` 选项模拟向介质进行写入，以验证系统是否可以充足的速率提供数据以便写入 CD 或 DVD。

写入错误可能是由以下问题之一导致的：

- 介质无法处理驱动器速度。例如，一些介质仅经过 2x 或 4x 速度的认证。
- 系统正在运行过多的超负荷进程，从而导致写入进程无法运行。
- 网络拥塞导致读取映像时产生延迟，并且映像位于远程系统上。
- 源驱动器的速度比目标驱动器慢。

如果出现上述任一问题，可以使用 `cdrw -p` 选项降低设备的写入速度。

例如，以下命令说明如何模拟以 4x 速度进行写入：

```
$ cdrw -iS -p 4 image.iso
```

也可以通过 `cdrw -c` 选项使用规定的介质容量复制 80 分钟的 CD。否则，`cdrw` 命令会使用缺省值 74 分钟来复制音频 CD。

有关更多信息，请参见 `cdrw(1)`。

通过 RBAC 限制用户对可移除介质的访问

缺省情况下，从 Solaris 9 发行版开始所有用户都可以访问可移除介质。但是，可以通过基于角色的访问控制 (role-based access control, RBAC) 设置角色，限制用户对可移除介质的访问。通过为一组有限的用户指定角色，可以限制对可移除介质的访问。

有关使用角色的说明，请参见《系统管理指南：安全性服务》中的“基于角色的访问控制（概述）”。

▼ 如何通过 RBAC 限制用户对可移除介质的访问

- 1 成为超级用户或承担等效角色。
- 2 启动 Solaris Management Console。

```
$ /usr/sadm/bin/smc &
```

有关启动控制台的更多信息，请参见《系统管理指南：基本管理》中的“启动 Solaris Management Console”。
- 3 设置具有设备管理权限的角色。
有关更多信息，请参见《系统管理指南：安全性服务》中的第 9 章“使用基于角色的访问控制（任务）”。
- 4 将需要使用 `cdwr` 命令的用户添加到新创建的角色。
- 5 注释 `/etc/security/policy.conf` 文件中的以下行：

```
AUTHS_GRANTED=solaris.device.cdwr
```

如果不执行此步骤，则所有用户仍可以访问 `cdwr` 命令，而不仅仅是作为设备管理角色的成员。

修改此文件后，只有设备管理角色的成员才可以使用 `cdwr` 命令。将拒绝所有其他人的访问，并显示以下消息：

```
Authorization failed, Cannot access disks.
```

▼ 如何标识 CD 或 DVD 写入者

- 1 在系统上标识 CD 或 DVD 写入者。
例如：

```
$ cdwr -l
```

```
Looking for CD devices...
```

Node	Connected Device	Device type
cdrom0	YAMAHA CRW8824S	1.0d CD Reader/Writer

- 2 标识特定的 CD 或 DVD 写入者。

例如：

```
$ cdrw -a filename.wav -d cdrom2
```

- 3 确定介质是否为空或介质上是否存在目录。

例如：

```
$ cdrw -M
```

```
Device : YAMAHA    CRW8824S
```

```
Firmware : Rev. 1.00 (26/04/00)
```

```
Media is blank
```

```
%
```

▼ 如何检查 CD 或 DVD 介质

无论 `vold` 是否运行都可以使用 `cdrw` 命令。但是，必须具有超级用户或等效角色的访问权限，才能停止和启动 `vold` 守护进程。

- 1 将 CD 或 DVD 插入驱动器。
CD 或 DVD 可以是驱动器可读取的任何 CD 或 DVD。

- 2 通过列出驱动器来检查是否已正确连接该驱动器。

```
$ cdrw -l
```

```
Looking for CD devices...
```

Node	Connected Device	Device type
-----+-----+-----		
cdrom1	YAMAHA CRW8824S	1.0d CD Reader/Writer

- 3 （可选）如果列表中未显示该驱动器，请选择以下操作之一以便系统可识别该驱动器。

- 执行重新配置引导。

```
# touch /reconfigure
```

```
# init 6
```

- 添加驱动器，而不重新引导系统。

```
# drvconfig
```

```
# disks
```

然后重新启动 `vold`。

```
# svcadm disable volfs
```

```
# svcadm enable volfs
```

创建数据 CD 或 DVD

首先使用 `mkisofs` 命令准备数据，将文件和文件信息转换为可在 CD 或 DVD 上使用的 High Sierra 格式。

▼ 如何为数据 CD 或 DVD 创建 ISO 9660 文件系统

- 1 将空白 CD 或 DVD 插入驱动器。
- 2 在新 CD 或 DVD 上创建 ISO 9660 文件系统。

```
$ mkisofs -r /pathname > cd-file-system
```

`-r` 创建 Rock Ridge 信息并将文件拥有权重置为零。

`/pathname` 标识用于创建 ISO 9660 文件系统的路径名。

`> cd-file-system` 标识要放置在 CD 或 DVD 上的文件系统的名称。

- 3 将文件系统复制到 CD 或 DVD 上。

```
$ cdrw -i cd-file-system
```

`-i cd-file-system` 指定用于创建数据 CD 或 DVD 的映像文件。

示例 4-1 为数据 CD 或 DVD 创建 ISO 9660 文件系统

以下示例说明如何为数据 CD 或 DVD 创建 ISO 9660 文件系统。

```
$ mkisofs -r /home/dubs/ufs_dir > ufs_cd
```

```
Total extents actually written = 56
```

```
Total translation table size: 0
```

```
Total rockridge attributes bytes: 329
```

```
Total directory bytes: 0
```

```
Path table size(bytes): 10
```

```
Max brk space used 8000
```

```
56 extents written (0 Mb)
```

然后，将文件系统复制到 CD 或 DVD 上。

```
$ cdrw -i ufs_cd
```

```
Initializing device...done.
```

```
Writing track 1...done.
```

```
Finalizing (Can take several minutes)...done.
```

▼ 如何创建多会话数据 CD

此过程介绍如何在 CD 上放置多个会话。此过程包括将 infoA 和 infoB 目录复制到 CD 上的示例。

1 为第一个 CD 会话创建文件系统。

```
$ mkisofs -o infoA -r -V my_infoA /data/infoA
```

```
Total translation table size: 0
```

```
Total rockridge attributes bytes: 24507
```

```
Total directory bytes: 34816
```

```
Path table size(bytes): 98
```

```
Max brk space used 2e000
```

```
8929 extents written (17 Mb)
```

-o infoA 标识 ISO 文件系统的名称。

-r 创建 Rock Ridge 信息并将文件拥有权重置为零。

-V my_infoA 标识要由 vold 用作挂载点的卷标号。

/data/infoA 标识要创建的 ISO 映像目录。

2 将第一个会话的 ISO 文件系统复制到 CD 上。

```
$ cdrw -i0 infoA
```

```
Initializing device...done.
```

```
Writing track 1...done.
```

```
done.
```

```
Finalizing (Can take several minutes)...done.
```

-i *infoA* 标识要写入 CD 的映像文件的名称。

-0 使 CD 保持打开状态以便进行写入。

3 弹出 CD 后将其重新插入。

4 标识要在下一个写入会话中包括的 CD 介质的路径名。

```
$ eject -n
```

```
.
```

```
.
```

```
.
```

```
cdrom0 -> /vol/dev/rdisk/c2t4d0/my_infoA
```

请注意 /vol/dev/... 路径名。

5 标识 CD 上下一个可写入地址以便写入下一个会话。

```
% cdrw -M /cdrom
```

```
Device : YAMAHA     CRW8424S
```

```
Firmware : Rev. 1.0d (06/10/99)
```

```
Track No. |Type     |Start address
```

```
-----+-----+-----
```

```
1        |Audio    |0
```


2	Audio	33057
3	Data	60887
4	Data	68087
5	Data	75287
Leadout	Data	84218

Last session start address: 75287

Next writable address: 91118

记下 Next writable address 输出中的地址，以便在写入下一个会话时可以提供此地址。

6 为下一个 CD 会话创建下一个 ISO 文件系统，然后将其写入 CD。

```
$ mkisofs -o infoB -r -C 0,91118 -M /vol/dev/rdisk/c2t4d0/my_infoA
```

/data/infoB

Total translation table size: 0

Total rockridge attributes bytes: 16602

Total directory bytes: 22528

Path table size(bytes): 86

Max brk space used 20000

97196 extents written (189 Mb)

- | | |
|--|------------------------------|
| -o <i>infoB</i> | 标识 ISO 文件系统的名称。 |
| -r | 创建 Rock Ridge 信息并将文件拥有权重置为零。 |
| -C <i>0,91118</i> | 标识第一个会话的起始地址和下一个可写入地址。 |
| -M <i>/vol/dev/rdisk/c2t4d0/my_infoA</i> | 指定要合并的现有 ISO 映像的路径。 |
| /data/infoB | 标识要创建的 ISO 映像目录。 |

创建音频 CD

可以使用 `cdwr` 命令从单独的声道或者从 `.au` 和 `.wav` 文件创建音频 CD。

下表介绍了支持的音频格式：

格式	说明
sun	数据为红皮书 CDDA 格式的 Sun <code>.au</code> 文件
wav	数据为红皮书 CDDA 格式的 RIFF (<code>.wav</code>) 文件
cda	包含原始 CD 音频数据的 <code>.cda</code> 文件，该音频数据是按小尾数法字节顺序以 44.1 kHz 抽样率进行抽样的 16 位 PCM 立体声
aur	包含原始 CD 数据的 <code>.aur</code> 文件，该数据以大尾数法字节顺序排序

如果未指定任何音频格式，则 `cdwr` 命令会尝试根据文件扩展名确定音频文件的格式。将忽略扩展名中字符的大小写。

▼ 如何创建音频 CD

此过程介绍如何将音频文件复制到 CD 上。

- 1 将空白 CD 插入 CD-RW 驱动器。
- 2 转到包含音频文件的目录。
`$ cd /myaudiodir`
- 3 将音频文件复制到 CD 上。
`$ cdwr -a track1.wav track2.wav track3.wav`
`-a` 选项用于创建音频 CD。

示例 4-2 创建音频 CD

以下示例说明如何创建音频 CD。

```
$ cdwr -a bark.wav chirp.au meow.wav

Initializing device...done.

Writing track 1...done.
```

```
done.

Writing track 2...done.

Writing track 3...done.

done.

Finalizing (Can take several minutes)...done.
```

以下示例说明如何创建多会话音频 CD。写入第一个会话后，将弹出 CD。在下一个写入会话之前，需要重新插入 CD。

```
$ cdrw -a0 groucho.wav chico.au harpo.wav

Initializing device...done.

Writing track 1...done.

done.

Writing track 2...done.

Writing track 3...done.

done.

Finalizing (Can take several minutes)...done.

<Re-insert CD>

$ cdrw -a zeppo.au

Initializing device...done.

Writing track 1...done.

done.

Finalizing (Can take several minutes)...done.
```

▼ 如何提取 CD 上的声道

使用以下过程可从 CD 提取声道并将该声道复制到新 CD 上。

如果不使用 `cdrw -T` 选项指定音频文件类型，则 `cdrw` 命令会使用文件扩展名来确定音频文件类型。例如，`cdrw` 命令检测到此文件是 `.wav` 文件。

```
$ cdrw -x 1 testme.wav
```

1 将音频 CD 插入 CD-RW 驱动器。

2 提取声道。

```
$ cdrw -x -T audio-type 1 audio-file
```

-x 从音频 CD 提取音频数据。

T *audio-type* 标识要提取的音频文件的类型。支持的音频类型包括 sun、wav、cda 或 aur。

audio-file 标识要提取的声道。

3 将声道复制到新 CD。

```
$ cdrw -a audio-file
```

示例 4-3 提取和创建音频 CD

以下示例说明如何从音频 CD 提取第一个声道并将文件命名为 `song1.wav`。

```
$ cdrw -x -T wav 1 song1.wav
```

```
Extracting audio from track 1...done.
```

本示例说明如何将声道复制到音频 CD。

```
$ cdrw -a song1.wav
```

```
Initializing device...done.
```

```
Writing track 1...done.
```

```
Finalizing (Can take several minutes)...done.
```

▼ 如何复制 CD

此过程说明如何将音频 CD 中的所有声道提取到一个目录中，然后将其全部复制到空白 CD 上。

注 – 缺省情况下，`cdrw` 命令会将 CD 复制到 `/tmp` 目录中。复制时可能需要多达 700 MB 的空闲空间。如果 `/tmp` 目录中用于复制 CD 的空间不足，请使用 `-m` 选项指定替代目录。

1 将音频 CD 插入 CD-RW 驱动器。

2 为音频文件创建目录。

```
$ mkdir /music_dir
```

3 从音频 CD 提取声道。

```
$ cdrw -c -m music_dir
```

对于每个声道，将显示 `Extracting audio ...` 消息。

提取所有声道后，将弹出 CD。

4 插入空白 CD，然后按回车键。

提取声道后，将弹出音频 CD。系统将提示插入空白 CD。

示例 4-4 复制 CD

本示例说明如何将一张 CD 复制到另一张 CD。必须具有两个 CD-RW 设备才能执行此任务。

```
$ cdrw -c -s cdrom0 -d cdrom1
```

▼ 如何删除 CD-RW 介质

必须删除现有的 CD-RW 数据后，才能重写 CD。

► 通过选择以下操作之一，删除整个介质或仅删除 CD 上的最后一个会话：

- 仅删除最后一个会话。

```
$ cdrw -d cdrom0 -b session
```

使用 `-b session` 选项仅删除最后一个会话比使用 `-b all` 选项删除整个介质更快。即使仅在一个会话中使用 `cdrw` 命令创建数据或音频 CD，仍可以使用 `-b session` 选项。

- 删除整个介质。

```
$ cdrw -d cdrom0 -b all
```


管理设备（概述/任务）

本章提供有关在 Solaris 发行版中管理外围设备（如磁盘、CD-ROM 和磁带设备）的概述信息和逐步说明。

以下是本章中概述信息的列表。

- 第 71 页中的 “Solaris 10 6/06 发行版在设备管理方面的新增功能”
- 第 73 页中的 “Solaris 10 发行版在设备管理方面的新增功能”
- 第 74 页中的 “有关设备管理任务的参考信息”
- 第 75 页中的 “关于设备驱动程序”
- 第 75 页中的 “设备的自动配置”
- 第 77 页中的 “显示设备配置信息”

以下是本章中的逐步说明列表。

- 第 77 页中的 “如何显示系统配置信息”
- 第 85 页中的 “如何添加设备驱动程序”
- 第 84 页中的 “如何添加外围设备”

有关访问外围设备的信息，请参见第 10 章。

Solaris 发行版中的设备管理通常涉及在系统中添加和删除外围设备，其中还可能涉及添加支持设备的第三方设备驱动程序，以及显示系统配置信息。

Solaris 10 6/06 发行版在设备管理方面的新增功能

本节提供有关 Solaris 10 6/06 发行版在设备管理方面新增功能的信息。

x86: 支持 PCI Express (PCIe)

Solaris 10 6/06：此 Solaris 发行版可提供对 PCI Express (PCIe) 互连的支持，此功能旨在将外围设备连接至桌面、企业、移动、通信和嵌入式应用程序。

PCIe 互连是行业标准的高性能串行 I/O 总线。有关 PCIe 技术的详细信息，请访问以下站点：

<http://www.pcisig.com>

此 Solaris 发行版中的 PCIe 软件提供了以下功能：

- 支持扩展的 PCIe 配置空间
- 支持 PCIe 基准线错误处理和 MSI 中断
- PCIe 设备的经修改的 IEEE-1275 属性
- 通过增强 `cfgadm` 命令的 `cfgadm_pci` 组成部分以支持 PCIe 热插拔（本机和基于 ACPI）
- 基于 ATTN 按钮使用情况的 PCIe 外围设备自动配置

热插拔 PCIe 外围设备与 PCI 外围设备的管理模型相同，也是使用 `cfgadm` 命令。

请检查硬件平台指南，确保系统提供 PCIe 以及 PCIe 热插拔支持。此外，请认真查阅有关以物理方式在系统中插入或移除适配器的说明以及设备自动配置的语义（如果适用）。

有关针对 PCIe 外围设备使用 `cfgadm` 命令的信息，请参见第 103 页中的“使用 `cfgadm` 命令执行 PCI 或 PCIe 热插拔（任务列表）”。

USB 和 1394 (FireWire) 支持增强功能

Solaris 10 6/06：在此 Solaris 发行版中，不可移除的 USB 存储设备和 1394 海量存储设备均标识为驱动程序级别的可热插拔设备。这种新行为意味着可以在不重新引导系统的情况下连接这些设备或断开其连接，并可在非交互情况下自动配置或取消配置这些设备。这些更改是在内核级别进行的，因此不会影响这些设备的使用。例如，挂载和取消挂载这些设备的工作仍然由 `vol` 控制。

此外，还可以使用 `format` 实用程序来访问和标记不可移除的 USB 设备和 1394 海量存储设备。但是，在 `/kernel/drv/scsa2usb.conf` 文件中将 `remvalue` 设置为 `true` 可以覆盖这些设备新的可热插拔行为。将此参数设置为 `true` 意味着将该设备视为驱动程序级别的可移除介质设备，前提是该行为是首选行为。

有关使用这些设备的更多信息，请参见 `scsa1394(7D)` 和第 138 页中的“使用 USB 海量存储设备（任务列表）”。

Solaris 10 发行版在设备管理方面新增功能

本节提供有关 Solaris 10 发行版在设备管理方面新增功能的信息。

USB 设备增强功能

有关 USB 设备新增功能的信息，请参见第 125 页中的“[Solaris 10 发行版在 USB 设备方面的新增功能](#)”

x86 系统的 1394 (FireWire) 和海量存储支持

在此 Solaris 发行版中，1394 OpenHCI 主机控制器驱动程序已进行了更新，从而提供对 x86 系统的支持。以前，仅有 SPARC 系统提供 1394 (FireWire) 支持。

IEEE 1394 也是因 Apple Computer 的商标名 Firewire 而得名。Sony 的 1394 商标名为 i.LINK。

1394 是一种行业标准串行总线，该总线可支持的数据速率为 100 Mbit/sec、200 Mbit/sec 或 400 Mbit/sec。由于该总线具有高带宽和同步（准时）功能，因此非常适用于处理用户电子设备（如摄像机）中的数据。

有关更多信息，请参见 hci1394(7D)。

在此 Solaris 发行版中，添加了 `scsa1394` 驱动程序，以支持符合串行总线协议 2 (Serial Bus Portocol 2, SBP-2) 规范的 1394 海量存储设备。该驱动程序支持总线供电和自供电两种 1394 海量存储设备。以前仅支持 1394 摄像机。

1394 海量存储设备被视为可移除介质设备。可以使用 `rmformat` 命令对 1394 海量存储设备进行格式化。1394 海量存储设备的使用方法与 USB 海量存储设备相同。这意味着您可以挂载、弹出、热拆卸和热插入 1394 海量存储设备。有关使用这些设备的更多信息，请参见 `scsa1394(7D)` 和 [第 8 章](#)。

设备文件系统 (devfs)

此 Solaris 发行版中 `devfs` 文件系统用于管理设备。请继续通过 `/dev` 目录中的各项来访问所有设备，这些项是指向 `/devices` 目录中各项的符号链接。`/devices` 目录的内容现在由 `devfs` 文件系统控制。`/devices` 目录中的各项可动态表示系统上可访问设备的当前状态，并且无需进行管理。

`devfs` 文件系统提供了下列增强功能：

- `/devices` 目录中进行的操作将导致附加设备项。未使用的设备条目被分离开来。
- 提高了系统启动性能，因为仅附加了引导系统所需的设备项。访问设备时将添加新的设备条目。

有关更多信息，请参见 `devfs(7FS)` 手册页。

光纤通道设备的电源管理

许多以前的 Solaris 发行版中都提供了 Sun 系统的电源管理。例如，缺省情况下，以下系统中的内部驱动器会进行电源管理：

- SunBlade 1000 或 2000
- SunBlade 100 或 150
- SunBlade 2500 或 1500

`/etc/power.conf` 文件中的缺省设置确保符合能源之星标准，并且完全支持这些系统的电源管理。

以下适配器用于连接外部光纤通道存储设备：

- Sun StorEdge PCI 双光纤通道主机适配器
- Sun StorEdge PCI 单光纤通道网络适配器

如果将上述适配器与 Sun 系统结合使用来连接外部光纤通道存储设备，则缺省情况下，外部存储设备也会进行电源管理。

在以下情况下，应禁用电源管理：

- 系统中的光纤通道连接的磁盘连接至存储区域网络 (storage area network, SAN)
- 系统中的光纤通道连接的磁盘用在多启动器配置中，如用于 SunCluster 软件
- 系统在光纤通道接口上使用 IP（请参见 `fcip(7D)`）

如果多个 Solaris 系统可能共享相同的设备（如上述情况下），则不应启用电源管理。

按以下方式更改 `/etc/power.conf` 文件中的 `autopm` 关键字，可以对系统禁用电源管理：

```
autopm          disable
```

然后，通过运行 `pmconfig` 命令或重新引导系统，重新配置电源管理。

有关更多信息，请参见 `power.conf(4)` 和 `pmconfig(1M)`。

有关设备管理任务的参考信息

下表介绍了有关热插拔设备以及添加串行设备（如打印机和调制解调器）和外围设备（如磁盘、CD-ROM 或磁带设备）的逐步说明。

表 5-1 有关添加设备的参考信息

设备管理任务	更多信息
添加不可热插拔的磁盘。	第 13 章 或 第 14 章
热插拔 SCSI 或 PCI 设备。	第 91 页中的 “使用 <code>cfgadm</code> 命令执行 SCSI 热插拔” 或 第 104 页中的 “使用 <code>cfgadm</code> 命令执行 PCI 或 PCIe 热插拔”
热插拔 USB 设备。	第 138 页中的 “使用 USB 海量存储设备（任务列表）”
添加 CD-ROM 或磁带设备。	第 84 页中的 “如何添加外围设备”
添加调制解调器。	《系统管理指南：高级管理》中的第 8 章 “管理终端和调制解调器（概述）”
添加打印机。	《系统管理指南：高级管理》中的第 1 章 “管理打印服务（概述）”
保护设备。	《系统管理指南：安全性服务》中的第 4 章 “控制对设备的访问（任务）”

关于设备驱动程序

一台计算机通常会使用大量外围设备和海量存储设备。例如，您的系统可能装有磁盘驱动器、键盘和鼠标，以及某种磁性备份介质。其他常用的设备包括以下各项：

- CD-ROM 驱动器
- 打印机和绘图仪
- 光笔
- 触控式屏幕
- 数字化仪
- 图形输入板和笔尖对

Solaris 软件并不直接与所有这些设备进行通信。每种类型的设备都要求不同的数据格式、协议和传输率。

设备驱动程序是一种低级程序，可用于使操作系统与某个特定的硬件进行通信。驱动程序可用作操作系统对该硬件的“解释程序”。

设备的自动配置

内核包含一个小的通用核心，其中带有一个平台特定的组件和一组模块。在此 Solaris 发行版中，内核是自动配置的。

内核模块是用于在系统上执行特定任务的一个硬件组件或软件组件。**可装入内核模块**的一个示例是访问设备时装入的设备驱动程序。

与平台无关的内核为 `/kernel/genunix`。平台特定的组件为 `/platform/'uname -m'/kernel/unix`。

内核模块在下表中介绍。

表 5-2 Solaris 内核模块的说明

位置	目录内容
<code>/platform/'uname -m'/kernel</code>	平台特定的内核组件
<code>/kernel</code>	引导系统所需的通用于所有平台的内核组件
<code>/usr/kernel</code>	特定指令集内通用于所有平台的内核组件

系统可确定在引导时与其连接的设备。然后，内核会对其自身进行动态配置，将所需的模块装入内存中。此时，在访问磁盘和磁带等设备时就会装入设备驱动程序。此过程称为**自动配置**，因为所有的内核模块在需要时都会自动装入。

通过修改 `/etc/system` 文件，可以自定义装入内核模块的方式。有关修改此文件的说明，请参见 `system(4)`。

自动配置的功能和优点

自动配置的优点如下：

- 由于模块会在需要时装入，因此可更有效地使用主内存。
- 向系统中添加新设备时，无需重新配置内核。
- 可以装入并测试驱动程序，而不必重新生成内核和重新引导系统。

向系统中添加新设备（和驱动程序）时会使用自动配置功能。此时，如果设备不是可热插拔的，则可能需要执行重新配置引导，以便系统可以识别新设备。有关热插拔设备的信息，请参见第 6 章。

不受支持的设备所需的内容

Solaris 发行版中包括支持大量标准设备所需的设备驱动程序。可以在 `/kernel/drv` 和 `/platform/'uname -m'/kernel/drv` 目录中找到这些驱动程序。

但是，如果购买了不受支持的设备，制造商应该提供正确安装、维护和管理设备所需的软件。

此软件至少应包括设备驱动程序及其关联的配置 `(.conf)` 文件。`.conf` 文件驻留在 `drv` 目录中。此软件可能还包括自定义维护和管理实用程序，因为该设备可能与 Solaris 实用程序不兼容。

有关不受支持的设备所需内容的更多信息，请与设备制造商联系。

显示设备配置信息

有三个命令可用于显示系统和设备的配置信息。

命令	说明	手册页
<code>prtconf</code>	显示系统配置信息，包括系统设备分层结构描述的内存总量和设备配置。通过此命令显示的输出取决于系统的类型。	<code>prtconf(1M)</code>
<code>sysdef</code>	显示设备配置信息，包括系统硬件、伪设备、可装入模块以及选定的内核参数。	<code>sysdef(1M)</code>
<code>dmesg</code>	显示系统诊断消息以及从上次系统重新引导以来与系统连接的设备的列表。	<code>dmesg(1M)</code>

有关用于标识系统上设备的设备名称的信息，请参见第 194 页中的“设备命名约定”。

driver not attached 消息

`prtconf` 和 `sysdef` 命令可能会显示以下与驱动程序有关的消息：

device, instance #number (driver not attached)

此消息并不总是表示驱动程序不可用于此设备。此消息表示**当前**没有向设备实例附加任何驱动程序，因为此节点中不存在任何设备或者设备未在使用中。驱动程序是在访问设备时自动装入的。如果设备未在使用中，则会取消挂载这些驱动程序。

▼ 如何显示系统配置信息

使用 `prtconf` 和 `sysdef` 命令的输出可确定与系统连接的磁盘、磁带和 CD-ROM 设备。这些命令的输出会在设备实例旁边显示 `driver not attached` 消息。由于这些设备始终受某个系统进程监视，因此 `driver not attached` 消息通常明确表示该设备实例中不存在任何设备。

使用 `sysdef` 命令可显示系统配置信息，其中包括伪设备、可装入模块以及选定的内核参数。

■ 显示系统和设备的配置信息。

- 显示与系统连接的所有设备。

例如，SunBlade 1000 上的以下 `prtconf -v` 输出确定了与系统连接的磁盘设备。详细的磁盘信息在 `ssd/fp` 驱动程序部分中的 `Device Minor Nodes` 一节介绍。

```
$ /usr/sbin/prtconf -v | more

.

.

.

Device Minor Nodes:

dev=(118,8)

dev_path=/pci@8,600000/SUNW,qlc@4/fp@0,0/ssd@w210000
2037bde864,0:a

spectype=blk type=minor

dev_link=/dev/dsk/c0t1d0s0

dev_path=/pci@8,600000/SUNW,qlc@4/fp@0,0/ssd@w210000
2037bde864,0:a,raw

spectype=chr type=minor

dev_link=/dev/rdisk/c0t1d0s0

dev=(118,9)

dev_path=/pci@8,600000/SUNW,qlc@4/fp@0,0/ssd@w210000
2037bde864,0:b

spectype=blk type=minor

dev_link=/dev/dsk/c0t1d0s1

dev_path=/pci@8,600000/SUNW,qlc@4/fp@0,0/ssd@w210000
2037bde864,0:b,raw

.

.

.
```

- 显示有关与系统连接的某个特定设备的信息。

例如，SunBlade 1000 上的以下 `prtconf` 输出显示了 `/dev/dsk/c0t1d0s0` 的 `ssd` 实例编号。

```
# prtconf -v /dev/dsk/c0t1d0s0
```

```
ssd, instance #1
```

- 仅显示与系统连接的设备。

```
# prtconf | grep -v not
```

- 显示设备使用信息。

例如，以下 `fuser` 命令显示了正在访问 `/dev/console` 设备的进程。

```
# fuser -d /dev/console
```

```
/dev/console:      346o      323o
```

```
#
```

示例 5-1 显示系统配置信息

基于 SPARC 的系统中会显示以下 `prtconf` 输出。

```
# prtconf
```

```
System Configuration:  Sun Microsystems  sun4u
```

```
Memory size: 512 Megabytes
```

```
System Peripherals (Software Nodes):
```

```
SUNW,Sun-Blade-1000
```

```
scsi_vhci, instance #0
```

```
packages (driver not attached)
```

```
SUNW,builtin-drivers (driver not attached)
```

```
deblocker (driver not attached)
```

```
disk-label (driver not attached)
```

```
terminal-emulator (driver not attached)
```

```
    obp-tftp (driver not attached)

    dropins (driver not attached)

    kbd-translator (driver not attached)

    ufs-file-system (driver not attached)

chosen (driver not attached)

openprom (driver not attached)

    client-services (driver not attached)

options, instance #0

aliases (driver not attached)

memory (driver not attached)

virtual-memory (driver not attached)

SUNW,UltraSPARC-III, instance #0

memory-controller, instance #0

SUNW,UltraSPARC-III, instance #1

memory-controller, instance #1

pci, instance #0

    ebus, instance #0

        flashprom (driver not attached)

        bbc (driver not attached)

        ppm, instance #0

        i2c, instance #0

            dimm-fru, instance #0

            dimm-fru, instance #1

            dimm-fru, instance #2
```



```
    dimm-fru, instance #3

    nvram, instance #4

    idprom (driver not attached)

i2c, instance #1

    cpu-fru, instance #5

    temperature, instance #0

    cpu-fru, instance #6

    temperature, instance #1

    fan-control, instance #0

    motherboard-fru, instance #7

    i2c-bridge (driver not attached)

beep, instance #0

rtc, instance #0

gpio (driver not attached)

pmc (driver not attached)

floppy (driver not attached)

parallel (driver not attached)

serial, instance #0

network, instance #0

firewire, instance #0

usb, instance #0

scsi (driver not attached)

    disk (driver not attached)

    tape (driver not attached)
```

```
scsi (driver not attached)

    disk (driver not attached)

    tape (driver not attached)

pci, instance #1

    SUNW,qlc, instance #0

        fp (driver not attached)

            disk (driver not attached)

        fp, instance #1

            ssd, instance #1

            ssd, instance #0 (driver not attached)

            ssd, instance #2 (driver not attached)

            ssd, instance #3 (driver not attached)

            ssd, instance #4 (driver not attached)

            ssd, instance #5 (driver not attached)

            ssd, instance #6 (driver not attached)

    upa, instance #0

        SUNW,ffb, instance #0 (driver not attached)

    ppm, instance #0

    pseudo, instance #0
```

基于 x86 的系统中会显示以下 sysdef 输出。

```
# sysdef

* Hostid

*

29f10b4d
```

```
*  
  
* i86pc Configuration  
  
*  
  
*  
  
* Devices  
  
*  
  
+boot (driver not attached)  
  
memory (driver not attached)  
  
aliases (driver not attached)  
  
chosen (driver not attached)  
  
i86pc-memory (driver not attached)  
  
i86pc-mmu (driver not attached)  
  
openprom (driver not attached)  
  
options, instance #0  
  
packages (driver not attached)  
  
delayed-writes (driver not attached)  
  
itu-props (driver not attached)  
  
isa, instance #0  
  
    motherboard (driver not attached)  
  
    pnpADP,1542, instance #0  
  
    asy, instance #0  
  
    asy, instance #1  
  
    lp, instance #0 (driver not attached)  
  
    fdc, instance #0
```

```
fd, instance #0

fd, instance #1 (driver not attached)

kd (driver not attached)

kdmouse (driver not attached)

.

.

.
```

向系统添加外围设备

添加不可热插拔的新外围设备通常涉及以下操作：

- 关闭系统
- 将设备连接至系统
- 重新引导系统

可以参阅第 84 页中的“如何添加外围设备”向系统添加以下不可热插拔的设备：

- CD-ROM
- 辅助磁盘驱动器
- 磁带机
- SBUS 卡

在某些情况下，为支持新设备，可能必须添加第三方设备驱动程序。

有关热插拔设备的信息，请参见第 6 章。

▼ 如何添加外围设备

- 1 成为超级用户。
- 2 （可选的）如果需要添加设备驱动程序以支持设备，请完成过程第 85 页中的“如何添加设备驱动程序”。
- 3 创建 `/reconfigure` 文件。

```
# touch /reconfigure
```

`/reconfigure` 文件将导致 Solaris 软件在下次打开或引导系统时检查所有新近安装的设备是否存在。

4 关闭系统。

```
# shutdown -i0 -g30 -y
```

- i0 使系统进入 0 初始状态，即适用于关闭系统电源以添加和删除设备的状态。
- g30 在 30 秒内关闭系统。缺省值为 60 秒。
- y 在没有用户介入的情况下，继续关闭系统。否则，系统会提示您继续执行关机过程。

5 在系统关闭后，选择以下方法之一关闭系统电源：

- 对于 SPARC 平台，如果显示 ok 提示，则可以安全关闭电源。
- 对于 x86 平台，如果显示 type any key to continue 提示，则可以安全关闭电源。

6 关闭所有外围设备的电源。

有关所有外围设备的电源开关的位置，请参阅外围设备附带的硬件安装指南。

7 安装外围设备，确保要添加的设备与系统上的其他设备具有不同的目标号码。

通常，磁盘背面有一个小开关，用于选择目标号码。

有关安装和连接设备的信息，请参阅外围设备附带的硬件安装指南。

8 打开系统电源。

系统将引导至多用户模式，并会显示登录提示。

9 通过尝试访问外围设备，检验是否已添加了该设备。

有关访问该设备的信息，请参见第 10 章。

▼ 如何添加设备驱动程序

此过程假定已将设备添加到系统中。否则，请参见第 76 页中的“不受支持的设备所需的内容”。

1 成为超级用户。

2 将磁带、软盘或 CD-ROM 放入驱动器中。

3 安装驱动程序。

```
# pkgadd [-d] device package-name
```

- d device 用于标识包含软件包的设备路径名。
- package-name 用于标识包含设备驱动程序的软件包名称。

4 检验软件包是否已正确添加。

```
# pkgchk package-name
```

```
#
```

如果正确安装了软件包，则系统提示不会返回任何响应。

示例 5-2 添加设备驱动程序

以下示例说明如何安装和检验名为 `XYZdrv` 的软件包。

```
# pkgadd XYZdrv
```

```
(licensing messages displayed)
```

```
.
```

```
.
```

```
.
```

```
Installing XYZ Company driver as <XYZdrv>
```

```
.
```

```
.
```

```
.
```

```
Installation of <XYZdrv> was successful.
```

```
# pkgchk XYZdrv
```

```
#
```

动态配置设备（任务）

本章提供有关在 Solaris OS 中动态配置设备的说明。如果系统组件支持热插拔，则在系统仍然处于运行状态时，可在 Solaris OS 中添加、移除或更换设备。如果系统组件不支持热插拔，则可以重新引导系统以重新配置设备。

有关与动态配置设备关联的过程信息，请参见以下内容：

- 第 90 页中的“使用 `cfgadm` 命令执行 SCSI 热插拔（任务列表）”
- 第 103 页中的“使用 `cfgadm` 命令执行 PCI 或 PCIe 热插拔（任务列表）”
- 第 113 页中的“应用程序开发者 RCM 脚本（任务列表）”
- 第 113 页中的“系统管理员 RCM 脚本（任务列表）”

有关使用 `cfgadm` 命令热插拔 USB 设备的信息，请参见第 168 页中的“使用 `cfgadm` 命令热插拔 USB 设备”。

有关访问设备的信息，请参见第 10 章。

动态重新配置和热插拔

热插拔是指在系统运行的同时以物理方式添加、移除或更换系统组件的能力。**动态重新配置**是指热插拔系统组件的能力。此术语还指在系统中四处移动系统资源（硬件和软件），或以某种方式禁用这些系统资源而不以物理方式将其从系统中移除的一般能力。

一般情况下，可以热插拔以下总线类型：

- USB
- 光纤通道
- 1394
- ATA
- SCSI

此外，还可以使用 `cfgadm` 命令热插拔以下设备：

- SPARC 和 x86 平台上的 USB 设备
- SPARC 和 x86 平台上的 SCSI 设备
- SPARC 和 x86 平台上的 PCI 设备
- x86 平台上的 PCIe 设备

cfgadm 命令包括以下功能：

- 显示系统组件状态
- 测试系统组件
- 更改组件配置
- 显示配置帮助消息

使用 `cfgadm` 命令重新配置系统组件的优点是可在系统运行的同时添加、移除或更换组件。另一优点是 `cfgadm` 命令可引导逐步完成添加、移除或更换系统组件所需执行的步骤。

有关热插拔组件的逐步说明，请参见以下内容：

- [第 91 页中的“使用 `cfgadm` 命令执行 SCSI 热插拔”](#)
- [第 104 页中的“使用 `cfgadm` 命令执行 PCI 或 PCIe 热插拔”](#)
- `cfgadm(1M)`

注 – 并非所有 SCSI 和 PCI 控制器都支持使用 `cfgadm` 命令进行热插拔。

作为 Sun 的高可用性策略的一部分，动态重新配置预计会与其他分层产品（如替代路径设置软件或故障转移软件）结合使用。这两种产品都可在出现设备故障的情况下提供容错功能。

如果没有任何高可用性软件，则可通过手动停止相应的应用程序，卸载非关键的文件系统，然后继续执行添加或删除操作来更换出现故障的设备。

注 – 某些系统同时具有可热插拔和不可热插拔的插槽。有关在特定硬件配置（如在企业级系统上）中热插拔设备的信息，请参阅硬件配置文档。

附加点

`cfgadm` 命令可以显示有关**附加点**的信息，附加点是指系统中可以执行动态重新配置操作的位置。

附加点由以下几项组成：

- **插卡**，表示可在系统中配置的硬件组件
- **插座**，是指插入插卡的位置

附加点由逻辑附加点 ID 和物理附加点 ID (Ap_Id) 表示。物理 Ap_Id 是附加点的物理路径名。逻辑 Ap_Id 是物理 Ap_Id 的用户友好替代项。有关 Ap_Id 的更多信息，请参阅 `cfgadm(1M)`。

SCSI 主机总线适配器 (Host Bus Adapter, HBA) 或 SCSI 控制器的逻辑 Ap_Id 通常由控制器编号表示，如 `c0`。

如果未向 SCSI HBA 指定控制器编号，则将提供内部生成的唯一标识符。以下是 SCSI 控制器的唯一标识符的示例：

```
fas1:scsi
```

SCSI 设备的逻辑 Ap_Id 通常具有以下格式：

HBA-logical-apid::device-identifier

在以下示例中，`c0` 是 SCSI HBA 的逻辑 Ap_Id：

```
c0::dsk/c0t3d0
```

设备标识符通常由 `/dev` 目录中的设备的逻辑设备名称派生而来。例如，逻辑设备名称为 `/dev/rmt/1` 的磁带设备具有以下逻辑 Ap_Id：

```
c0::rmt/1
```

如果 SCSI 设备的逻辑 Ap_Id 无法从 `/dev` 目录中的逻辑名称派生而来，则将提供内部生成的唯一标识符。以下是 `/dev/rmt/1` 磁带设备的标识符的示例：

```
c0::st4
```

有关 SCSI Ap_Id 的更多信息，请参阅 `cfgadm_scsi(1M)`。

`cfgadm` 命令按照一组通用状态（如已配置和未配置）和操作（如连接、配置、取消配置等）来表示所有资源和动态重新配置操作。有关这些通用状态和操作的更多信息，请参见 `cfgadm(1M)`。

下表列出了 SCSI HBA 附加点的插座和插卡的状态。

插座状态	说明	插卡状态	说明
empty	N/A（对于 SCSI HBA）	configured	在总线上配置了一个或多个设备
disconnected	总线处于静止状态	unconfigured	未配置任何设备
connected	总线处于活动状态		

下表列出了 SCSI 设备附加点的插座和插卡的状态。

插座状态	说明	插卡状态	说明
empty	N/A（对于 SCSI 设备）	configured	设备已配置
disconnected	总线处于静止状态	unconfigured	设备未配置
connected	总线处于活动状态		

如果特殊硬件未另行指示其他状态，则 SCSI 附加点的状态是未知的。有关显示 SCSI 组件信息的说明，请参见第 91 页中的[“如何显示有关 SCSI 设备的信息”](#)。

拆离 PCI 或 PCIe 适配卡

如果设备驱动程序支持热插拔，则可以移除承载非重要系统资源的 PCI 适配卡。如果 PCI 适配卡是重要的系统资源，则不可将其拆离。要使 PCI 适配卡可拆离，必须满足以下条件：

- 设备驱动程序必须支持热插拔。
- 必须可通过替代路径访问关键资源。

例如，如果系统中仅安装了一块以太网卡，则拆离该以太网卡必然会断开网络连接。此拆离操作需要其他分层软件支持，以使网络连接保持活动状态。

连接 PCI 或 PCIe 适配卡

只要满足以下条件，便可以向系统中添加 PCI 适配卡。

- 具有可用插槽。
- 设备驱动程序支持对此适配卡进行热插拔。

有关添加或移除 PCI 适配卡的逐步说明，请参见第 104 页中的[“使用 `cfgadm` 命令执行 PCI 或 PCIe 热插拔”](#)。

使用 `cfgadm` 命令执行 SCSI 热插拔（任务列表）

任务	说明	参考
显示有关 SCSI 设备的信息。	显示有关 SCSI 控制器和设备的信息。	第 91 页中的“如何显示有关 SCSI 设备的信息”
取消配置 SCSI 控制器。	取消配置 SCSI 控制器。	第 92 页中的“如何取消配置 SCSI 控制器”

任务	说明	参考
配置 SCSI 控制器。	配置以前取消配置的 SCSI 控制器。	第 93 页中的 “如何配置 SCSI 控制器”
配置 SCSI 设备。	配置特定的 SCSI 设备。	第 94 页中的 “如何配置 SCSI 设备”
断开 SCSI 控制器的连接。	断开特定 SCSI 控制器的连接。	第 95 页中的 “如何断开 SCSI 控制器连接”
连接 SCSI 控制器。	连接以前断开的特定 SCSI 控制器。	第 96 页中的 “SPARC: 如何连接 SCSI 控制器”
将 SCSI 设备添加到 SCSI 总线中。	将特定的 SCSI 设备添加到 SCSI 总线中。	第 97 页中的 “SPARC: 如何将 SCSI 设备添加到 SCSI 总线中”
更换 SCSI 控制器上的相同设备。	将 SCSI 总线上的设备更换为同一类型的其他设备。	第 98 页中的 “SPARC: 如何更换 SCSI 控制器上的相同设备”
移除 SCSI 设备。	从系统中移除 SCSI 设备。	第 100 页中的 “SPARC: 如何移除 SCSI 设备”
对 SCSI 配置问题进行疑难解答。	解析失败的 SCSI 取消配置操作。	第 103 页中的 “如何解析失败的 SCSI 取消配置操作”

使用 `cfgadm` 命令执行 SCSI 热插拔

本节介绍可使用 `cfgadm` 命令执行的各种 SCSI 热插拔过程。

注 – 一般情况下，SCSI 框架支持 SCSI 设备的热插拔。但是，您应该查询硬件文档，以确认 SCSI 设备是否支持热插拔。

这些过程以特定设备为例，说明如何使用 `cfgadm` 命令来热插拔 SCSI 组件。您提供的以及 `cfgadm` 命令显示的设备信息取决于系统配置。

▼ 如何显示有关 SCSI 设备的信息

以下过程在使用 `cfgadm` 命令显示设备类型配置信息的示例中使用 SCSI 控制器 `c0` 和 `c1` 以及与其连接的设备。

注 – 如果 `cfgadm` 命令不支持 SCSI 设备，则该设备不会显示在 `cfgadm` 命令输出中。

- 1 成为超级用户。

2 显示有关系统上的附加点的信息。

```
# cfmadm -l
```

Ap_Id	Type	Receptacle	Occupant	Condition
c0	scsi-bus	connected	configured	unknown
c1	scsi-bus	connected	configured	unknown

在本示例中，c0 和 c1 表示两个 SCSI 控制器。

3 显示有关系统的 SCSI 控制器及其连接设备的信息。

```
# cfmadm -al
```

Ap_Id	Type	Receptacle	Occupant	Condition
c0	scsi-bus	connected	configured	unknown
c0::dsk/c0t0d0	disk	connected	configured	unknown
c0::rmt/0	tape	connected	configured	unknown
c1	scsi-bus	connected	configured	unknown
c1::dsk/c1t3d0	disk	connected	configured	unknown
c1::dsk/c1t4d0	unavailable	connected	unconfigured	unknown

注 - cfmadm -l 命令显示有关 SCSI HBA 而不是 SCSI 设备的信息。使用 cfmadm -al 命令可显示有关 SCSI 设备（如磁盘和磁带）的信息。

▼ 如何取消配置 SCSI 控制器

以下过程在取消配置 SCSI 控制器的示例中使用 SCSI 控制器 c1。

1 成为超级用户。

2 取消配置 SCSI 控制器。

```
# cfmadm -c unconfigure c1
```

3 验证是否已取消配置 SCSI 控制器。

```
# cfmadm -al
```

Ap_Id	Type	Receptacle	Occupant	Condition
-------	------	------------	----------	-----------

c0	scsi-bus	connected	configured	unknown
c0::dsk/c0t0d0	disk	connected	configured	unknown
c0::rmt/0	tape	connected	configured	unknown
c1	scsi-bus	connected	unconfigured	unknown

请注意，c1 的 Occupant 列指定 unconfigured，表明 SCSI 总线没有已配置的插卡。
如果取消配置操作失败，请参见第 103 页中的[“如何解析失败的 SCSI 取消配置操作”](#)。

▼ 如何配置 SCSI 控制器

以下过程在配置 SCSI 控制器的示例中使用 SCSI 控制器 c1。

- 1 成为超级用户。
- 2 配置 SCSI 控制器。
- 3 验证 SCSI 控制器是否已配置。

```
# cfgadm -c configure c1  
  
# cfgadm -al
```

Ap_Id	Type	Receptacle	Occupant	Condition
c0	scsi-bus	connected	configured	unknown
c0::dsk/c0t0d0	disk	connected	configured	unknown
c0::rmt/0	tape	connected	configured	unknown
c1	scsi-bus	connected	configured	unknown
c1::dsk/c1t3d0	disk	connected	configured	unknown
c1::dsk/c1t4d0	unavailable	connected	unconfigured	unknown

前面的取消配置过程移除了 SCSI 总线上的所有设备。现在，所有设备都已配置回系统中。

▼ 如何配置 SCSI 设备

以下过程在配置 SCSI 设备的示例中使用 SCSI 磁盘 c1t4d0。

- 1 成为超级用户。
- 2 确定要配置的设备。

cfgadm -al

Ap_Id	Type	Receptacle	Occupant	Condition
c0	scsi-bus	connected	configured	unknown
c0::dsk/c0t0d0	disk	connected	configured	unknown
c0::rmt/0	tape	connected	configured	unknown
c1	scsi-bus	connected	configured	unknown
c1::dsk/c1t3d0	disk	connected	configured	unknown
c1::dsk/c1t4d0	unavailable	connected	unconfigured	unknown

- 3 配置 SCSI 设备。
- # **cfgadm -c configure c1::dsk/c1t4d0**

- 4 验证 SCSI 设备是否已配置。
- # **cfgadm -al**

Ap_Id	Type	Receptacle	Occupant	Condition
c0	scsi-bus	connected	configured	unknown
c0::dsk/c0t0d0	disk	connected	configured	unknown
c0::rmt/0	tape	connected	configured	unknown
c1	scsi-bus	connected	configured	unknown
c1::dsk/c1t3d0	disk	connected	configured	unknown
c1::dsk/c1t4d0	disk	connected	configured	unknown

▼ 如何断开 SCSI 控制器连接



注意 - 断开 SCSI 设备连接时必须非常谨慎，特别是在处理包含关键文件系统（如根 (/)、`usr`、`var` 和 `swap` 分区）的控制器任务时。动态重新配置软件无法检测到系统挂起可能导致的所有情况。使用此过程时应谨慎。

以下过程在断开 SCSI 设备连接的示例中使用 SCSI 控制器 `c1`。

- 1 成为超级用户。
- 2 验证断开设备连接之前是否已连接该设备。

```
# cfgadm -al
```

Ap_Id	Type	Receptacle	Occupant	Condition
c0	scsi-bus	connected	configured	unknown
c0::dsk/c0t0d0	disk	connected	configured	unknown
c0::rmt/0	tape	connected	configured	unknown
c1	scsi-bus	connected	configured	unknown
c1::dsk/c1t3d0	disk	connected	configured	unknown
c1::dsk/c1t4d0	disk	connected	configured	unknown

- 3 断开 SCSI 控制器的连接。

```
# cfgadm -c disconnect c1
```

```
WARNING: Disconnecting critical partitions may cause system hang.
```

```
Continue (yes/no)? y
```



注意 - 此命令可以暂停 SCSI 总线上的所有 I/O 活动，直到使用 `cfgadm -c connect` 命令为止。`cfgadm` 命令可执行一些基本检查，以防止断开关键分区的连接，但不能检测到所有情况。如果此命令使用不当，则可能会导致系统挂起并且可能需要重新引导系统。

- 4 验证 SCSI 总线是否已断开连接。

```
# cfgadm -al
```

Ap_Id	Type	Receptacle	Occupant	Condition
-------	------	------------	----------	-----------

c0	scsi-bus	connected	configured	unknown
c0::dsk/c0t0d0	disk	connected	configured	unknown
c0::rmt/0	tape	connected	configured	unknown
c1	unavailable	disconnected	configured	unknown
c1::dsk/c1t3d0	unavailable	disconnected	configured	unknown
c1::dsk/c1t4d0	unavailable	disconnected	configured	unknown

控制器及其连接的所有设备都将与系统断开连接。

▼ **SPARC: 如何连接 SCSI 控制器**

以下过程在连接 SCSI 控制器的示例中使用 SCSI 控制器 c1。

- 1 成为超级用户。
- 2 验证连接设备之前该设备是否已断开连接。

```
# cfmadm -al
```

Ap_Id	Type	Receptacle	Occupant	Condition
c0	scsi-bus	connected	configured	unknown
c0::dsk/c0t0d0	disk	connected	configured	unknown
c0::rmt/0	tape	connected	configured	unknown
c1	unavailable	disconnected	configured	unknown
c1::dsk/c1t3d0	unavailable	disconnected	configured	unknown
c1::dsk/c1t4d0	unavailable	disconnected	configured	unknown

- 3 连接 SCSI 控制器。
- 4 验证 SCSI 控制器是否已连接。

```
# cfmadm -al
```

Ap_Id	Type	Receptacle	Occupant	Condition
-------	------	------------	----------	-----------

c0	scsi-bus	connected	configured	unknown
c0::dsk/c0t0d0	disk	connected	configured	unknown
c0::rmt/0	tape	connected	configured	unknown
c1	scsi-bus	connected	configured	unknown
c1::dsk/c1t3d0	disk	connected	configured	unknown
c1::dsk/c1t4d0	disk	connected	configured	unknown

▼ **SPARC: 如何将 SCSI 设备添加到 SCSI 总线中**

在如何将 SCSI 设备添加到 SCSI 总线的示例中使用的是 SCSI 控制器 `c1`。

注 – 添加设备时，应指定与设备连接的 SCSI HBA（控制器）的 `Ap_Id`，而不是设备本身的 `Ap_Id`。

- 1 成为超级用户。
- 2 确定当前的 SCSI 配置。

`cfgadm -al`

Ap_Id	Type	Receptacle	Occupant	Condition
c0	scsi-bus	connected	configured	unknown
c0::dsk/c0t0d0	disk	connected	configured	unknown
c0::rmt/0	tape	connected	configured	unknown
c1	scsi-bus	connected	configured	unknown
c1::dsk/c1t3d0	disk	connected	configured	unknown

- 3 将 SCSI 设备添加到 SCSI 总线。

- a. 键入以下 `cfgadm` 命令。

例如：

`cfgadm -x insert_device c1`

Adding device to SCSI HBA: /devices/sbus@1f,0/SUNW,fas@1,8800000

This operation will suspend activity on SCSI bus: c1

- b. 在 Continue (yes/no)? 提示符下键入 **y** 以继续操作。
- ```
Continue (yes/no)? y
```
- SCSI bus quiesced successfully.
- It is now safe to proceed with hotplug operation.
- 正在执行热插拔操作的同时，SCSI 总线上的 I/O 活动会暂停。
- c. 连接设备，然后打开设备电源。
- d. 在 Enter y if operation is complete or n to abort (yes/no)? 提示符下键入 **y**。
- ```
Enter y if operation is complete or n to abort (yes/no)? y
```

4 验证是否已添加了设备。

```
# cfgadm -al
```

Ap_Id	Type	Receptacle	Occupant	Condition
c0	scsi-bus	connected	configured	unknown
c0::dsk/c0t0d0	disk	connected	configured	unknown
c0::rmt/0	tape	connected	configured	unknown
c1	scsi-bus	connected	configured	unknown
c1::dsk/c1t3d0	disk	connected	configured	unknown
c1::dsk/c1t4d0	disk	connected	configured	unknown

新磁盘已添加到控制器 c1 中。

▼ **SPARC: 如何更换 SCSI 控制器上的相同设备**

以下过程在 SCSI 控制器上更换相同设备的示例中使用 SCSI 磁盘 c1t4d0。

- 1 成为超级用户。
- 2 确定当前的 SCSI 配置。

```
# cfgadm -al
```

Ap_Id	Type	Receptacle	Occupant	Condition
c0	scsi-bus	connected	configured	unknown

c0::dsk/c0t0d0	disk	connected	configured	unknown
c0::rmt/0	tape	connected	configured	unknown
c1	scsi-bus	connected	configured	unknown
c1::dsk/c1t3d0	disk	connected	configured	unknown
c1::dsk/c1t4d0	disk	connected	configured	unknown

3 将 SCSI 总线上的设备更换为同一类型的其他设备。

a. 键入以下 cfmadm 命令。

例如：

```
# cfmadm -x replace_device c1::dsk/c1t4d0
```

```
Replacing SCSI device: /devices/sbus@1f,0/SUNW,fas@1,88000000/sd@4,0
```

```
This operation will suspend activity on SCSI bus: c1
```

b. 在 Continue (yes/no)? 提示符下键入 y 以继续操作。

正在执行热插拔操作的同时，SCSI 总线上的 I/O 活动会暂停。

```
Continue (yes/no)? y
```

```
SCSI bus quiesced successfully.
```

```
It is now safe to proceed with hotplug operation.
```

c. 关闭要移除的设备的电源，然后将其移除。

d. 添加替换设备。然后，关闭该设备的电源。

替换设备应该与要移除的设备类型相同并且具有同一地址（目标和 LUN）。

e. 在 Enter y if operation is complete or n to abort (yes/no)? 提示符下键入 y。

```
Enter y if operation is complete or n to abort (yes/no)? y
```

4 验证是否已更换了设备。

```
# cfmadm -al
```

Ap_Id	Type	Receptacle	Occupant	Condition
c0	scsi-bus	connected	configured	unknown
c0::dsk/c0t0d0	disk	connected	configured	unknown

c0::rmt/0	tape	connected	configured	unknown
c1	scsi-bus	connected	configured	unknown
c1::dsk/clt3d0	disk	connected	configured	unknown
c1::dsk/clt4d0	disk	connected	configured	unknown

▼ SPARC: 如何移除 SCSI 设备

以下过程在移除 SCSI 控制器上的设备示例中使用 SCSI 磁盘 c1t4d0。

- 1 成为超级用户。
- 2 确定当前的 SCSI 配置。

```
# cfgadm -al
```

Ap_Id	Type	Receptacle	Occupant	Condition
c0	scsi-bus	connected	configured	unknown
c0::dsk/c0t0d0	disk	connected	configured	unknown
c0::rmt/0	tape	connected	configured	unknown
c1	scsi-bus	connected	configured	unknown
c1::dsk/clt3d0	disk	connected	configured	unknown
c1::dsk/clt4d0	disk	connected	configured	unknown

- 3 从系统中移除 SCSI 设备。

- a. 键入以下 cfgadm 命令。

例如：

```
# cfgadm -x remove_device c1::dsk/clt4d0
```

```
Removing SCSI device: /devices/sbus@1f,0/SUNW,fas@1,8800000/sd@4,0
```

```
This operation will suspend activity on SCSI bus: c1
```

- b. 在 Continue (yes/no)? 提示符下键入 y 以继续操作。

```
Continue (yes/no)? y
```

SCSI bus quiesced successfully.

It is now safe to proceed with hotplug operation.

正在执行热插拔操作的同时，SCSI 总线上的 I/O 活动会暂停。

- c. 关闭要移除的设备的电源，然后将其移除。
- d. 在 Enter y if operation is complete or n to abort (yes/no)? 提示符下键入 y。
Enter y if operation is complete or n to abort (yes/no)? y

4 验证是否已从系统中移除了设备。

cfigadm -al

Ap_Id	Type	Receptacle	Occupant	Condition
c0	scsi-bus	connected	configured	unknown
c0::dsk/c0t0d0	disk	connected	configured	unknown
c0::rmt/0	tape	connected	configured	unknown
c1	scsi-bus	connected	configured	unknown
c1::dsk/clt3d0	disk	connected	configured	unknown

对 SCSI 配置问题进行疑难解答

本节提供有关 SCSI 配置问题的错误消息和可能的解决方案。有关对 SCSI 配置问题进行疑难解答的更多信息，请参见 cfigadm(1M)。

错误消息

cfigadm: Component system is busy, try again: failed to offline:

device-path

Resource Information

/dev/dsk/clt0d0s0 mounted filesystem "/file-system"

原因

尝试移除或更换已挂载文件系统的设备。

解决方案

取消挂载错误消息中列出的文件系统，并重试 cfigadm 操作。

错误消息

cfigadm: Component system is busy, try again: failed to offline:

<i>device-path</i>	
Resource	Information

/dev/dsk/ <i>device-name</i>	swap area

原因

如果使用 cfigadm 命令移除系统资源（如交换设备或专用转储设备），则在系统资源仍然处于活动状态时将显示一条类似的错误消息。

解决方案

取消配置指定设备上的交换区域，并重试 cfigadm 操作。

错误消息

cfigadm: Component system is busy, try again: failed to offline:

<i>device-path</i>	
Resource	Information

/dev/dsk/ <i>device-name</i>	dump device (swap)

原因

尝试移除或更换在交换区域中配置的转储设备。

解决方案

取消配置在交换区域中配置的转储设备，并重试 cfigadm 操作。

错误消息

cfigadm: Component system is busy, try again: failed to offline:

<i>device-path</i>	
Resource	Information

```
/dev/dsk/device-name    dump device (dedicated)
```

原因
尝试移除或更换专用转储设备。

解决方案
取消配置该专用转储设备，并重试 `cfgadm` 操作。

▼ 如何解析失败的 SCSI 取消配置操作

如果一个或多个目标设备繁忙并且 SCSI 取消配置操作失败，请使用此过程。否则，将来对此控制器和目标设备进行动态重新配置操作会失败，并会显示 `dr in progress` 消息。

- 1 成为超级用户。
- 2 重新配置控制器。

```
# cfgadm -c configure device-name
```

使用 `cfgadm` 命令执行 PCI 或 PCIe 热插拔（任务列表）

以下任务列表介绍了在系统上管理 PCI 或 PCIe 设备的任务。

任务	说明	参考
显示 PCI 插槽配置信息。	显示系统上的 PCI 可热插拔设备和插槽的状态。	第 104 页中的“如何显示 PCI 插槽配置信息”
移除 PCI 适配卡。	取消配置该卡，断开插槽的电源，然后从系统中移除该卡。	第 106 页中的“如何移除 PCI 适配卡”
添加 PCI 适配卡。	将适配卡插入可热插拔的插槽中，连接插槽的电源，然后配置该卡。	第 108 页中的“如何添加 PCI 适配卡”
对 PCI 配置问题进行疑难解答。	确定错误消息和可能的解决方案，以解析 PCI 配置问题。	第 110 页中的“对 PCI 配置问题进行疑难解答”

使用 cfgadm 命令执行 PCI 或 PCIe 热插拔

本节提供有关热插拔 x86 系统上的 PCI 或 PCIe 适配卡的逐步说明。

除了 `cfgadm` 命令以外，在热插拔操作过程中，`prtconf` 命令也非常有帮助。`prtconf` 命令可以显示与硬件有关的其他配置信息。

添加硬件后，可以使用 `prtconf` 命令来验证该硬件是否正确配置。例如，在配置操作后，使用 `prtconf -D` 命令可验证是否已向新安装的硬件设备附加了驱动程序。如果在配置硬件之前未将设备驱动程序添加到系统中，则可以使用 `add_drv` 命令手动添加该驱动程序。

有关更多信息，请参见 `prtconf(1M)` 和 `add_drv(1M)`。

在这些示例中，为了简便起见，仅列出了 PCI 附加点。系统上显示的附加点取决于系统配置。

PCIe LED 指示灯行为

您可能需要观察系统上的 LED 指示灯，以直观地了解有关插槽热插拔操作的状态。如果使用 PCI Express，则 LED 的行为符合 PCI Express 规范中定义的行为，否则该行为可能与平台相关。

有关具体的详细信息，请参阅平台指南。如果使用 PCI Express，则按“注意”按钮时，电源指示灯将会闪烁，表示状态转换开始。状态转换结束后，即会停止闪烁。

▼ 如何显示 PCI 插槽配置信息

此过程在 Solaris 10 6/06 发行版中已进行了更新，包括 PCIe 配置信息。

`cfgadm` 命令可显示系统上的 PCI 可热插拔设备和插槽的状态。有关更多信息，请参见 `cfgadm(1M)`。

- 1 成为超级用户。
- 2 显示 PCI 配置信息。
 - 显示 PCI 插槽配置信息。

例如：

```
# cfgadm

Ap_Id                Type                Receptacle  Occupant      Condition
```


pci1:hpc0_slot0	unknown	empty	unconfigured	unknown
pci1:hpc0_slot1	unknown	empty	unconfigured	unknown
pci1:hpc0_slot2	unknown	empty	unconfigured	unknown
pci1:hpc0_slot3	ethernet/hp	connected	configured	ok
pci1:hpc0_slot4	unknown	empty	unconfigured	unknown

- 显示特定的 PCI 设备信息。
例如：

```
# cfgadm -s "cols=ap_id:type:info" pci
```

Ap_Id	Type	Information
pci1:hpc0_slot0	unknown	Slot 7
pci1:hpc0_slot1	unknown	Slot 8
pci1:hpc0_slot2	unknown	Slot 9
pci1:hpc0_slot3	ethernet/hp	Slot 10
pci1:hpc0_slot4	unknown	Slot 11

逻辑 Ap_Id pci1:hpc0_slot0 是可热插拔插槽 Slot 7 的逻辑 Ap_Id。组件 hpc0 表示此插槽的可热插拔适配卡，pci1 表示 PCI 总线实例。Type 字段表示插槽中存在的 PCI 适配卡的类型。

- 显示 PCIe 插槽配置信息。
例如：

```
# cfgadm pci
```

Ap_Id	Type	Receptacle	Occupant	Condition
pcie1	unknown	empty	unconfigured	unknown
pcie2	unknown	empty	unconfigured	unknown
pcie3	unknown	empty	unconfigured	unknown
pcie4	etherne/hp	connected	configured	ok
pcie5	pci-pci/hp	connected	configured	ok
pcie6	unknown	disconnected	unconfigured	unknown

- 显示特定的 PCIe 设备信息。

例如：

```
# cfgadm -s "cols=ap_id:busy:o_state" pci

Ap_Id          Busy    Occupant
pcie1          n      unconfigured
pcie2          n      unconfigured
pcie3          n      unconfigured
pcie4          n      configured
pcie5          n      configured
pcie6          n      configured
```

注 – 在大多数情况下，逻辑 Ap_Id 都应该与系统机箱上用丝网印花法标记的插槽标签匹配。有关可热插拔插槽的 cfgadm 输出，请参阅平台指南。显示 Busy 字段可确保尝试热插拔操作之前 Ap_Id 未转换为其他状态。

▼ 如何移除 PCI 适配卡

在 Solaris 10 6/06 发行版中，以下过程已经更新，可用于移除 PCIe 适配卡。但是，无论使用 PCI 还是 PCIe，移除适配卡的过程都是相同的。

- 1 成为超级用户。
- 2 确定 PCI 适配卡所在的插槽。

例如：

```
# cfgadm pci

Ap_Id          Type      Receptacle  Occupant    Condition
pcie1          unknown  empty       unconfigured unknown
pcie2          unknown  empty       unconfigured unknown
pcie3          unknown  empty       unconfigured unknown
pcie4          etherne/hp  connected  configured  ok
```

pcie5	pci-pci/hp	connected	configured	ok
pcie6	unknown	disconnected	unconfigured	unknown

- 3 停止用于打开设备的应用程序。
例如，如果设备是一块以太网卡，请使用 `ifconfig` 命令停用相应的接口，并取消检测该接口。

- 4 按照如下方式使用 `cfgadm(1M)` 命令手动取消配置设备。或者，如果您有 PCIe 适配卡，也可使用自动配置方法，如按平台指南中定义的方式按插槽的“注意”按钮。

```
# cfgadm -c unconfigure pcie4
```

- 5 确认是否已取消配置设备。

例如：

```
# cfgadm pci
```

Ap_Id	Type	Receptacle	Occupant	Condition
pcie1	unknown	empty	unconfigured	unknown
pcie2	unknown	empty	unconfigured	unknown
pcie3	unknown	empty	unconfigured	unknown
pcie4	unknown	connected	unconfigured	unknown
pcie5	pci-pci/hp	connected	configured	ok
pcie6	unknown	disconnected	unconfigured	unknown

注 - 如果取消配置设备，则 `Type` 和 `Condition` 也将成为 `unknown`。

- 6 手动断开插槽的电源。如果使用自动配置方法，则此步骤是不必要的。有关更多信息，请参阅平台指南。

```
# cfgadm -c disconnect pcie4
```

- 7 确认是否已断开设备连接。

例如：

```
# cfgadm pci
```

Ap_Id	Type	Receptacle	Occupant	Condition
pcie1	unknown	empty	unconfigured	unknown

pcie2	unknown	empty	unconfigured	unknown
pcie3	unknown	empty	unconfigured	unknown
pcie4	unknown	disconnected	unconfigured	unknown
pcie5	pci-pci/hp	connected	configured	ok
pcie6	unknown	disconnected	unconfigured	unknown

- 8 遵照平台指南中的相应说明移除 PCI 适配卡。移除该卡后，插座状态即为空。
例如：

cfgadm pci

Ap_Id	Type	Receptacle	Occupant	Condition
pcie1	unknown	empty	unconfigured	unknown
pcie2	unknown	empty	unconfigured	unknown
pcie3	unknown	empty	unconfigured	unknown
pcie4	unknown	empty	unconfigured	unknown
pcie5	pci-pci/hp	connected	configured	ok
pcie6	unknown	disconnected	unconfigured	unknown

注- 可以在引导时启用或禁用自动配置方法，具体取决于平台的实现。请针对环境设置合适的自动配置方法。

▼ 如何添加 PCI 适配卡

在 Solaris 10 6/06 发行版中，以下过程已经更新，可用于添加 PCIe 适配卡。但是，无论使用 PCI 还是 PCIe，添加适配卡的过程都是相同的。

- 1 成为超级用户。
- 2 标识可热插拔的插槽并打开锁。

例如，pcie3。

cfgadm pci

Ap_Id	Type	Receptacle	Occupant	Condition
-------	------	------------	----------	-----------

pcie1	unknown	empty	unconfigured	unknown
pcie2	unknown	empty	unconfigured	unknown
pcie3	unknown	empty	unconfigured	unknown
pcie4	unknown	empty	unconfigured	unknown
pcie5	pci-pci/hp	connected	configured	ok
pcie6	unknown	disconnected	unconfigured	unknown

3 遵照平台指南中的相应说明将 PCI 适配卡插入插槽。

4 确定插入 PCI 适配卡后该卡所在的插槽。

例如：

cfgadm pci

Ap_Id	Type	Receptacle	Occupant	Condition
pcie1	unknown	empty	unconfigured	unknown
pcie2	unknown	empty	unconfigured	unknown
pcie3	unknown	disconnected	unconfigured	unknown
pcie4	unknown	empty	unconfigured	unknown
pcie5	pci-pci/hp	connected	configured	ok
pcie6	unknown	disconnected	unconfigured	unknown

5 使用 **cfgadm** 命令手动连接插槽的电源。或者，如果您有 PCIe 适配卡，也可使用自动配置方法，如按平台指南中定义的方式按插槽的“注意”按钮。

例如：

cfgadm -c connect pcie3

6 确认附加点已连接。

例如：

cfgadm pci

Ap_Id	Type	Receptacle	Occupant	Condition
pcie1	unknown	empty	unconfigured	unknown
pcie2	unknown	empty	unconfigured	unknown

pcie3	unknown	connected	unconfigured	unknown
pcie4	unknown	empty	unconfigured	unknown
pcie5	pci-pci/hp	connected	configured	ok
pcie6	unknown	disconnected	unconfigured	unknown

- 7 按照如下方式使用 ckgadm 命令手动配置 PCI 适配卡。如果使用自动配置方法，此步骤应该是不必要的。有关更多信息，请参阅平台指南。

例如：

```
# ckgadm -c configure pcie3
```

- 8 验证插槽中的 PCI 适配卡的配置。

例如：

```
# ckgadm pci
```

Ap_Id	Type	Receptacle	Occupant	Condition
pcie1	unknown	empty	unconfigured	unknown
pcie2	unknown	empty	unconfigured	unknown
pcie3	etherne/hp	connected	configured	unknown
pcie5	pci-pci/hp	connected	configured	ok
pcie6	unknown	disconnected	unconfigured	unknown

- 9 如果此设备为新设备，请配置任何支持的软件。

例如，如果此设备是一块以太网卡，请使用 ifconfig 命令来设置接口。

注 - 可以在引导时启用或禁用自动配置方法，具体取决于平台的实现。请针对环境设置合适的自动配置方法。

对 PCI 配置问题进行疑难解答

错误消息

```
ckgadm: Configuration operation invalid: invalid transition
```

原因

尝试进行无效转换。

解决方案

检查是否正确发出了 `cfgadm -c` 命令。使用 `cfgadm` 命令检查当前的插座和插卡状态，并确保 `Ap_Id` 正确。

错误消息

```
cfgadm: Attachment point not found
```

原因

未找到指定的附加点。

解决方案

检查附加点是否正确。使用 `cfgadm` 命令显示可用附加点的列表。另外，请检查物理路径，以查看附加点是否仍然在正确的位置。

重新配置调整管理器 (Reconfiguration Coordination Manager, RCM) 脚本概述

重新配置调整管理器 (Reconfiguration Coordination Manager, RCM) 是用于管理动态移除系统组件的框架。通过使用 RCM，可以按顺序注册和释放系统资源。

可以使用新的 RCM 脚本功能来编写您自己的脚本，以关闭应用程序，或在动态重新配置过程中从应用程序中完全释放设备。如果重新配置请求影响通过脚本注册的资源，则 RCM 框架将自动启动脚本来响应该请求。

动态移除资源之前，还可以手动从应用程序中释放资源。或者，也可以使用带有 `-f` 选项的 `cfgadm` 命令来强制执行重新配置操作。但是，此操作可能会使应用程序处于未知状态。另外，从应用程序中手动释放资源通常还会导致错误。

RCM 脚本功能可以简化并更好地控制动态重新配置过程。通过创建 RCM 脚本，可以执行以下操作：

- 动态移除设备时，自动释放该设备。如果设备是通过应用程序打开，则此过程还将关闭该设备。
- 从系统中动态移除设备时，运行特定于站点的任务。

什么是 RCM 脚本？

- RCM 守护进程运行的可执行 shell 脚本（Perl、sh、csh 或 ksh）或二进制程序。Perl 是建议使用的语言。
- 通过使用脚本文件属主的用户 ID 在其自身的地址空间中运行的脚本。
- 使用 `cfgadm` 命令动态重新配置系统资源时，由 RCM 守护进程运行的脚本。

RCM 脚本的功能

动态移除设备时，可以使用 RCM 脚本从应用程序中释放该设备。如果设备当前处于打开状态，则 RCM 脚本还会将其关闭。

例如，磁带备份应用程序的 RCM 脚本可以通知磁带备份应用程序关闭磁带机或磁带备份应用程序。

RCM 脚本流程的工作原理

可以按照以下方式调用 RCM 脚本：

```
$ script-name command [args ...]
```

RCM 脚本可执行以下基本步骤：

1. 通过命令行参数获取 RCM 命令。
2. 执行该命令。
3. 将结果作为名称-值对写入 `stdout` 中。
4. 以适当的退出状态退出。

RCM 守护进程每次只运行一个脚本实例。例如，如果某一脚本正在运行，则在该脚本退出之前，RCM 守护进程不会运行该脚本的其他实例。

RCM 脚本命令

必须在 RCM 脚本中包括以下 RCM 命令：

- `scriptinfo`—收集脚本信息
- `register`—注册重要资源
- `resourceinfo`—收集资源信息

可以包括以下某些或所有 RCM 命令：

- `queryremove`—查询是否可以释放资源
- `preremove`—释放资源
- `postremove`—提供资源移除后的通知
- `undoremove`—撤消在 `preremove` 中执行的操作

有关这些 RCM 命令的完整说明，请参见 `rcmscript(4)`。

RCM 脚本处理环境

动态移除设备时，RCM 守护进程将运行以下命令：

- 脚本的 `register` 命令，用于收集脚本中标识的资源（设备名称）的列表。
- 脚本的 `queryremove` 和 `preremove` 命令，在移除资源之前运行，前提是脚本的已注册资源受到动态移除操作的影响。

- 脚本的 `postremove` 命令，前提是移除操作成功。但是，如果移除操作失败，RCM 守护进程会运行该脚本的 `undoremove` 命令。

RCM 脚本任务

以下各节介绍了应用程序开发者和系统管理员的 RCM 脚本任务。

应用程序开发者 RCM 脚本（任务列表）

以下任务列表介绍了创建 RCM 脚本的应用程序开发者的任务。

任务	说明	参考
1. 确定应用程序使用的资源。	确定应用程序使用的可能要动态移除的资源（设备名称）。	<code>cfgadm(1M)</code>
2. 确定用于释放资源的命令。	确定用于通知应用程序从应用程序中正常释放资源的命令。	应用程序文档
3. 确定用于资源移除后进行后续处理的命令。	包括用于通知应用程序资源移除的命令。	<code>rcmscript(4)</code>
4. 确定资源移除失败时使用的命令。	包括用于通知应用程序可用资源的命令。	<code>rcmscript(4)</code>
5. 编写 RCM 脚本。	根据任务 1-4 中标识的信息，编写 RCM 脚本。	第 116 页中的“磁带备份 RCM 脚本示例”
6. 安装 RCM 脚本。	将脚本添加到相应的脚本目录中。	第 115 页中的“如何安装 RCM 脚本”
7. 测试 RCM 脚本	通过手动运行脚本命令和启动动态重新配置操作来测试脚本。	第 115 页中的“如何测试 RCM 脚本”

系统管理员 RCM 脚本（任务列表）

以下任务列表介绍了创建 RCM 脚本进行站点自定义的系统管理员的任务。

任务	说明	参考
1. 确定要动态删除的资源。	使用 <code>cfgadm -l</code> 命令确定可能要删除的资源（设备名称）。	<code>cfgadm(1M)</code>

任务	说明	参考
2. 确定要停止的应用程序。	确定用于正常停止应用程序的命令。	应用程序文档
3. 确定用于在删除资源前后进行各种处理的命令。	确定删除资源前后要执行的操作。	<code>rcmscript(4)</code>
4. 编写 RCM 脚本。	根据任务 1-3 中标识的信息，编写 RCM 脚本。	第 116 页中的“磁带备份 RCM 脚本示例”
5. 安装 RCM 脚本。	将脚本添加到相应的脚本目录中。	第 115 页中的“如何安装 RCM 脚本”
6. 测试 RCM 脚本。	通过手动运行脚本命令和启动动态重新配置操作来测试脚本。	第 115 页中的“如何测试 RCM 脚本”

命名 RCM 脚本

必须按以下形式命名脚本，*vendor,service*，其中：

vendor 提供脚本的供应商的股票代号，或标识供应商的任何独特名称。

service 脚本表示的服务名称。

安装或删除 RCM 脚本

必须是超级用户 (root) 才能安装或删除 RCM 脚本。使用下表可确定应安装 RCM 脚本的位置。

表 6-1 RCM 脚本目录

目录位置	脚本类型
<code>/etc/rcm/scripts</code>	特定系统的脚本
<code>/usr/platform/'uname -i'/lib/rcm/scripts</code>	特定硬件实现的脚本
<code>/usr/platform/'uname -m'/lib/rcm/scripts</code>	特定硬件类的脚本
<code>/usr/lib/rcm/scripts</code>	任何硬件的脚本

▼ 如何安装 RCM 脚本

- 1 成为超级用户。
- 2 将脚本复制到相应目录。
请参见表 6-1。

例如：

```
# cp SUNW,sample.pl /usr/lib/rcm/scripts
```

- 3 将脚本的用户 ID 和组 ID 更改为所需的值。
chown user:group /usr/lib/rcm/scripts/SUNW,sample.pl
- 4 将 SIGHUP 发送到 RCM 守护进程。
pkill -HUP -x -u root rcm_daemon

▼ 如何删除 RCM 脚本

- 1 成为超级用户。
- 2 从 RCM 脚本目录中删除脚本。

例如：

```
# rm /usr/lib/rcm/scripts/SUNW,sample.pl
```

- 3 将 SIGHUP 发送到 RCM 守护进程。
pkill -HUP -x -u root rcm_daemon

▼ 如何测试 RCM 脚本

- 1 运行脚本之前，在命令行 shell 中设置环境变量，如 RCM_ENV_FORCE。
例如，在 Korn shell 中，请使用以下命令：

```
$ export RCM_ENV_FORCE=TRUE
```

- 2 通过从命令行手动运行脚本命令来测试脚本。
例如：

```
$ script-name scriptinfo
```

```
$ script-name register
```

```
$ script-name preremove resource-name
```

```
$ script-name postremove resource-name
```

- 3 确保脚本中的每个 RCM 脚本命令都会列显 `stdout` 的相应输出。
- 4 在相应的脚本目录中安装脚本。
有关更多信息，请参见第 115 页中的“如何安装 RCM 脚本”。
- 5 通过启动动态删除操作来测试脚本。
例如，假定脚本注册了设备 `/dev/dsk/clt0d0s0`。请尝试键入以下命令。

```
$ cfgadm -c unconfigure cl::dsk/clt0d0
```

```
$ cfgadm -f -c unconfigure cl::dsk/clt0d0
```

```
$ cfgadm -c configure cl::dsk/clt0d0
```



注意 - 请确保您熟悉这些命令，因为这些命令可以改变系统的状态，并会导致系统故障。

磁带备份 RCM 脚本示例

本示例说明如何使用 RCM 脚本执行磁带备份。

磁带备份 RCM 脚本的功能

磁带备份 RCM 脚本可执行以下步骤：

1. 设置 RCM 命令的分发表。
2. 调用与指定的 RCM 命令对应的分发例程，并对于未实现的 RCM 命令以状态 2 退出。
3. 设置 `scriptinfo` 部分。

```
rcm_script_func_info=Tape backup appl script for DR
```

4. 通过在 `stdout` 中列显所有磁带机设备名称，在系统中注册所有磁带机。

```
rcm_resource_name=/dev/rmt/$f
```

如果出现错误，则该脚本将在 `stdout` 中列显错误信息。

```
rcm_failure_reason=$errmsg
```

5. 为磁带设备设置资源信息。

```
rcm_resource_usage_info=Backup Tape Unit Number $unit
```

6. 通过检查备份应用程序是否使用该设备，设置 `preremove` 信息。如果备份应用程序未使用该设备，则动态重新配置操作将继续进行。如果备份应用程序使用该设备，则该脚本将检查 `RCM_ENV_FORCE`。如果将 `RCM_ENV_FORCE` 设置为 `FALSE`，则该脚本将拒绝动态重新配置操作，并列显以下消息：

```
rcm_failure_reason=tape backup in progress pid=...
```

如果将 `RCM_ENV_FORCE` 设置为 `TRUE`，则将停止备份应用程序，重新配置操作则继续进行。

磁带备份重新配置方案的结果

以下是使用 `cfgadm` 命令移除不包含 RCM 脚本的磁带设备时的各种结果。

- 如果使用 `cfgadm` 命令并且备份应用程序未使用磁带设备，则该操作将成功。
- 如果使用 `cfgadm` 命令并且备份应用程序使用磁带设备，则该操作将失败。

以下是使用 `cfgadm` 命令移除包含 RCM 脚本的磁带设备时的各种结果。

- 如果使用 `cfgadm` 命令并且备份应用程序未使用磁带设备，则该操作将成功。
- 如果使用不带 `-f` 选项的 `cfgadm` 命令并且备份应用程序使用磁带设备，则该操作将失败，并会显示以下错误消息：

```
tape backup in progress pid=...
```

- 如果使用 `cfgadm -f` 命令并且备份应用程序使用磁带设备，则该脚本会停止备份应用程序，`cfgadm` 操作将成功。

示例一磁带备份 RCM 脚本

```
#!/usr/bin/perl -w

#

# A sample site customization RCM script.

#

# When RCM_ENV_FORCE is FALSE this script indicates to RCM that it cannot

# release the tape drive when the tape drive is being used for backup.

#

# When RCM_ENV_FORCE is TRUE this script allows DR removing a tape drive

# when the tape drive is being used for backup by killing the tape
```

```
# backup application.

#

use strict;

my ($cmd, %dispatch);

$cmd = shift(@ARGV);

# dispatch table for RCM commands

%dispatch = (

    "scriptinfo"    =>    \&do_scriptinfo,

    "register"      =>    \&do_register,

    "resourceinfo"  =>    \&do_resourceinfo,

    "queryremove"   =>    \&do_preremove,

    "preremove"     =>    \&do_preremove

);

if (defined($dispatch{$cmd})) {

    &{$dispatch{$cmd}};

} else {

    exit (2);

}

sub do_scriptinfo
```

```
{

    print "rcm_script_version=1\n";

    print "rcm_script_func_info=Tape backup appl script for DR\n";

    exit (0);

}

sub do_register

{

    my ($dir, $f, $errmsg);

    $dir = opendir(RMT, "/dev/rmt");

    if (!$dir) {

        $errmsg = "Unable to open /dev/rmt directory: $!";

        print "rcm_failure_reason=$errmsg\n";

        exit (1);

    }

    while ($f = readdir(RMT)) {

        # ignore hidden files and multiple names for the same device

        if (($f !~ /\^\.\/) && ($f =~ /^[0-9]+$/)) {

            print "rcm_resource_name=/dev/rmt/$f\n";

        }

    }

}
```

```
        closedir(RMT);

        exit (0);

    }

sub do_resourceinfo

{

    my ($rsrc, $unit);

    $rsrc = shift(@ARGV);

    if ($rsrc =~ /^\/dev\/rmt\/([0-9]+)$/) {

        $unit = $1;

        print "rcm_resource_usage_info=Backup Tape Unit Number $unit\n";

        exit (0);

    } else {

        print "rcm_failure_reason=Unknown tape device!\n";

        exit (1);

    }

}

sub do_preremove

{

    my ($rsrc);

    $rsrc = shift(@ARGV);
```



```
# check if backup application is using this resource

# if (the backup application is not running on $rsrc) {

    # allow the DR to continue

#     exit (0);

# }

#

# If RCM_ENV_FORCE is FALSE deny the operation.

# If RCM_ENV_FORCE is TRUE kill the backup application in order

# to allow the DR operation to proceed

#

if ($ENV{RCM_ENV_FORCE} eq 'TRUE') {

    if ($cmd eq 'preremove') {

        # kill the tape backup application

    }

    exit (0);

} else {

    #

    # indicate that the tape drive can not be released

    # since the device is being used for backup by the

    # tape backup application

    #

    print "rcm_failure_reason=tape backup in progress pid=...\n"
```

;

```
        exit (3);  
    }  
}
```

使用 USB 设备（概述）

本章概述了 Solaris OS 中的通用串行总线 (Universal Serial Bus, USB) 设备。

以下是本章中概述信息的列表。

- 第 123 页中的 “Solaris 10 6/06 发行版在 USB 设备方面的新增功能”
- 第 124 页中的 “Solaris 10 1/06 发行版在 USB 设备方面的新增功能”
- 第 125 页中的 “Solaris 10 发行版在 USB 设备方面的新增功能”
- 第 126 页中的 “USB 设备概述”
- 第 131 页中的 “关于 Solaris OS 中的 USB”

有关 USB 设备的最新信息，请访问以下站点：

http://www.sun.com/io_technologies/USB-Faq.html

有关 USB 设备的常规信息，请访问以下站点：

<http://developers.sun.com/solaris/developer/support/driver/usb.html>

有关在 Solaris OS 中使用 USB 设备的逐步说明，请参见第 8 章。

有关动态重新配置和热插拔的常规信息，请参见第 6 章。

有关配置 USB 打印机的信息，请参见《系统管理指南：高级管理》中的“打印方面的新增功能”。

Solaris 10 6/06 发行版在 USB 设备方面的新增功能

Solaris 10 6/06：可以在与 EHCI 控制器连接的 USB 海量存储设备上创建并挂载 ZFS 文件系统。请勿在与 OHCI 或 UHCI 控制器连接的 USB 设备上使用 ZFS。

有关使用 USB 海量存储设备的信息，请参见第 139 页中的 “使用 USB 海量存储设备”。

有关创建和挂载 ZFS 文件系统的信息，请参见 [zfs\(1M\)](#) 和 [zpool\(1M\)](#)。

Solaris 10 6/06：在本 Solaris 发行版中，不可移除的 USB 存储设备和 1394 海量存储设备在驱动程序级别上都被标识为可热插拔的设备。此新增功能意味着，USB 存储设备可以在不重新引导系统的情况下连接或断开，并且可以无需用户干预地自动配置或取消配置。这些更改是在内核级别进行的，因此不会影响这些设备的使用。例如，挂载和取消挂载这些设备的工作仍然由 `vol` 控制。

有关更多信息，请参见第 72 页中的“[USB 和 1394 \(FireWire\) 支持增强功能](#)”。

Solaris 10 1/06 发行版在 USB 设备方面的新增功能

以下各节介绍了 Solaris 10 1/06 发行版中 USB 的新增功能。

- 第 124 页中的“[x86: 在基于 GRUB 的引导中支持 USB CD 和 DVD](#)”
- 第 124 页中的“[USB 虚拟键盘和鼠标支持](#)”
- 第 125 页中的“[vol 用于识别热插拔 USB 设备](#)”

x86: 在基于 GRUB 的引导中支持 USB CD 和 DVD

Solaris 10 1/06：可以在基于 GRUB 的引导环境中使用以下 USB 功能：

- 从 USB CD 驱动器或 DVD 驱动器安装
- 从 USB 存储设备引导

有关基于 GRUB 的引导的更多信息，请参见《系统管理指南：基本管理》中的第 8 章“关闭和引导系统（概述）”。

USB 虚拟键盘和鼠标支持

Solaris 10 1/06：借助 USB 虚拟键盘和鼠标支持，可以挂接多个键盘和多个鼠标，这种情况下该键盘组或鼠标组的行为如同一个虚拟键盘或鼠标。这意味着每个物理设备的输入都会并入单个输入流中。例如，如果在一个键盘上键入 SHIFT，而在另一个键盘上键入 A，则回显的字符为大写的 A。

另外，本发行版还支持将 USB 键盘或鼠标添加到膝上型计算机中，并将这些设备作为一个整体设备用于膝上型计算机的 PS/2 键盘和触摸板。

有关更多信息，请参阅 [virtualkm\(7D\)](#)。

vold 用于识别热插拔 USB 设备

Solaris 10 1/06：可移除介质管理器 (vold) 现在可识别热插拔。无需重新启动此守护进程，即可挂载已热插入的 USB 海量存储设备。但是，由于 vold 并不总是成功，因此对于某些设备可能仍然需要手动进行挂载。如果 vold 无法自动挂载 USB 设备，请按如下所示停止 vold：

```
# /etc/init.d/volmgt stop
```

有关在不运行 vold 的情况下使用 USB 设备的信息，请参见第 146 页中的“如何在^{不运行 vold 的情况下准备使用 USB 海量存储设备}”。

有关手动挂载 USB 海量存储设备的信息，请参见第 160 页中的“如何在^{不运行 vold 的情况下挂载或取消挂载 USB 海量存储设备}”。

Solaris 10 发行版在 USB 设备方面的新增功能

有关 Solaris 10 发行版中 USB 新增功能的信息，请参见以下参考资料：

- 第 134 页中的“USB 滚轮鼠标支持”
- 第 131 页中的“USB 2.0 功能”
- 第 129 页中的“USB 驱动程序增强功能”

有关 Solaris 新增功能的完整列表以及 Solaris 发行版的描述，请参见《Solaris 10 新增功能》。

Solaris 对 USB 设备的支持

从 Solaris 10 发行版开始，USB 2.0 集线器上可支持所有 USB 1.1 设备，包括音频设备。使用下表可确定特定 USB 1.1 和 USB 2.0 设备的 Solaris 支持信息。

	Solaris 8 HW 5/03 和更高发行版	Solaris 9 发行版	Solaris 10
常规的 USB 1.1 设备支持	SPARC 和 x86	SPARC 和 x86	SPARC 和 x86
常规的 USB 2.0 设备支持	仅限于 SPARC	SPARC 和 x86 (Solaris 9 4/04)	SPARC 和 x86
特定的 USB 1.1 和 USB 2.0 设备支持			

	Solaris 8 HW 5/03 和更高发行版	Solaris 9 发行版	Solaris 10
音频设备（请参见下面的注释。）	仅限于 USB 1.1： 在 USB 2.0 集线器上不受支持	仅限于 USB 1.1： 在 USB 2.0 集线器上不受支持	仅限于 USB 1.1： 在 USB 2.0 集线器上受支持
通用 USB 驱动程序 (ugen(7D))	仅限于 SPARC	SPARC 和 x86	SPARC 和 x86
hid 设备（键盘和鼠标设备，hid(7D)）	SPARC 和 x86	SPARC 和 x86	SPARC 和 x86
集线器 (hubd(7D))	SPARC 和 x86	SPARC 和 x86	SPARC 和 x86
打印机	SPARC 和 x86	SPARC 和 x86	SPARC 和 x86
串行设备（Edgeport (usbser_edge(7D))、Prolific (usbprl(7D))、Keyspan (usbksp(7D))）	SPARC 和 x86	SPARC 和 x86	SPARC 和 x86
存储设备 (scsa2usb(7D))	SPARC 和 x86	SPARC 和 x86	SPARC 和 x86
用户空间 USB 设备管理库 (libusb(3LIB))	不支持	不支持	SPARC 和 x86

- 注：
- 系统仅支持 USB 1.x 音频设备，不支持任何 USB 2.0 音频设备。
 - 与 USB 2.0 集线器连接的 USB 1.x 音频设备只能在 Solaris 10 发行版中使用，该集线器与 USB 2.0 端口连接。有关更多信息，请参见 usb_ac(7D) 和 usb_as(7D)。
 - USB 驱动程序不支持的设备可能包含 libusb 应用程序，如 gphoto2、gtkam 和 pilotlink。有关更多信息，请参阅 /usr/sfw/share/doc/libusb/libusb.txt。
 - **Solaris 8 和 Solaris 9 发行版**—有关 USB 双框架问题，请访问以下站点：
http://www.sun.com/io_technologies/USB-Faq.html

有关与海量存储设备关联的任务信息，请参见第 8 章。

有关 ugen 的更多信息，请参见第 129 页中的“USB 驱动程序增强功能”。

USB 设备概述

通用串行总线 (Universal Serial Bus, USB) 由 PC 行业开发，用于提供将外围设备（如键盘、鼠标设备和打印机）连接到系统的低成本解决方案。

USB 连接器适用于以某种方式连接的唯一一种类型的电缆。USB 的主要设计意图是缓解对不同设备的多种连接器类型的需求。此设计可减少系统后面板上各种连接的混乱。

设备与外部 USB 集线器或计算机本身的根集线器上的 USB 端口连接。由于集线器有多个端口，因此从集线器可以引出多个设备树分支。

有关更多信息，请参见 `usb(7D)` 或转至以下站点：

<http://www.usb.org>

常用的 USB 首字母缩略词

下表介绍了 Solaris OS 中使用的 USB 首字母缩略词。有关 USB 组件和对应的首字母缩略词的完整说明，请访问以下站点：

<http://www.usb.org>

首字母缩略词	定义
UGEN	USB 通用驱动程序
USB	通用串行总线
USBA	通用串行总线体系结构 (Solaris)
USBAI	USBA 客户机驱动程序接口 (Solaris)
HCD	USB 主机控制器驱动程序
EHCI	增强型主机控制器接口
OHCI	开放主机控制器接口
UHCI	通用主机控制器接口

USB 总线说明

USB 规范是公用免费规范。该规范定义了总线和连接器的电子和机械接口。

USB 采用由集线器为 USB 设备提供附着点的拓扑结构。主机控制器包含根集线器，该集线器是系统中所有 USB 端口的起点。有关集线器的更多信息，请参见第 134 页中的“USB 主机控制器和集线器”。

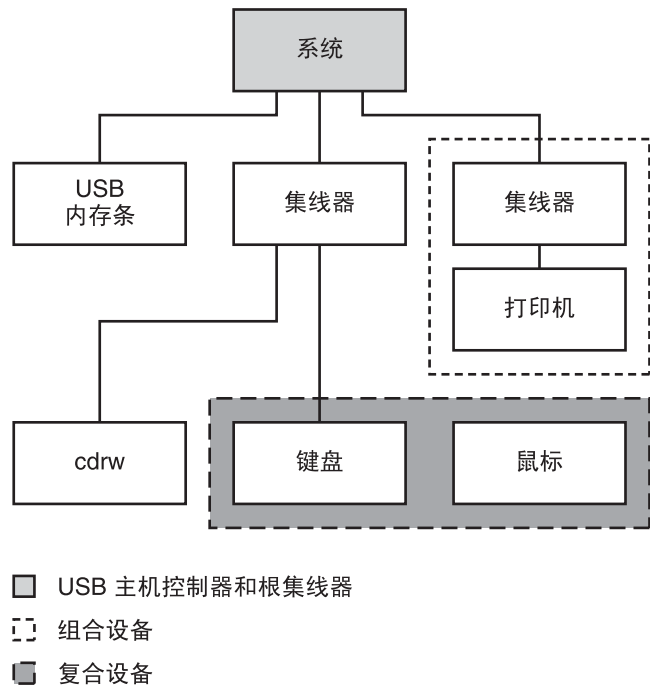


图 7-1 USB 物理设备分层结构

图 7-1 显示了带有三个活动 USB 端口的系统。第一个 USB 端口连接 USB 内存条。第二个 USB 端口连接外部集线器，该集线器又连接 cdrw 设备和键盘/鼠标复合设备。作为**复合设备**，此键盘包含 USB 控制器，该控制器可以操作键盘和连接的鼠标。键盘和鼠标共享公用的 USB 总线地址，因为它们由同一个 USB 控制器定向。

图 7-1 还显示了作为**组合设备**的集线器和打印机的示例。该集线器是与打印机封装在同一个箱子中的外部集线器。打印机与集线器永久连接。集线器和打印机具有各自独立的 USB 总线地址。

以下列出了图 7-1 中显示的一些设备的设备树路径名。

内存条	/pci@1f,4000/usb@5/storage@1
键盘	/pci@1f,4000/usb@5/hub@2/device@1/keyboard@0
鼠标	/pci@1f,4000/usb@5/hub@2/device@1/mouse@1
cdrw 设备	/pci@1f,4000/usb@5/hub@2/storage@3
打印机	/pci@1f,4000/usb@5/hub@3/printer@1

USB 设备和驱动程序

具有类似属性和服务的 USB 服务组成设备类。每个设备类都有对应的驱动程序。类中的设备由同一个设备驱动程序进行管理。但是，USB 规范也允许使用不属于特定类的供应商专有设备。

人工接口设备 (Human Interface Device, HID) 类包含用户控制的设备，如以下设备：

- 键盘
- 鼠标设备
- 游戏控制器

通信设备类包含以下设备：

- 调制解调器
- 以太网适配器

其他设备类包括以下类：

- 音频
- 显示器
- 打印机
- 存储设备

每个 USB 设备都包含反映其所属类的描述符。设备类可指定其成员在配置和数据传输中的行为方式。可以从以下站点获取其他的类信息：

<http://www.usb.org>

有关 Solaris 发行版支持的 USB 设备的更多信息，请参见 `usb(7D)`。

USB 驱动程序增强功能

从 Solaris 9 4/04 发行版开始，包括以下 USB 驱动程序增强功能。

- **通用 USB 驱动程序**—现在可通过使用标准的 UNIX® `read(2)` 和 `write(2)` 系统调用的应用程序来访问和处理 USB 设备，并且无需编写专用内核驱动程序。其他功能包括：
 - 应用程序可访问原始设备数据和设备状态。
 - 驱动程序支持控制、批处理和中断（出入）传输。

有关更多信息，请参阅 `ugen(7D)`。

- **USB 串行驱动程序支持**
 - Digi Edgeport USB 支持—Edgeport USB 驱动程序只能用于 Edgeport 设备，而不能用于其他 USB 串行设备。
 - 新设备可以作为 `/dev/term/[0-9]*` 和 `/dev/cua/[0-9]*` 进行访问。

- USB 串行端口可以像任何其他串行端口一样使用，但不可用作本地串行控制台。用户可以很容易地理解这样一个事实，即数据是通过 USB 端口运行的。

有关更多信息，请参见 `usbser_edge(7D)` 或转至以下站点：

- <http://www.digi.com>
- <http://www.sun.com/io>

有关 Edgeport 兼容性问题的更多信息，请访问以下站点：

http://www.sun.com/io_technologies/USB-Faq.html

- Keyspan—Keyspan USB 串行驱动程序只能用于 Keyspan 设备，该驱动程序当前支持 USA-19HS 和 USA-49WLC 模型。
有关更多信息，请参见 `usbsksp(7D)`。
- Prolific—Prolific USB 串行驱动程序只能用于基于 PL2303 芯片组的设备。
有关更多信息，请参见 `usbsprl(7D)`。
- 对用户编写的内核和用户级驱动程序的文档和二进制数据支持—有关 USB 驱动程序开发的最新信息，请访问：
 - http://www.sun.com/io_technologies/USB-Faq.html
 - <http://developers.sun.com/solaris/developer/support/driver/usb.html>
 - 《编写设备驱动程序》中的第 19 章“USB 驱动程序”。
 - 《编写设备驱动程序》中的附录 C“使设备驱动程序支持 64 位”。
 - 《Device Driver Tutorial》
 - `Intro(7)`、`Intro(9F)` 和 `Intro(9S)`
 - http://developers.sun.com/prodtech/solaris/driverdev/reference/codesamples/usb_security/index.html

EHCI、OHCI 和 UHCI 驱动程序

EHCI 驱动程序的功能包括：

- 符合支持 USB 2.0 的增强主机控制器接口的标准。
- 支持高速控制、批处理和中断传输。
- 目前尚不支持高速同步事务。
- USB 2.0 芯片包含一个 EHCI 控制器以及一个或多个 OHCI 或 UHCI 控制器。
- USB 1.1 设备插入后会动态分配给 OHCI 或 UHCI 控制器。USB 2.0 设备插入后会动态分配给 EHCI 控制器。

使用 `prtconf` 命令输出可确定系统是支持 USB 1.1 设备还是 USB 2.0 设备。例如：

```
# prtconf -D | egrep "ehci|ohci|uhci"
```

如果prtconf 输出标识了 EHCI 控制器，则系统支持 USB 2.0 设备。

如果prtconf 输出标识了 OHCI 或 UHCI 控制器，则系统支持 USB 1.1 设备。

Solaris USB 体系结构 (Solaris USB Architecture, USB A)

USB 设备可以表示为两级设备树节点。设备节点表示整个 USB 设备。一个或多个子接口节点表示设备上的各个 USB 接口。

驱动程序绑定是通过使用兼容的名称属性实现的。有关更多信息，请参阅 IEEE 1275 USB 绑定的 3.2.2.1 节以及《编写设备驱动程序》。驱动程序可以绑定至整个设备并控制所有接口，也可以只绑定至一个接口。如果供应商或类驱动程序不要求绑定至整个设备，则可以将通用 USB 多接口驱动程序绑定至设备级节点。此驱动程序会通过使用 IEEE 1275 绑定规范的 3.3.2.1 节中定义的兼容名称属性，尝试将各驱动程序绑定至每个接口。

Solaris USB 体系结构 (USB Architecture, USB A) 遵循 USB 1.1 和 USB 2.0 规范，并且是 Solaris 设备驱动程序接口 (Device Driver Interface, DDI) 的一部分。USB A 模型与 Sun 公用 SCSI 体系结构 (Sun Common SCSI Architecture, SCSI A) 类似。如下图所示，USB A 是一个薄层，用于为客户机驱动程序提供通用抽象 USB 传输层，进而为这些驱动程序提供实现核心通用 USB 功能的服务。

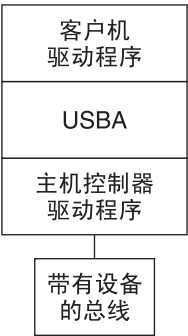


图 7-2 Solaris USB 体系结构 (Solaris USB Architecture, USB A)

关于 Solaris OS 中的 USB

本节介绍您应了解的有关 Solaris OS 中的 USB 的信息。

USB 2.0 功能

从 Solaris 9 4/04 发行版开始，包括以下 USB 2.0 功能：

- **更好的性能**—提高了连接到 USB 2.0 控制器的设备的数据吞吐量，速度最快可达 USB 1.1 设备的 40 倍。
访问高速海量存储设备（如 DVD 和硬盘）时，可以利用高速 USB 协议。
- **向下兼容性**—与 1.0 和 1.1 版本的设备和驱动程序兼容，因而可以使用相同的电缆、连接器和软件界面。

有关 USB 设备和术语的说明，请参见第 126 页中的“USB 设备概述”。

USB 2.0 设备功能和兼容性问题

USB 2.0 设备的定义是遵循 USB 2.0 规范的高速设备。您可以参阅 <http://www.usb.org> 中的 USB 2.0 规范。

要确定 USB 设备在 Solaris 10 以及更高发行版中的速度，请检查 `/var/adm/messages` 文件中类似以下内容的消息：

```
Dec 13 17:05:57 mysystem usba: [ID 912658 kern.info] USB 2.0 device
(usb50d,249) operating at hi speed (USB 2.x) on USB 2.0 external hub:
storage@4, scsa2usb0 at bus address 4
```

以下是本 Solaris 发行版中支持的一些 USB 设备：

- 海量存储设备，如 CD-RW、硬盘、DVD、数码相机、软盘、磁带机、内存条以及多格式读卡器
- 键盘和鼠标设备
- 音频设备，如扬声器和麦克风

有关 Solaris 发行版中已验证的 USB 设备的完整列表，请访问：

http://www.sun.com/io_technologies/USB.html

通过修改 `scsa2usb.conf` 文件，可以使用其他存储设备。有关更多信息，请参见 `scsa2usb(7D)`。

Solaris USB 2.0 设备支持包括以下功能：

- USB 总线速度从 12 MB/秒提高到 480 MB/秒。这种提高意味着如果将支持 USB 2.0 规范的设备连接到 USB 2.0 端口，其运行速度将大大高于支持 USB 1.1 规范的同类设备。
USB 2.0 端口可能是以下端口之一：
 - USB 2.0 PCI 卡上的端口
 - 连接到 USB 2.0 端口的 USB 2.0 集线器上的端口
 - SPARC 或 x86 计算机主板上的端口

- 要提供 USB 2.0 端口，需要一块 USB 2.0 PCI 卡。有关已针对 Solaris 发行版验证过的 USB 2.0 PCI 卡的列表，请访问：
http://www.sun.com/io_technologies/USB.html
- USB 1.1 设备的工作方式与过去一样，即使同一系统中同时有 USB 1.1 和 USB 2.0 设备也是如此。
- 尽管 USB 2.0 设备能够在 USB 1.x 端口上运行，但如果将其连接到 USB 2.0 端口，则性能会大大提高。
- USB 2.0 主机控制器具有一个高速的增强主机控制器接口 (Enhanced Host Controller Interface, EHCI) 以及一个或多个内嵌了 OpenHCI 主机控制器接口 (OpenHCI Host Controller Interface, OHCI) 或通用主机控制器接口 (Universal Host Controller Interface, UHCI) 的控制器。连接到 USB 2.0 端口的设备会动态分配给 EHCI 或 OHCI 控制器，具体取决于它们是否支持 USB 2.0。

注 - 对于连接到 USB 2.0 PCI 卡上某一端口的 USB 2.0 存储设备，以及在同一硬件配置中用于早期 Solaris 发行版的 USB 2.0 存储设备，在升级到本发行版后，可以更改设备名称。进行这种更改是由于上述设备现在被视为 USB 2.0 设备，并且由 EHCI 控制器接管。上述设备的控制器编号（`/dev/[r]dsk/cwtxdysz` 中的 `w`）将被更改。

另请注意，USB 设备的速度限制为父端口可支持的速度。例如，如果 USB 2.0 外部集线器的下游设备依次是 USB 1.x 集线器和 USB 2.0 设备，则与 USB 2.0 外部集线器连接的设备将以全速而非高速运行。

有关 USB 2.0 设备支持的更多信息，请参见 `ehci(7D)` 和 `usba(7D)`。

总线供电设备

总线供电的集线器使用来自与之连接的 USB 总线的电力，为与其连接的设备供电。必须特别小心不要使这些集线器过载，因为这些集线器为其下游设备提供的电力有限。

从 Solaris 10 6/06 发行版开始，已实现了对 USB 设备进行电能预算。此功能具有以下限制：

- 禁止层叠两个总线供电的集线器。
- 每个总线供电的集线器允许每个接口上的最大电流仅为 100 mA。
- 仅允许将自备电源或低总线供电的设备连接至总线供电的集线器。不允许连接高总线供电的设备。某些集线器或设备会报告虚假的电源来源，这种连接可能是不可预测的。

USB 键盘和鼠标设备

使用 USB 键盘和鼠标设备时，请牢记以下问题：

- 在重新引导期间或在 SPARC 系统中出现 ok 提示符时，请勿移动键盘和鼠标。系统重新引导后，可以随时将键盘和鼠标移动到另一个集线器。插入键盘和鼠标后，它们便会再次完全正常运行。
- 在某些第三方 USB 键盘上，紧靠小键盘左侧的键可能不起作用。
- **SPARC**—在 SPARC 系统中使用 USB 键盘和鼠标设备时，请牢记以下问题：
 - USB 键盘上的电源键与 Sun 类型 5 键盘上的电源键作用不同。在 USB 键盘上，可以使用 SUSPEND/SHUTDOWN 键暂停或关闭系统。但是，不能使用该键来打开系统电源。
 - 引导过程完成之前，OpenBoot PROM (OBP) 限制键盘和鼠标设备只能与主板根集线器端口连接。
 - USB 键盘和鼠标设备不能与 Sun 类型 3、4 或 5 键盘同时用于传统 SPARC 系统（如 Ultra 80）。
- 有关多键盘和鼠标设备支持的信息，请参见 `virtualkm(7D)`。

USB 滚轮鼠标支持

从 Solaris 9 9/04 发行版开始，支持以下滚轮鼠标功能：

- 支持超过 3 个按键的 USB 或 PS/2 鼠标设备。
- 支持 USB 或 PS/2 鼠标设备的滚轮鼠标滚动功能。此项支持表明滚动 USB 或 PS/2 鼠标的滚轮会在鼠标焦点所在的应用程序或窗口中产生滚动效果。StarSuite™、Firefox 和 GNOME 应用程序均支持滚轮鼠标的滚动功能。但其他应用程序可能不支持。

USB 主机控制器和集线器

USB 集线器负责以下操作：

- 监视在其端口上插入或移除设备
- 对其端口上的各个设备进行电源管理
- 控制传输到其端口的电源

USB 主机控制器具有一个嵌入式集线器，称为**根集线器**。系统的后面板上显示的端口是根集线器的端口。USB 主机控制器负责以下操作：

- 定向 USB 总线。各个设备都不能任意定向总线。
- 使用该设备确定的轮询间隔轮询设备。为说明轮询间隔的时间，假定设备具有足够缓冲。
- 在 USB 主机控制器和与其连接的设备之间发送数据。不支持对等通信。

USB 集线器设备

- 请勿在基于 SPARC 的系统或基于 x86 的系统上将集线器层叠超过四级。在 SPARC 系统上，OpenBoot™ PROM 不能可靠地探测超过四级的设备。
- 请勿以层叠样式将一个总线供电的集线器插入另一个总线供电的集线器中。总线供电的集线器没有自己的电源。
- 请勿将需要大功率电源的设备连接至总线供电的集线器。这些设备可能会被拒绝与总线供电的集线器连接，或者可能会消耗其他设备的集线器的电源。这类设备的示例为 USB 软盘设备。

SPARC: USB 电源管理

SPARC 系统完全支持暂停和恢复 USB 设备。但是，请勿暂停繁忙的设备，并且请勿在暂停关机的情况下关闭系统电源时移除设备。

USB 框架可通过启用电源管理，尽可能地对基于 SPARC 的系统上的所有设备进行电源管理。对 USB 设备进行电源管理意味着集线器驱动程序将暂停与设备连接的端口。支持**远程唤醒**的设备可以通知系统唤醒设备路径中的所有内容，以便可以使用该设备。如果应用程序将 I/O 发送到该设备，则主机系统也可以唤醒该设备。

缺省情况下，会对所有的 HID 设备（键盘、鼠标、集线器和存储设备）、集线器设备和存储设备进行电源管理，前提是这些设备支持远程唤醒功能。仅在两个打印作业之间对 USB 打印机进行电源管理。通用 USB 启动程序 (UGEN) 管理的设备仅在关闭时才会进行电源管理。

实施电源管理以减少能耗时，首先会关闭 USB 叶设备的电源。关闭与集线器端口连接的所有设备的电源后，集线器电源会在一定延迟后关闭。为实现最有效的电源管理，请勿层叠多个集线器。

有关在 SPARC 系统中使用 SUSPEND/SHUTDOWN 键的信息，请参见第 134 页中的“[USB 键盘和鼠标设备](#)”。

USB 电缆连接原则

连接 USB 电缆时，请牢记以下原则：

- 始终使用与 USB 2.0 兼容并且以全速（480 Mbit/sec）运行的 20/28 AWG 电缆来连接 USB 2.0 设备。
- 支持的最大电缆长度为 5 米。
- 请勿使用电缆延伸架。为获得最佳效果，请使用自备电源的集线器来延伸电缆长度。

有关更多信息，请访问：

<http://www.usb.org/about/faq/ans5>

使用 USB 设备（任务）

本章提供有关在 Solaris OS 中使用 USB 设备的逐步说明。

有关与使用 USB 设备关联的过程信息，请参见以下主题：

- 第 137 页中的 “在 Solaris OS 中管理 USB 设备（指南）”
- 第 138 页中的 “使用 USB 海量存储设备（任务列表）”
- 第 163 页中的 “使用 USB 音频设备（任务列表）”
- 第 167 页中的 “使用 `cfgadm` 命令热插拔 USB 设备（任务列表）”

有关 USB 设备的最新信息，请转至以下站点：

http://www.sun.com/io_technologies/USB-Faq.html

有关使用 USB 设备的概述信息，请参见第 7 章。

在 Solaris OS 中管理 USB 设备（指南）

使用本指南可以确定用于在 Solaris OS 中管理 USB 设备的所有任务。每个任务都指向一系列其他任务，如使用 USB 设备、热插拔 USB 设备以及添加 USB 音频设备。

有关在 Solaris OS 中使用 USB 组件的信息，请参见第 131 页中的 “关于 Solaris OS 中的 USB”。

任务	说明	参考
使用 USB 海量存储设备。	USB 海量存储设备必须包含文件系统，才能向其中添加数据。此外，必须先格式化 USB 软盘，然后才能在这些软盘上创建和挂载文件系统。 本节还介绍如何以物理方式在系统中添加或移除 USB 设备。	第 138 页中的 “使用 USB 海量存储设备（任务列表）”
添加 USB 音频设备。	使用此任务列表可确定与添加 USB 音频设备关联的任务。	第 163 页中的 “使用 USB 音频设备（任务列表）”
使用 <code>cfgadm</code> 命令在系统中添加或删除 USB 设备。	可以使用 <code>cfgadm</code> 命令以逻辑方式在系统中添加或删除 USB 设备。	第 167 页中的 “使用 <code>cfgadm</code> 命令热插拔 USB 设备（任务列表）”

使用 USB 海量存储设备（任务列表）

任务	说明	参考
添加或删除 USB 海量存储设备。	选择以下操作之一可添加 USB 海量存储设备： 在不运行 <code>vol</code> d 的情况下添加 USB 海量存储设备。 添加用于访问数字图像的 USB 相机。 在不运行 <code>vol</code> d 的情况下删除 USB 海量存储设备。	第 143 页中的 “如何在不运行 <code>vol</code> d 的情况下添加 USB 海量存储设备” 第 143 页中的 “如何添加 USB 相机” 第 145 页中的 “如何在不运行 <code>vol</code> d 的情况下移除 USB 海量存储设备”
添加不兼容的 USB 海量存储设备。	通过向 <code>scsa2usb.conf</code> 中添加项，可以添加不兼容的 USB 海量存储设备。	第 141 页中的 “如何使用不兼容的 USB 海量存储设备”
准备使用 USB 海量存储设备。	在运行 <code>vol</code> d 的情况下准备使用 USB 海量存储设备。 在不运行 <code>vol</code> d 的情况下准备使用 USB 海量存储设备。	第 145 页中的 “在运行 <code>vol</code> d 的情况下准备使用 USB 海量存储设备” 第 146 页中的 “如何在不运行 <code>vol</code> d 的情况下准备使用 USB 海量存储设备”

任务	说明	参考
显示 USB 设备信息。	显示有关 USB 设备的信息。	第 147 页中的 “如何显示 USB 设备信息”
在 USB 海量存储设备中创建文件系统。	必须先和设备中创建文件系统，然后才能向其中添加数据。	第 148 页中的 “如何在运行 <code>vol</code> 的情况下在 USB 海量存储设备中创建文件系统”
在 USB 海量存储设备中修改分区并创建文件系统。	在 USB 海量存储设备中创建文件系统之前，可能需要修改现有的分区。	第 151 页中的 “如何在运行 <code>vol</code> 的情况下在 USB 海量存储设备中修改分区和创建 PCFS 文件系统”
在 USB 海量存储设备中创建 Solaris 分区并修改相应的片。	使用 <code>rmformat</code> 修改片信息之前，需要使用片信息创建一个文件。	第 157 页中的 “如何在运行 <code>vol</code> 的情况下在 USB 海量存储设备中创建 Solaris 分区和修改片”
在不运行 <code>vol</code> 的情况下挂载 USB 海量存储设备。	在不运行 <code>vol</code> 的情况下挂载 USB 海量存储设备。	第 160 页中的 “如何在运行 <code>vol</code> 的情况下挂载或取消挂载 USB 海量存储设备”
（可选）禁用 USB 设备驱动程序。	如果不需要系统支持 USB，可以禁用 USB 设备驱动程序。	第 162 页中的 “如何禁用特定的 USB 驱动程序”
（可选）删除未使用的 USB 设备链路。	使用 <code>devfsadm</code> 命令删除 USB 设备链路。	第 162 页中的 “如何删除未使用的 USB 设备链路”

使用 USB 海量存储设备

从 Solaris 9 发行版开始，支持以下 USB 可移除海量存储设备：

- CD-RW
- 硬盘
- DVD
- 数码相机
- 软盘设备
- SmartMedia 和 CompactFlash 设备

有关将 USB 海量存储设备用于 Solaris ZFS 文件系统的信息，请参见第 123 页中的 “Solaris 10 6/06 发行版在 USB 设备方面的新增功能”。

有关 Solaris OS 所支持的 USB 设备的完整列表，请参见：

http://www.sun.com/io_technologies/USB.html

在以前的 Solaris 发行版中，所有的 USB 存储设备都被标识为可移除介质设备，这些设备可提供以下许多优点，包括通过 `vol` 自动挂载。在 Solaris 10 6/06 发行版中，USB 海

量存储设备被标识为可热插拔设备，而且具有 USB 可移除设备的优点（如下所述）。有关可热插拔行为的更多信息，请参见第 72 页中的“USB 和 1394 (FireWire) 支持增强功能”。

- 从 Solaris 10 1/06 发行版开始，`vol` 可自动挂载热插拔设备。有关更多信息，请参见第 125 页中的“`vol` 用于识别热插拔 USB 设备”。
- 支持使用标准 MS-DOS 或 Windows (FAT) 文件系统的 USB 存储设备。
- 可以使用用户友好的 `rmformat` 命令来创建片，还可以使用 `fdisk` 命令对 USB 设备进行分区，但决不能使用 `format` 实用程序或 `rmformat -F` 命令以物理方式格式化 USB 驱动器。
- 可使用 `rmformat` 命令显示插入了介质的所有 USB 设备。有关示例，请参见第 147 页中的“如何显示 USB 设备信息”。
- 由于不再需要 `mount` 命令，因此非超级用户现在可以访问 USB 存储设备。设备将通过 `vol` 自动挂载并且位于 `/rmdisk` 目录下。如果在系统关闭时连接了一个新设备，可使用 `boot -r` 命令进行重新配置引导，以便 `vol` 可以识别该设备。
- 可以通过卷管理来管理这些设备，也可以不通过卷管理。
- 可以挂载和访问使用 FAT 文件系统的磁盘。例如：

```
mount -F pcfs /dev/dsk/c2t0d0s0:c /mnt
```

- 现在，除了支持 LOG SENSE 页面的 USB 存储设备之外，其他所有 USB 存储设备都会进行电源管理。包含 LOG SENSE 页面的设备通常是通过 USB-to-SCSI 桥接设备进行连接的 SCSI 驱动器。
- 使用 USB 海量存储设备时，应用程序的工作方式可能有所不同。如果要将应用程序用于 USB 存储设备，请牢记以下问题：
 - 应用程序可能对介质的存储容量作出不正确的假定，因为以前仅有较小存储容量的设备（如软盘）才是可移除设备。
 - 当应用程序发出请求，要求设备弹出介质，而此移除操作却不适用于该设备（如硬盘驱动器）时，请求会成功，但不会产生任何结果。
 - 如果要使用以前的 Solaris 发行版中的性能，即不将所有的 USB 海量存储设备都视为可移除介质设备，则可通过更新 `/kernel/drv/scsa2usb.conf` 文件来强制使用旧性能。

有关使用 USB 海量存储设备的更多信息，请参见 `scsa2usb(7D)`。

使用 USB 软盘设备

USB 软盘设备显示为可移除介质设备。USB 软盘设备不是通过 `fd`（软盘）驱动程序进行管理的。发出旨在用于 `fd`（本机软盘）驱动程序的操作调用的应用程序将失败。仅发出 `read(2)` 和 `write(2)` 调用的应用程序将成功。其他应用程序（如 SunPCI 和 `rmformat`）也将成功。

卷管理 (vold) 会将 USB 软盘设备视为 SCSI 可移除介质设备。通过卷管理可在 /rmdisk 目录下访问该设备。

有关如何使用 USB 软盘设备的更多信息，请参见第 1 章。

使用不兼容的 USB 海量存储设备

即使某些设备未将其自身标识为与 USB 海量存储类兼容，或以不正确的方式标识其自身，USB 海量存储驱动程序也仍可能支持这些设备。scsa2usb.conf 文件包含一个属性覆盖列表，其中列有供应商 ID、产品 ID、用于匹配海量存储设备的修订版，以及用于覆盖缺省设备属性的字段。缺省情况下，会注释掉此列表中的项。要启用对特定设备的支持，可以复制并取消注释这些项。

如果将 USB 海量存储设备连接至运行此 Solaris 发行版的系统，但该系统无法使用它，则可以检查 /kernel/drv/scsa2usb.conf 文件，查看是否存在与该设备对应的注释掉的项。根据 scsa2usb.conf 文件中给出的信息，查看是否可以使用覆盖信息支持特定设备。

有关推荐的 USB 海量存储设备的列表，请转至：

http://www.sun.com/io_technologies/USB.html

有关更多信息，请参见 scsa2usb(7D)。

▼ 如何使用不兼容的 USB 海量存储设备

- 1 成为超级用户。
- 2 向 /kernel/drv/scsa2usb.conf 文件中添加项。

以下项适用于 USB 内存条。

```
attribute-override-list = "vid=* reduced-cmd-support=true";
```

- 3 重新引导系统或执行以下操作：

- a. 手动更新驱动程序。

```
# update_drv -f scsa2usb
```

- b. 移除该设备。

- c. 添加该设备。

热插拔 USB 海量存储设备

热插拔设备是指在未关闭操作系统或未关闭系统电源的情况下添加或移除设备。所有的 USB 设备都是可热插拔设备。

可移除介质管理器 (vold) 现在可以识别热插拔设备。一旦插入该设备，vold 便会在几秒钟内尝试对其进行挂载。如果未发生任何情况，请查看该设备是否已挂载。

确保 vold 正在运行。

```
# svcsvcs voldfs

STATE          STIME          FMRI

online          10:39:12 svc:/system/filesystem/voldfs:default
```

如果文件系统有效并且可由 vold 进行识别，则可以从该设备将其挂载。

如果无法挂载，请停止 vold。

```
# svcadm disable voldfs
```

然后，尝试手动挂载。

热移除设备之前，应在 eject -n 命令的别名中找到该设备的名称。然后，弹出该设备的介质。如果不执行此操作，vold 仍会释放该设备，并且端口可再次使用，但是该设备中的文件系统可能已被破坏。

热插拔 USB 设备时，系统的设备分层结构中会立即显示该设备，如 prtconf 命令输出所示。移除 USB 设备时，如果未使用该设备，则会从系统的设备分层结构中删除该设备。

如果拔除设备时正在使用该设备，则该设备节点仍然存在，但控制该设备的驱动程序将停止该设备中的所有活动。向此设备发出的任何新的 I/O 活动都会返回错误。

在这种情况下，系统会提示插入原始设备。如果该设备不再可用，请停止应用程序。几秒钟后，端口将再次可用。

注-移除活动的设备或打开的设备可能会损害数据完整性。移除设备之前应始终先将其关闭，但连接的键盘和鼠标除外（可在活动状态时将其移除）。

以只读方式挂载 USB 海量存储设备

通过更改 /etc/rmmount.conf 文件中的以下行，可以采用只读方式挂载 USB 海量存储设备：

```
mount * hsfs udfs ufs -o nosuid
```

更改为：

```
mount * hsfs udfs ufs -o nosuid,ro
```

然后重新启动 `vold`：

```
# svcadm disable volfs
```

```
# svcadm enable volfs
```

此时会以只读方式挂载包含这些文件系统的可移除设备。

▼ 如何在不运行 `vold` 的情况下添加 **USB 海量存储设备**

- 1 （可选）有关禁用 `vold` 的信息，请参见第 146 页中的“[如何在不运行 `vold` 的情况下准备使用 USB 海量存储设备](#)”。
- 2 检验 `vold` 是否未运行。

```
# svcs volfs
```

```
STATE          STIME      FMRI
```

```
disabled       10:39:12  svc:/system/filesystem/volfs:default
```

- 3 连接 USB 海量存储设备。

- 4 检验是否已添加 USB 设备。

查找 USB 磁盘设备链路，这些链路可能在非 USB 存储设备的设备链路中，如下所示：

```
$ rmformat
```

```
Looking for devices...
```

```
1. Logical Node: /dev/rdisk/c3t0d0p0
```

```
Physical Node: /pci@0,0/pci108e,4131@2,1/storage@4/disk@0,0
```

```
Connected Device: USB2.0    Flash Disk        2.00
```

```
Device Type: Removable
```

▼ 如何添加 **USB 相机**

如果相机的介质使用 PCFS 文件系统，则 `vold` 会自动挂载该相机。如果该设备未绑定至 `scsa2usb` 驱动程序，请使用 `libusb` 应用程序来传送图片。有关更多信息，请参阅 `/usr/sfw/share/doc/libusb/libusb.txt`。

1 成为超级用户。**2 插入并打开 USB 相机。**

系统将为相机创建一个逻辑设备。插入相机后，会将输出写入 `/var/adm/messages` 文件，以确认设备的连接。系统会将相机视为存储设备。

3 检查写入 `/var/adm/messages` 文件的输出。

```
# more /var/adm/messages
```

通过检查此输出，可以确定已创建的逻辑设备，这样便可以使用该设备来访问图像。其输入与以下内容类似：

```
Jul 15 09:53:35 buffy usba: [ID 349649 kern.info]    OLYMPUS, C-3040ZOOM,
000153719068
```

```
Jul 15 09:53:35 buffy genunix: [ID 936769 kern.info] scsa2usb1 is
/pci@0,0/pci925,1234@7,2/storage@2
```

```
Jul 15 09:53:36 buffy scsi: [ID 193665 kern.info] sd3 at scsa2usb1:
target 0 lun 0
```

请通过执行以下操作，使该设备与可挂载的 `/dev/dsk` 链路项匹配：

```
# ls -l /dev/dsk/c*0 | grep /pci@0,0/pci925,1234@7,2/storage@2

lrwxrwxrwx  1 root    root          58 Jun 30  2004 c3t0d0p0 ->
../../../../devices/pci@0,0/pci925,1234@7,2/storage@2/disk@0,0:a
```

4 挂载 USB 相机文件系统。

该相机的文件系统很可能是 PCFS 文件系统。如果文件系统是 PCFS，则该文件系统应由 `vold` 自动挂载。

要在 x86 系统上手动挂载文件系统，可以使用如下语法：

```
# mount -F pcfs /dev/dsk/c3t0d0p0:c /mnt
```

要在 SPARC 系统上手动挂载文件系统，可以使用如下语法：

```
# mount -F pcfs /dev/dsk/c3t0d0s0:c /mnt
```

有关挂载文件系统的信息，请参见第 19 章。

有关挂载其他 PCFS 文件系统的信息，请参见 `mount_pcfs(1M)`。

- 5 检验图像文件是否可用。

例如：

```
# ls /mnt/DCIM/100OLYMP/
```

```
P7220001.JPG* P7220003.JPG* P7220005.JPG*
```

```
P7220002.JPG* P7220004.JPG* P7220006.JPG*
```

- 6 查看并处理 USB 相机创建的图像文件。

例如：

```
# /usr/dt/bin/sdtimage P7220001.JPG &
```

- 7 断开相机连接之前，取消挂载文件系统。

例如：

```
# umount /mnt
```

- 8 （可选的）关闭并断开相机连接。

▼ 如何在不运行 `vold` 的情况下移除 USB 海量存储设备

- 1 （可选）有关禁用 `vold` 的信息，请参见第 146 页中的“[如何在不运行 `vold` 的情况下准备使用 USB 海量存储设备](#)”。

- 2 成为超级用户。

- 3 检验 `vold` 是否未运行。

```
# svcs volfs
```

```
STATE          STIME      FMRI
```

```
disabled       Sep_29    svc:/system/filesystem/volfs:default
```

- 4 停止使用该设备的所有活动应用程序。

- 5 取消挂载该设备。

- 6 移除该设备。

在运行 `vold` 的情况下准备使用 USB 海量存储设备

无论是否使用卷管理，都可以访问有关可移除介质的信息。有关使用 GNOME 的文件管理器访问可移除介质相关信息的信息，请参见 GNOME 桌面文档。

格式化 USB 设备后，通常会将该设备挂载于 `/rmdisk/label` 目录下。有关配置可移除存储设备的更多信息，请参见 `rmmount.conf(4)` 或 `vold.conf(4)`。

设备节点是在 `/vol/dev` 目录下创建的。有关更多信息，请参见 `scsa2usb(7D)`。

以下过程介绍如何在不运行 `vold` 的情况下管理 USB 海量存储设备。对于字符设备，对应的设备节点是在 `/dev/rdsk` 目录下创建的，而对于块设备，对应的设备节点是在 `/dev/dsk` 目录下创建的。设备链路是在热插拔设备时创建的。有关更多信息，请参见 `scsa2usb(7D)`。

▼ 如何在不运行 vold 的情况下准备使用 USB 海量存储设备

可以在不运行卷管理 (`vold`) 的情况下使用 USB 海量存储设备。可通过发出以下命令停止 `vold`：

```
# svcadm disable volfs
```

或者，使用以下过程使 `vold` 保持运行，但不使用 `vold` 来注册 USB 海量存储设备。

1 成为超级用户。

2 检验 `vold` 是否未运行。

```
# svcs volfs
```

```
STATE          STIME      FMRI
disabled        10:28:23  svc:/system/filesystem/volfs:default
```

3 通过对 `/etc/vold.conf` 文件中的以下行进行注释，删除 USB 海量存储设备的卷管理注册功能。

```
# use rmdisk drive /dev/rdsk/c*s2 dev_rmdisk.so rmdisk%d
```

4 注释该行后，重新启动 `vold`。

```
# svcadm enable volfs
```



注意 – 如果注释掉该行，且其他可移除设备（如 SCSI 或 Peerless）位于系统上，则这些设备的 `vold` 注册功能也会被禁用。

有关更多信息，请参见 `vold.conf(4)`。

▼ 如何显示 USB 设备信息

► 显示有关 USB 设备的信息。

例如，使用 `prtconf` 命令可显示 USB 设备信息。本示例中的 `prtconf` 输出已截断为仅显示 USB 设备信息。

\$ prtconf

```
usb, instance #0

hub, instance #2

device, instance #8

interface (driver not attached)

printer (driver not attached)

mouse, instance #14

device, instance #9

keyboard, instance #15

mouse, instance #16

storage, instance #7

disk (driver not attached)

communications, instance #10

modem (driver not attached)

data (driver not attached)

storage, instance #0

disk (driver not attached)

storage, instance #1

disk (driver not attached)
```

例如，使用 `rmformat` 命令可显示 USB 设备信息。

\$ rmformat

Looking for devices...

1. Volmgt Node: /vol/dev/aliases/rmdisk3

Logical Node: /dev/rdisk/c1t0d3s2

Physical Node: /pci@1e,600000/usb@a/storage@2/disk@0,3

Connected Device: SMSC 223 U HS-SD/MMC 1.95

Device Type: Removable

2. Volmgt Node: /vol/dev/aliases/cdrom0

Logical Node: /dev/rdisk/c2t2d0s2

Physical Node: /pci@1e,600000/ide@d/sd@2,0

Connected Device: JLMS XJ-HD166S D3S4

Device Type: DVD Reader

3. Volmgt Node: /vol/dev/aliases/rmdisk4

Logical Node: /dev/rdisk/c4t0d0s2

Physical Node: /pci@1e,600000/usb@a/storage@1/disk@0,0

Connected Device: IC25T048 ATDA05-0 DA80

Device Type: Removable

▼ 如何在不运行 **vold** 的情况下在 **USB 海量存储设备** 中创建文件系统

使用此过程可向 USB 海量存储设备中添加文件系统。

必须先格式化 USB 软盘，然后才能向其中添加文件系统。所有其他的 USB 海量存储设备只需 PCFS 或 UFS 文件系统即可使用。格式化 USB 软盘时，请牢记以下要点：

- 除在 USB 软盘上以外，请勿使用 **rmformat -F**。
- 如果缺省的片不可接受，请使用 **rmformat -s** 命令来创建片。使用 **fdisk** 实用程序对 USB 设备进行分区（如果需要）。有关逐步说明，请参见：
 - [第 151 页中的“如何在不运行 **vold** 的情况下在 USB 海量存储设备中修改分区和创建 PCFS 文件系统”](#)

- 第 157 页中的“如何在 `不运行 vold` 的情况下在 USB 海量存储设备中创建 Solaris 分区和修改片”

注 – 仅在需要格式化 USB 软盘时，才执行步骤 5–6。

- 1 (可选) 有关禁用 `vold` 的信息，请参见第 146 页中的“如何在 `不运行 vold` 的情况下准备使用 USB 海量存储设备”。

- 2 检验 `vold` 是否未运行。

```
# svcs volfs
```

```
STATE          STIME      FMRI

disable        10:39:12 svc:/system/filesystem/volfs:default
```

- 3 向系统中添加 USB 设备。

有关热插拔 USB 设备的信息，请参见：

- 第 142 页中的“热插拔 USB 海量存储设备”
- 第 167 页中的“使用 `cfgadm` 命令热插拔 USB 设备（任务列表）”

- 4 (可选) 标识 USB 设备。

以下示例标识了 USB 软盘设备。

```
# cd /dev/rdisk
```

```
# ls -l c*0 | grep usb
```

```
lrwxrwxrwx  1 root  root   55 Mar  5 10:35 c2t0d0s0 ->
```

```
../../../../devices/pci@1f,0/usb@c,3/storage@3/disk@0,0:a,raw
```

在本示例中，软盘设备为 `c2t0d0s0`。

- 5 将软盘插入软盘驱动器中（如果必要）。

- 6 格式化该软盘（如果必要）。

```
% rmformat -Flong raw-device
```

例如，在 SPARC 系统上：

```
% rmformat -Flong /dev/rdisk/c2t0d0s2
```

例如，在 x86 系统上：

```
% rmformat -Flong /dev/rdisk/c3t0d0p0
```

7 确定文件系统类型并选择以下操作之一：

- 创建 PCFS 文件系统。

```
# mkfs -F pcfs -o nofdisk,size=size raw-device
```

以 512 字节块为单位指定 `-size` 选项。

以下示例说明如何在 SPARC 系统中的 1.4 MB 软盘上创建 PCFS 文件系统：

```
# mkfs -F pcfs -o nofdisk,size=2880 /dev/rdisk/c4t0d0s2
```

以下示例说明如何在 x86 系统中的 1.4 MB 软盘上创建 PCFS 文件系统：

```
# mkfs -F pcfs -o nofdisk,size=2880 /dev/rdisk/c4t0d0p0
```

以下示例说明如何在 SPARC 系统上的 100 MB USB 内存条上创建 PCFS 文件系统：

```
# mkfs -F pcfs /dev/rdisk/c5t0d0s2:c
```

以下示例说明如何在 x86 系统上的 100 MB USB 内存条上创建 PCFS 文件系统：

```
# mkfs -F pcfs /dev/rdisk/c5t0d0p0:c
```

完成此命令可能要用几分钟时间。

- 创建 UFS 文件系统。

```
# newfs raw-device
```

例如：

```
# newfs /dev/rdisk/c4t0d0s7
```

对于大容量 USB 硬盘，请考虑使用 `newfs -f 4096` 选项或 `newfs -T` 选项。

注 - 由于软盘的存储容量有限，因此 UFS 文件系统开销会占用软盘上的大部分空间。

有关在 USB 海量存储设备中创建 PCFS 文件和修改片的详细示例，请参见以下两个过程。

▼ 如何在`不运行 vold`的情况下在**USB 海量存储设备**中修改分区和创建**PCFS**文件系统

以下步骤介绍如何删除现有分区，创建新分区，然后在 USB 设备中创建 PCFS 文件系统。执行此任务之前，请确保先备份所有数据。

- 1 成为超级用户。
- 2 启动 `fdisk` 实用程序。

例如：

```
# fdisk /dev/rdisk/c3t0d0p0
```

- 3 通过选择选项 3 来删除分区。

例如：

Total disk size is 29 cylinders

Cylinder size is 2048 (512 byte) blocks

Cylinders						
Partition	Status	Type	Start	End	Length	%
=====	=====	=====	=====	=====	=====	=====
1	Active	Solaris2	1	28	28	97

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: 3

4 选择要删除的分区号。

例如：

Total disk size is 29 cylinders

Cylinder size is 2048 (512 byte) blocks

Cylinders						
Partition	Status	Type	Start	End	Length	%
=====	=====	=====	=====	=====	=====	=====
1	Active	Solaris2	1	28	28	97

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)

Specify the partition number to delete (or enter 0 to exit): 1

Partition deleted.

5 创建分区。

例如：

Total disk size is 29 cylinders

Cylinder size is 2048 (512 byte) blocks

Cylinders

Partition	Status	Type	Start	End	Length	%
=====	=====	=====	=====	=====	=====	=====

WARNING: no partitions are defined!

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: 1

6 选择 FAT32 分区类型。

Total disk size is 29 cylinders

Cylinder size is 2048 (512 byte) blocks

Cylinders						
Partition	Status	Type	Start	End	Length	%
=====	=====	=====	=====	=====	=====	=====

WARNING: no partitions are defined!

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)

Select the partition type to create:

- 1=SOLARIS2 2=UNIX 3=PCIXOS 4=Other
- 5=DOS12 6=DOS16 7=DOSEXT 8=DOSBIG
- 9=DOS16LBA A=x86 Boot B=Diagnostic C=FAT32
- D=FAT32LBA E=DOSEXTLBA F=EFI 0=Exit? **c**

7 指定此分区所使用的磁盘百分比。

Total disk size is 29 cylinders

Cylinder size is 2048 (512 byte) blocks

Cylinders						
Partition	Status	Type	Start	End	Length	%
=====	=====	=====	=====	===	=====	===

WARNING: no partitions are defined!

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)

Select the partition type to create:

Specify the percentage of disk to use for this partition (or type "c" to specify the size in cylinders). **100**

8 选择新分区应为活动分区还是非活动分区。

Total disk size is 29 cylinders

Cylinder size is 2048 (512 byte) blocks

Cylinders						
Partition	Status	Type	Start	End	Length	%
=====	=====	=====	=====	===	=====	===

WARNING: no partitions are defined!

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)

Select the partition type to create:

Should this become the active partition? If yes, it will be activated
each time the computer is reset or turned on.
Please type "y" or "n". n

9 更新磁盘配置并退出。

Total disk size is 29 cylinders
Cylinder size is 2048 (512 byte) blocks

Cylinders						
Partition	Status	Type	Start	End	Length	%
=====	=====	=====	=====	=====	=====	=====
1		Win95 FAT32	1	28	28	97

- 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

10 在此分区中创建 PCFS 文件系统。

例如：

```
# mkfs -F pcfs -o fat=32 /dev/rdisk/c3t0d0p0:c
```

Construct a new FAT file system on /dev/rdisk/c3t0d0p0:c: (y/n)? y

▼ 如何在不运行 `vold` 的情况下在 **USB 海量存储设备** 中创建 **Solaris** 分区和修改片

以下步骤说明如何创建 Solaris 分区和修改片。

有关禁用 `vold` 的信息，请参见第 146 页中的“如何在不运行 `vold` 的情况下准备使用 **USB 海量存储设备**”。

执行此任务之前，请确保先备份所有数据。

- 1 成为超级用户。
- 2 启动 `fdisk` 实用程序。

例如：

```
# fdisk /dev/rdisk/c5t0d0s2
```

No fdisk table exists. The default partition for the disk is:

```
a 100% "SOLARIS System" partition
```

Type "y" to accept the default partition, otherwise type "n" to edit the partition table.

```
y
```

- 3 显示当前的片。

例如：

```
# prtvtoc /dev/rdisk/c5t0d0s2
```

```
* /dev/rdisk/c5t0d0s2 partition map
```

```
*
```

```
* Dimensions:
```

```
*      512 bytes/sector
```

```
*      63 sectors/track
```

```
*      255 tracks/cylinder
```

```
* 16065 sectors/cylinder

* 5836 cylinders

* 5836 accessible cylinders

*

* Flags:

* 1: unmountable

* 10: read-only

*

*                               First      Sector      Last

* Partition  Tag  Flags      Sector      Count      Sector  Mount Directory

      0        0    00          0  93755340  93755339

      2        0    00          0  93755340  93755339
```

4 创建包含片信息的文本文件。

例如：

```
slices: 0 = 0, 5GB, "wm", "home" :
```

```
1 = 8225280000, 6GB :
```

```
2 = 0, 44GB, "wm", "backup" :
```

```
6 = 16450560000, 15GB
```

请确保每个片都从柱面边界开始。例如，片 1 的初始大小为 822280000 字节，即柱面大小与 1000 的乘积（以字节为单位）。

有关更多信息，请参见 `rmformat(1)` 中的 `-s` 选项说明。

5 通过包含以上创建的片文件来创建片。

例如：

```
# rmformat -s slice_file /dev/rdisk/c5t0d0s2
```

6 查看新的片信息。

例如：

```
# prtvtoc /dev/rdisk/c5t0d0s2
```

```
* /dev/rdisk/c5t0d0s2 partition map
*
* Dimensions:
*     512 bytes/sector
*     63 sectors/track
*     255 tracks/cylinder
*    16065 sectors/cylinder
*     5836 cylinders
*     5836 accessible cylinders
*
* Flags:
*     1: unmountable
*    10: read-only
*
* Unallocated space:
*
*      First      Sector      Last
*      Sector      Count      Sector
*
*    10485760    5579240    16064999
*
*    28647912    3482088    32129999
*
*    63587280    30168060    93755339
*
*
*                               First      Sector      Last
*
* Partition Tag  Flags      Sector      Count      Sector Mount Directory
*
*         0      8      00          0    10485760    10485759
```

1	3	01	16065000	12582912	28647911
2	5	00	0	92274688	92274687
6	4	00	32130000	31457280	63587279

▼ 如何在`不运行 vold`的情况下挂载或取消挂载 **USB 海量存储设备**

- 1 有关禁用 `vold` 的信息，请参见第 146 页中的 “如何在`不运行 vold`的情况下准备使用 **USB 海量存储设备**”。
- 2 成为超级用户。

- 3 检验 `vold` 是否未运行。

```
# svcs volfs

STATE          STIME      FMRI

disabled       10:39:12  svc:/system/filesystem/volfs:default
```

- 4 （可选）标识该设备。

```
例如：

# cd /dev/rdisk

# ls -l c*0 | grep usb

lrwxrwxrwx    1 root  root   55 Mar  5 10:35 c2t0d0s0 ->
../../../../devices/pci@1f,0/usb@c,3/storage@3/disk@0,0:a,raw

在本示例中，软盘设备为 c2t0d0s0。
```

- 5 选择以下操作之一以挂载或取消挂载 **USB 海量存储设备**：

- 挂载 USB 海量存储设备。

```
# mount [ -F fstype ] block-device mount-point

本示例说明如何挂载包含 UFS 文件系统的设备：
```

```
# mount /dev/dsk/c1t0d0s2 /mnt

本示例说明如何在 SPARC 系统上挂载包含 PCFS 文件系统的设备：
```



```
# mount -F pcfs /dev/dsk/c3t0d0s2:c /mnt
```

本示例说明如何在 x86 系统上挂载包含 PCFS 文件系统的设备：

```
# mount -F pcfs /dev/dsk/c3t0d0p0:c /mnt
```

本示例说明如何挂载包含只读 HSFS 文件系统的 CD：

```
# mount -F hsfs -o ro /dev/dsk/c1t0d0s2 /mnt
```

- 取消挂载 USB 海量存储设备。

首先，确保无人正在使用该设备中的文件系统。

例如：

```
# fuser -c -u /mnt
```

```
# umount /mnt
```

- 6 弹出该设备（对于 CD 或软盘设备，此操作可选）。

```
# eject /dev/[r]dsk/cntndnsn
```

例如：

```
# eject /dev/rdisk/c1t0d0s2
```

USB 海量存储设备的疑难解答提示

如果在添加或移除 USB 海量存储设备时遇到问题，请牢记以下提示。

请检查 `/var/adm/messages` 文件中枚举该设备时出现的故障。对于枚举故障，可能需要插入 USB 集线器，或移除集线器并将其直接连接至根 USB 集线器。

- 如果在系统关闭时添加或移除 USB 设备，则必须执行重新配置引导。

```
ok boot -r
```

如果系统正在运行时访问连接的设备遇到问题，请尝试以下命令：

```
# devfsadm
```

- 如果因暂停操作关闭了系统电源，请勿来回移动设备。有关更多信息，请参见第 135 页中的“SPARC:USB 电源管理”。
- 如果应用程序在设备使用时已将其移除并且该设备不再可用，请停止应用程序。使用 `prtconf` 命令可查看是否已删除了设备节点。

禁用特定的 USB 驱动程序

通过禁用 USB 设备的客户机驱动程序，可以禁用特定类型的 USB 设备。例如，通过禁用驱动 USB 打印机的 `usbprn` 驱动程序，可以禁用这些打印机。禁用 `usbprn` 不会影响其他种类的设备，如 USB 存储设备。

下表提供了一些 USB 设备类型及其对应的驱动程序。

设备类型	要禁用的驱动程序
音频	<code>usb_ac</code> 和 <code>usb_as</code>
HID（通常为键盘和鼠标）	<code>hid</code>
存储	<code>scsa2usb</code>
打印机	<code>usbprn</code>
串行	<code>usbser_edge</code>

如果禁用仍与系统连接的 USB 设备的驱动程序，则会显示类似如下的控制台消息：

```
usb10: WARNING: usba: no driver found for device name
```

▼ 如何禁用特定的 USB 驱动程序

- 1 成为超级用户。
- 2 从 `/etc/system` 文件中排除驱动程序别名项。
例如，包含以下排除语句，用于排除 `usbprn` 驱动程序。
`exclude: usbprn`
- 3 重新引导系统。
`# init 6`

▼ 如何删除未使用的 USB 设备链路

如果在系统电源关闭时移除 USB 设备，请使用此过程。在系统电源关闭时移除 USB 设备可能会保留不存在设备的设备链路。

- 1 成为超级用户。
- 2 关闭可能正在访问该设备的所有应用程序。

3 删除特定 USB 类的未使用的链路。

例如：

```
# devfsadm -C -c audio
```

或者，直接删除悬而未决的链路：

```
# devfsadm -C
```

使用 USB 音频设备（任务列表）

任务	说明	参考
添加 USB 音频设备。	添加 USB 麦克风和扬声器。	第 165 页中的“如何添加 USB 音频设备”
确定系统的主音频设备。	确定哪个音频设备是主音频设备。	第 165 页中的“如何确定系统的主音频设备”
更改主 USB 音频设备。	如果移除或更改 USB 音频设备，则可能需要使某个音频设备成为主音频设备。	第 167 页中的“如何更改主 USB 音频设备”
删除未使用的 USB 设备链路。	如果在系统电源关闭时移除 USB 音频设备，则 <code>/dev/audio</code> 设备可能会指向不存在的 <code>/dev/sound/*</code> 设备。	第 162 页中的“如何删除未使用的 USB 设备链路”
解决 USB 音频问题。	如果 USB 扬声器无法发声，请使用本部分来解决此问题。	第 167 页中的“USB 音频设备问题疑难解答”

使用 USB 音频设备

有关特定 Solaris 发行版中的 USB 音频支持的信息，请参见[第 125 页中的“Solaris 对 USB 设备的支持”](#)。

此 Solaris 发行版提供了由一对协作驱动程序 `usb_ac` 和 `usb_as` 实现的 USB 音频支持。音频控制器驱动程序 `usb_ac` 是与 Solaris USB 体系结构兼容的客户机驱动程序，用于向用户应用程序提供控制接口。音频流驱动程序 `usb_as` 可在播放和录制期间处理音频数据消息。该驱动程序将设置样例频率和精度，并对来自 `usb_ac` 驱动程序请求进行编码。这两个驱动程序都符合 USB 音频类 1.0 规范。

某些音频设备可以在软件控制下设置音量。为了管理此功能，可以将 STREAMS 模块 `usb_ah` 推送至 HID 驱动程序的顶部。

Solaris 支持仅播放、仅录制或同时播放和录制的 USB 音频设备，还支持热插拔 USB 音频设备。

- 具有 USB 连接器的 SPARC Ultra™ 和 x86 平台支持 USB 音频设备。
- Solaris 8 10/01、Solaris 8 2/02 或 Solaris 9 发行版支持的 USB 音频设备必须支持固定的 44100 Hz 或 48000 Hz 抽样频率以进行播放或录制。Solaris 10 发行版中不再要求 44100 Hz 或 48000 Hz 抽样频率。
- 有关完全支持的音频数据格式的信息，请参见 `usb_ac(7D)`。

主音频设备为 `/dev/audio`。通过使用以下命令，可以检验 `/dev/audio` 是否指向 USB 音频：

```
% mixerctl
```

```
Device /dev/audiocctl:
```

```
    Name      = USB Audio
```

```
    Version = 1.0
```

```
    Config  = external
```

```
Audio mixer for /dev/audiocctl is enabled
```

连接 USB 音频设备后，可以通过 `/dev/sound/N` 设备链路使用 `audioplay` 和 `audiorecord` 命令对其进行访问。

请注意，`/dev/audio` 和 `/dev/sound/N` 设备可能是指扬声器、麦克风或组合设备。如果引用不正确的设备类型，则该命令将失败。例如，如果尝试将其与麦克风一同使用，则 `audioplay` 命令将失败。

对于大多数 Sun 音频应用程序（如 `audioplay` 和 `audiorecord`），通过设置 `AUDIODEV` shell 变量或为这些命令指定 `-d` 选项，可以选择特定的缺省音频设备。但是，设置 `AUDIODEV` 不适用于将 `/dev/audio` 硬编码为音频文件的第三方应用程序。

插入 USB 音频设备时，如果没有正在使用 `/dev/audio`，则该设备将自动成为主音频设备 `/dev/audio`。有关将 `/dev/audio` 从板载音频更改为 USB 音频或从 USB 音频更改为板载音频的说明，请参阅第 167 页中的“[如何更改主 USB 音频设备](#)”和 `usb_ac(7D)`。

热插拔多个 USB 音频设备

如果将 USB 音频设备插入系统，该设备将成为主音频设备 `/dev/audio`。即使在重新引导系统后，该设备也仍为主音频设备。如果插入其他 USB 音频设备，则最后一个设备将成为主音频设备。

有关 USB 音频设备问题疑难解答的其他信息，请参见 `usb_ac(7D)`。

▼ 如何添加 USB 音频设备

1 插入 USB 扬声器。

主音频设备 `/dev/audio` 指向 USB 扬声器。

```
% ls -l /dev/audio
```

```
lrwxrwxrwx  1 root    root      10 Feb 13 08:46 /dev/audio -> usb/audio0
```

2 （可选）移除扬声器。然后，重新插入扬声器。

如果移除扬声器，则 `/dev/audio` 设备将恢复回板载音频。

```
% ls -l /dev/audio
```

```
lrwxrwxrwx  1 root    root      7 Feb 13 08:47 /dev/audio -> sound/0
```

3 添加 USB 麦克风。

```
% ls -l /dev/audio
```

```
lrwxrwxrwx  1 root    root      10 Feb 13 08:54 /dev/audio -> usb/audio1
```

▼ 如何确定系统的主音频设备

此过程假定已连接 USB 音频设备。

■ 检查系统的新音频链路。

- 使用 `ls` 命令显示系统的新音频链路。

例如：

```
% ls -lt /dev/audio*
```

```
lrwxrwxrwx  1 root  root      7 Jul 23 15:46 /dev/audio -> usb/audio0
```

```
lrwxrwxrwx  1 root  root     10 Jul 23 15:46 /dev/audiocctl ->
```

```
usb/audiocctl0/
```

```
% ls -lt /dev/sound/*
```

```
lrwxrwxrwx  1 root  root     74 Jul 23 15:46 /dev/sound/1 ->
```

```
../../../../devices/pci@1f,4000/usb@5/hub@1/device@3/sound-control@0:...
```

```
lrwxrwxrwx  1 root  root     77 Jul 23 15:46 /dev/sound/1ctl ->
```

```

.././devices/pci@1f,4000/usb@5/hub@1/device@3/sound-control@0:...

lrwxrwxrwx   1 root   other    66 Jul 23 14:21 /dev/sound/0 ->

.././devices/pci@1f,4000/ebus@1/SUNW,CS4231@14,200000:sound, audio

lrwxrwxrwx   1 root   other    69 Jul 23 14:21 /dev/sound/0ctl ->

.././devices/pci@1f,4000/ebus@1/SUNW,CS4231@14,200000:sound,audioctl

%

```

请注意，主音频设备 `/dev/audio` 指向新插入的 USB 音频设备 `/dev/usb/audio0`。

- 还可以使用 `prtconf` 命令来检查系统的 USB 音频设备，并查找 USB 设备信息。

```

% prtconf

.

.

.

usb, instance #0

    hub, instance #0

        mouse, instance #0

        keyboard, instance #1

        device, instance #0

            sound-control, instance #0

            sound, instance #0

            input, instance #0

.

.

.

```

▼ 如何更改主 USB 音频设备

- ▮ 选择以下操作之一可更改主 USB 音频设备。
 - 如果要使板载音频设备成为主音频设备，请移除 USB 音频设备。然后，`/dev/audio` 链接即会指向 `/dev/sound/0` 项。如果 `/dev/sound/0` 项不是主音频设备，请关闭系统并使用 `boot -r` 命令，或以超级用户身份运行 `devfsadm -i` 命令。
 - 如果要使 USB 音频设备成为主音频设备，只需插入 USB 音频设备并检查设备链路即可。

USB 音频设备问题疑难解答

有时，USB 扬声器不会发出任何声音，即使连接了驱动程序并将音量设置为高也是如此。热插拔该设备可能不会更改该行为。

解决方法是关闭然后再打开 USB 扬声器的电源。

音频设备拥有权要点

使用音频设备时，请牢记音频设备拥有权的以下要点：

- 插入 USB 音频设备并登录控制台后，控制台即是 `/dev/*` 项的属主。此情况表明，只要登录控制台，便可使用音频设备。
- 如果插入 USB 音频设备时未登录控制台，则超级用户会成为该设备的属主。但是，如果登录控制台并尝试访问 USB 音频设备，则设备拥有权将归控制台所有。有关更多信息，请参见 `logindevperm(4)`。
- 如果使用 `rlogin` 命令远程进行登录并尝试访问 USB 音频设备，则拥有权不会更改。这表明，未经授权的用户无法通过他人拥有的麦克风侦听会话。

使用 cfgadm 命令热插拔 USB 设备（任务列表）

任务	说明	参考
显示 USB 总线信息。	显示有关 USB 设备和总线的信息。	第 169 页中的“如何显示 USB 总线信息 (cfgadm)”
取消配置 USB 设备。	以逻辑方式取消配置仍通过物理方式连接至系统的 USB 设备。	第 170 页中的“如何取消配置 USB 设备”
配置 USB 设备。	配置先前取消配置的 USB 设备。	第 171 页中的“如何配置 USB 设备”

任务	说明	参考
以逻辑方式断开 USB 设备连接。	如果物理位置远离系统，则可以以逻辑方式断开 USB 设备连接。	第 171 页中的 “如何以逻辑方式断开 USB 设备连接”
以逻辑方式连接 USB 设备。	以逻辑方式连接先前通过逻辑方式断开连接或取消配置的 USB 设备。	第 172 页中的 “如何以逻辑方式连接 USB 设备”
断开 USB 设备子树连接。	断开 USB 设备子树连接，该子树是集线器下的设备分层结构（或树）。	第 172 页中的 “如何以逻辑方式断开 USB 设备子树连接”
重置 USB 设备。	重置 USB 设备，以便以逻辑方式移除和重新创建设备。	第 173 页中的 “如何重置 USB 设备”
更改多重配置 USB 设备的缺省配置。	更改多重配置 USB 设备的缺省配置。	第 173 页中的 “如何更改多重配置 USB 设备的缺省配置”

使用 cfgadm 命令热插拔 USB 设备

可以在正在运行的系统中添加或移除 USB 设备，而不必使用 `cfgadm` 命令。不过，USB 设备也可以**逻辑方式**热插拔，而不必以物理方式移除。在远程位置工作并且需要禁用或重置未正常运行的 USB 设备时，这种方案比较方便。`cfgadm` 命令还可提供显示 USB 设备树（包括制造商和产品信息）的方法。

`cfgadm` 命令可以显示有关**附着点**的信息，附着点是指系统中可以执行动态重新配置操作的位置。附着点由以下几项组成：

- 插卡，表示可以配置到系统中的硬件资源，如 USB 设备。
- 插座，接受插卡的位置，如 USB 端口

附着点由逻辑附着点 ID 和物理附着点 ID (`Ap_Id`) 表示。物理 `Ap_Id` 是附着点的物理路径名。逻辑 `Ap_Id` 是物理 `Ap_Id` 的用户友好替代项。有关 `Ap_Id` 的更多信息，请参见 `cfgadm_usb(1M)`。

`cfgadm` 命令提供了以下 USB 设备状态信息。

插座状态	说明
<code>empty/unconfigured</code>	设备不是通过物理方式连接的。
<code>disconnected/unconfigured</code>	设备以逻辑方式断开连接且不可用，即使仍可以通过物理方式连接设备也是如此。
<code>connected/unconfigured</code>	设备以逻辑方式建立连接但不可用。设备在 <code>prtconf</code> 输出中可见。

插座状态	说明
connected/configured	设备已连接并且可用。

以下各小节介绍如何使用 `cfgadm` 命令通过软件热插拔 USB 设备。这些小节中的所有样例 USB 设备信息都已截断，以便集中阐明相关信息。

▼ 如何显示 USB 总线信息 (cfgadm)

有关使用 `prtconf` 命令显示 USB 配置信息的示例，请参见第 147 页中的“[如何显示 USB 设备信息](#)”。

1 显示 USB 总线信息。

例如：

```
% cfgadm
```

Ap_Id	Type	Receptacle	Occupant	Condition
usb0/4.5	usb-hub	connected	configured	ok
usb0/4.5.1	usb-device	connected	configured	ok
usb0/4.5.2	usb-printer	connected	configured	ok
usb0/4.5.3	usb-mouse	connected	configured	ok
usb0/4.5.4	usb-device	connected	configured	ok
usb0/4.5.5	usb-storage	connected	configured	ok
usb0/4.5.6	usb-communi	connected	configured	ok
usb0/4.5.7	unknown	empty	unconfigured	ok

在前面的示例中，`usb0/4.5.1` 标识连接至二级外部集线器的端口 1 的设备，二级外部集线器连接至一级外部集线器的端口 5，而一级外部集线器连接至第一个 USB 控制器的根集线器端口 4。

2 显示特定 USB 设备信息。

例如：

```
% cfgadm -l -s "cols=ap_id:info"
```

Ap_Id	Information
usb0/4.5.1	Mfg: Inside Out Networks Product: Edgeport/421 NConfigs: 1

```
Config: 0 : ...

usb0/4.5.2  Mfg: <undef> Product: <undef>  NConfigs: 1 Config: 0 ...

usb0/4.5.3  Mfg: Mitsumi Product: Apple USB Mouse NConfigs: 1

Config: 0 ...

usb0/4.5.4  Mfg: NMB   Product: NMB USB KB/PS2 M NConfigs: 1 Config: 0

usb0/4.5.5  Mfg: Hagiwara Sys-Com  Product: SmartMedia R/W  NConfigs: 1

Config: 0 : ...

usb0/4.5.6  Mfg: 3Com Inc.  Product: U.S.Robotics 56000 Voice USB Modem

NConfigs: 2 ...

usb0/4.5.7
```

▼ 如何取消配置 USB 设备

可以取消配置仍通过物理方式连接至系统的 USB 设备。但是，无法再将驱动程序连接至该设备。请注意，prtconf 输出中会始终显示 USB 设备，即使已取消配置该设备也是如此。

- 1 成为超级用户。
- 2 取消配置 USB 设备。

例如：

```
# cfigadm -c unconfigure usb0/4.7

Unconfigure the device: /devices/pci@8,700000/usb@5,3/hub@4:4.7

This operation will suspend activity on the USB device

Continue (yes/no)? y
```

- 3 验证是否已取消配置该设备。

例如：

```
# cfigadm
```

Ap_Id	Type	Receptacle	Occupant	Condition
-------	------	------------	----------	-----------

usb0/4.5	usb-hub	connected	configured	ok
usb0/4.5.1	usb-device	connected	configured	ok
usb0/4.5.2	usb-printer	connected	configured	ok
usb0/4.5.3	usb-mouse	connected	configured	ok
usb0/4.5.4	usb-device	connected	configured	ok
usb0/4.5.5	usb-storage	connected	configured	ok
usb0/4.5.6	usb-communi	connected	configured	ok
usb0/4.5.7	unknown	empty	unconfigured	ok
usb0/4.6	usb-storage	connected	configured	ok
usb0/4.7	usb-storage	connected	unconfigured	ok

▼ 如何配置 USB 设备

- 1 成为超级用户。
- 2 配置 USB 设备。
例如：
`cfgadm -c configure usb0/4.7`
- 3 检验是否已配置 USB 设备。
例如：
`cfgadm usb0/4.7`

Ap_Id	Type	Receptacle	Occupant	Condition
usb0/4.7	usb-storage	connected	configured	ok

▼ 如何以逻辑方式断开 USB 设备连接

如果要从系统和 `prtconf` 输出中删除 USB 设备，但物理位置远离系统，则只需以逻辑方式断开 USB 设备连接即可。从物理角度来看，该设备仍然处于连接状态。但是，从逻辑角度来看，对系统而言设备已断开连接、不可用且不可见。

- 1 成为超级用户。

2 断开 USB 设备连接。

例如：

```
# cfgadm -c disconnect -y usb0/4.7
```

3 验证是否已断开该设备的连接。

例如：

```
# cfgadm usb0/4.7
```

Ap_Id	Type	Receptacle	Occupant	Condition
usb0/4.7	unknown	disconnected	unconfigured	ok

▼ **如何以逻辑方式连接 USB 设备**

使用此过程可以逻辑方式连接先前通过逻辑方式断开连接或取消配置的 USB 设备。

1 成为超级用户。

2 连接 USB 设备。

例如：

```
# cfgadm -c configure usb0/4.7
```

3 验证是否已连接该设备。

例如：

```
# cfgadm usb0/4.7
```

Ap_Id	Type	Receptacle	Occupant	Condition
usb0/4.7	usb-storage	connected	configured	ok

对系统而言设备现在可用且可见。

▼ **如何以逻辑方式断开 USB 设备子树连接**

使用此过程可以断开 USB 设备子树连接，该子树是集线器下的设备分层结构（或树）。

1 成为超级用户。

2 删除 USB 设备子树。

例如：

```
# cfgadm -c disconnect -y usb0/4
```

3 检验是否已断开 USB 设备子树连接。

例如：

```
# cfgadm usb0/4
```

Ap_Id	Type	Receptacle	Occupant	Condition
usb0/4	unknown	disconnected	unconfigured	ok

▼ 如何重置 USB 设备

如果 USB 设备行为异常，请使用 `cfgadm` 命令重置该设备，即以逻辑方式删除并重新创建该设备。

- 1 成为超级用户。
- 2 确保设备未被使用。

3 重置设备。

例如：

```
# cfgadm -x usb_reset -y usb0/4.7
```

4 检验是否已连接设备。

例如：

```
# cfgadm usb0/4.7
```

Ap_Id	Type	Receptacle	Occupant	Condition
usb0/4.7	usb-storage	connected	configured	ok

▼ 如何更改多重配置 USB 设备的缺省配置

使用多重配置 USB 设备时，请牢记以下要点：

- USB 设备配置定义了设备向操作系统显示其本身的方式。此方法不同于其他 `cfgadm` 部分中介绍的系统设备配置。
- 某些 USB 设备支持多种配置，但一次只能有一种配置处于活动状态。
- 通过检查 `cfgadm -lv` 输出，可以确定多重配置设备。Nconfigs 将大于 1。
- 缺省 USB 配置为配置 1。当前配置在 `cfgadm -lv` 输出中反映为 Config。
- 只要将设备重新连接至同一个端口，对缺省配置所做的更改在重新引导、热移除以及重新配置设备期间就将始终保持不变。

1 确保设备未被使用。

2 更改缺省 USB 配置。

例如：

```
# cfmadm -x usb_config -o config=2 usb0/4

Setting the device: /devices/pci@1f,0/usb@c,3:4

to USB configuration 2

This operation will suspend activity on the USB device

Continue (yes/no)? yes
```

3 检验设备是否已更改。

例如：

```
# cfmadm -lv usb0/4

Ap_Id  Receptacle  Occupant      Condition  Information  When  Type
      Busy      Phys_Id

usb0/4 connected  unconfigured ok          Mfg: Sun  2000

Product: USB-B0B0 aka Robotech

With 6 EPPS High Clk Mode  NConfigs: 7  Config: 2  : EVAL Board Setup

unavailable

usb-device  n          /devices/pci@1f,0/usb@c,3:4
```

请注意，Config: 现在显示 2。

使用 InfiniBand 设备（概述/任务）

InfiniBand (IB) 是一项新的基于 Solaris 10 发行版中引入的交换光纤结构的 I/O 技术。它为将 I/O 设备连接到主机以及主机到主机通信提供了高带宽、低延迟的互连。

以下是本章中概述信息的列表。

- 第 175 页中的 “InfiniBand 设备概述”
- 第 178 页中的 “动态重新配置 IB 设备 (cfgadm)”

有关与使用 IB 设备关联的过程的信息，请参见以下内容：

- 第 176 页中的 “动态重新配置 IB 设备（任务列表）”
- 第 189 页中的 “将 uDAPL 应用程序接口用于 InfiniBand 设备”

有关动态重新配置和热插拔的常规信息，请参见第 6 章。

InfiniBand 设备概述

IB 设备由 Solaris IB 结点驱动程序来管理。此驱动程序支持 5 种类型的设备：

- IB 端口设备
- IB 虚拟物理附着点 (virtual physical point of attachment, VPPA) 设备
- IB HCA 服务 (HCA_SVC) 设备
- 伪设备
- I/O 控制器 (I/O controller, IOC) 设备

IB 结点驱动程序可查询 Solaris IB 设备管理器 (IB Device Manager, IBDM) 中的服务（在本指南中称为**通信服务**），以枚举 IB 端口设备、HCA_SVC 设备和 IB VPPA 设备。

端口设备会将通信服务绑定至主机通道适配器 (Host Channel Adapter, HCA) 的给定 port#，而 VPPA 设备会将通信服务绑定至 port#、p_key# 组合。HCA_SVC 设备会将通信服务绑定至给定的 HCA。请注意，端口设备和 HCA_SVC 设备始终使用其值为零的 p_key（分区密钥）。端口、HCA_SVC 和 VPPA 设备都是 HCA 的子设备，并且通过 ib.conf 文件进行枚举。有关更多信息，请参见 ib(7D)。

IOC 设备是 IB 结点驱动程序子程序的子程序，同时也是 I/O 单元的一部分。伪设备也是 IB 结点驱动程序子程序的子程序，并且指代提供各自要枚举的配置文件的其它所有设备。有关更多信息，请参见 `ib(4)`。

下表列出了可能的 IB 设备树路径名。

IOC 设备	<code>/ib/ioc@1730000007F510C,1730000007F50</code>
IB 伪设备	<code>/ib/<driver>@<unit-address></code>
IB VPPA 设备	<code>/pci@1f,2000/pci@1/pci15b3,5a44@0/ibport@<port#>,<p_key>,<service></code>
IB HCA_SVC 设备	<code>/pci@1f,2000/pci@1/pci15bc,5a44@0/ibport@0,0,<service></code>
IB 端口设备	<code>/pci@1f,2000/pci@1/pci15b3,5a44@0/ibport@<port#>,0,<service></code>
HCA	<code>/pci@1f,2000/pci@1/pci15b3,5a44@0</code>

请注意，IB HCA_SVC 设备使用零作为 `port#` 和 `p_key` 的值。

下面介绍了上表中的 IB 组件：

- `<services>`通信服务。例如，`ipib` 是 `ibd` 内核客户机驱动程序所使用的通信服务。
- `<p_key>`使用的分区密钥值。
- `<port>`端口号。
- `<unit-address>`是指以其 `driver.conf` 文件中指定的名称命名的 IB 内核客户机驱动程序的属性。有关更多信息，请参见 `driver.conf(4)`。

动态重新配置 IB 设备（任务列表）

任务	说明	参考
显示 IB 设备信息。	显示有关系统中的 IB 设备的信息。	第 178 页中的“如何显示 IB 设备信息”
配置或取消配置 IOC 设备。	选择以下操作之一： 取消配置 IOC 设备。	第 181 页中的“如何取消配置 IOC 设备”

任务	说明	参考
	配置 IOC 设备。	第 181 页中的 “如何配置 IOC 设备”
配置或取消配置端口或 VPPA 设备。	选择以下操作之一： 取消配置端口或 VPPA 设备。 配置端口或 VPPA 设备。	第 182 页中的 “如何取消配置 IB 端口、HCA_SVC 或 VPPA 设备” 第 182 页中的 “如何配置 IB 端口、HCA_SVC 或 VPPA 设备”
配置或取消配置 IB 伪设备。	选择以下操作之一： 取消配置 IB 伪设备。 配置 IB 伪设备。	第 183 页中的 “如何取消配置 IB 伪设备” 第 183 页中的 “如何配置 IB 伪设备”
显示 HCA 的内核 IB 客户机。	您可能需要显示有关 HCA 的内核 IP 客户机的信息，特别是要取消配置 HCA 时。	第 184 页中的 “如何显示 HCA 的内核 IB 客户机”
配置或取消配置 IB HCA。	选择以下操作之一： 取消配置与 HCA 连接的 IB 设备。 配置与 HCA 连接的 IB 设备。	第 185 页中的 “如何取消配置与 HCA 连接的 IB 设备” 第 186 页中的 “配置 IB HCA”
更新 IB p_key 表。	如果 HCA 端口的 p_key 表信息发生更改，则需要通知 IBTF 和 IBDM，以便更新其内部 p_key 数据库。	第 186 页中的 “如何更新 IB p_key 表”
显示 IB 通信服务	显示当前由 IBTF 使用的 IB 通信服务。	第 186 页中的 “如何显示 IB 通信服务”
添加或删除 VPPA 通信服务。	选择以下操作之一： 添加 VPPA 通信服务。 删除 VPPA 通信服务。	第 187 页中的 “如何添加 VPPA 通信服务” 第 188 页中的 “如何删除现有的 IB 端口、HCA_SVC 或 VPPA 通信服务”
更新 IOC 配置。	可以更新所有 IOC 设备节点的属性，也可以更新特定 IOC 的 Ap_Id。	第 188 页中的 “如何更新 IOC 配置”

动态重新配置 IB 设备 (cfgadm)

只能使用 `cfgadm` CLI 配置或取消配置正在运行的系统中的 IB 设备。此命令还提供了显示 IB 光纤结构、管理通信服务以及更新 `p_key` 表数据库的方法。有关更多信息，请参见 `cfgadm_ib(1M)`。

`cfgadm` CLI 可以管理被主机视为整个 IB 光纤结构的动态重新配置（在本指南中称为 DR）。`cfgadm` 操作在端口、VPPA、HCA_SVC、IOC 和伪设备等所有 IB 设备上都受支持。

`cfgadm` 命令可以显示有关附着点 (`Ap_Id`) 的信息，附着点是指系统中可以进行 DR 操作的位置。有关 `cfgadm` 支持的 `Ap_Id` 的详细信息，请参见 `cfgadm_ib.1M`。请注意，所有 IB `Ap_Id` 都显示为 `connected`。

`cfgadm` 命令提供了以下 IB 设备状态信息。

插座状态	说明
connected/configured/ok	设备已连接并且可用。存在 <code>devinfo</code> 节点。
connected/unconfigured/unknown	设备不可用，并且不存在与设备对应的任何 <code>devinfo</code> 节点或设备驱动程序。或者，设备从未配置为供 <code>ib</code> 结点驱动程序使用。设备可能会由 IB 设备管理器识别。

以下各节介绍如何使用 `cfgadm` 命令动态重新配置 (DR) IB 设备。这些小节中的所有的样例 IB 设备的信息都已截断，以便集中介绍相关信息。

▼ 如何显示 IB 设备信息

可以使用 `prtconf` 命令来显示有关 IB 设备的常规信息。例如：

```
$ prtconf

pci, instance #0

    pci15b3,5a44, instance #0

        ibport, instance #253

        ibport, instance #254

        ibport, instance #255

    .
```

```
.  
.  
  
ib, instance #0  
  
ioc, instance #243  
  
ioc, instance #244  
  
ioc, instance #245  
  
ioc, instance #246  
  
ioc, instance #247  
  
ioc, instance #248  
  
ibgen, instance #249
```

在前面的示例中，pci15b3,5a44 是指 IB HCA。
执行以下步骤可显示特定 IB 设备的信息。

- 1 成为超级用户。
- 2 显示 IB 光纤结构信息。

例如：

```
# cfgadm -a
```

Ap_Id	Type	Receptacle	Occupant	Condition
ib	IB-Fabric	connected	configured	ok
hca:1730000008070	IB-HCA	connected	configured	ok
ib::1730000007F5198	IB-IOC	connected	configured	ok
ib::1730000007F5199	IB-IOC	connected	configured	ok
ib::1730000008070,0,hnfs	IB-HCA_SVC	connected	configured	ok
ib::1730000008071,0,sdp	IB-PORT	connected	configured	ok
ib::1730000008072,0,sdp	IB-PORT	connected	configured	ok
ib::1730000008071,8001,ipib	IB-VPPA	connected	configured	ok

```
ib::1730000008072,8001,ipib IB-VPPA    connected  configured  ok

ib::ibgen,0                      IB-PSEUDO  connected  configured  ok

#
```

在前面的示例输出中，组件的描述如下：

Ap_Id ib::1730000008072,0,sdp	标识与端口 2 连接并且绑定至 sdp 服务的 IB 端口设备。
Ap_Id ib::1730000008072,8001,ipib	标识使用 p_key 值 0x8001 与端口 2 连接并且绑定至 ibd 服务的 IB VPPA 设备。
Ap_Id ib:: 1730000008070,0,hnfs	标识绑定至 hnfs 服务的 IB HCA_SVC 设备。
Ap_Id ib::1730000007F5198	标识 IOC 设备。
Ap_Id ib::ibgen,0	标识伪设备。

3 显示特定 IB 设备的信息。

例如，对于 IB VPPA 设备，将显示以下信息：

```
# cfgadm -al -s "cols=ap_id:info" ib::1730000008072,8001,ipib

Ap_Id                      Information

ib::1730000008072,8001,ipib  ipib

#
```

例如，对于 IB HCA 设备，将显示以下信息：

```
# cfgadm -al -s "cols=ap_id:info" hca::1730000008070

Ap_Id                      Information

hca::1730000008070          VID: 0x15b3, PID: 0x5a44, #ports: 0x2,

port1 GUID: 0x1730000008071, port2 GUID: 0x1730000008072

#
```

前面的输出显示了端口号和端口的 GUID。

▼ 如何取消配置 IOC 设备

可以取消配置仍然与系统物理连接的 IB 设备，但决不能向该设备附加驱动程序。

1 成为超级用户。

2 取消配置 IB 设备。

例如：

```
# cfgadm -c unconfigure ib::1730000007F5198

Unconfigure the device: /devices/ib:fabric::1730000007F5198

This operation will suspend activity on the IB device

Continue (yes/no)? y

#
```

3 验证是否已取消配置该设备。

例如：

```
# cfgadm -a ib::1730000007F5198

ib::1730000007F5198      IB-IOC      connected  unconfigured unknown

#
```

▼ 如何配置 IOC 设备

1 成为超级用户。

2 配置 IB 设备。

例如：

```
# cfgadm -yc configure ib::1730000007F5198
```

3 验证是否已配置 IB 设备。

例如：

```
# cfgadm -al ib::1730000007F5198

Ap_Id      Type      Receptacle  Occupant    Condition
ib::1730000007F5198  IB-IOC  connected   configured  ok
```

▼ 如何取消配置 IB 端口、HCA_SVC 或 VPPA 设备

如果要从系统中删除 IB 端口、HCA_SVC 或 VPPA 设备，请执行以下步骤。

以下示例说明如何取消配置 VPPA 设备，但同样的过程也适用于端口设备和 HCA_SVC 设备。

1 成为超级用户。

2 取消配置 IB VPPA 设备。

例如：

```
# cfgadm -c unconfigure ib::1730000007F51,8001,ipib

Unconfigure the device: /devices/ib:fabric::1730000007F51,8001,ipib

This operation will suspend activity on the IB device

Continue (yes/no)? Y

#
```

3 验证是否已断开该设备的连接。

例如：

```
# cfgadm -a ib::1730000007F51,8001,ipib

Ap_Id                      Type      Receptacle Occupant    Condition
ib::1730000007F51,8001,ipib IB-VPPA  connected  unconfigured unknown

#
```

▼ 如何配置 IB 端口、HCA_SVC 或 VPPA 设备

如果要在系统中配置 IB 端口、HCA_SVC 或 VPPA 设备，请执行以下步骤。

以下示例说明如何配置 VPPA 设备，但类似的步骤也可用于配置端口设备和 HCA_SVC 设备。

1 成为超级用户。

2 配置 IB VPPA 设备。

例如：

```
# cfgadm -c configure ib::1730000007F51,8001,ipib
```

3 验证是否已连接该设备。

例如：

```
# cfgadm -a ib::1730000007F51,8001,ipib

Ap_Id                                Type      Receptacle Occupant  Condition
ib::1730000007F51,8001,ipib IB-VPPA   connected  configured ok
```

注 - IB 端口设备和 HCA_SVC 设备的基于 `cfgadm` 的配置或取消配置操作与前面的 IB VPPA 设备示例类似。

▼ 如何取消配置 IB 伪设备

如果要从系统中删除 IB 伪设备，请执行以下步骤。

1 成为超级用户。

2 取消配置 IB 伪设备。

例如：

```
# cfgadm -c unconfigure ib::ibgen,0

Unconfigure the device: /devices/ib:fabric::ibgen,0

This operation will suspend activity on the IB device

Continue (yes/no)? Y

#
```

3 验证是否已断开该设备的连接。

```
# cfgadm -a ib::ibgen,0

Ap_Id                                Type      Receptacle Occupant  Condition
ib::ibgen,0                          IB-PSEUDO connected  unconfigured unknown
```

▼ 如何配置 IB 伪设备

执行以下步骤可配置 IB 伪设备。

1 成为超级用户。

2 配置 IB 伪设备。

例如：

```
# cfgadm -yc configure ib::ibgen,0
```

3 验证是否已连接该设备。

例如：

```
# cfgadm -a ib::ibgen,0
```

Ap_Id	Type	Receptacle	Occupant	Condition
ib::ibgen,0	IB-PSEUDO	connected	configured	ok

▼ 如何显示 HCA 的内核 IB 客户机

可以调用以下 IB `cfgadm` 插件命令来列出使用此 HCA 的内核 IB 客户机。请注意，如果内核 IB 客户机使用其他 HCA，则最后一列会显示 "yes"。不使用 HCA 的 IB 管理器和内核客户机以 Ap_Id “-” 显示。

► 显示 HCA 的内核 IB 客户机。

例如：

```
$ cfgadm -x list_clients hca:1730000007F50
```

Ap_Id	IB Client	Alternate HCA
ib::1730000007F51D0	ibgen	no
ib::1730000007F51D1	ibgen	no
ib::1730000007F51,8001,ipib	ibd	no
ib::ibgen,0	ibgen	no
-	ibdm	no
-	ibmf	no
-	nfs/ib	no
\$		

▼ 如何取消配置与 HCA 连接的 IB 设备

HCA 的实际 DR 超出了 IB cfgadm 插件的范围。尽管如此，仍可以使用基础总线的插件来实现 HCA 的 DR。例如，基于 PCI 的 HCA 可以使用 `cfgadm_pci` 命令。有关更多信息，请参见 `cfgadm_pci(1M)`。

但是，IB cfgadm 插件通过按以下步骤说明列出其内核 IB 客户机，可协助执行 HCA DR。

- 1 成为超级用户。
- 2 列出 HCA 的内核 IB 客户机。

例如：

```
# cfgadm -x list_clients hca:173000007F50
```

Ap_Id	IB Client	Alternate HCA
ib::1730000007F51D0	ibgen	no
ib::1730000007F51D1	ibgen	no
ib::1730000007F51,8001,ipib	ibd	no
ib::ibgen,0	ibgen	no
-	ibdm	no
-	ibmf	no
-	nfs/ib	no

- 3 取消配置不存在备用 HCA 的内核 IB 客户机，如端口设备、VPPA 设备、HCA_SVC 设备或 IOC 设备。

例如：

```
# cfgadm -x unconfig_clients hca:1730000008070
```

```
Unconfigure Clients of HCA /devices/ib:1730000008070

This operation will unconfigure IB clients of this HCA

Continue (yes/no)? y
```

- 4 验证是否已取消配置 HCA 的内核 IB 客户机。

```
# cfgadm -x list_clients hca:173000007F50
```

Ap_Id	IB Client	Alternate HCA
-	ibdm	no
-	ibmf	no
-	nfs/ib	no
#		

配置 IB HCA

调用总线特定的 `cfgadm` 插件可配置 HCA。确切的详细信息超出了本章的范围。

▼ 如何更新 IB p_key 表

如果 HCA 端口的 `p_key` 表信息发生更改（例如，启用或禁用了其他 `p_key`），则需要通知 InfiniBand 传输框架 (InfiniBand Transport Framework, IBTF) 和 IBDM，以便更新其内部 `p_key` 数据库。`cfgadm` 命令可帮助更新 IBTF 和 IBDM 的 `p_key` 数据库。有关更多信息，请参见 `ibt1(7D)` 和 `ibdm(7D)`。

- 1 成为超级用户。
- 2 更新 `p_key` 表。

例如：

```
# cfgadm -x update_pkey_tbls -y ib
```

▼ 如何显示 IB 通信服务

执行以下步骤可显示当前由 IBTF 使用的通信服务。

- 1 成为超级用户。
- 2 显示 IB 通信服务。

例如：

```
# cfgadm -x list_services ib
```

```
Port communication services:
```

```
srp
```

```
VPPA communication services:
```

```
    ibd
```

```
HCA_SVC communication services:
```

```
    hnfs
```

▼ 如何添加 VPPA 通信服务

执行以下步骤可添加新的 VPPA 通信服务。

可以使用类似的步骤来添加新的 HCA_SVC 或端口通信服务。

- 1 成为超级用户。
- 2 添加新的 VPPA 通信服务。

例如：

```
# cfgadm -o comm=vppa,service=new -x add_service ib
```

- 3 验证是否已添加新的服务。

例如：

```
# cfgadm -x list_services ib
```

```
Port communication services:
```

```
    srp
```

```
VPPA communication services:
```

```
    ibd
```

```
    new
```

```
HCA_SVC communication services:
```

```
    nfs_service
```

```
#
```

▼ 如何删除现有的 IB 端口、HCA_SVC 或 VPPA 通信服务

执行以下步骤可删除现有的 IB 端口、HCA_SVC 或 VPPA 通信服务。

- 1 成为超级用户。

- 2 删除 VPPA 通信服务。

例如：

```
# cfgadm -o comm=vppa,service=new -x delete_service ib
```

- 3 验证是否已删除该通信服务。

例如：

```
# cfgadm -x list_services ib
```

Port communication services:

srp

VPPA communication services:

ibd

HCA_SVC communication services:

hnfs

#

▼ 如何更新 IOC 配置

执行以下步骤可更新所有 IOC 设备节点或特定 IOC Ap_Id 的属性。可更新的属性如下：

- port-list
- port-entries
- service-id
- service-name

有关这些属性的更多信息，请参见 [ib\(7D\)](#)。

请注意，如果配置没有更改，则这些属性可能不会更新。以下示例说明如何更新特定 IOC 的配置。如果需要更新所有 IOC 的配置，请指定静态的 `ibAp_Id`，而不是特定的 `IOCAp_Id`。

1 成为超级用户。

2 更新 IOC 的配置。

例如：

```
# cfgadm -x update_ioc_conf ib::1730000007F5198
```

This operation can update properties of IOC devices.

Continue (yes/no)? y

```
#
```

3 通过运行 `prtconf -v` 验证是否已更新这些属性。

将 uDAPL 应用程序接口用于 InfiniBand 设备

用户直接访问编程库 (User Direct Access Programming Library, uDAPL) 是标准的 API，可以通过具有远程直接内存访问 (Remote Direct Memory Access, RDMA) 功能的互连（如 InfiniBand）来提高数据中心应用程序数据消息处理的性能、伸缩性和可靠性。uDAPL 接口由 DAT collaborative 定义。有关 DAT collaborative 的更多信息，请转至以下站点：

<http://www.datcollaborative.org>

Solaris 发行版提供了以下 uDAPL 功能：

- 标准 DAT 注册表库 `libdat`。有关更多信息，请参见 `libdat(3LIB)`。
- 标准注册文件 `dat.conf`。有关更多信息，请参见 `dat.conf(4)`。
- 支持多个，以便每个提供商都在各自的 `service_provider.conf` 文件中指定各自的 uDAPL 库路径、版本号等。有关更多信息，请参见 `service_provider.conf(4)`。
- 管理工具，即 `datadm` 命令，用于配置 `dat.conf`。有关更多信息，请参见 `datadm(1M)`。
- 新的资源控制属性 `project.max-device-locked-memory`，用于控制锁定的物理内存量。
- 使用 IPv4 或 IPv6 地址的命名方案，这些地址利用 IP 基础结构（如 IPv4 中的 ARP 以及 IPv6 中的相邻节点搜索）来进行地址解析。Solaris uDAPL 接口适配器可直接映射到 IPoIB 设备实例。
- 支持 DAT collaborative 团体使用的标准地址转换方案。

- uDAPL 库，用于支持 Mellanox Tavor 主机通道适配器，该适配器具有向 `dat.conf` 注册文件自动注册的功能。
- 支持 SPARC 平台和 x86 平台。

▼ 如何启用 uDAPL

- 1 成为超级用户。
- 2 确认是否已安装以下软件包，也可以根据需要安装这些软件包。
 - SUNWib – Sun InfiniBand 框架
 - SUNWtavor – Sun Tavor HCA 驱动程序
 - SUNWipoib – Sun IP over InfiniBand
 - SUNWudaplr – 直接访问传输 (Direct Access Transport, DAT) 注册软件包 (root)
 - SUNWudaplu – 直接访问传输 (Direct Access Transport, DAT) 注册软件包 (usr)
 - SUNWudapltr – Service Provider for Tavor packages (root)
 - SUNWudapltu – Service Provider for Tavor packages (usr)
- 3 选择以下操作之一来探测 IPoIB 接口。
 - 使用 `ifconfig` 和 `datadm` 命令手动探测接口。
例如：

```
# ifconfig ibd1 plumb

# ifconfig ibd1 192.168.0.1/24 up

# datadm -a /usr/share/dat/SUNWudaplt.conf
```

- 通过执行以下操作自动探测接口：
 - 创建以下包含相应 IP 地址的文件。

`/etc/hostname.ibd1`
 - 重新引导系统。

更新 DAT 静态注册表

可以使用 `datadm` 命令来维护 DAT 静态注册表，即 `dat.conf` 文件。有关此文件的更多信息，请参见 `dat.conf(4)`。

`datadm` 命令还可用于注册或注销 `dat.conf` 文件的。有关更多信息，请参见 `datadm(1M)`。

添加或删除 IPoIB 接口适配器时，请运行 `datadm` 命令来更新 `dat.conf` 文件，以反映系统的当前状态。此时将重新产生当前安装的所有的一组新的接口适配器。

▼ 如何更新 DAT 静态注册表

- 1 成为超级用户。
- 2 在系统中添加或删除 IPoIP 接口适配器后，更新 DAT 静态注册表。

```
# datadm -u
```

- 3 显示更新的 DAT 静态注册表。

```
# datadm
```

▼ 如何在 DAT 静态注册表中注册

- 1 成为超级用户。
- 2 添加 Sun 的 Mellanox Tavor 主机通道适配器后，更新 DAT 静态注册表。

```
# datadm -a /usr/share/dat/SUNWudaplt.conf
```

- 3 显示更新的 DAT 静态注册表。

```
# datadm -v
```

▼ 如何从 DAT 静态注册表中注销

- 1 成为超级用户。
- 2 从系统中删除 Sun 的 Mellanox Tavor 主机通道适配器后，更新 DAT 静态注册表。

```
# datadm -r /usr/share/dat/SUNWudaplt.conf
```

- 3 显示更新的 DAT 静态注册表。

```
# datadm -v
```


访问设备（概述）

本章提供有关如何访问系统上的设备的信息。

以下是本章中概述信息的列表。

- 第 193 页中的“访问设备”
- 第 194 页中的“逻辑磁盘设备名称”
- 第 197 页中的“逻辑磁带设备名称”
- 第 197 页中的“逻辑可移除介质设备名称”

有关配置设备的概述信息，请参见第 5 章。

访问设备

使用命令管理磁带、文件系统和其他设备时，需要了解如何指定设备名称。在大多数情况下，可以使用逻辑设备名称来表示与系统连接的设备。逻辑设备名称和物理设备名称在系统中分别由逻辑设备文件和物理设备文件表示。

如何创建设备信息

首次引导系统时，将会创建设备分层结构，用于表示与系统连接的所有设备。内核使用设备分层结构信息将驱动程序与其相应的设备关联起来。内核还提供了一组指向执行特定操作的驱动程序的指针。有关设备分层结构的更多信息，请参见《OpenBoot 3.x Command Reference Manual》。

如何管理设备

`devfsadm` 命令用于管理 `/dev` 和 `/devices` 目录中的特殊设备文件。缺省情况下，`devfsadm` 命令会尝试在系统中装入每个驱动程序，并连接至所有可能的设备实例。然

后，`devfsadm` 将在 `/devices` 目录中创建设备文件并在 `/dev` 目录中创建逻辑链接。除了管理 `/dev` 和 `/devices` 目录以外，`devfsadm` 命令还可维护 `path_to_inst` 实例数据库。有关更多信息，请参见 `path_to_inst(4)`。

为响应动态重新配置事件而执行的重新配置引导处理以及对 `/dev` 和 `/devices` 目录的更新，都将通过 `devfsadmd`（即 `devfsadm` 命令的守护进程版本）进行处理。引导系统时将从 `/etc/rc*` 脚本启动此守护进程。

由于 `devfsadmd` 守护进程会自动检测任何重新配置事件产生的设备配置变化，因此无需交互运行此命令。

有关更多信息，请参见 `devfsadm(1M)`。

设备命名约定

在 Solaris OS 中以三种方式对设备进行引用。

- **物理设备名称**—表示设备信息分层结构中的设备全路径名。物理设备名称是首次将设备添加到系统时创建的。物理设备名称位于 `/devices` 目录中。
- **实例名称**—表示系统上每个可能设备的内核的缩写名称。例如，`sd0` 和 `sd1` 表示两个磁盘设备的实例名称。实例名称会在 `/etc/path_to_inst` 文件中进行映射。
- **逻辑设备名称**—逻辑设备名称是首次将设备添加到系统时创建的。在大多数文件系统命令中，可以使用逻辑设备名称引用相应的设备。有关使用逻辑设备名称的文件命令列表，请参见表 10-1。`/dev` 目录中的逻辑设备文件以符号形式链接至 `/devices` 目录中的物理设备文件。

前面的设备名称信息是使用以下命令显示的：

- `dmesg`
- `format`
- `sysdef`
- `prtconf`

逻辑磁盘设备名称

执行以下任务时可使用逻辑设备名称访问磁盘设备：

- 向系统中添加新磁盘。
- 在系统间移动磁盘。
- 访问或挂载驻留在本地磁盘上的文件系统。
- 备份本地文件系统。

许多管理命令都会使用引用磁盘片或文件系统的参数。

通过指定以符号形式与磁盘设备链接的子目录（`/dev/dsk` 或 `/dev/rdisk`），后跟标识特定控制器、磁盘和片的字符串来引用磁盘设备。

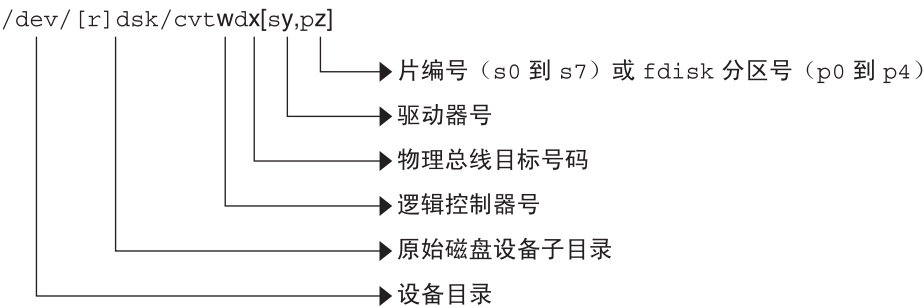


图 10-1 逻辑设备名称的说明

指定磁盘子目录

磁盘和文件管理命令需要使用**原始**（或**字符**）设备接口，或**块**设备接口。该区别是由从设备读取数据的方式导致的。

原始设备接口一次仅传输少量数据。块设备接口包括一次从中读取大数据块的缓冲区。

不同的命令需要不同的接口：

- 如果命令需要原始设备接口，请指定 `/dev/rdsk` 子目录。（`rdsk` 中的 "r" 代表 "raw" [原始]。）
- 如果命令需要块设备接口，请指定 `/dev/dsk` 子目录。
- 如果不确定命令需要使用 `/dev/dsk` 还是 `/dev/rdsk`，请检查该命令的手册页。

下表显示了某些常用磁盘和文件系统命令所需的接口。

表 10-1 一些常用命令所需的设备接口类型

命令参考	接口类型	用法示例
<code>df(1M)</code>	块	<code>df /dev/dsk/c0t3d0s6</code>
<code>fsck(1M)</code>	原始	<code>fsck -p /dev/rdsk/c0t0d0s0</code>
<code>mount(1M)</code>	块	<code>mount /dev/dsk/clt0d0s7 /export/home</code>
<code>newfs(1M)</code>	原始	<code>newfs /dev/rdsk/c0t0d1s1</code>
<code>prtvtoc(1M)</code>	原始	<code>prtvtoc /dev/rdsk/c0t0d0s2</code>

直接控制器和面向总线的控制器

根据磁盘设备连接至直接控制器还是面向总线的控制器，可能要采用不同方式访问磁盘分区或磁盘片。通常，直接控制器在逻辑设备名称中不包括目标标识符。

有关这两种类型控制器的约定将在以下小节中加以说明。

注 - 控制器编号是在系统初始化过程中自动指定的。这些编号为严格的逻辑编号，表示不会直接映射到物理控制器。

x86: 带有直接控制器的磁盘

要在基于 x86 的系统中带有 IDE 控制器的磁盘上指定片，请遵循下图中显示的命名约定。

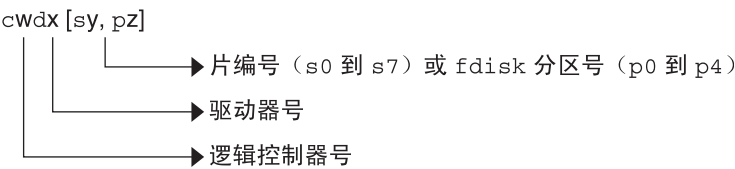


图 10-2 x86: 带有直接控制器的磁盘

要指示整个 Solaris fdisk 分区，请指定片 2 (s2)。

如果系统中仅有一个控制器，则 w 通常为 0。

带有面向总线的控制器的磁盘

要在带有面向总线的控制器（例如 SCSI）的磁盘上指定片，请遵循下图中显示的命名约定。

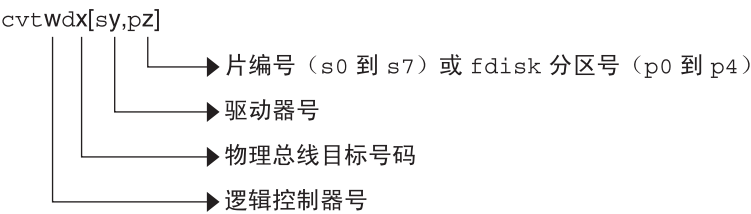


图 10-3 带有面向总线的控制器的磁盘

在带有直接连接磁盘（如 UltraSPARC® 系统上的 IDE 磁盘）的基于 SPARC 的系统上，

命名约定与带有面向总线的控制器的系统的命名约定相同。

如果系统中仅有一个控制器，则 *w* 通常为 0。

对于 SCSI 控制器，*x* 是通过装置背面的开关设置的目标地址，*y* 是连接至目标的驱动器的逻辑单元号 (logical unit number, LUN)。如果磁盘具有嵌入式控制器，则 *y* 通常为 0。有关基于 SPARC 的系统上的 SCSI 地址的更多信息，请参见 SunSolveSM 信息文档 48041 和 `scsi_address(9S)`。

要指示整个磁盘，请指定片 2 (`s2`)。

逻辑磁带设备名称

逻辑磁带设备文件位于 `/dev/rmt/*` 目录中，作为 `/devices` 目录的符号链接。

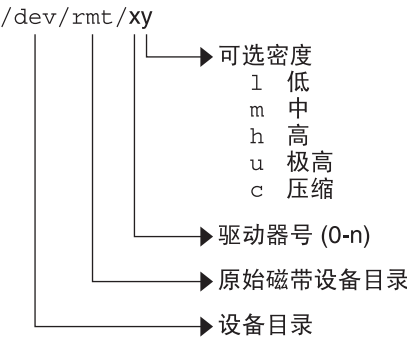


图 10-4 逻辑磁带设备名称

连接至系统的第一个磁带设备为 0 (`/dev/rmt/0`)。磁带密度值 (`l`、`m`、`h`、`c` 和 `u`) 在 [第 30 章](#) 中介绍。

逻辑可移除介质设备名称

由于可移除介质由卷管理 (`vold`) 来进行管理，因此除非要手动挂载介质，否则通常不会使用逻辑设备名称。

表示系统上的可移除介质设备的逻辑设备名称在 [第 3 章](#) 中加以介绍。

管理磁盘（概述）

本章提供有关 Solaris 磁盘片的概述信息并介绍 `format` 实用程序。

以下是本章中概述信息的列表。

- 第 199 页中的 “Solaris 10 6/06 发行版在磁盘管理方面的新增功能”
- 第 200 页中的 “Solaris 10 1/06 发行版在磁盘管理方面的新增功能”
- 第 201 页中的 “Solaris 10 发行版在磁盘管理方面的新增功能”
- 第 206 页中的 “有关磁盘管理任务的参考信息”
- 第 207 页中的 “磁盘管理概述”
- 第 207 页中的 “磁盘术语”
- 第 207 页中的 “关于磁盘片”
- 第 211 页中的 “`format` 实用程序”
- 第 214 页中的 “关于磁盘标号”
- 第 217 页中的 “对磁盘分区”

有关如何将磁盘添加到系统的说明，请参见第 13 章或第 14 章。

Solaris 10 6/06 发行版在磁盘管理方面的新增功能

本节介绍 Solaris 10 6/06 发行版中新增的磁盘管理功能。

iSCSI 启动器支持增强功能

Solaris 10 6/06：已在 Solaris iSCSI 启动器支持中添加了以下增强功能：

- 动态目标删除支持—提供在不重新引导系统的情况下删除（或注销）iSCSI 目标的功能。如果尝试删除或禁用搜索方法或地址，且目标未被使用，则可以删除目标并释放相关的资源。如果正在使用目标，则搜索地址或方法始终处于启用状态，并将显示正在使用消息。

有关更多信息，请参见第 289 页中的 “如何删除搜索到的 iSCSI 目标”。

- Internet 存储名称服务 (Internet Storage Name Service, iSNS) 客户机支持—启用 iSCSI 启动器，以搜索其使用尽可能少的配置即可访问的目标。另外，该支持还提供状态更改通知功能，在存储节点的操作状态发生更改时通知 iSCSI 启动器。iscsiadm 命令已得到增强，支持 iSNS 搜索。

有关更多信息，请参见第 288 页中的“如何配置 iSCSI 目标搜索”。

- 多会话目标 (Multiple Session Target, MS/T) 支持—提供按照需要创建目标的多个 iSCSI 会话或路径的功能。在特定配置中（例如支持登录重定向的 iSCSI 数组），其他 iSCSI 路径可提供更高的总带宽和可用性。iSCSI MS/T 功能应该与 MPxIO 或其他多路径软件组合使用。iscsiadm 命令已得到增强，支持 MS/T。

有关配置 Solaris iSCSI 启动器的更多信息，请参见第 15 章和 iscsiadm(1M)。

Solaris 10 1/06 发行版在磁盘管理方面的新增功能

本节介绍 Solaris 10 1/06 发行版中新增的磁盘管理功能。

- 第 200 页中的“x86: GRUB 引导环境中的磁盘管理”
- 第 201 页中的“Solaris iSCSI 启动器支持”
- 第 201 页中的“支持容量超过 2 TB 的 SCSI 磁盘”

x86: GRUB 引导环境中的磁盘管理

Solaris 10 1/06：GRUB 引导菜单取代了以前的 x86 系统引导方法。在磁盘管理方面，从备用设备引导以更换系统磁盘或安装引导块时，可以使用 GRUB 界面。

GRUB 引导环境提供了以下功能：

- **Solaris 故障安全引导**—Solaris 故障安全引导选项，用于引导到 miniroot，以便可以从阻止系统引导的问题中恢复，而不必从备用设备进行引导。使用方向键从 GRUB 引导菜单中选择以下选项，然后按回车键：

```
Solaris failsafe
```

使用 Solaris 故障安全引导选项后，需要重新引导系统。

- **网络引导**—通过在 BIOS 配置阶段按 F12 键从网络进行引导。
- **单用户引导**—通过从 Solaris 故障安全引导菜单中选择此选项将系统引导至单用户模式：

```
kernel /platform/i86pc/multiboot
```

然后，使用 e（编辑）选项添加 -s 单用户选项。例如：

```
kernel /platform/i86pc/multiboot -s
```


按回车键，然后按 **b** 键来引导系统。按 **Control-D** 组合键，将系统引导回多用户模式。

如果运行的是 **x86** 系统，则在 **GRUB** 环境中不能使用 **fmthard** 命令自动安装引导块。必须单独安装引导块。

有关详细的功能信息以及在 **x86** 系统上使用基于 **GRUB** 的新式引导的说明，请参见《系统管理指南：基本管理》中的第 11 章“基于 **GRUB** 的引导（任务）”。

有关在 **GRUB** 引导环境下管理磁盘的说明，请参见以下参考资料：

- 第 278 页中的“**x86: 如何在系统磁盘上安装引导块**”
- 第 262 页中的“**x86: 如何连接系统磁盘**”

此功能在 **SPARC** 系统上不可用。

Solaris iSCSI 启动器支持

Solaris 10 1/06：iSCSI 是用于链接数据存储子系统的基于 Internet 协议 (Internet Protocol, IP) 的存储联网标准。iSCSI 协议通过 IP 网络执行 SCSI 命令，使您可以将磁盘设备从网络挂载到本地系统。在本地系统上，可以使用诸如块设备的设备。

有关更多信息，请参见第 15 章。

支持容量超过 2 TB 的 SCSI 磁盘

Solaris 10 1/06：现在，运行 64 位内核的 **SPARC** 和 **x86** 系统支持容量超过 2 TB 的 SCSI、光纤通道和 iSCSI 磁盘。

可以使用 **format** 实用程序对这些较大的磁盘进行标记、配置和分区。有关在大磁盘上使用 EFI 磁盘标号的信息以及使用 **fdisk** 实用程序的限制，请参见第 203 页中的“**EFI 磁盘标号限制**”。

Solaris 10 发行版在磁盘管理方面的新增功能

本节介绍 Solaris 10 发行版中新增的磁盘管理功能。

- 第 202 页中的“带有 EFI 磁盘标号的多 TB 磁盘支持”
- 第 206 页中的“用于 **SPARC** 和 **x86** 系统的通用 SCSI 驱动程序”
- 第 206 页中的“新的 **fdisk** 分区标识符”

带有 EFI 磁盘标号的多 TB 磁盘支持

Solaris 10：对于运行 64 位 Solaris 内核的系统，支持容量超过 1 TB 的磁盘。可扩展固件接口 GUID 分区表 (Extensible Firmware Interface GUID Partition Table, EFI GPT) 磁盘标号还可用于与运行 32 位 Solaris 内核的系统连接的容量小于 1 TB 的磁盘。

可以下载位于以下位置的 EFI 规范：

http://www.intel.com/technology/efi/main_specification.htm

如果系统正在运行适当的 Solaris 发行版，则可以使用 `format -e` 命令将 EFI 标号应用于磁盘。但是，尝试应用 EFI 标号之前，应该先查阅第 203 页中的“EFI 磁盘标号限制”中的重要信息。

EFI 标号支持物理磁盘和虚拟磁盘卷。该发行版还提供了更新的磁盘实用程序，用于管理容量超过 1 TB 的磁盘。UFS 文件系统与 EFI 磁盘标号兼容，您可以创建容量超过 1 TB 的 UFS 文件系统。有关创建多 TB UFS 文件系统的信息，请参见第 340 页中的“64 位：对多 TB UFS 文件系统的支持”。

如果需要创建容量超过 1 TB 的文件系统，也可以使用非绑定 Sun QFS 文件系统。有关 Sun QFS 文件系统的信息，请参见《Sun QFS, Sun SAM-FS, and Sun SAM-QFS File System Administrator's Guide》。

在此 Solaris 发行版中，Solaris 卷管理器软件也可用于管理容量超过 1 TB 的磁盘。有关使用 Solaris 卷管理器的信息，请参见《Solaris Volume Manager 管理指南》。

VTOC 标号仍可用于容量小于 1 TB 的磁盘。如果在系统中仅使用容量小于 1 TB 的磁盘，则磁盘管理与以前的 Solaris 发行版中的方法相同。此外，还可以使用 `format -e` 命令以 EFI 标号来标记容量小于 1 TB 的磁盘。有关更多信息，请参见示例 12-6。

比较 EFI 标号与 VTOC 标号

在以下方面 EFI 磁盘标号不同于 VTOC 磁盘标号：

- 支持容量超过 1 TB 的磁盘。
- 提供可用的片 0-6，其中仅片 2 是不同的片。
- 分区（或片）不能与主标号或备份标号重叠，也不能与任何其他分区重叠。EFI 标号的大小通常为 34 个扇区，因此分区从第 34 个扇区开始。此特性意味着任何分区都不能从扇区零 (0) 开始。
- EFI 标号中不存储柱面、磁头或扇区信息。大小以块数报告。
- 在备用柱面区域（磁盘的最后两个柱面）中存储的信息现在存储在片 8 中。
- 如果使用 `format` 实用程序来更改分区大小，则 `unassigned` 分区标记将被指定给大小等于零的分区。缺省情况下，`format` 实用程序将 `usr` 分区标记指定给大小大于零的任何分区。更改分区后，可以使用分区更改菜单重新指定分区标记。但是，不能将大小不为零的分区更改为 `unassigned` 分区标记。

EFI 磁盘标号限制

确定使用容量超过 1 TB 的磁盘是否适合于环境时，应牢记以下限制：

- SCSI 驱动程序 `ssd` 或 `sd` 最多仅支持 2 TB。
- 专门为安装了带有 EFI 标号的磁盘的系统而设计的分层软件产品，可能无法访问带有 EFI 磁盘标号的磁盘。
- 运行以前的 Solaris 发行版的系统不能识别带有 EFI 标号的磁盘。
- 无法从带有 EFI 磁盘标号的磁盘进行引导。
- 在容量大于 1 TB 且带有 EFI 标号的磁盘上，不能使用 `fdisk` 命令。
- 不能使用 Solaris Management Console 的磁盘管理器工具来管理带有 EFI 标号的磁盘。使用 `format` 实用程序对带有 EFI 标号的磁盘进行分区。然后，可以使用 Solaris Management Console 的增强存储工具来管理卷和带有 EFI 标号的磁盘的磁盘集。
- EFI 规范禁止重叠片。整个磁盘以 `cxydz` 表示。
- EFI 磁盘标号以扇区和块（而不是柱面和磁头）为单位提供磁盘或分区大小的信息。
- 在带有 EFI 标号的磁盘上不支持或不适用以下 `format` 选项。
 - 不支持 `save` 选项，因为带有 EFI 标号的磁盘不需要在 `format.dat` 文件中包含项。
 - 不适用 `backup` 选项，因为磁盘驱动程序会发现主标号，并将其写回磁盘。

支持 x86 系统上带有 EFI 标号的磁盘

在 x86 系统中提供对 EFI 磁盘标号的 Solaris 支持。在 x86 系统上可以使用以下命令添加 EFI 标号：

```
# format -e

> [0] SMI Label

> [1] EFI Label

> Specify Label type[0]: 1

> WARNING: converting this device to EFI labels will erase all current

> fdisk partition information. Continue? yes
```

以前的标号信息不会转换为 EFI 磁盘标号。

必须使用 `format` 命令手动重新创建标号的分区信息。在容量大于 1 TB 且带有 EFI 标号的磁盘上，不能使用 `fdisk` 命令。`fdisk` 命令不适用于容量大于 1 TB 的磁盘。有关 EFI 磁盘标号的更多信息，请参见上一节。

使用带有 EFI 标号的磁盘安装系统

Solaris 安装实用程序将自动识别带有 EFI 标号的磁盘。但是，不能使用 Solaris 安装程序对这些磁盘重新分区。安装之前或安装之后，必须使用 `format` 实用程序对带有 EFI 标号的磁盘重新分区。Solaris Upgrade 和 Live Upgrade 实用程序还可以识别带有 EFI 标号的磁盘。但是，不能从带有 EFI 标号的磁盘引导系统。

在带有 EFI 标号的磁盘的系统上安装 Solaris 发行版后，分区表将显示以下类似信息：

Current partition table (original):

Total disk sectors available: 2576924638 + 16384 (reserved sectors)

Part	Tag	Flag	First Sector	Size	Last Sector
0	root	wm	34	1.20TB	2576924636
1	unassigned	wm	0	0	0
2	unassigned	wm	0	0	0
3	unassigned	wm	0	0	0
4	unassigned	wm	0	0	0
5	unassigned	wm	0	0	0
6	unassigned	wm	0	0	0
8	reserved	wm	2576924638	8.00MB	2576941021

管理带有 EFI 磁盘标号的磁盘

使用下表查找有关带有 EFI 磁盘标号的磁盘管理信息

任务	更多信息
如果已安装系统，请将磁盘连接至系统并执行重新配置引导。	第 247 页中的“SPARC: 添加系统磁盘或辅助磁盘（任务列表）”
使用 <code>format</code> 实用程序对磁盘重新分区（如果需要）。	第 250 页中的“SPARC: 如何创建磁盘片和标记磁盘”
使用 Solaris 卷管理器创建磁盘卷并创建软分区（如果需要）。	《Solaris Volume Manager 管理指南》中的第 2 章“存储管理概念”。

任务	更多信息
使用 <code>newfs</code> 命令为新磁盘创建 UFS 文件系统。	第 258 页中的 “SPARC: 如何创建 UFS 文件系统”
或者，创建 QFS 文件系统。	http://docs.sun.com/db/coll/20445.2
克隆带有 EFI 标号的磁盘	示例 29-2

EFI 磁盘标号问题的疑难解答

使用以下错误消息和解决方案解决带有 EFI 标号的磁盘问题。

错误消息

```
The capacity of this LUN is too large.

Reconfigure this LUN so that it is < 2TB.
```

原因

您尝试在容量超过 2 TB 的 SCSI 设备上创建分区。

解决方案

在容量小于 2 TB 的 SCSI 设备上创建分区。

错误消息

```
Dec  3 09:26:48 holoship scsi: WARNING: /sbus@a,0/SUNW,socal@d,10000/
sf@1,0/ssd@w50020f23000002a4,0 (ssd1):

Dec  3 09:26:48 holoship disk has 2576941056 blocks, which is too large
for a 32-bit kernel
```

原因

您尝试使用容量超过 1 TB 的磁盘引导运行 32 位 SPARC 或 x86 内核的系统。

解决方案

使用容量大于 1 TB 的磁盘引导运行 64 位 SPARC 或 x86 内核的系统。

错误消息

```
Dec  3 09:12:17 holoship scsi: WARNING: /sbus@a,0/SUNW,socal@d,10000/
sf@1,0/ssd@w50020f23000002a4,0 (ssd1):

Dec  3 09:12:17 holoship corrupt label - wrong magic number
```

原因

您尝试向运行早期 Solaris 发行版的系统中添加磁盘。

解决方案
向运行 Solaris 发行版且支持 EFI 磁盘标号的系统中添加磁盘。

用于 SPARC 和 x86 系统的通用 SCSI 驱动程序

在此 Solaris 发行版中，将分别用于 SPARC 平台和 x86 平台的磁盘驱动程序合并成一个驱动程序。这一更改为以下 3 种驱动程序创建了一个源文件：

- 用于 SCSI 设备的 SPARC sd
- 用于光纤通道和 SCSI 设备的 x86 sd
- 用于光纤通道设备的 SPARC ssd

在以前的 Solaris 发行版中，要支持 SPARC 平台和 x86 平台上的 SCSI 和光纤通道磁盘设备，需要 3 种不同的驱动程序。

已对所有的磁盘实用程序（例如 `format`、`fmthard` 和 `fdisk` 命令）进行了更新，以便支持这些更改。有关更多信息，请参见 `sd.7D` 和 `ssd.7D`。

新的 fdisk 分区标识符

x86 系统上的 Solaris `fdisk` 分区标识符已从 130 (0x82) 更改为 191 (0xbf)。已对所有的 Solaris 命令、实用程序和驱动程序进行了更新，以便可以使用上述每一种 `fdisk` 标识符。`fdisk` 功能没有任何变化。

通过新的 `fdisk` 菜单选项，可以在新标识符与旧标识符之间来回切换。即使挂载了分区中包含的文件系统，也可以更改 `fdisk` 标识符。

`fdisk` 菜单中的两个 `type` 值反映旧标识符和新标识符，如下所示：

- Solaris 标识 0x82
- Solaris2 标识 0xbf

有关更改 Solaris `fdisk` 分区标识符的逐步说明，请参见第 263 页中的“[如何更改 Solaris fdisk 标识符](#)”。

有关磁盘管理任务的参考信息

使用这些参考信息可以查找有关管理磁盘的逐步说明。

磁盘管理任务	更多信息
格式化磁盘并检查磁盘标号。	第 12 章

磁盘管理任务	更多信息
向 SPARC 系统中添加新磁盘。	第 13 章
向 x86 系统中添加新磁盘。	第 14 章
热插拔 SCSI 或 PCI 磁盘。	第 6 章

磁盘管理概述

在 Solaris OS 中管理磁盘通常涉及设置系统并运行 Solaris 安装程序，以创建适当的磁盘片和文件系统并安装 Solaris OS。有时，可能需要使用 `format` 实用程序添加新磁盘驱动器或更换有故障的磁盘驱动器。

磁盘术语

您应熟悉基本的磁盘体系结构，这样才能有效地使用本节中的信息。需要特别指出的是，应熟悉以下术语：

磁盘术语	说明
Track（磁轨）	磁盘上的同心环，在磁盘旋转时在单个固定磁头下穿过。
Cylinder（柱面）	距磁盘旋转所围绕的轴的距离相同的一组磁轨。
Sector（扇区）	每个磁盘片的段。一个扇区能容纳 512 字节。
Block（块）	磁盘上的数据存储区域。磁盘块为 512 字节。
Disk controller（磁盘控制器）	用于控制磁盘驱动器的芯片及其关联的电路。
Disk label（磁盘标号）	磁盘中包含磁盘几何参数和分区信息的第一个扇区。
Device driver（设备驱动程序）	控制硬件或虚拟设备的内核模块。

有关其他信息，请参见磁盘制造商提供的产品信息。

关于磁盘片

文件系统中包含存储在磁盘上的文件。磁盘上的每个文件系统都被指定给片，片是专门保留供该文件系统使用的一组扇区。对于 Solaris OS（和系统管理员）而言，每个磁盘片看起来就如同单独的磁盘驱动器一样。

有关文件系统的信息，请参见[第 17 章](#)。

注 - 片有时称为**分区**。某些界面（如 `format` 实用程序）将片称为分区。

设置片时，请记住以下规则：

- 每个磁盘片仅容纳一个文件系统。
- 任何文件系统都不能跨多个片。

片在 SPARC 和 x86 平台上的设置方式略有不同。下表总结了这些差异。

表 11-1 SPARC 和 x86 平台上的片差异

SPARC 平台	x86 平台
整个磁盘供 Solaris OS 专用。	磁盘被分成 <code>fdisk</code> 分区，每个操作系统都有一个 <code>fdisk</code> 分区。
VTOC —磁盘被分成 8 片，编号为 0-7。	VTOC —Solaris <code>fdisk</code> 分区被分成 10 片，编号为 0-9。
EFI —磁盘被分成 7 片，编号为 0-6。	EFI —磁盘被分成 7 片，编号为 0-6

Solaris 卷管理器（以前称为 Solstice DiskSuite™）具有分区功能，即**软分区**。软分区允许每个磁盘包含八个以上分区。

有关 Solaris 卷管理器的常规信息，请参见《Solaris Volume Manager 管理指南》中的第 2 章“存储管理概念”。有关软分区的信息，请参见《Solaris Volume Manager 管理指南》中的第 12 章“软分区（概述）”。

磁盘片

下表介绍了在运行 Solaris OS 的系统上可能找到的片。

在 x86 系统上：

- 磁盘被分成 `fdisk` 分区。`fdisk` 分区是为特定操作系统（如 Solaris OS）保留的磁盘段。
- Solaris OS 将十个片（编号为 0-9）置于 Solaris `fdisk` 分区上。

表 11-2 通常的磁盘片

片	文件系统	通常位于客户机系统还是服务器系统？	注释
0	根目录 (/)	二者皆有	承载构成操作系统的文件和目录。 EFI —无法从带有 EFI 标号的磁盘进行引导。
1	交换	二者皆有	提供虚拟内存或 交换空间 。
2	—	二者皆有	VTOC —习惯上指代整个磁盘。不应更改此片的大小。 EFI —根据站点需要定义的可选片。
3	例如， /export	二者皆有	可根据站点需要定义的可选片。 在服务器上可用于承载客户机系统所需的备用操作系统。
4		二者皆有	根据站点需要定义的可选片。
5	例如， /opt	二者皆有	根据站点需要定义的可选片。 可用于承载添加到系统中的应用程序软件。如果未在安装期间为 /opt 文件系统分配片，则 /opt 目录将被置于片 0 中。
6	/usr	二者皆有	承载 OS 命令（也称为 可执行文件 ）。此片还可以承载文档、系统程序（例如， init 和 syslogd）以及库例程。
7	/home 或 /export/home	二者皆有	承载用户创建的文件。
8	N/A	N/A	VTOC —不适用。 EFI —缺省情况下创建的保留片。此区域类似于 VTOC 的备用柱面。请勿修改或删除此片。
9（仅限于 x86）	—	二者皆有	EFI —不适用。 VTOC —提供为备用磁盘块保留的区域。片 9 称为备用扇区片。

使用原始数据片

磁盘标号存储在每个磁盘的块 0 中。因此，用于创建原始数据片的第三方数据库应用程序决不能从块 0 启动。否则，将覆写磁盘标号，且磁盘上的数据将无法访问。

请勿对原始数据片使用以下磁盘区域，这些原始数据片有时是由第三方数据库应用程序创建的：

- 存储磁盘标号的块 0
- 片 2，表示带有 VTOC 标号的整个磁盘

多个磁盘上的片安排

尽管单个大磁盘可以承载所有的片及其对应的文件系统，但通常使用两个或更多磁盘来承载系统的片和文件系统。

注 - 不能在两个或更多磁盘之间拆分片。但是，允许单个磁盘上包含多个交换片。

例如，单个磁盘可能承载根 (/) 文件系统、交换区域和 /usr 文件系统，而另一个磁盘承载 /export/home 文件系统和包含用户数据的其他文件系统。

在多个磁盘安排中，包含 OS 和交换空间的磁盘（即承载根 (/) 文件系统、/usr 文件系统以及交换空间片的磁盘）称为**系统磁盘**。其他磁盘称为**辅助磁盘**或**非系统磁盘**。

在多个磁盘上安排系统的文件系统时，可以修改辅助磁盘上的文件系统和片，而不必关闭系统或重新装入 OS。

如果您具有多个磁盘，还可以增大输入-输出 (I/O) 卷。通过将磁盘负载分布于多个磁盘上，可以避免 I/O 瓶颈。

确定要使用的片

设置磁盘的文件系统时，不仅要选择每个片的大小，而且还要选择要使用的片。这些决定取决于磁盘连接至的系统的配置以及要在磁盘上安装的软件。

以下是需要磁盘空间的系统配置：

- 服务器
- 独立系统

每个系统配置都会以不同的方式使用片。下表列出了一些示例。

表 11-3 系统配置和片

片	服务器	独立系统
0	根	根
1	交换	交换
2	—	—
3	/export	—
6	/usr	/usr

表 11-3 系统配置和片 (续)

片	服务器	独立系统
7	/export/home	/home

有关系统配置的更多信息，请参见《系统管理指南：基本管理》中的“系统类型概述”。

注 - Solaris 安装实用程序根据选择安装的软件提供缺省的片大小。

format 实用程序

继续学习操作方法或参考部分之前，请先阅读以下 `format` 实用程序及其用法的概述。

`format` 实用程序是系统管理工具，用于准备要在 Solaris 系统上使用的硬盘驱动器。

下表说明了 `format` 实用程序提供的功能和关联优势。

表 11-4 `format` 实用程序的功能和优势

功能	优势
搜索系统中所有连接的磁盘驱动器	报告以下内容： <ul style="list-style-type: none">■ 目标位置■ 磁盘几何参数■ 磁盘是否已格式化■ 磁盘是否具有已挂载的分区
检索磁盘标号	便于修复操作
修复有故障的扇区	允许管理员修复具有可恢复错误的磁盘驱动器，而不用将驱动器发回给制造商
格式化和分析磁盘	在磁盘上创建扇区并验证每个扇区
对磁盘分区	将磁盘分成片，以便可以在不同的片上创建各个文件系统
标记磁盘	将磁盘名称和配置信息写入磁盘，以备将来检索（通常用于修复操作）

`format` 实用程序选项在 [第 16 章](#) 中予以介绍。

何时使用 `format` 实用程序

安装 Solaris 发行版时，Solaris 安装实用程序会对磁盘驱动器进行分区和标记。可以使用 `format` 实用程序执行以下操作：

- 显示片信息

- 对磁盘分区
- 向现有系统添加磁盘驱动器
- 格式化磁盘驱动器
- 标记磁盘
- 修复磁盘驱动器
- 分析磁盘错误

系统管理员使用 `format` 实用程序的主要目的是对磁盘分区。这些步骤将在第 13 章和第 14 章中介绍。

有关使用 `format` 实用程序的指南，请参见下一节。

format 实用程序使用指南

表 11-5 format 实用程序指南

任务	指南	更多信息
格式化磁盘。	<ul style="list-style-type: none">■ 重新格式化磁盘时，任何现有数据都将被破坏。■ 越来越多的制造商提供经过格式化和分区的磁盘驱动器，因此对磁盘驱动器格式化的需求就减少了。向现有系统添加磁盘驱动器时，可能不需要使用 <code>format</code> 实用程序。■ 如果重新放置磁盘且显示许多磁盘错误，则可以尝试对其重新格式化。重新格式化将自动重映射所有错误扇区。	第 224 页中的 “如何格式化磁盘”
更换系统磁盘。	<ul style="list-style-type: none">■ 必须从备份介质恢复受损系统磁盘中的数据。否则，必须使用安装实用程序重新安装系统。	第 248 页中的 “SPARC: 如何连接系统磁盘并进行引导”、第 262 页中的 “x86: 如何连接系统磁盘” 或（如果必须重新安装系统）《Solaris 10 安装指南：基本安装》
将磁盘分成片。	<ul style="list-style-type: none">■ 重新分区和重新标记包含现有片的磁盘时，将破坏所有的现有数据。■ 重新分区和恢复磁盘之前，必须将现有数据复制到备份介质。	第 250 页中的 “SPARC: 如何创建磁盘片和标记磁盘” 或第 276 页中的 “x86: 如何创建磁盘片和标记磁盘”
向现有系统中添加辅助磁盘。	<ul style="list-style-type: none">■ 如果对辅助磁盘重新格式化或重新分区，则必须从备份介质中恢复所有的现有数据。	第 249 页中的 “SPARC: 如何连接辅助磁盘并进行引导” 或第 265 页中的 “x86: 如何连接辅助磁盘并进行引导”

表 11-5 format 实用程序指南 (续)

任务	指南	更多信息
修复磁盘驱动器。	<ul style="list-style-type: none">■ 有些客户站点希望更换而不是修复有故障的驱动器。如果站点与磁盘驱动器制造商之间具有修复合同，则可能不需要使用 format 实用程序来修复磁盘驱动器。■ 修复磁盘驱动器通常意味着将错误扇区添加到缺陷列表。新的控制器将重映射错误扇区，而不会发生系统中断。■ 如果系统具有以前的控制器，则可能需要重映射错误扇区并恢复所有丢失的数据。	第 241 页中的“修复有缺陷的扇区”

格式化磁盘

在大多数情况下，磁盘由制造商或经销商来格式化。因此，安装驱动器时，不需要重新格式化磁盘。要确定磁盘是否已格式化，请使用 format 实用程序。有关更多信息，请参见[第 223 页中的“如何确定磁盘是否已格式化”](#)。

如果确定磁盘未格式化，请使用 format 实用程序来格式化磁盘。

格式化磁盘时，需要完成两个步骤：

- 准备好磁盘介质以便随时使用。
- 根据表面分析编译磁盘缺陷列表。



注意 - 格式化磁盘是一个破坏性过程，因为它覆写磁盘上的数据。由于此原因，磁盘通常仅由制造商或转售商进行格式化。如果认为磁盘缺陷是导致某个问题反复出现的原因，则可以使用 format 实用程序执行表面分析。但是，应非常谨慎地仅使用不会破坏数据的命令。有关详细信息，请参见[第 224 页中的“如何格式化磁盘”](#)。

用于存储数据的总磁盘空间中有一小部分用于存储缺陷和格式化信息。此部分的百分比随磁盘几何参数的变化而变化，而且会因磁盘使用年限增加，产生的缺陷增多，而不断增大。

格式化磁盘可能需要几分钟到几小时时间，具体取决于磁盘类型和大小。

关于磁盘标号

专门为每个磁盘保留一个特殊区域，用于存储有关磁盘的控制器、几何参数和片的信息。此信息称为磁盘的**标号**。用于描述磁盘标号的另一个术语是带有 VTOC 标号的磁盘上的 VTOC（**卷目录**）。**标记磁盘**意味着将片信息写入磁盘。更改磁盘片后，通常要标记磁盘。

如果在创建片后无法标记磁盘，则片将不可用，因为 OS 无法“了解”有关片的信息。

分区表术语

磁盘标号的一个重要部分是**分区表**。分区表标识磁盘片、片边界（在柱面中）以及片的总大小。可以使用 `format` 实用程序来显示磁盘的分区表。下面介绍了分区表术语。

表 11-6 分区表术语

分区术语	值	说明
Number（编号）	0-7	VTOC —分区或片，编号为 0-7。 EFI —分区或片，编号为 0-6。
Tag（标记）	0=UNASSIGNED 1=BOOT 2=ROOT 3=SWAP 4=USR 5=BACKUP 7=VAR 8=HOME 11=RESERVED	通常用于描述此分区中挂载的文件系统的数值。
Flag（标志）	wm wu rm rm	分区是可写且可挂载的。 分区可写，但不能挂载。此状态是专用于交换区域的分区的缺省状态。（但是， <code>mount</code> 命令不会检查“不可挂载”标志。） 分区是只读且可挂载的。

分区标志和标记按约定指定且不需要进行维护。

有关显示分区表的更多信息，请参见以下参考信息：

- 第 214 页中的“显示分区表信息”
- 第 227 页中的“如何显示磁盘片信息”
- 第 233 页中的“如何检查磁盘标号”

显示分区表信息

以下 `format` 实用程序输出给出了带有 VTOC 标号的 74 GB 磁盘中的分区表示例：

Total disk cylinders available: 38756 + 2 (reserved cylinders)

Part	Tag	Flag	Cylinders	Size	Blocks	
0	root	wm	3 - 2083	4.00GB	(2081/0/0)	8390592
1	swap	wu	2084 - 3124	2.00GB	(1041/0/0)	4197312
2	backup	wm	0 - 38755	74.51GB	(38756/0/0)	156264192
3	unassigned	wm	0	0	(0/0/0)	0
4	unassigned	wm	0	0	(0/0/0)	0
5	unassigned	wm	0	0	(0/0/0)	0
6	unassigned	wm	0	0	(0/0/0)	0
7	home	wm	3125 - 38755	68.50GB	(35631/0/0)	143664192
8	boot	wu	0 - 0	1.97MB	(1/0/0)	4032
9	alternates	wu	1 - 2	3.94MB	(2/0/0)	8064

partition>

通过 format 实用程序显示的分区表包含以下信息。

列名	说明
Part	分区编号或片编号。有关此列的说明，请参见表 11-6。
Tag	分区标记。有关此列的说明，请参见表 11-6。
Flag	分区标志。有关此列的说明，请参见表 11-6。
Cylinders	片的起始和结束柱面编号。不显示在带有 EFI 标号的磁盘上。
Size	片大小 (MB)。
Blocks	每个片的柱面总数和扇区总数。不显示在带有 EFI 标号的磁盘上。
First Sector	EFI 一起始块编号。不显示在带有 VTOC 标号的磁盘上。
Last Sector	EFI 一结束块编号。不显示在带有 VTOC 标号的磁盘上。

以下是使用 `prtvtoc` 命令显示的 EFI 磁盘标号的示例。

```
# prtvtoc /dev/rdisk/c4t1d0s0

* /dev/rdisk/c4t1d0s0 partition map

*

* Dimensions:

*   512 bytes/sector

* 2576941056 sectors

* 2576940989 accessible sectors

*

* Flags:

*   1: unmountable

*  10: read-only

*

*                               First   Sector   Last
* Partition Tag  Flags      Sector    Count    Sector  Mount Directory
*
*      0      2    00          34   629145600  629145633
*      1      4    00   629145634   629145600 1258291233
*      6      4    00 1258291234 1318633404 2576924637
*      8     11    00 2576924638      16384 2576941021
```

`prtvtoc` 命令的输出按以下三个部分提供信息：

- 尺寸
- 标志
- 分区表

prtvtoc 列名	说明
Partition	分区编号或片编号。有关此列的说明，请参见表 11-6。
Tag	分区标记。有关此列的说明，请参见表 11-6。
Flags	分区标志。有关此列的说明，请参见表 11-6。
First Sector	片的第一个扇区。
Sector Count	片中的扇区总数。
Last Sector	片的最后一个扇区。
Mount Directory	文件系统的最后一个挂载点目录。

对磁盘分区

系统管理员经常会使用 `format` 实用程序来对磁盘分区。步骤如下：

- 确定所需的片
- 确定每个片或分区的大小
- 使用 `format` 实用程序对磁盘分区
- 使用新的分区信息标记磁盘
- 为每个分区创建文件系统

最简单的磁盘分区方法是使用 `format` 实用程序分区菜单中的 `modify` 命令。借助 `modify` 命令，通过指定每个分区的大小可以创建分区，而不必跟踪起始柱面边界。`modify` 命令还可用于跟踪“浮动 (free hog)”片中的任何磁盘空间。

使用浮动片 (free hog slice)

使用 `format` 实用程序更改一个或多个磁盘片的大小时，可以指定为适应大小调整操作而伸缩的临时片。

扩展片时，此临时片将提供或“释放”空间；收缩片时，此临时片将接收或“占据”放弃的空间。出于此原因，提供片有时称为**浮动片 (free hog slice)**。

浮动片 (free hog slice) 仅在安装期间或运行 `format` 实用程序时存在。日常操作期间没有永久的浮动片 (free hog slice)。

有关使用浮动片 (free hog slice) 的信息，请参见第 250 页中的“SPARC: 如何创建磁盘片和标记磁盘”或第 276 页中的“x86: 如何创建磁盘片和标记磁盘”。

管理磁盘（任务）

本章介绍管理磁盘的过程。如果您已经熟悉如何在运行 Solaris™ OS 的系统中管理磁盘，则本章介绍的许多过程是可选的。

有关与管理磁盘关联的过程的信息，请参见第 219 页中的“管理磁盘（任务列表）”。

有关磁盘管理的概述信息，请参见第 11 章。

管理磁盘（任务列表）

任务	说明	参考
确定系统中的磁盘。	如果不能确定系统中的磁盘类型，请使用 <code>format</code> 实用程序确定磁盘类型。	第 220 页中的“如何确定系统中的磁盘”
格式化磁盘。	使用 <code>format</code> 实用程序确定磁盘是否已格式化。 大多数情况下，磁盘已经格式化。如果需要格式化磁盘，请使用 <code>format</code> 实用程序。	第 223 页中的“如何确定磁盘是否已格式化” 第 224 页中的“如何格式化磁盘”
显示片信息。	使用 <code>format</code> 实用程序显示片信息。	第 227 页中的“如何显示磁盘片信息”
标记磁盘。	使用 <code>format</code> 实用程序创建磁盘标号。	第 230 页中的“如何标记磁盘”
检查磁盘标号。	使用 <code>prtvtoc</code> 命令检查磁盘标号。	第 233 页中的“如何检查磁盘标号”

任务	说明	参考
恢复损坏的磁盘标号。	可以尝试恢复因系统或电源故障而损坏的磁盘标号。	第 235 页中的 “如何恢复损坏的磁盘标号”
创建 <code>format.dat</code> 项。	创建 <code>format.dat</code> 项以支持第三方磁盘。	第 239 页中的 “如何创建 <code>format.dat</code> 项”
自动配置 SCSI 磁盘。	可以自动配置符合磁盘设备模式感知页面的 SCSI-2 规范的 SCSI 磁盘，即使 <code>/etc/format.dat</code> 文件中未列出该特定驱动器类型也是如此。	第 240 页中的 “如何自动配置 SCSI 驱动器”
确定有缺陷的磁盘扇区。	使用 <code>format</code> 实用程序确定有缺陷的磁盘扇区。	第 242 页中的 “如何使用表面分析确定有缺陷的扇区”
如有必要，请修复有缺陷的磁盘扇区。	使用 <code>format</code> 实用程序修复有缺陷的磁盘扇区。	第 243 页中的 “如何修复有缺陷的扇区”

确定系统中的磁盘

使用 `format` 实用程序搜索已连接到系统的磁盘的类型。还可以使用 `format` 实用程序验证磁盘是否是系统可识别的。有关使用 `format` 实用程序的详细信息，请参见[第 16 章](#)。

▼ 如何确定系统中的磁盘

- 1 成为超级用户或承担等效角色。
角色包含授权和具有一定权限的命令。有关角色的更多信息，请参见《系统管理指南：安全性服务》中的“配置 RBAC（任务列表）”。
- 2 使用 `format` 实用程序确定系统可识别的磁盘。

```
# format
```

`format` 实用程序在 `AVAILABLE DISK SELECTIONS` 下显示它所识别的磁盘的列表。

示例 12-1 确定系统中的磁盘

以下示例说明 `format` 命令的输出来自具有一个磁盘的系统。

```
# format

AVAILABLE DISK SELECTIONS:

0. c0t1d0 <FUJITSU MAN3367M SUN36G 1804 43d671f>
```

```
/pci@1f,0/pci@1,1/scsi@2/sd@1,0
```

该输出将磁盘的物理设备名称和逻辑设备名称与磁盘的市场名称（位于尖括号 <> 中）相关联。请参见下面的示例。使用此方法可以轻松地确定哪些逻辑设备名称表示已连接到系统的磁盘。有关逻辑设备名称和物理设备名称的说明，请参见第 10 章。

以下示例使用通配符显示已连接到第二个控制器的四个磁盘。

```
# format /dev/rdisk/c2*
```

```
AVAILABLE DISK SELECTIONS:
```

```
0. /dev/rdisk/c2t10d0s0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
```

```
/sbus@3,0/SUNW,fas@3,88000000/sd@a,0
```

```
1. /dev/rdisk/c2t11d0s0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
```

```
/sbus@3,0/SUNW,fas@3,88000000/sd@b,0
```

```
2. /dev/rdisk/c2t14d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
```

```
/sbus@3,0/SUNW,fas@3,88000000/sd@e,0
```

```
3. /dev/rdisk/c2t15d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
```

```
/sbus@3,0/SUNW,fas@3,88000000/sd@f,0
```

```
Specify disk (enter its number):
```

以下示例说明如何确定基于 SPARC 的系统中的磁盘。

```
# format
```

```
0. c0t1d0 <FUJITSU MAN3367M SUN36G 1804 43d671f>
```

```
/pci@1f,0/pci@1,1/scsi@2/sd@1,0
```

```
Specify disk (enter its number):
```

该输出表明磁盘 0（目标 1）已连接到第二个 SCSI 主机适配器 (scsi@2)，该适配器已连接到第二个 PCI 接口 (/pci@1f0/pci@1,1...)。该输出还将物理设备名称和逻辑设备名称与磁盘的市场名称 SUN36G 相关联。

以下示例说明如何确定基于 x86 的系统中的磁盘。

```
# format
```

AVAILABLE DISK SELECTIONS:

0. c0d0 <DEFAULT cyl 615 alt 2 hd 64 sec 63>

/pci@0,0/pci-ide@7,1/ata@0/cmdk@0,0

1. c0d1 <DEFAULT cyl 522 alt 2 hd 32 sec 63>

/pci@0,0/pci-ide@7,1/ata@0/cmdk@1,0

2. c1d0 <DEFAULT cyl 817 alt 2 hd 256 sec 63>

/pci@0,0/pci-ide@7,1/ata@1/cmdk@0,0

Specify disk (enter its number):

该输出说明磁盘 0 已连接到第一个 PCI 主机适配器 (pci-ide@7...), 该适配器已连接到 ATA 接口 (ata...)。基于 x86 的系统上的 `format` 输出不根据其市场名称确定磁盘。

更多信息 如果 `format` 实用程序无法识别磁盘...

- 请转至第 13 章或第 14 章。
- 请转至第 238 页中的“创建 `format.dat` 项”。
- 请转至第 230 页中的“如何标记磁盘”。
- 请通过参阅磁盘硬件文档将磁盘连接到系统。

格式化磁盘

磁盘通常由制造商或转售商进行格式化。在安装驱动器时，通常无需重新格式化磁盘。

必须已对磁盘进行格式化后才能执行以下操作：

- 将数据写入磁盘。不过，大多数磁盘都已经格式化。
- 使用 Solaris 安装实用程序安装系统。



注意 - 格式化磁盘是一个破坏性过程，因为它覆写磁盘上的数据。由于此原因，磁盘通常仅由制造商或转售商进行格式化。如果认为磁盘缺陷是导致某个问题反复出现的原因，则可以使用 `format` 实用程序执行表面分析。但是，应非常谨慎地仅使用不会破坏数据的命令。

▼ 如何确定磁盘是否已格式化

1 成为超级用户或承担等效角色。

2 调用 `format` 实用程序。

```
# format
```

将显示可用磁盘的编号列表。

3 键入要检查的磁盘的编号。

```
Specify disk (enter its number): 0
```

4 通过查看以下消息，验证所选磁盘是否已格式化：

```
[disk formatted]
```

示例 12-2 确定磁盘是否已格式化

以下示例说明磁盘 `c1t0d0` 已格式化。

```
# format /dev/rdisk/c1*
```

```
AVAILABLE DISK SELECTIONS:
```

```
0. /dev/rdisk/c1t0d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
```

```
    /sbus@2,0/QLGC,isp@2,10000/sd@0,0
```

```
1. /dev/rdisk/c1t1d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
```

```
    /sbus@2,0/QLGC,isp@2,10000/sd@1,0
```

```
2. /dev/rdisk/c1t8d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
```

```
    /sbus@2,0/QLGC,isp@2,10000/sd@8,0
```

```
3. /dev/rdisk/c1t9d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
```

```
    /sbus@2,0/QLGC,isp@2,10000/sd@9,0
```

```
Specify disk (enter its number): 0
```

```
selecting /dev/rdisk/c1t0d0s0
```

```
[disk formatted]
```

▼ 如何格式化磁盘

- 1 成为超级用户或承担等效角色。

- 2 调用 `format` 实用程序。

```
# format
```

将显示可用磁盘的编号列表。

- 3 键入要格式化的磁盘的编号。

```
Specify disk (enter its number): 0
```



注意 - 请勿选择系统磁盘。如果格式化系统磁盘，则会删除此磁盘上的 OS 和任何数据。

- 4 要开始格式化磁盘，请在 `format>` 提示符下键入 `format`。通过键入 `y` 确认该命令。

```
format> format
```

```
Ready to format. Formatting cannot be interrupted
```

```
and takes 23 minutes (estimated). Continue? yes
```

- 5 通过查看以下消息，验证磁盘格式化是否已成功：

```
Beginning format. The current time Tue ABC xx xx:xx:xx xxxx
```

```
Formatting...
```

```
done
```

```
Verifying media...
```

```
pass 0 - pattern = 0xc6dec6de
```

```
2035/12/18
```

```
pass 1 - pattern = 0x6db6db6d
```

```
2035/12/18
```


Total of 0 defective blocks repaired.

6 退出 format 实用程序。

format> quit

示例 12-3 格式化磁盘

以下示例说明如何格式化磁盘 c0t6d0。

format

Searching for disks...done

AVAILABLE DISK SELECTIONS:

- 0. c0t0d0 <SUNW18G cyl 7506 alt 2 hd 19 sec 248
/pci@1f,0/pci@1,1/scsi@2/sd@0,0
- 1. c0t1d0 <FUJITSU MAN3367M SUN36G 1804 43d671f>
/pci@1f,0/pci@1,1/scsi@2/sd@1,0
- 2. c0t2d0 <FUJITSU MAN3367M SUN36G 1804 43d671f>
/pci@1f,0/pci@1,1/scsi@2/sd@2,0
- 3. c0t3d0 <FUJITSU MAN3367M SUN36G 1804 43d671f>
/pci@1f,0/pci@1,1/scsi@2/sd@3,0
- 4. c0t4d0 <FUJITSU MAN3367M SUN36G 1804 43d671f>
/pci@1f,0/pci@1,1/scsi@2/sd@4,0
- 5. c0t5d0 <FUJITSU MAN3367M SUN36G 1804 43d671f>
/pci@1f,0/pci@1,1/scsi@2/sd@5,0
- 6. c0t6d0 <FUJITSU MAN3367M SUN36G 1804 43d671f>

```
                /pci@1f,0/pci@1,1/scsi@2/sd@6,0

Specify disk (enter its number): 6

selecting c0t6d0

[disk formatted]

format> format

Ready to format.  Formatting cannot be interrupted
and takes 332 minutes (estimated).  Continue? y

Beginning format.  The current time is Wed Jan  7 16:16:05 2004

Formatting...

    99% complete (00:00:21 remaining) done

Verifying media...

    pass 0 - pattern = 0xc6dec6de

    71132922

    pass 1 - pattern = 0x6db6db6d

    71132922

Total of 0 defective blocks repaired.

format> quit
```

显示磁盘片

可以使用 `format` 实用程序检查磁盘是否具有适当的磁盘片。如果确定磁盘不包含要使用的片，请使用 `format` 实用程序重新创建它们并标记磁盘。有关创建磁盘片的信息，请参见第 250 页中的“SPARC: 如何创建磁盘片和标记磁盘”或第 276 页中的“x86: 如何创建磁盘片和标记磁盘”。

注 – `format` 实用程序使用术语 *partition* 而不是 *slice*。

▼ 如何显示磁盘片信息

1 成为超级用户或承担等效角色。

2 调用 `format` 实用程序。

```
# format
```

将显示可用磁盘的编号列表。

3 键入要显示其片信息的磁盘的编号。

```
Specify disk (enter its number):1
```

4 选择 `partition` 菜单。

```
format> partition
```

5 显示选定磁盘的片信息。

```
partition> print
```

6 退出 `format` 实用程序。

```
partition> q
```

```
format> q
```

```
#
```

7 通过确定特定的片标记和片，验证所显示的片信息。

如果屏幕输出说明未指定片大小，则磁盘很可能没有片。

示例 12-4 显示磁盘片信息

以下示例显示标号为 VTOC 的磁盘的片信息。

```
# format

Searching for disks...done

Specify disk (enter its number):1

Selecting c0t0d0

format> partition

partition> print

Current partition table (original):

Total disk cylinders available: 8892 + 2 (reserved cylinders)


```

Part	Tag	Flag	Cylinders	Size	Blocks
0	root	wm	1110 - 4687	1.61GB	(0/3578/0) 3381210
1	swap	wu	0 - 1109	512.00MB	(0/1110/0) 1048950
2	backup	wm	0 - 8891	4.01GB	(0/8892/0) 8402940
3	unassigned	wm	0	0	(0/0/0) 0
4	unassigned	wm	0	0	(0/0/0) 0
5	unassigned	wm	0	0	(0/0/0) 0
6	unassigned	wm	0	0	(0/0/0) 0
7	home	wm	4688 - 8891	1.89GB	(0/4204/0) 3972780

```
partition> q

format> q

#
```

有关这些示例中片信息的详细说明，请参见第 11 章。

以下示例显示标号为 EFI 的磁盘的片信息。

```
# format

Searching for disks...done
```

```
Specify disk (enter its number): 9
```

```
selecting c4t1d0
```

```
[disk formatted]
```

```
format> partition
```

```
partition> print
```

```
Current partition table (original):
```

```
partition> q
```

```
format> q
```

Part	Tag	Flag	First Sector	Size	Last Sector
0	root	wm	34	300.00GB	629145633
1	usr	wm	629145634	300.00GB	1258291233
2	unassigned	wm	0	0	0
3	unassigned	wm	0	0	0
4	unassigned	wm	0	0	0
5	unassigned	wm	0	0	0
6	usr	wm	1258291234	628.77GB	2576924637
8	reserved	wm	2576924638	8.00MB	2576941021

创建和检查磁盘标号

标记磁盘通常是在系统安装过程中或者创建新磁盘片时进行的。如果磁盘标号已损坏，则可能需要重新标记磁盘。例如，磁盘标号因电源故障而损坏。

`format` 实用程序会尝试自动配置任何未标记的 SCSI 磁盘。如果 `format` 实用程序能够自动配置未标记的磁盘，则它将显示与以下内容类似的消息：

```
c0t0d1: configured with capacity of 4.00GB
```

提示 – 有关标记具有相同磁盘标号的多个磁盘的信息，请参见第 246 页中的“使用 `prtvtoc` 和 `fmthard` 命令标记多个磁盘”。

▼ 如何标记磁盘

可以使用接下来的过程执行以下操作：

- 用标号 VTOC 标记磁盘。
- 用标号 EFI 标记大于 1 TB 的磁盘。

如果要用标号 EFI 标记小于 1 TB 的磁盘，请参见[示例 12-6](#)。

1 成为超级用户或承担等效角色。

2 调用 `format` 实用程序。

```
# format
```

将显示可用磁盘的编号列表。

3 键入要标记的磁盘的编号。

```
Specify disk (enter its number):1
```

如果 `format` 实用程序识别该磁盘类型，则下一步是搜索备份标号以标记磁盘。使用备份标号标记磁盘时，会以正确的分区信息、磁盘类型和磁盘几何参数标记磁盘。

4 选择以下内容之一以标记磁盘：

- 如果磁盘是未标记的但已成功配置，请转至步骤 5 标记磁盘。
`format` 实用程序将询问您是否要标记磁盘。
- 如果已标记磁盘但您要更改磁盘类型，或者如果 `format` 实用程序无法自动配置磁盘，请继续执行步骤 6 设置磁盘类型并标记磁盘。

5 通过在 `Label it now?` 提示符下键入 `y` 来标记磁盘。

```
Disk not labeled. Label it now? y
```

现在已标记磁盘。转至步骤 10 以退出 `format` 实用程序。

6 在 `format>` 提示符下输入 `type`。

```
format> type
```

将显示 "Available Drive Types" 菜单。

7 从可能的磁盘类型列表中选择一种磁盘类型。

```
Specify disk type (enter its number)[12]: 12
```

或者，选择 **0** 自动配置 SCSI-2 磁盘。有关更多信息，请参见第 240 页中的“如何自动配置 SCSI 驱动器”。

- 8 标记磁盘。**如果磁盘未标记，则显示以下消息。

```
Disk not labeled. Label it now? y
```

否则，将显示以下消息提示您：

```
Ready to label disk, continue? y
```

- 9 验证磁盘标号。**

```
format> verify
```

- 10 退出 format 实用程序。**

```
format> q
```

```
#
```

示例 12-5 标记磁盘

以下示例说明如何自动配置并标记 1.05 GB 磁盘。

```
# format
```

```
clt0d0: configured with capacity of 1002.09MB
```

```
AVAILABLE DISK SELECTIONS:
```

```
0. c0t3d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
```

```
/iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@1,0
```

```
1. clt0d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
```

```
/iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@1,0
```

```
Specify disk (enter its number): 1
```

```
Disk not labeled. Label it now? yes
```

```
format> verify
```

```
format> q
```

```
#
```

示例 12-6 用标号 EFI 标记小于 1 TB 的磁盘

以下示例说明如何使用 `format -e` 命令以标号 EFI 标记小于 1 TB 的磁盘。请记住验证分层软件产品是否继续在包含标号 EFI 的磁盘的系统上起作用。有关 EFI 标号限制的常规信息，请参见第 203 页中的“EFI 磁盘标号限制”。

```
# format -e

Searching for disks...done

AVAILABLE DISK SELECTIONS:

    1. clt0d0 <SUNW18g cyl 7506 alt 2 hd 19 sec 248>

        /sbus@2,0/QLGC,isp@2,10000/sd@0,0

    2. clt1d0 <SUNW18g cyl 7506 alt 2 hd 19 sec 248>

        /sbus@2,0/QLGC,isp@2,10000/sd@1,0

    3. clt8d0 <SUNW18g cyl 7506 alt 2 hd 19 sec 248>

        /sbus@2,0/QLGC,isp@2,10000/sd@8,0

    4. clt9d0 <SUNW18g cyl 7506 alt 2 hd 19 sec 248>

        /sbus@2,0/QLGC,isp@2,10000/sd@9,0

Specify disk (enter its number): 4

selecting clt9d0

[disk formatted]

format> label

[0] SMI Label

[1] EFI Label

Specify Label type[0]: 1

Ready to label disk, continue? yes

format> quit
```


▼ 如何检查磁盘标号

使用 `prtvtoc` 命令检查磁盘标号信息。有关磁盘标号的详细说明和 `prtvtoc` 命令所显示的信息，请参见第 11 章。

- 1 成为超级用户或承担等效角色。
- 2 显示磁盘标号信息。

```
# prtvtoc /dev/rdisk/device-name
```

其中 `device-name` 是要检查的原始磁盘设备。

示例 12-7 检查磁盘标号

以下示例说明标号为 VTOC 的磁盘的磁盘标号信息。

```
# prtvtoc /dev/rdisk/c0t0d0s0

* /dev/rdisk/c0t0d0s0 partition map
*
* Dimensions:
*
*      512 bytes/sector
*
*      63 sectors/track
*
*      15 tracks/cylinder
*
*      945 sectors/cylinder
*
*      8894 cylinders
*
*      8892 accessible cylinders
*
*
* Flags:
*
*      1: unmountable
*
*      10: read-only
*
*
*                               First      Sector      Last
```

* Partition	Tag	Flags	Sector	Count	Sector	Mount Directory
0	2	00	1048950	3381210	4430159	/
1	3	01	0	1048950	1048949	
2	5	00	0	8402940	8402939	
7	8	00	4430160	3972780	8402939	/export/home

以下示例说明标号为 EFI 的磁盘的磁盘标号信息。

```
# prtvtoc /dev/rdisk/c3t1d0s0

* /dev/rdisk/c3t1d0s0 partition map

*

* Dimensions:

*   512 bytes/sector

* 2479267840 sectors

* 2479267773 accessible sectors

*

* Flags:

*   1: unmountable

*  10: read-only

*

*           First      Sector      Last
* Partition Tag  Flags  Sector    Count      Sector  Mount Directory
  0      2    00        34    262144    262177
  1      3    01    262178    262144    524321
  6      4    00    524322 2478727100 2479251421
  8     11    00 2479251422    16384 2479267805
```

恢复损坏的磁盘标号

有时，电源或系统故障会导致磁盘标号变得无法识别。损坏的磁盘标号并不总是意味着，必须重新创建或恢复片信息或磁盘数据。

要恢复损坏的磁盘标号，首先应使用正确的几何参数和磁盘类型信息标记磁盘。可以通过常规的磁盘标记方法完成该步骤，即使用自动配置或手动指定磁盘类型。

如果 `format` 实用程序识别该磁盘类型，则下一步是搜索备份标号以标记磁盘。使用备份标号标记磁盘时，会以正确的分区信息、磁盘类型和磁盘几何参数来标记磁盘。

▼ 如何恢复损坏的磁盘标号

1 将系统引导到单用户模式。

如有必要，请在单用户模式下从本地 CD-ROM 或网络引导系统以访问磁盘。

有关引导系统的信息，请参见《系统管理指南：基本管理》中的第 10 章“引导系统（任务）”或《系统管理指南：基本管理》中的第 11 章“基于 GRUB 的引导（任务）”。

2 重新标记磁盘。

```
# format
```

`format` 实用程序会尝试自动配置任何未标记的 SCSI 磁盘。如果 `format` 实用程序能够配置未标记的已损坏磁盘，则它将显示以下消息：

```
cwtxdy: configured with capacity of abcMB
```

`format` 实用程序然后显示系统中磁盘的编号列表。

3 键入需要恢复的磁盘的编号。

```
Specify disk (enter its number): 1
```

4 选择以下内容之一以确定如何标记磁盘。

- 如果已成功配置磁盘，则按照步骤 5 和 6 进行操作，然后转至步骤 12。
- 如果磁盘未成功配置，则按照步骤 7–11 操作，然后转至步骤 12。

5 搜索备份标号。

```
format> verify
```

```
Warning: Could not read primary label.
```

```
Warning: Check the current partitioning and 'label' the disk or
```

```
use the 'backup' command.
```

Backup label contents:

Volume name = < >

ascii name = <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>

pcyl = 2038

ncyl = 2036

acyl = 2

nhead = 14

nsect = 72

Part	Tag	Flag	Cylinders	Size	Blocks	
0	root	wm	0 - 300	148.15MB	(301/0/0)	303408
1	swap	wu	301 - 524	110.25MB	(224/0/0)	225792
2	backup	wm	0 - 2035	1002.09MB	(2036/0/0)	2052288
3	unassigned	wm	0	0	(0/0/0)	0
4	unassigned	wm	0	0	(0/0/0)	0
5	unassigned	wm	0	0	(0/0/0)	0
6	usr	wm	525 - 2035	743.70MB	(1511/0/0)	1523088
7	unassigned	wm	0	0	(0/0/0)	0

- 6 如果 format 实用程序找到备份标号，而且备份标号的内容符合要求，请使用 backup 命令以备份标号标记磁盘。

format> **backup**

Disk has a primary label, still continue? **y**

Searching for backup labels...found.

Restoring primary label

磁盘标号已恢复。转至步骤 12。

- 7 如果 `format` 实用程序无法自动配置磁盘，请使用 `type` 命令指定磁盘类型。

```
format> type
```

将显示 "Available Drive Types" 菜单。

- 8 选择 0 以自动配置磁盘。或者，从可能的磁盘类型列表中选择一种磁盘类型。

```
Specify disk type (enter its number)[12]: 12
```

- 9 如果成功配置磁盘，则在 `format` 实用程序询问您是否要标记磁盘时应答 `no`。

```
Disk not labeled. Label it now? no
```

- 10 使用 `verify` 命令搜索备份标号。

```
format> verify
```

```
Warning: Could not read primary label.
```

```
Warning: Check the current partitioning and 'label' the disk
```

```
or use the 'backup' command.
```

```
.
```

```
.
```

```
.
```

- 11 如果 `format` 实用程序找到备份标号，而且备份标号的内容符合要求，请使用 `backup` 命令以备份标号标记磁盘。

```
format> backup
```

```
Disk has a primary label, still continue? y
```

```
Searching for backup labels...found.
```

```
Restoring primary label
```

磁盘标号已恢复。

- 12 退出 `format` 实用程序。

```
format> q
```

- 13 使用 `fsck` 命令验证已恢复磁盘上的文件系统。

有关使用 `fsck` 命令的信息，请参见第 22 章。

添加第三方磁盘

Solaris OS 支持许多第三方磁盘。但是，为了识别磁盘，您可能需要提供设备驱动程序和/或 `format.dat` 项。用于添加磁盘的其他选项如下：

- 如果要添加 SCSI 磁盘，则可能尝试使用 `format` 实用程序的自动配置功能。有关更多信息，请参见第 239 页中的“自动配置 SCSI 磁盘驱动器”。
- 可能尝试热插拔 PCI、SCSI 或 USB 磁盘。有关更多信息，请参见第 5 章。

如果第三方磁盘设计为使用标准的 SunOS 兼容设备驱动程序，则创建适当的 `format.dat` 项应该足以使 `format` 实用程序能够识别该磁盘。在其他情况下，需要装入第三方设备驱动程序以支持磁盘。

注 - Sun 无法保证其 `format` 实用程序将与所有第三方磁盘驱动程序兼容。如果磁盘驱动程序与 Solaris `format` 实用程序不兼容，则磁盘驱动器供应商应该为您提供自定义的磁盘格式化程序。

本节讨论在缺少某些此软件支持时需要执行的操作。通常，在调用 `format` 实用程序如果发现未识别出磁盘类型，则说明缺少软件支持。

如本节所述提供缺少的软件。然后，参阅第 13 章或第 14 章中添加系统磁盘或辅助磁盘的相应配置过程。

创建 `format.dat` 项

如果没有有关磁盘几何参数和操作参数的准确信息，则无法格式化无法识别的磁盘。此信息在 `/etc/format.dat` 文件中提供。

注 - SCSI-2 磁盘不需要 `format.dat` 项。如果在重新配置引导过程中打开了磁盘电源，则 `format` 实用程序自动配置 SCSI-2 驱动程序。有关自动配置 SCSI 磁盘驱动器的逐步说明，请参见第 240 页中的“如何自动配置 SCSI 驱动器”。

如果磁盘是无法识别的，请使用文本编辑器在 `format.dat` 文件中为该磁盘创建项。在开始之前，需要收集与磁盘及其控制器有关的所有技术规格。磁盘应该附带有此信息。如果没有此信息，请与磁盘制造商或供应商联系。

▼ 如何创建 format.dat 项

- 1 成为超级用户或承担等效角色。
- 2 生成 /etc/format.dat 文件的副本。
cp /etc/format.dat /etc/format.dat.gen
- 3 修改 /etc/format.dat 文件以包括第三方磁盘的项。
使用第 16 章中所述的 format.dat 信息。
此外，使用磁盘的硬件产品文档收集所需的信息。

自动配置 SCSI 磁盘驱动器

format 实用程序自动配置 SCSI 磁盘驱动器，即使 /etc/format.dat 文件中未列出该特定类型的驱动器也将如此。通过此功能，可以对符合磁盘设备模式感知页面的 SCSI-2 规范的任何磁盘驱动程序进行格式化、创建片和标记。

以下是用于添加磁盘的其他选项：

- 如果要添加 SCSI 磁盘，则可能尝试使用 format 实用程序的自动配置功能。
- 可能尝试热插拔 PCI、SCSI 或 USB 磁盘。有关更多信息，请参见第 5 章。

使用自动配置功能配置 SCSI 驱动器涉及以下步骤：

- 关闭系统
- 将 SCSI 磁盘驱动器连接到系统
- 打开磁盘驱动器
- 执行重新配置引导
- 使用 format 实用程序自动配置 SCSI 磁盘驱动器

在执行重新配置引导后，调用 format 实用程序。format 实用程序将尝试配置磁盘；如果配置成功，则提醒用户已配置磁盘。有关自动配置 SCSI 磁盘驱动器的逐步说明，请参见第 240 页中的“如何自动配置 SCSI 驱动器”。

以下是 format 实用程序所显示的 1.3 GB SCSI 磁盘驱动器分区表的示例。

Part	Tag	Flag	Cylinders	Size	Blocks
0	root	wm	0 - 96	64.41MB	(97/0/0)
1	swap	wu	97 - 289	128.16MB	(193/0/0)
2	backup	wu	0 - 1964	1.27GB	(1965/0/0)
6	usr	wm	290 - 1964	1.09GB	(1675/0/0)

▼ 如何自动配置 SCSI 驱动器

1 成为超级用户或等效角色。

2 创建将在引导系统时读取的 `/reconfigure` 文件。

```
# touch /reconfigure
```

3 关闭系统。

```
# shutdown -i0 -gn -y
```

-i0 将系统降至 init 级别 0（即电源关闭状态）。

-gn 通知已登录的用户，*n* 秒之后系统将开始关闭。

-y 指定命令应该在没有用户干预的情况下运行。

在关闭系统后将显示 ok 提示符。

4 关闭系统和所有外部外围设备的电源。

5 确保所添加的磁盘与系统中的其他设备具有不同的目标号码。

通常，磁盘背面有一个用于此目的的小开关。

6 将磁盘连接到系统，并检查物理连接。

有关详细信息，请参阅磁盘的硬件安装指南。

7 打开所有外部外围设备的电源。

8 打开系统的电源。

系统将引导并显示登录提示。

9 再次作为超级用户登录或承担等效角色。

10 调用 `format` 实用程序，然后选择要自动配置的磁盘。

```
# format
```

```
Searching for disks...done
```

```
clt0d0: configured with capacity of 1002.09MB
```

```
AVAILABLE DISK SELECTIONS:
```

```
0. c0t1d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
```

```
/iommu@f,e0000000/sbus@f,e0001000/espdma@f,4000000/esp@f,8000000/sd@1,0
```



```
1. c0t3d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>

/iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@3,0

Specify disk (enter its number): 1
```

11 作为对提示的响应，键入 y 以标记磁盘。
键入 y 时，将使用 SCSI 自动配置生成磁盘标号并将其写入磁盘。
Disk not labeled. Label it now? y

12 验证磁盘标号。
format> verify

13 退出 format 实用程序。
format> q

修复有缺陷的扇区

如果系统中的磁盘存在有缺陷的扇区，可以按照本节中的过程修复磁盘。在执行以下操作时，可能会发现存在有缺陷的扇区：

- 对磁盘运行表面分析
有关 format 实用程序的分析功能的更多信息，请参见第 314 页中的“analyze 菜单”。

如果系统中的磁盘包含有缺陷的扇区，可以按照本节中的过程修复磁盘。由于系统同时对磁盘上的许多扇区执行操作，因此，通常很难准确查明是哪个扇区导致了给定的错误。要找到确切的一个或多个扇区，请使用第 242 页中的“如何使用表面分析确定有缺陷的扇区”。

- 系统运行时，从磁盘驱动程序中获得有关磁盘特定部分的多个错误消息。
与磁盘错误有关的控制台消息与以下内容类似：

```
WARNING: /io-unit@f,e0200000/sbi@0,0/QLGC,isp@1,10000/sd@3,0 (sd33):

Error for command 'read' Error Level: Retryable

Requested Block 126, Error Block: 179

Sense Key: Media Error

Vendor 'name':
```

```
ASC = 0x11 (unrecovered read error), ASCQ = 0x0, FRU = 0x0
```

此消息指示块 179 可能有缺陷。使用 `format` 实用程序的 `repair` 命令，可以重新定位有缺陷的块。或者，可以在启用修复选项的情况下使用 `analyze` 命令。

▼ 如何使用表面分析确定有缺陷的扇区

- 1 成为超级用户或承担等效角色。

- 2 取消挂载包含有缺陷扇区的片中的文件系统。

```
# umount /dev/dsk/device-name
```

有关更多信息，请参见 `mount(1M)`。

- 3 调用 `format` 实用程序。

```
# format
```

- 4 选择受影响的磁盘。

```
Specify disk (enter its number):1
```

```
selecting c0t2d0:
```

```
[disk formatted]
```

```
Warning: Current Disk has mounted partitions.
```

- 5 选择 `analyze` 菜单。

```
format> analyze
```

- 6 通过在 `analyze>` 提示符下键入 `setup` 设置分析参数。

使用如下所示的参数：

```
analyze> setup
```

```
Analyze entire disk [yes]? n
```

```
Enter starting block number [0, 0/0/0]: 12330
```

```
Enter ending block number [2052287, 2035/13/71]: 12360
```

```
Loop continuously [no]? y
```

```
Repair defective blocks [yes]? n
```

```

Stop after first error [no]? n

Use random bit patterns [no]? n

Enter number of blocks per transfer [126, 0/1/54]: 1

Verify media after formatting [yes]? y

Enable extended messages [no]? n

Restore defect list [yes]? y

Create defect label [yes]? y

```

7 使用 read 命令查找缺陷。

```

analyze> read

Ready to analyze (won't harm SunOS). This takes a long time,
but is interruptible with Control-C. Continue? y

    pass 0

2035/12/1825/7/24

    pass 1

Block 12354 (18/4/18), Corrected media error (hard data ecc)

25/7/24

^C

Total of 1 defective blocks repaired.

```

▼ 如何修复有缺陷的扇区

- 1 成为超级用户或承担等效角色。
 - 2 调用 format 实用程序。
- ```
format
```

**3 选择存在有缺陷扇区的磁盘。**

```
Specify disk (enter its number): 1
```

```
selecting c0t3d0
```

```
[disk formatted]
```

```
format>
```

**4 选择 repair 命令。**

```
format> repair
```

**5 键入有缺陷块的编号。**

```
Enter absolute block number of defect: 12354
```

```
Ready to repair defect, continue? y
```

```
Repairing block 12354 (18/4/18)...ok.
```

```
format>
```

如果不能确定用于确定有缺陷扇区的格式，请参见第 242 页中的[“如何使用表面分析确定有缺陷的扇区”](#)以了解更多信息。

## 管理磁盘的提示和技巧

使用以下提示可帮助您更有效地管理磁盘。

### 调试 format 会话

调用 `format -M` 命令为 ATA 和 SCSI 设备启用扩展消息和诊断消息。

示例 12-8 调试 format 会话

在此示例中，Inquiry 下的数字系列表示在数字右侧显示的 inquiry 数据的十六进制值。

```
format -M
```

```
Searching for disks...done
```

```
AVAILABLE DISK SELECTIONS:
```

示例 12-8 调试 format 会话 (续)

```

0. c0t1d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>

 /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@1,0

1. c0t3d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>

 /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@3,0

Specify disk (enter its number): 0

selecting c0t3d0

[disk formatted]

format> inquiry

Inquiry:

00 00 02 02 8f 00 00 12 53 45 41 47 41 54 45 20 NAME....
53 54 31 31 32 30 30 4e 20 53 55 4e 31 2e 30 35 ST11200N SUN1.05
38 33 35 38 30 30 30 33 30 32 30 39 00 00 00 00 835800030209....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 43 6f 70 79 72 69 67 68 74 20 28 63 29 20 31 .Copyright (c) 1
39 39 32 20 53 65 61 67 61 74 65 20 41 6c 6c 20 992 NAME All
72 69 67 68 74 73 20 72 65 73 65 72 76 65 64 20 rights reserved
30 30 30 000

Vendor: name

Product: ST11200N SUN1.05

Revision: 8358

```

示例 12-8 调试 format 会话 (续)

```
format>
```

## 使用 prtvtoc 和 fmthard 命令标记多个磁盘

使用 prtvtoc 和 fmthard 命令标记具有相同磁盘几何参数的多个磁盘。

在脚本中使用以下 for 循环，从一个磁盘复制磁盘标号，然后在多个磁盘上复制它。

```
for i in xyz
> do
> prtvtoc /dev/rdisk/cwtxdysz | fmthard -s - /dev/rdisk/cwt${i}d0s2
> done
```

示例 12-9 标记多个磁盘

在此示例中，将来自 c2t0d0s0 的磁盘标号复制到其他四个磁盘。

```
for i in 1 2 3 5
> do
> prtvtoc /dev/rdisk/c2t0d0s0 | fmthard -s - /dev/rdisk/c2t${i}d0s2
> done

fmthard: New volume table of contents now in place.

fmthard: New volume table of contents now in place.

fmthard: New volume table of contents now in place.

fmthard: New volume table of contents now in place.

#
```

## SPARC：添加磁盘（任务）

---

本章介绍如何将磁盘添加到 SPARC 系统。

有关与将磁盘添加到 SPARC 系统关联的过程的信息，请参见第 247 页中的“[SPARC: 添加系统磁盘或辅助磁盘（任务列表）](#)”。

有关磁盘管理的概述信息，请参见第 11 章。有关将磁盘添加到基于 x86 的系统的逐步说明，请参见第 14 章。

### SPARC: 添加系统磁盘或辅助磁盘（任务列表）

以下任务列表介绍了将磁盘添加到基于 SPARC 的系统的过程。

| 任务            | 说明                                                                                                  | 参考                                                                                                         |
|---------------|-----------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------|
| 1. 连接磁盘并进行引导。 | <b>系统磁盘</b><br>连接新磁盘并从本地或远程 Solaris CD 或 DVD 进行引导。<br><br><b>辅助磁盘</b><br>连接新磁盘并执行重新配置引导，以便系统可识别该磁盘。 | 第 248 页中的“ <a href="#">SPARC: 如何连接系统磁盘并进行引导</a> ”<br><br>第 249 页中的“ <a href="#">SPARC: 如何连接辅助磁盘并进行引导</a> ” |
| 2. 创建片并标记磁盘。  | 如果磁盘制造商尚未创建磁盘片和标记磁盘，请创建磁盘片并标记磁盘。                                                                    | 第 250 页中的“ <a href="#">SPARC: 如何创建磁盘片和标记磁盘</a> ”                                                           |
| 3. 创建文件系统。    | 使用 <code>newfs</code> 命令在磁盘片上创建 UFS 文件系统。必须为系统磁盘创建根 (/) 文件系统和/或 <code>/usr</code> 文件系统。             | 第 258 页中的“ <a href="#">SPARC: 如何创建 UFS 文件系统</a> ”                                                          |

| 任务         | 说明                                                  | 参考                                              |
|------------|-----------------------------------------------------|-------------------------------------------------|
| 4. 恢复文件系统。 | 在系统磁盘上恢复根 (/) 文件系统和/或 /usr 文件系统。如有必要，请在辅助磁盘上恢复文件系统。 | <a href="#">第 27 章</a>                          |
| 5. 安装引导块。  | <b>仅限系统磁盘。</b> 在根 (/) 文件系统中安装引导块，以便该系统可以引导。         | <a href="#">第 259 页中的“SPARC: 如何在系统磁盘上安装引导块”</a> |

## SPARC: 添加系统磁盘或辅助磁盘

系统磁盘包含根 (/) 文件系统和/或 /usr 文件系统。如果包含其中任一文件系统的磁盘损坏，则可以使用以下两种方法恢复文件系统：

- 可以重新安装整个 Solaris OS。
- 或者，可以更换系统磁盘，然后从备份介质恢复文件系统。

辅助磁盘不包含根 (/) 文件系统和 /usr 文件系统。辅助磁盘通常包含用于存储用户文件的空間。可以将辅助磁盘添加到系统以获得更多磁盘空间。或者，可以更换损坏的辅助磁盘。如果更换系统中的辅助磁盘，则可以在新磁盘上恢复旧磁盘的数据。

### ▼ SPARC: 如何连接系统磁盘并进行引导

此过程假定系统已关闭。

- 1 断开损坏的系统磁盘与系统之间的连接。
- 2 确保所添加的磁盘与系统中的其他设备具有不同的目标号码。  
通常，磁盘背面有一个用于此目的的小开关。
- 3 将用来替换的系统磁盘连接到系统，并检查物理连接。  
有关详细信息，请参阅磁盘的硬件安装指南。
- 4 根据从本地 Solaris CD 或 DVD 还是从网络中的远程 Solaris CD 或 DVD 引导，按照下表中的说明进行操作。



| 引导类型                      | 操作                                                                                                      |
|---------------------------|---------------------------------------------------------------------------------------------------------|
| 从本地驱动器中的 Solaris CD 或 DVD | 1. 确保 Solaris 软件 1 CD 或 Solaris DVD 位于驱动器中。<br><br>2. 从介质引导到单用户模式：<br><br><code>ok boot cdrom -s</code> |
| 从网络                       | 从网络引导到单用户模式：<br><br><code>ok boot net -s</code>                                                         |

几分钟后，将显示超级用户提示符 (#)。

更多信息 **连接系统磁盘并进行引导后...**

引导系统后，可以在磁盘上创建片和磁盘标号。请转至第 250 页中的“[SPARC: 如何创建磁盘片和标记磁盘](#)”。

▼ **SPARC: 如何连接辅助磁盘并进行引导**

如果要添加磁盘标号为 EFI 的磁盘，请参见第 202 页中的“[带有 EFI 磁盘标号的多 TB 磁盘支持](#)”以了解更多信息。

通常，大多数新式总线类型支持热插拔。如果系统的总线类型支持热插拔，则可能无需执行下面的步骤 2 或 3。

有关热插拔设备的更多信息，请参见第 6 章。

- 1 成为超级用户或承担等效角色。
- 2 （可选的）如果 Solaris 软件不支持该磁盘类型，请按照硬件附带的说明，添加磁盘的设备驱动程序。  
有关为磁盘创建 `format.dat` 项的信息，请参见第 239 页中的“[如何创建 `format.dat` 项](#)”（如有必要）。

- 3 （可选的）创建将在引导系统时读取的 `/reconfigure` 文件。

```
touch /reconfigure
```

稍后打开系统电源或引导系统时，`/reconfigure` 文件会使 SunOS™ 软件检查是否存在任何新安装的外围设备。

- 4 关闭系统。

```
shutdown -i0 -g1 -y
```

`-i0` 更改为运行级 0（即关闭电源状态）。

-gn 通知已登录的用户，*n* 秒之后系统将开始关闭。

-y 指定命令应该在没有任何用户干预的情况下运行。

在关闭 Solaris OS 后将显示 ok 提示符。

- 5 关闭系统和所有外部外围设备的电源。
- 6 确保所添加的磁盘与系统中的其他设备具有不同的目标号码。  
通常，磁盘背面有一个用于此目的的小开关。
- 7 将磁盘连接到系统并检查物理连接。  
有关详细信息，请参阅磁盘的硬件安装指南。
- 8 打开所有外部外围设备的电源。
- 9 打开系统的电源。  
系统将引导并显示登录提示。

#### 更多信息 连接辅助磁盘并进行引导之后

引导系统后，可以在磁盘上创建片和磁盘标号。请转至第 250 页中的“[SPARC: 如何创建磁盘片和标记磁盘](#)”。

## ▼ SPARC: 如何创建磁盘片和标记磁盘

- 1 成为超级用户或承担等效角色。
- 2 调用 `format` 实用程序。  

```
format
```

将显示可用磁盘的编号列表。有关更多信息，请参见 `format(1M)`。
- 3 键入要重新分区的磁盘的编号。  

```
Specify disk (enter its number): disk-number
```

*disk-number* 是要重新分区的磁盘的编号。
- 4 选择 `partition` 菜单。  

```
format> partition
```

**5 显示当前分区 (片) 表。**

```
partition> print
```

**6 开始修改过程。**

```
partition> modify
```

**7 将磁盘全部设置为浮动片 (free hog slice)。**

```
Choose base (enter number) [0]? 1
```

有关浮动片 (free hog slice) 的更多信息，请参见第 217 页中的“使用浮动片 (free hog slice)”。

**8 通过在提示您继续时应答 y 创建新分区表。**

```
Do you wish to continue creating a new partition table based on
```

```
above table[yes]? y
```

**9 出现提示时，标识浮动 (free hog) 分区 (片) 和片大小。**

添加系统磁盘时，必须为以下项设置片：

- 根文件系统 (片 0) 和交换区域 (片 1)
- /usr (片 6)

标识片后，将显示新的分区表。

有关创建磁盘片的示例，请参见示例 13-1。

**10 通过在提示时应答 y，使显示的分区表成为当前分区表。**

```
Okay to make this the current partition table[yes]? y
```

如果不需要当前分区表但希望对其进行更改，请应答 no 并转至步骤 6。

**11 命名分区表。**

```
Enter table name (remember quotes): "partition-name"
```

其中 *partition-name* 是新分区表的名称。

**12 在新磁盘上完成片的分配后，用新分区表标记磁盘。**

```
Ready to label disk, continue? yes
```

**13 退出 partition 菜单。**

```
partition> q
```

**14 验证磁盘标号。**

```
format> verify
```

**15 退出 format 实用程序。**

```
format> q
```

**示例 13-1 SPARC: 创建磁盘片和标记系统磁盘**

以下示例说明如何使用 `format` 实用程序将 18 GB 的磁盘分成三个片：其中一个片用于根 (/) 文件系统，一个片用于交换区域，还有一个片用于 /usr 文件系统。

```
format
```

```
AVAILABLE DISK SELECTIONS:
```

```
0. /dev/rdisk/clt0d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
```

```
 /sbus@2,0/QLGC,isp@2,10000/sd@0,0
```

```
1. /dev/rdisk/clt1d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
```

```
 /sbus@2,0/QLGC,isp@2,10000/sd@1,0
```

```
2. /dev/rdisk/clt8d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
```

```
 /sbus@2,0/QLGC,isp@2,10000/sd@8,0
```

```
3. /dev/rdisk/clt9d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
```

```
 /sbus@2,0/QLGC,isp@2,10000/sd@9,0
```

```
Specify disk (enter its number): 0
```

```
selecting clt0d0
```

```
[disk formatted]
```

```
format> partition
```

```
partition> print
```

```
partition> modify
```

```
Select partitioning base:
```

```
0. Current partition table (original)
```

```
1. All Free Hog
```

| Part | Tag        | Flag | Cylinders | Size    | Blocks     |          |
|------|------------|------|-----------|---------|------------|----------|
| 0    | root       | wm   | 0         | 0       | (0/0/0)    | 0        |
| 1    | swap       | wu   | 0         | 0       | (0/0/0)    | 0        |
| 2    | backup     | wu   | 0 - 7505  | 16.86GB | (7506/0/0) | 35368272 |
| 3    | unassigned | wm   | 0         | 0       | (0/0/0)    | 0        |
| 4    | unassigned | wm   | 0         | 0       | (0/0/0)    | 0        |
| 5    | unassigned | wm   | 0         | 0       | (0/0/0)    | 0        |
| 6    | usr        | wm   | 0         | 0       | (0/0/0)    | 0        |
| 7    | unassigned | wm   | 0         | 0       | (0/0/0)    | 0        |

Choose base (enter number) [0]? 1

table based on above table[yes]? yes

Free Hog partition[6]? 6

Enter size of partition '0' [0b, 0c, 0.00mb, 0.00gb]: 4gb

Enter size of partition '1' [0b, 0c, 0.00mb, 0.00gb]: 4gb

Enter size of partition '3' [0b, 0c, 0.00mb, 0.00gb]:

Enter size of partition '4' [0b, 0c, 0.00mb, 0.00gb]:

Enter size of partition '5' [0b, 0c, 0.00mb, 0.00gb]:

Enter size of partition '7' [0b, 0c, 0.00mb, 0.00gb]:

| Part | Tag        | Flag | Cylinders   | Size    | Blocks     |          |
|------|------------|------|-------------|---------|------------|----------|
| 0    | root       | wm   | 0 - 1780    | 4.00GB  | (1781/0/0) | 8392072  |
| 1    | swap       | wu   | 1781 - 3561 | 4.00GB  | (1781/0/0) | 8392072  |
| 2    | backup     | wu   | 0 - 7505    | 16.86GB | (7506/0/0) | 35368272 |
| 3    | unassigned | wm   | 0           | 0       | (0/0/0)    | 0        |

|   |            |    |             |        |            |          |
|---|------------|----|-------------|--------|------------|----------|
| 4 | unassigned | wm | 0           | 0      | (0/0/0)    | 0        |
| 5 | unassigned | wm | 0           | 0      | (0/0/0)    | 0        |
| 6 | usr        | wm | 3562 - 7505 | 8.86GB | (3944/0/0) | 18584128 |
| 7 | unassigned | wm | 0           | 0      | (0/0/0)    | 0        |

Okay to make this the current partition table[yes]? **yes**

Enter table name (remember quotes): **"disk0"**

Ready to label disk, continue? **yes**

partition> **quit**

format> **verify**

format> **quit**

**示例 13-2 SPARC: 创建磁盘片和标记辅助磁盘**

以下示例说明如何使用 `format` 实用程序将一个 18 GB 的磁盘分成用于 `/export/home` 文件系统的一个片。

**# format /dev/rdisk/c1\***

AVAILABLE DISK SELECTIONS:

- 0. /dev/rdisk/c1t0d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>  
/sbus@2,0/QLGC,isp@2,10000/sd@0,0
- 1. /dev/rdisk/c1t1d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>  
/sbus@2,0/QLGC,isp@2,10000/sd@1,0
- 2. /dev/rdisk/c1t8d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>  
/sbus@2,0/QLGC,isp@2,10000/sd@8,0
- 3. /dev/rdisk/c1t9d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>  
/sbus@2,0/QLGC,isp@2,10000/sd@9,0

Specify disk (enter its number): 1

selecting cltld0

[disk formatted]

format> partition

partition> print

partition> modify

Select partitioning base:

0. Current partition table (original)

1. All Free Hog

Choose base (enter number) [0]? 1

| Part | Tag        | Flag | Cylinders | Size    | Blocks     |          |
|------|------------|------|-----------|---------|------------|----------|
| 0    | root       | wm   | 0         | 0       | (0/0/0)    | 0        |
| 1    | swap       | wu   | 0         | 0       | (0/0/0)    | 0        |
| 2    | backup     | wu   | 0 - 7505  | 16.86GB | (7506/0/0) | 35368272 |
| 3    | unassigned | wm   | 0         | 0       | (0/0/0)    | 0        |
| 4    | unassigned | wm   | 0         | 0       | (0/0/0)    | 0        |
| 5    | unassigned | wm   | 0         | 0       | (0/0/0)    | 0        |
| 6    | usr        | wm   | 0         | 0       | (0/0/0)    | 0        |
| 7    | unassigned | wm   | 0         | 0       | (0/0/0)    | 0        |

Do you wish to continue creating a new partition

table based on above table[yes]? y

Free Hog partition[6]? 7

```
Enter size of partition '0' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '1' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '3' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '4' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '5' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '6' [0b, 0c, 0.00mb, 0.00gb]:
```

| Part | Tag        | Flag | Cylinders | Size    | Blocks              |
|------|------------|------|-----------|---------|---------------------|
| 0    | root       | wm   | 0         | 0       | (0/0/0) 0           |
| 1    | swap       | wu   | 0         | 0       | (0/0/0) 0           |
| 2    | backup     | wu   | 0 - 7505  | 16.86GB | (7506/0/0) 35368272 |
| 3    | unassigned | wm   | 0         | 0       | (0/0/0) 0           |
| 4    | unassigned | wm   | 0         | 0       | (0/0/0) 0           |
| 5    | unassigned | wm   | 0         | 0       | (0/0/0) 0           |
| 6    | usr        | wm   | 0         | 0       | (0/0/0) 0           |
| 7    | unassigned | wm   | 0 - 7505  | 16.86GB | (7506/0/0) 35368272 |

```
Okay to make this the current partition table[yes]? yes
```

```
Enter table name (remember quotes): "home"
```

```
Ready to label disk, continue? y
```

```
partition> q
```

```
format> verify
```

```
format> q
```

```
#
```

以下示例说明如何使用 `format` 实用程序，将标号为 EFI 的 1.15 TB 磁盘分成三个片。



```
format

.

.

.

partition> modify

Select partitioning base:

 0. Current partition table (original)

 1. All Free Hog

Choose base (enter number) [0]? 1

Part Tag Flag First Sector Size Last Sector

 0 root wm 0 0 0

 1 usr wm 0 0 0

 2 unassigned wm 0 0 0

 3 unassigned wm 0 0 0

 4 unassigned wm 0 0 0

 5 unassigned wm 0 0 0

 6 usr wm 0 0 0

 8 reserved wm 2576924638 8.00MB 2576941021

Do you wish to continue creating a new partition
table based on above table[yes]? y

Free Hog partition[6]? 4

Enter size of partition 0 [0b, 34e, 0mb, 0gb, 0tb]:

Enter size of partition 1 [0b, 34e, 0mb, 0gb, 0tb]:

Enter size of partition 2 [0b, 34e, 0mb, 0gb, 0tb]: 400gb
```

```
Enter size of partition 3 [0b, 838860834e, 0mb, 0gb, 0tb]: 400gb

Enter size of partition 5 [0b, 1677721634e, 0mb, 0gb, 0tb]:

Enter size of partition 6 [0b, 1677721634e, 0mb, 0gb, 0tb]:

Part Tag Flag First Sector Size Last Sector

0 unassigned wm 0 0 0
1 unassigned wm 0 0 0
2 usr wm 34 400.00GB 838860833
3 usr wm 838860834 400.00GB 1677721633
4 usr wm 1677721634 428.77GB 2576924637
5 unassigned wm 0 0 0
6 unassigned wm 0 0 0
8 reserved wm 2576924638 8.00MB 2576941021

Ready to label disk, continue? yes

partition> q
```

更多信息 **创建磁盘片和标记磁盘之后**

创建磁盘片并标记磁盘后，可以在磁盘上创建文件系统。请转至第 258 页中的“SPARC: 如何创建 UFS 文件系统”。

▼ **SPARC: 如何创建 UFS 文件系统**

- 1 成为超级用户或承担等效角色。
- 2 为每个片创建文件系统。

```
newfs /dev/rdisk/cwtxdysz
```

其中 /dev/rdisk/cwtxdysx 是要创建的文件系统的原始设备。

有关 `newfs` 命令的更多信息，请参见第 18 章或 `newfs(1M)`。

### 3 通过挂载新文件系统对其进行验证。

```
mount /dev/dsk/cwtxdysz /mnt
```

```
ls
```

```
lost+found
```

## 更多信息 创建 UFS 文件系统之后

- **系统磁盘**—需要在磁盘上恢复根 (/) 文件和 `/usr` 文件系统。
  - 请转至第 27 章。
  - 恢复根 (/) 文件和 `/usr` 文件系统后，安装引导块。请转至第 259 页中的“SPARC: 如何在系统磁盘上安装引导块”。
- **辅助磁盘**—可能需要在新的磁盘上恢复文件系统。请转至第 27 章。如果不在新磁盘上恢复文件系统，则完成了辅助磁盘的添加。
- 有关使文件系统可被用户使用的信息，请参见第 19 章。

## ▼ SPARC: 如何在系统磁盘上安装引导块

### 1 成为超级用户或承担等效角色。

### 2 在系统磁盘上安装引导块。

```
installboot /usr/platform/'uname -i'/lib/fs/ufs/bootblk
```

```
/dev/rdsk/cwtxdys0
```

```
/usr/platform/'uname -i'/lib/fs /ufs/bootblk
```

是引导块代码。

```
/dev/rdsk/cwtxdy s0
```

是根 (/) 文件系统的原始设备。

有关更多信息，请参见 `installboot(1M)`。

### 3 通过将系统重新引导到运行级 3，验证是否已安装引导块。

```
init 6
```

## 示例 13-3 SPARC: 在系统磁盘上安装引导块

以下示例说明如何在 Ultra™ 10 系统中安装引导块。

```
installboot /usr/platform/sun4u/lib/fs/ufs/bootblk
/dev/rdisk/c0t0d0s0
```

## x86：添加磁盘（任务）

---

本章介绍如何将磁盘添加到基于 x86 的系统。

有关与将磁盘添加到基于 x86 的系统关联的过程的信息，请参见第 261 页中的“[x86: 添加系统磁盘或辅助磁盘（任务列表）](#)”。

有关磁盘管理的概述信息，请参见第 11 章。有关将磁盘添加到基于 SPARC 的系统的逐步说明，请参见第 13 章。

### x86: 添加系统磁盘或辅助磁盘（任务列表）

以下任务列表提供了将磁盘添加到基于 x86 的系统的过程。

| 任务                     | 说明                                                                                                   | 参考                                                                                                |
|------------------------|------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|
| 1. 连接磁盘并进行引导。          | <b>系统磁盘</b><br>连接新磁盘并从本地或远程 Solaris CD 或 DVD 进行引导。<br><br><b>辅助磁盘</b><br>连接新磁盘并执行重新配置引导，以便使系统识别该磁盘。  | 第 262 页中的“ <a href="#">x86: 如何连接系统磁盘</a> ”<br><br>第 265 页中的“ <a href="#">x86: 如何连接辅助磁盘并进行引导</a> ” |
| 2. （可选）更改 fdisk 分区标识符。 | x86 系统上的 Solaris 10 fdisk 分区标识符已从 130 (0x82) 更改为 191 (0xbf)。<br><br>可以使用新的 fdisk 菜单选项在新标识符和旧标识符之间切换。 | 第 263 页中的“ <a href="#">如何更改 Solaris fdisk 标识符</a> ”                                               |

| 任务           | 说明                                                                                      | 参考                                                                    |
|--------------|-----------------------------------------------------------------------------------------|-----------------------------------------------------------------------|
| 3. 创建片并标记磁盘。 | 如果磁盘制造商尚未创建磁盘片和标记磁盘，请创建磁盘片并标记磁盘。                                                        | 第 267 页中的 “x86: 如何创建 Solaris fdisk 分区” 和第 276 页中的 “x86: 如何创建磁盘片和标记磁盘” |
| 4. 创建文件系统。   | 使用 <code>newfs</code> 命令在磁盘片上创建 UFS 文件系统。必须为系统磁盘创建根 (/) 文件系统和/或 <code>/usr</code> 文件系统。 | 第 277 页中的 “x86: 如何创建文件系统”                                             |
| 5. 恢复文件系统。   | 在系统磁盘上恢复根 (/) 文件系统和/或 <code>/usr</code> 文件系统。如有必要，请在辅助磁盘上恢复文件系统。                        | 第 27 章                                                                |
| 6. 安装引导块。    | 仅限系统磁盘。在根 (/) 文件系统中安装引导块，以便该系统可以引导。                                                     | 第 278 页中的 “x86: 如何在系统磁盘上安装引导块”                                        |

## x86: 添加系统磁盘或辅助磁盘

系统磁盘包含根 (/) 文件系统和/或 `/usr` 文件系统。如果包含其中任一文件系统的磁盘损坏，则可以使用以下两种方法恢复文件系统：

- 可以重新安装整个 Solaris OS。
- 或者，可以更换系统磁盘，然后从备份介质恢复文件系统。

辅助磁盘不包含根 (/) 文件系统和 `/usr` 文件系统。辅助磁盘通常包含用于存储用户文件的空間。可以将辅助磁盘添加到系统以获得更多磁盘空间。或者，可以更换损坏的辅助磁盘。如果更换系统中的辅助磁盘，则可以在新磁盘上恢复旧磁盘的数据。

### ▼ x86: 如何连接系统磁盘

此过程假定已关闭操作系统。

- 1 断开损坏的系统磁盘与系统之间的连接。
- 2 确保所添加的磁盘与系统中的其他设备具有不同的目标号码。  
通常，磁盘背面有一个用于此目的的小开关。
- 3 将用来替换的系统磁盘连接到系统，并检查物理连接。  
有关详细信息，请参阅磁盘的硬件安装指南。

#### 4 引导系统。

此过程假定将从 GRUB 的 Solaris 故障安全引导选项进行引导。

- a. 如果系统显示 `Press any key to reboot` 提示，请按任意键重新引导系统。或者，使用复位按钮重新启动系统（如果已关闭系统）。  
几分钟后将显示 GRUB 菜单。
- b. 使用方向键选择 Solaris 故障安全引导选项。
- c. 按回车键。
- d. 出现提示 `Do you wish to automatically update boot archives?` 时，应答 `no`。  
此时将显示超级用户提示符 (`#`)。

---

注 - 要退出 Solaris 故障安全引导模式时，必须重新引导系统。在成功添加系统磁盘、恢复数据并安装引导块后，才能将系统重新引导到多用户模式。

---

#### 更多信息 连接系统磁盘之后

如果磁盘小于 1 TB，则可以创建 `fdisk` 分区。请转至第 267 页中的“[x86: 如何创建 Solaris `fdisk` 分区](#)”。

### ▼ 如何更改 Solaris `fdisk` 标识符

通过新的 `fdisk` 菜单选项，可以在新标识符和旧标识符之间来回切换。即使挂载了分区中包含的文件系统，也可以更改 `fdisk` 标识符。

`fdisk` 菜单中的两个 `type` 值反映旧标识符和新标识符，如下所示：

- Solaris 标识 `0x82`
- Solaris2 标识 `0xbf`

- 1 成为超级用户。
- 2 显示当前的 `fdisk` 标识符。

例如：

```
Total disk size is 39890 cylinders
```

```
Cylinder size is 4032 (512 byte) blocks
```

Cylinders

| Partition | Status | Type     | Start | End   | Length | %     |
|-----------|--------|----------|-------|-------|--------|-------|
| =====     | =====  | =====    | ===== | ===== | =====  | ===== |
| 1         | Active | x86 Boot | 1     | 6     | 6      | 0     |
| 2         |        | Solaris2 | 7     | 39889 | 39883  | 100   |

3 从 fdisk 菜单中选择选项 4，将 fdisk 分区标识符更改回 0x82

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: 4

4 选择选项 5 更新磁盘配置并退出。

5 如有必要，请从 fdisk 菜单中选择选项 4，将 fdisk 分区标识符更改回 0xbf。

例如：

Total disk size is 39890 cylinders

Cylinder size is 4032 (512 byte) blocks

| Cylinders |        |          |       |       |        |       |
|-----------|--------|----------|-------|-------|--------|-------|
| Partition | Status | Type     | Start | End   | Length | %     |
| =====     | =====  | =====    | ===== | ===== | =====  | ===== |
| 1         | Active | x86 Boot | 1     | 6     | 6      | 0     |
| 2         |        | Solaris  | 7     | 39889 | 39883  | 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: 4

- 6 选择选项 5 更新磁盘配置并退出。

## ▼ x86: 如何连接辅助磁盘并进行引导

有关在 x64 系统上添加磁盘标号为 EFI 的磁盘的信息，请参见第 202 页中的“带有 EFI 磁盘标号的多 TB 磁盘支持”以了解更多信息。

通常，大多数新式总线类型支持热插拔。如果系统的总线类型支持热插拔，则可能无需执行下面的步骤 2 或 3。

有关热插拔设备的更多信息，请参见第 6 章。

- 1 成为超级用户或承担等效角色。
- 2 （可选的）如果磁盘不受 Solaris 软件的支持，请按照硬件附带的说明，添加磁盘的设备驱动程序。
- 3 （可选的）创建将在引导系统时读取的 `/reconfigure` 文件。

```
touch /reconfigure
```

稍后打开系统电源或引导系统时，`/reconfigure` 文件会使 SunOS™ 软件检查是否存在任何新安装的外围设备。

- 4 关闭系统。

```
shutdown -i0 -gn -y
```

-i0 将系统降至运行级 0（即电源关闭状态）。

-gn 通知已登录的用户，*n* 秒之后系统将开始关闭。

-y 指定命令应该在没有用户干预的情况下运行。

将显示提示 `Press any key to reboot`。

- 5 关闭系统和所有外部外围设备的电源。
- 6 确保所添加的磁盘与系统中的其他设备具有不同的目标号码。  
通常，磁盘背面有一个用于此目的的小开关。
- 7 将磁盘连接到系统并检查物理连接。  
有关详细信息，请参阅磁盘的硬件安装指南。
- 8 打开所有外部外围设备的电源。
- 9 打开系统的电源。  
系统将引导并显示登录提示。

#### 更多信息 连接辅助磁盘并进行引导之后

引导系统后，如果磁盘小于 1 TB，则可以创建 `fdisk` 分区。请转至第 267 页中的“[x86: 如何创建 Solaris fdisk 分区](#)”。

## x86: fdisk 分区创建准则

设置一个或多个 `fdisk` 分区时，请遵循以下准则。

- `fdisk` 命令不能在大于 1 TB、标号为 EFI 的磁盘上使用。
- 磁盘最多可以分成四个 `fdisk` 分区。其中一个分区必须是 Solaris 分区。
- Solaris 分区必须是磁盘上的活动分区。系统启动时，缺省引导的操作系统所在的分区即为活动分区。
- Solaris `fdisk` 分区必须从柱面边界开始。
- Solaris `fdisk` 分区必须从第一个磁盘上的柱面 1 而不是柱面 0 开始，因为其他引导信息（包括主引导记录）写在扇区 0 中。
- Solaris `fdisk` 分区可以是整个磁盘。或者，可能希望使其变小以便为 DOS 分区留出空间。也可以在不干扰现有分区（如果有充足的可用空间）的情况下，在磁盘上设置新的 `fdisk` 分区以创建新分区。

仅适用于 x86 – Solaris 片也称为分区。某些界面可能将片称为分区。

只有基于 x86 的系统支持 fdisk 分区。为避免混淆，Solaris 文档尝试区分 fdisk 分区和 Solaris fdisk 分区中的实体。这些实体可能称为片或分区。

## ▼ x86: 如何创建 Solaris fdisk 分区

开始之前 如果需要有关 fdisk 分区的信息，请参见第 266 页中的 “x86: fdisk 分区创建准则”。

1 成为超级用户或承担等效角色。

2 调用 format 实用程序。

# format

将显示可用磁盘的编号列表。

有关更多信息，请参见 format(1M)。

3 键入要创建 Solaris fdisk 分区的磁盘的编号。

Specify disk (enter its number): *disk-number*

其中 *disk-number* 是要创建 Solaris fdisk 分区的磁盘的编号。

4 选择 fdisk 菜单。

format> fdisk

所显示的 fdisk 菜单取决于磁盘上是否已存在 fdisk 分区。使用下表确定下一步操作。

| 任务                                                   | 请转至  | 更多信息                    |
|------------------------------------------------------|------|-------------------------|
| 创建一个跨整个磁盘的 Solaris fdisk 分区。                         | 步骤 5 | <a href="#">示例 14-1</a> |
| 创建一个 Solaris fdisk 分区，并保留一个或多个现有的非 Solaris fdisk 分区。 | 步骤 6 | <a href="#">示例 14-2</a> |
| 创建一个 Solaris fdisk 分区以及一个或多个其他非 Solaris fdisk 分区。    | 步骤 6 | <a href="#">示例 14-3</a> |

5 通过在提示符下指定 y，创建并激活跨整个磁盘的 Solaris fdisk 分区。然后，转至步骤 13。

No fdisk table exists. The default partition for the disk is:

a 100% "SOLARIS System" partition

Type "y" to accept the default partition, otherwise type "n" to edit the partition table.

y

- 6 如果不希望 Solaris fdisk 分区占满整个磁盘，请在提示符下指定 n。
- Type "y" to accept the default partition, otherwise type "n" to edit the partition table.

n

Total disk size is 3498 cylinders

Cylinder size is 1199 (512 byte) blocks

| Cylinders |        |       |       |       |        |       |
|-----------|--------|-------|-------|-------|--------|-------|
| Partition | Status | Type  | Start | End   | Length | %     |
| =====     | =====  | ===== | ===== | ===== | =====  | ===== |

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:

- 7 选择选项 1 (Create a partition) 以创建 fdisk 分区。
- Enter Selection: 1

**8 通过选择 1(=Solaris2) 创建 Solaris fdisk 分区。**

Indicate the type of partition you want to create

```

1=SOLARIS2 2=UNIX 3=PCIXOS 4=Other

5=DOS12 6=DOS16 7=DOSEXT 8=DOSBIG

9=DOS16LBA A=x86 Boot B=Diagnostic C=FAT32

D=FAT32LBA E=DOSEXTLBA F=EFI 0=Exit? 1

```

**9 确定要为 Solaris fdisk 分区保留的磁盘百分比。计算此百分比时，请牢记任何现有 fdisk 分区的大小。**

Specify the percentage of disk to use for this partition

(or type "c" to specify the size in cylinders). *nn*

**10 通过在提示符下键入 y，激活 Solaris fdisk 分区。**

Should this to become the active partition? If yes, it will be

activated each time the computer is reset or turned on.

Please type "y" or "n". **y**

激活 fdisk 分区后，将显示提示 Enter Selection。

**11 选择选项 1 Create a partition 以创建另一个 fdisk 分区。**

有关创建 fdisk 分区的说明，请参见步骤 8-10。

**12 更新磁盘配置，然后从 selection 菜单退出 fdisk 菜单。**

Selection: **5**

**13 使用 label 命令重新标记磁盘。**

```
format> label
```

Ready to label disk, continue? **yes**

```
format>
```

**14 退出 format 实用程序。**

```
format> quit
```

**示例 14-1 x86: 创建一个占满整个驱动器的 Solaris fdisk 分区**

以下示例使用 `format` 实用程序的 `fdisk` 选项创建一个占满整个驱动器的 Solaris `fdisk` 分区。

```
format
```

```
Searching for disks...done
```

```
AVAILABLE DISK SELECTIONS:
```

```
0. c0d0 <DEFAULT cyl 2466 alt 2 hd 16 sec 63>
```

```
 /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0
```

```
1. c0d1 <DEFAULT cyl 522 alt 2 hd 32 sec 63>
```

```
 /pci@0,0/pci-ide@7,1/ide@0/cmdk@1,0
```

```
2. c1d0 <DEFAULT cyl 13102 alt 2 hd 16 sec 63>
```

```
 /pci@0,0/pci-ide@7,1/ide@1/cmdk@0,0
```

```
Specify disk (enter its number): 0
```

```
selecting c0d0
```

```
Controller working list found
```

```
[disk formatted]
```

```
format> fdisk
```

```
No fdisk table exists. The default partitioning for your disk is:
```

```
 a 100% "SOLARIS System" partition.
```

```
Type "y" to accept the default partition, otherwise type "n" to edit the
partition table. y
```

```
format> label
```

```
Ready to label disk, continue? yes

format> quit
```

示例 14-2 x86: 创建 Solaris fdisk 分区并保留现有的 fdisk 分区

以下示例说明如何在包含现有 DOS-BIG fdisk 分区的磁盘上创建 Solaris fdisk 分区。

```
format> fdisk

Total disk size is 3498 cylinders

Cylinder size is 1199 (512 byte) blocks

Cylinders
Partition Status Type Start End Length %
===== ===== =====
1 Active DOS-BIG 1 699 699 20

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: 1

Indicate the type of partition you want to create

1=SOLARIS2 2=UNIX 3=PCIXOS 4=Other
5=DOS12 6=DOS16 7=DOSEXT 8=DOSBIG
9=DOS16LBA A=x86 Boot B=Diagnostic C=FAT32
```

```
D=FAT32LBA E=DOSEXTLBA F=EFI 0=Exit?1

Indicate the percentage of the disk you want this partition
to use (or enter "c" to specify in cylinders). 80

Should this become the active partition? If yes, it will be
activated each time the computer is or turned on.

Please type "y" or "n". y

Total disk size is 3498 cylinders

Cylinder size is 1199 (512 byte) blocks

Cylinders

Partition Status Type Start End Length %
===== ===== =====
1 DOS-BIG 1 699 699 20
2 Active Solaris2 700 3497 2798 80

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

Partition 2 is now the active partition

format> label
```



```
Ready to label disk, continue? yes

format> q
```

**示例 14-3    x86: 创建 Solaris fdisk 分区和另一个 fdisk 分区**

以下示例说明如何创建 Solaris fdisk 分区和 DOSBIG fdisk 分区。

```
format> fdisk

No fdisk table exists. The default partitioning for your disk is:

 a 100% "SOLARIS System" partition.

Type "y" to accept the default partition, otherwise type "n" to edit the
partition table.

n
```

Total disk size is 3498 cylinders

Cylinder size is 1199 (512 byte) blocks

Cylinders

| Partition | Status | Type  | Start | End | Length | %   |
|-----------|--------|-------|-------|-----|--------|-----|
| =====     | =====  | ===== | ===== | === | =====  | === |

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: **1**

Indicate the type of partition you want to create

1=SOLARIS2    2=UNIX            3=PCIXOS       4=Other  
5=DOS12       6=DOS16       7=DSEXT       8=DOSBIG  
9=DOS16LBA    A=x86 Boot      B=Diagnostic C=FAT32  
D=FAT32LBA    E=DSEXTLBA    F=EFI           0=Exit?    8

Specify the percentage of disk to use for this partition  
(or type "c" to specify the size in cylinders)20  
Should this to become the Active partition? If yes, it will be  
activated each time the computer is reset or turned on.  
again. Please type "y" or "n". n

Total disk size is 3498 cylinders  
Cylinder size is 1199 (512 byte) blocks

| Cylinders |        |         |       |       |        |       |
|-----------|--------|---------|-------|-------|--------|-------|
| Partition | Status | Type    | Start | End   | Length | %     |
| =====     | =====  | =====   | ===== | ===== | =====  | ===== |
| 1         |        | DOS-BIG | 1     | 699   | 699    | 20    |

- 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: 1  
Indicate the type of partition you want to create

1=SOLARIS2    2=UNIX            3=PCIXOS        4=Other  
5=DOS12        6=DOS16           7=DOSEXT        8=DOSBIG  
9=DOS16LBA    A=x86 Boot        B=Diagnostic C=FAT32  
D=FAT32LBA    E=DOSEXTLBA      F=EFI            0=Exit? **1**

Indicate the percentage of the disk you want this partition  
to use (or enter "c" to specify in cylinders). **80**  
  
Should this become the active partition? If yes, it will be  
activated each time the computer is reset or turned on.  
  
Please type "y" or "n". **y**

Total disk size is 3498 cylinders

Cylinder size is 1199 (512 byte) blocks

| Cylinders |        |          |       |       |        |       |
|-----------|--------|----------|-------|-------|--------|-------|
| Partition | Status | Type     | Start | End   | Length | %     |
| =====     | =====  | =====    | ===== | ===== | =====  | ===== |
| 1         |        | DOS-BIG  | 1     | 699   | 699    | 20    |
| 2         | Active | Solaris2 | 700   | 3497  | 2798   | 80    |

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**

```
Partition 2 is now the Active partition
```

```
format> q
```

## 更多信息 创建 Solaris fdisk 分区之后

在磁盘上创建 Solaris fdisk 分区后，可以在该磁盘上创建片。请转至第 276 页中的[“x86: 如何创建磁盘片和标记磁盘”](#)

## ▼ x86: 如何创建磁盘片和标记磁盘

- 1 成为超级用户或承担等效角色。

- 2 调用 format 实用程序。

```
format
```

将显示可用磁盘的编号列表。

- 3 键入要重新分区的磁盘的编号。

```
Specify disk (enter its number): disk-number
```

其中 *disk-number* 是要重新分区的磁盘的编号。

- 4 选择 partition 菜单。

```
format> partition
```

- 5 显示当前分区（片）表。

```
partition> print
```

- 6 开始修改过程。

```
partition> modify
```

- 7 将磁盘全部设置为浮动片 (free hog slice)。

```
Choose base (enter number) [0]? 1
```

有关浮动片 (free hog slice) 的更多信息，请参见第 217 页中的[“使用浮动片 \(free hog slice\)”](#)。

- 8 在提示继续时应答 yes 创建新分区表。

```
Do you wish to continue creating a new partition
```

```
table based on above table[yes]? yes
```

## 9 出现提示时，标识浮动 (free hog) 分区 (片) 和片大小。

添加系统磁盘时，必须为以下项设置片：

- 根文件系统 (片 0) 和交换区域 (片 1) 和/或
- /usr (片 6)

标识片后，将显示新的分区表。

## 10 通过在出现提示时应答 yes，使显示的分区表成为当前分区表。

Okay to make this the current partition table[yes]? **yes**

如果不需要当前分区表但希望对其进行更改，请应答 no 并转至[步骤 6](#)。

## 11 命名分区表。

Enter table name (remember quotes): "*partition-name*"

其中 *partition-name* 是新分区表的名称。

## 12 在新磁盘上完成片的分配后，用新分区表标记磁盘。

Ready to label disk, continue? **yes**

## 13 退出 partition 菜单。

partition> **quit**

## 14 检验新的磁盘标号。

format> **verify**

## 15 退出 format 实用程序。

format> **quit**

## 更多信息 创建磁盘片和标记磁盘之后

创建磁盘片并标记磁盘后，可以在磁盘上创建文件系统。请转至第 277 页中的[“x86: 如何创建文件系统”](#)。

# ▼ x86: 如何创建文件系统

## 1 成为超级用户或承担等效角色。

## 2 为每个片创建文件系统。

```
newfs /dev/rdisk/cwtxdysz
```

其中 /dev/rdisk/cwtxdysz 是要创建的文件系统的原始设备。

有关 newfs 命令的更多信息，请参见[第 18 章](#)或 newfs(1M)。

### 3 通过挂载新文件系统对其进行检验。

```
mount /dev/dsk/cwtxdysz /mnt

ls /mnt

lost+found
```

## 更多信息 创建文件系统之后

- **系统磁盘**—需要在磁盘上恢复根 (/) 文件和 /usr 文件系统。
  - 请转至第 27 章。
  - 恢复根 (/) 文件和 /usr 文件系统后，安装引导块。请转至第 278 页中的“x86: 如何在系统磁盘上安装引导块”。
- **辅助磁盘**—可能需要在新的磁盘上恢复文件系统。请转至第 27 章。如果不在新磁盘上恢复文件系统，则完成了辅助磁盘的添加。
- 有关使文件系统可被用户使用的信息，请参见第 19 章。

## ▼ x86: 如何在系统磁盘上安装引导块

### 1 成为超级用户或承担等效角色。

### 2 在系统磁盘上安装引导块。

```
/sbin/installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdisk/cwtxdysz
```

/boot/grub/stage1 是分区引导文件。

/boot/grub/stage2 是引导块代码。

/dev/rdisk/cwtxdysz 是原始设备名称，该名称表示 GRUB 菜单 /boot/grub/menu.lst 在 Solaris 根片上的位置。

有关更多信息，请参见 installgrub(1M)。

### 3 通过将系统重新引导到运行级 3，验证是否已安装引导块。

```
init 6
```

## 示例 14-4 x86: 在系统磁盘上安装引导块

以下示例说明如何在 x86 系统上安装引导块。

```
/sbin/installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdsk/c1d0s0

stage1 written to partition 0 sector 0 (abs 2016)
```

stage2 written to to partition 0, 227 sectors starting 50 (abs 2066)





## 配置 Solaris iSCSI 启动器（任务）

---

本章介绍如何从 Solaris 10 1/06 发行版开始配置 Solaris iSCSI 启动器。有关与配置 iSCSI 启动器关联的过程信息，请参见第 283 页中的“设置 Solaris iSCSI 启动器（任务列表）”。

有关 Solaris 10 6/06 发行版中新增的 Solaris iSCSI 启动器功能的信息，请参见第 199 页中的“Solaris 10 6/06 发行版在磁盘管理方面的新增功能”。

有关 Solaris iSCSI 配置问题的疑难解答，请参见第 301 页中的“iSCSI 配置问题的疑难解答”。

## iSCSI 技术（概述）

iSCSI 是 Internet SCSI（Small Computer System Interface，小型计算机系统接口）的缩写，是用于链接数据存储子系统的基于 Internet 协议 (Internet Protocol, IP) 的存储网络标准。此网络标准是由 Internet 工程任务组 (Internet Engineering Task Force, IETF) 开发的。有关 iSCSI 技术的更多信息，请参见 RFC 3720：

<http://www.ietf.org/rfc/rfc3720.txt>

通过在 IP 网络上传输 SCSI 命令，iSCSI 协议可用于访问网络中的块设备，就像这些设备连接至本地系统一样。

如果要使用现有 TCP/IP 网络中的存储设备，则可以使用以下解决方案：

- iSCSI 块设备或磁带—将 SCSI 命令和数据从块级别转换到 IP 数据包。在网络中使用 iSCSI 的优点体现在需要在一个系统与目标设备（如磁带设备或数据库）之间具有块级别访问权限时。对块级别设备的访问权限不会被锁定，因此不会有多个用户或系统访问块级别设备，如 iSCSI 目标设备。
- NFS—通过 IP 传输文件数据。在网络中使用 NFS 的优点在于可以在多个系统之间共享文件数据。当很多用户访问 NFS 环境中可用的数据时，对文件数据的访问就会相应地被锁定。

以下是使用 Solaris iSCSI 启动器的益处：

- iSCSI 协议可以跨现有的以太网网络运行。
  - 可以使用任何支持的网络接口卡 (network interface card, NIC)、以太网集线器或交换机。
  - 一个 IP 端口可以处理多个 iSCSI 目标设备。
  - 可以对 IP 网络使用现有的基础结构和管理工具。
- 配置的 iSCSI 目标设备的最大数目没有上限。
- 可以使用该协议连接至光纤通道或包含适当硬件的 iSCSI 存储区域网络 (Storage Area Network, SAN) 环境。

以下是当前使用 Solaris iSCSI 启动器软件的限制：

- 当前不支持使用 SLP 的 iSCSI 设备。
- 当前不支持 iSCSI 设备进行引导。
- 请勿将 iSCSI 目标配置为转储设备。
- iSCSI 支持每个会话建立多个连接，但当前的 Solaris 实现仅支持每个会话建立一个连接。

有关更多信息，请参见 RFC 3720。

- 应考虑在现有网络上传输大量数据的影响。

## iSCSI 软件和硬件要求

- iSCSI 目标软件和设备
- Solaris 10 1/06 或更高发行版
- 以下软件包：
  - SUNWiscsir—Sun iSCSI 设备驱动程序 (root)
  - SUNWiscsiu—Sun iSCSI 管理实用程序 (usr)

---

注—Solaris iSCSI 技术仅包括 iSCSI 启动器软件。

---

- 任何支持的 NIC

# 设置 Solaris iSCSI 启动器（任务列表）

| 任务                              | 说明                                                                                              | 参考                                                                           |
|---------------------------------|-------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------|
| 1. 确定 iSCSI 软件和硬件要求。            | 确定有关设置基于 iSCSI 的存储网络的软件和硬件要求。                                                                   | 第 282 页中的 “iSCSI 软件和硬件要求”                                                    |
| 2. 设置 iSCSI 目标设备。               | 连接并设置 iSCSI 目标设备。                                                                               | 有关设置说明，请参见供应商的 iSCSI 目标设备文档。                                                 |
| 3. 准备 Solaris iSCSI 配置          | 确保已安装正确的软件版本和硬件。                                                                                | 第 285 页中的 “如何准备 Solaris iSCSI 配置”                                            |
| 4. （可选）在 Solaris iSCSI 配置中设置验证。 | 决定是否要在 Solaris iSCSI 配置中使用验证：<br><br>考虑使用单向 CHAP 还是双向 CHAP<br><br>考虑使用第三方 RADIUS 服务器简化 CHAP 管理。 | 第 285 页中的 “如何配置对 iSCSI 配置的 CHAP 验证”<br><br>第 287 页中的 “如何为 iSCSI 配置配置 RADIUS” |
| 5. 配置 iSCSI 目标搜索方法。             | 选择最适合于您所在环境的 iSCSI 目标搜索方法。                                                                      | 第 288 页中的 “如何配置 iSCSI 目标搜索”                                                  |
| 6. （可选）删除已搜索到的 iSCSI 目标。        | 可能需要删除已搜索到的 iSCSI 目标。                                                                           | 第 289 页中的 “如何删除搜索到的 iSCSI 目标”                                                |
| 7. 监视 iSCSI 配置。                 | 使用 <code>iscsiadm</code> 命令监视 iSCSI 配置。                                                         | 第 291 页中的 “监视 iSCSI 配置”                                                      |
| 8. （可选）修改 iSCSI 配置。             | 可能需要更改 iSCSI 目标设置，如头和数据摘要参数。                                                                    | 第 295 页中的 “如何修改 iSCSI 启动器和目标参数”                                              |
| 9. （可选）对目标启用多个 iSCSI 会话         | 使用此过程可以创建与单个目标连接的多个 iSCSI 会话。                                                                   | 第 299 页中的 “如何为目标启用多个 iSCSI 会话”                                               |

## 配置 Solaris iSCSI 启动器

基本上，配置 Solaris iSCSI 启动器包含以下步骤：

- 确定硬件和软件要求
- 配置 IP 网络
- 连接并设置 iSCSI 目标设备
- （可选）在 iSCSI 启动器与 iSCSI 目标之间配置 iSCSI 验证（如有必要）
- 配置 iSCSI 目标搜索方法
- 在 iSCSI 磁盘上创建文件系统
- 监视 iSCSI 配置

iSCSI 配置信息存储在 `/etc/iscsi` 目录中。此信息不需要进行管理。

## iSCSI 术语

配置 iSCSI 启动器之前，请查阅以下术语。

| 术语   | 说明                                                                                                                                                                                                                                                                   |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 启动器  | 用于启动对 iSCSI 目标的 SCSI 请求的驱动程序。                                                                                                                                                                                                                                        |
| 目标设备 | 表示 iSCSI 存储组件。                                                                                                                                                                                                                                                       |
| 搜索   | 搜索是指为启动器提供可用目标列表的过程                                                                                                                                                                                                                                                  |
| 搜索方法 | <p>介绍可找到 iSCSI 目标的方法。当前有三种可用方法：</p> <ul style="list-style-type: none"><li>■ Internet 存储名称服务 (Internet Storage Name Service, iSNS) — 通过与一个或多个 iSNS 服务器进行交互来搜索潜在目标。</li><li>■ SendTargets — 通过使用 <i>discovery-address</i> 来搜索潜在目标。</li><li>■ 静态 — 静态目标地址已配置。</li></ul> |

## 配置动态或静态目标搜索

确定是要配置其中一种动态功能，还是要使用静态 iSCSI 启动器目标来执行设备搜索。

### ■ 动态设备搜索

有两种可用的动态设备搜索方法：SendTargets 和 iSNS。

- SendTargets — 如果 iSCSI 节点显示了大量目标，如 iSCSI 到光纤通道网桥，则可以提供 iSCSI 节点 IP 地址/端口组合，并允许 iSCSI 启动器使用 SendTargets 功能执行设备搜索。
- iSNS — iSNS（Internet Storage Name Service，Internet 存储名称服务）允许 iSCSI 启动器搜索其有权访问的使用尽可能少的配置的目标。另外，该支持还提供状态更改通知功能，在存储节点的操作状态发生更改时通知 iSCSI 启动器。要使用 iSNS 搜索选项，可以提供 iSNS 服务器地址/端口组合，并允许 iSCSI 启动器查询为执行设备搜索而指定的 iSNS 服务器。iSNS 服务器的缺省端口为 3205。有关 iSNS 的更多信息，请参见 RFC 4171：

<http://www.ietf.org/rfc/rfc4171.txt>

- 静态设备搜索 — 如果 iSCSI 节点具有较少目标，或者如果要限制启动器尝试访问的目标，则可以使用以下静态目标地址命名约定以静态方式配置 *target-name*：

*target,target-address[:port-number]*

可以使用数组的管理工具来确定静态目标地址。

注 - 请勿将 iSCSI 目标配置为同时使用静态和动态设备搜索方法进行搜索。使用冗余搜索方法的后果是可能会在与 iSCSI 目标设备进行通信时减缓系统性能。

## ▼ 如何准备 Solaris iSCSI 配置

1 成为超级用户。

2 验证是否已安装 iSCSI 软件包。

```
pkginfo SUNWiscsiu SUNWiscsir
```

```
system SUNWiscsiu Sun iSCSI Device Driver (root)
```

```
system SUNWiscsir Sun iSCSI Management Utilities (usr)
```

3 验证是否运行的是 Solaris 10 1/06 或更高发行版。

4 确认已设置了 TCP/IP 网络。

5 连接第三方 iSCSI 目标设备并确定已对其进行了配置。

例如，确定是否可以使用 telnet 命令访问 iSCSI 目标设备，以便使用端口 3260 连接至 iSCSI 目标设备。如果拒绝连接，请参见第 301 页中的“iSCSI 配置问题的疑难解答”。

有关连接第三方 iSCSI 目标设备的信息，请参见硬件文档。

## 在基于 iSCSI 的存储网络中配置验证

为 iSCSI 设备设置验证是可选操作。

在安全环境下，由于仅有受信任的启动器才能访问目标，因此不需要进行验证。

在安全性较低的环境中，目标不能确定连接请求是否真正来自给定主机。在这种情况下，目标可以使用质询握手身份验证协议 (Challenge-Handshake Authentication Protocol, CHAP) 来验证启动器。

CHAP 验证使用质询和响应的概念，这意味着目标将质询启动器以查验其身份。要使质询/响应方法可行，目标必须知道启动器的密钥，并且必须将启动器设置为响应质询。有关在数组中设置密钥的说明，请参阅数组供应商的文档。

iSCSI 支持单向和双向验证：

- 单向验证使目标可以验证启动器的身份。
- 双向验证通过为启动器提供一种验证目标身份的方法提供了更高级别的安全性。

## ▼ 如何配置对 iSCSI 配置的 CHAP 验证

此过程假定您已登录到本地系统，并且要在此系统中安全地访问已配置的 iSCSI 目标设备。

1 成为超级用户。

## 2 确定要配置单向 CHAP 还是双向 CHAP。

- 目标可以使用单向验证来验证启动器。此方法为缺省方法。只需完成步骤 3–5 即可。
- 双向验证通过为启动器提供一种验证目标的方式提供了更高级别的安全性。请完成步骤 3–9。

## 3 单向 CHAP—设置启动器的密钥。

例如，以下命令将启动一个用于定义 CHAP 密钥的对话框。

```
iscsiadm modify initiator-node --CHAP-secret
```

---

注 -CHAP 密钥长度必须最少为 12 个字符，最多为 16 个字符。

---

## 4 （可选的）单向 CHAP—设置启动器的 CHAP 名称。

缺省情况下，启动器的 CHAP 名称会设置为启动器节点名称。

可以使用以下命令来更改启动器的 CHAP 名称。

```
iscsiadm modify initiator-node --CHAP-name new-CHAP-name
```

## 5 单向 CHAP—设置密钥后在启动器上启用 CHAP 验证。

```
iscsiadm modify initiator-node --authentication CHAP
```

CHAP 要求启动器节点具有用户名和口令。用户名通常由目标用于查找给定用户名的密钥。在 Solaris 环境中，缺省情况下 CHAP 名称会始终设置为启动器节点名称。可以将 CHAP 名称设置为小于 512 字节的任意长度的文本。512 字节长度限制是 Solaris 限制。但是，如果未设置 CHAP 名称，则系统会在初始化时将其设置为启动器节点名称。

## 6 双向 CHAP—在目标上启用双向验证参数。

例如：

```
iscsiadm modify target-param -B enable eui.5000ABCD78945E2B
```

## 7 双向 CHAP—在目标上将验证方法设置为 CHAP。

例如：

```
iscsiadm modify target-param --authentication CHAP eui.5000ABCD78945E2B
```

## 8 双向 CHAP—在目标上设置目标设备密钥。

例如，以下命令将启动一个用于定义 CHAP 密钥的对话框。

```
iscsiadm modify target-param --CHAP-secret eui.5000ABCD78945E2B
```

## 9 双向 CHAP—设置目标的 CHAP 名称。

缺省情况下，目标的 CHAP 名称会设置为目标名称。

可以使用以下命令来更改目标的 CHAP 名称。

```
iscsiadm modify target-param --CHAP-name target-CHAP-name
```

## 使用第三方 Radius 服务器简化 iSCSI 配置中的 CHAP 管理

可以使用第三方 RADIUS 服务器来简化 CHAP 密钥管理。RADIUS 服务器是集中验证服务。尽管仍然必须指定启动器的 CHAP 密钥，但是，在将双向验证与 RADIUS 服务器结合使用时，不再需要在每个启动器上指定每个目标的 CHAP 密钥。

有关更多信息，请参见：

- CHAP—<http://www.ietf.org/rfc/rfc1994.txt>
- RADIUS—<http://www.ietf.org/rfc/rfc2865.txt>

### ▼ 如何为 iSCSI 配置配置 RADIUS

- 1 成为超级用户。
- 2 使用 RADIUS 服务器的 IP 地址和端口（缺省端口为 1812）配置启动器节点。  
例如：

```
iscsiadm modify initiator-node --radius-server 10.0.0.72:1812
```

- 3 使用 RADIUS 服务器的共享密钥配置启动器节点。

```
iscsiadm modify initiator-node --radius-shared-secret
```

---

注 - Solaris iSCSI 实现要求在 Solaris iSCSI 软件与 RADIUS 服务器进行交互之前先使用共享密钥配置 RADIUS 服务器。

---

- 4 启用 RADIUS 服务器。

```
iscsiadm modify initiator-node --radius-access enable
```

## Solaris iSCSI 和 RADIUS 服务器错误消息

本节介绍与 Solaris iSCSI 和 RADIUS 服务器配置相关的消息以及可能的恢复解决方案。

empty RADIUS shared secret

**原因:** 已在启动器上启用了 RADIUS 服务器，但未设置 RADIUS 共享密钥。

**解决方法:** 使用 RADIUS 共享密钥配置启动器。有关更多信息，请参见第 287 页中的“如何为 iSCSI 配置配置 RADIUS”。

WARNING: RADIUS packet authentication failed

原因: 启动器无法验证 RADIUS 数据包。如果在启动器节点上配置的共享密钥不同于 RADIUS 服务器上的共享密钥, 则会出现此错误。

解决方法: 使用正确的 RADIUS 共享密钥重新配置启动器。有关更多信息, 请参见第 287 页中的“如何为 iSCSI 配置配置 RADIUS”。

## ▼ 如何配置 iSCSI 目标搜索

此过程假定您已登录到本地系统, 并且要在此系统中配置对 iSCSI 目标设备的访问权限。

### 1 成为超级用户。

### 2 配置要动态或静态搜索的目标设备：

- 配置动态搜索的设备 (SendTargets)。

例如：

```
iscsiadm add discovery-address 10.0.0.1:3260
```

启用搜索方法之前, 不会启动 iSCSI 连接。请参见下一步。

- 配置动态搜索的设备 (iSNS)。

例如：

```
iscsiadm add iSNS-server 10.0.0.1:3205
```

启用搜索方法之前, 不会启动 iSCSI 连接。请参见下一步。

- 配置静态搜索的设备。

例如：

```
iscsiadm add static-config eui.5000ABCD78945E2B,10.0.0.1
```

启用搜索方法之前, 不会启动 iSCSI 连接。请参见下一步。

### 3 使用以下命令之一启用 iSCSI 目标搜索方法：

- 如果已配置动态搜索 (SendTargets) 设备, 请启用 SendTargets 搜索方法。

```
iscsiadm modify discovery --sendtargets enable
```

- 如果已配置动态搜索 (iSNS) 设备, 请启用 iSNS 搜索方法。

```
iscsiadm modify discovery --isns enable
```

- 如果已配置静态目标, 请启用静态目标搜索方法。



```
iscsiadm modify discovery --static enable
```

- 4 为本地系统创建 iSCSI 设备链接。

```
devfsadm -i iscsi
```

## ▼ 如何删除搜索到的 iSCSI 目标

此可选过程假定您已登录到本地系统，在此系统上已配置了对 iSCSI 目标设备的访问权限。

删除搜索地址、iSNS 服务器、静态配置或禁用搜索方法后，关联的目标即被注销。如果这些关联目标仍在（例如，已挂载文件系统），则注销这些设备将失败，设备仍然位于活动目标列表中。

- 1 成为超级用户。
- 2 (可选的) 使用以下操作之一禁用 iSCSI 目标搜索方法：
  - 如果需要禁用 SendTargets 搜索方法，请使用以下命令：

```
iscsiadm modify discovery --sendtargets disable
```

- 如果需要禁用 iSNS 搜索方法，请使用以下命令：

```
iscsiadm modify discovery --iSNS disable
```

- 如果需要禁用静态目标，请使用以下命令：

```
iscsiadm modify discovery --static disable
```

- 3 删除 iSCSI 设备搜索项：

- 删除 iSCSI SendTargets 搜索项。

例如：

```
iscsiadm remove discovery-address 10.0.0.1:3260
```

- 删除 iSCSI iSNS 搜索项。

例如：

```
iscsiadm remove isns-server 10.0.0.1:3205
```

- 删除静态的 iSCSI 启动器项。

例如：

```
iscsiadm remove static-config eui.5000ABCD78945E2B,10.0.0.1
```

---

注 – 如果尝试禁用或删除关联逻辑单元正在使用的搜索信息，则禁用或删除操作将失败，并会显示以下消息：

```
logical unit in use
```

如果出现这种情况，请停止逻辑单元中的所有关联 I/O，取消挂载文件系统等。然后，重复执行禁用或删除操作。

---

## 访问 iSCSI 磁盘

如果要使 iSCSI 驱动器在重新引导系统时可用，请创建文件系统，并向 `/etc/vfstab` 文件中添加项，类似于对 SCSI 设备上的 UFS 文件系统执行相同操作。

Solaris iSCSI 启动器搜索到设备后，即会自动进行登录协商。Solaris iSCSI 驱动程序可确定可用的 LUN 数量并创建设备节点。然后，即可将 SCSI 设备视为与其他任何 iSCSI 设备一样进行处理。

可以使用 `format` 实用程序查看本地系统上的 iSCSI 磁盘。

在以下 `format` 输出中，磁盘 2 和 3 是不受 MPxIO 控制的 iSCSI LUN。磁盘 21 和 22 是受 MPxIO 控制的 iSCSI LUN。

**# format**

AVAILABLE DISK SELECTIONS:

- ```
0. c0t1d0 <SUN72G cyl 14087 alt 2 hd 24 sec 424>

   /pci@8,600000/SUNW,qlc@4/fp@0,0/ssd@w500000e010685cf1,0

1. c0t2d0 <SUN72G cyl 14087 alt 2 hd 24 sec 424>

   /pci@8,600000/SUNW,qlc@4/fp@0,0/ssd@w500000e0106e3ba1,0

2. c3t0d0 <ABCSTORAGE-100E-00-2.2 cyl 20813 alt 2 hd 16 sec 63>

   /iscsi/disk@0000iqn.2001-05.com.abcstorage%3A6-8a0900-477d70401-
   b0fff044352423a2-hostname-020000,0

3. c3t1d0 <ABCSTORAGE-100E-00-2.2 cyl 20813 alt 2 hd 16 sec 63>

   /iscsi/disk@0000iqn.2001-05.com.abcstorage%3A6-8a0900-3fcd70401-
   -085ff04434f423a2-hostname-010000,0
```

.

.

.

```
21. c4t60A98000686F694B2F59775733426B77d0 <ABCSTORAGE-LUN-0.2 cyl
    4606 alt 2 hd 16 sec 256>

    /scsi_vhci/ssd@g60a98000686f694b2f59775733426b77

22. c4t60A98000686F694B2F59775733434C41d0 <ABCSTORAGE-LUN-0.2 cyl
    4606 alt 2 hd 16 sec 256>

    /scsi_vhci/ssd@g60a98000686f694b2f59775733434c41
```

▼ 监视 iSCSI 配置

通过使用 `iscsiadm list` 命令，可以显示有关 iSCSI 启动器和目标设备的信息。

- 1 成为超级用户。
- 2 显示有关 iSCSI 启动器的信息。

例如：

```
# iscsiadm list initiator-node
```

```
Initiator node name: iqn.1986-03.com.sun:01:0003ba4d233b.425c293c
```

```
Initiator node alias: zzr1200
```

```
    Login Parameters (Default/Configured):
```

```
        Header Digest: NONE/-
```

```
        Data Digest: NONE/-
```

```
    Authentication Type: NONE
```

```
    RADIUS Server: NONE
```

```
    RADIUS access: unknown
```

```
    Configured Sessions: 1
```

3 显示有关当前使用的搜索方法的信息。

例如：

```
# iscsiadm list discovery
```

Discovery:

Static: enabled

Send Targets: enabled

iSNS: enabled

示例 15-1 列出 iSCSI 目标信息

以下示例显示如何列出特定 iSCSI 目标的参数设置。

```
# iscsiadm list target-param iqn.1992-08.com.abcstorage:sn.33592219
```

Target: iqn.1992-08.com.abcstorage:sn.33592219

iscsiadm list target-param -v 命令将显示以下信息：

- 目标的验证设置
- 目标登录参数的缺省设置
- 每个登录参数的已配置值

iscsiadm list target-param -v 命令将在 / 指示符前显示缺省参数值，并在 / 指示符后显示已配置的参数值。如果未配置任何参数，则已配置的参数值将显示为连字符 (-)。有关更多信息，请参见以下示例：

```
# iscsiadm list target-param -v eui.50060e8004275511 Target: eui.50060e8004275511
```

Alias: -

Bi-directional Authentication: disabled

Authentication Type: NONE

Login Parameters (Default/Configured):

Data Sequence In Order: yes/-

Data PDU In Order: yes/-

Default Time To Retain: 20/-

Default Time To Wait: 2/-

```

Error Recovery Level: 0/-
First Burst Length: 65536/-
Immediate Data: yes/-
Initial Ready To Transfer (R2T): yes/-
Max Burst Length: 262144/-
Max Outstanding R2T: 1/-
Max Receive Data Segment Length: 65536/-
Max Connections: 1/-
Header Digest: NONE/-
Data Digest: NONE/-

```

```
Configured Sessions: 1
```

以下示例 `iscsiadm list target-param -v` 输出显示了目标与启动器之间协商的参数。

```
# iscsiadm list target -v eui.50060e8004275511
```

```
Target: eui.50060e8004275511
```

```
TPGT: 1
```

```
ISID: 4000002a0000
```

```
Connections: 1
```

```
CID: 0
```

```
IP address (Local): 172.90.101.71:32813
```

```
IP address (Peer): 172.90.101.40:3260
```

```
Discovery Method: Static
```

```
Login Parameters (Negotiated):
```

```
Data Sequence In Order: yes
```

```
Data PDU In Order: yes
```

Default Time To Retain: 0

Default Time To Wait: 3

Error Recovery Level: 0

First Burst Length: 65536

Immediate Data: yes

Initial Ready To Transfer (R2T): yes

Max Burst Length: 262144

Max Outstanding R2T: 1

Max Receive Data Segment Length: 65536

Max Connections: 1

Header Digest: NONE

Data Digest: NONE

修改 iSCSI 启动器和目标参数

可以同时修改 iSCSI 启动器和 iSCSI 目标设备上的参数。但是，只能在 iSCSI 启动器上修改的参数包括以下各项：

- iSCSI 启动器节点名 可以将启动器节点名更改为不同的名称。如果更改了启动器节点名，则系统可能会从启动器的目标列表中删除 iSNS 搜索到的目标，具体取决于更改名称时 iSNS 服务器上的搜索域配置。有关更多信息，请参见第 295 页中的“[如何修改 iSCSI 启动器和目标参数](#)”。
- 头摘要一值可以为 None（缺省值），也可以为 CRC32。
- 数据摘要一值可以为 None（缺省值），也可以为 CRC32。
- 验证和 CHAP 密钥—有关设置验证的更多信息，请参见第 285 页中的“[如何配置对 iSCSI 配置的 CHAP 验证](#)”。
- 已配置的会话—有关配置多个会话的更多信息，请参见第 299 页中的“[如何为目标启用多个 iSCSI 会话](#)”。

iSCSI 驱动程序为 iSCSI 启动器和 iSCSI 目标设备参数提供缺省值。如果修改了 iSCSI 启动器的参数，则修改后的参数将由 iSCSI 目标设备继承，除非已将 iSCSI 目标设备设置为不同的值。



注意 – 请确保目标软件支持要修改的参数。否则，可能无法登录到 iSCSI 目标设备。有关支持的参数的列表，请参见数组文档。

应在启动器与目标之间的 I/O 完成后再修改 iSCSI 参数。使用 `iscsiadm modify` 命令进行更改后，iSCSI 驱动程序将重新连接会话。

▼ 如何修改 iSCSI 启动器和目标参数

此过程的第一部分说明了 iSCSI 目标设备如何继承 iSCSI 启动器的参数修改。此过程的第二部分说明如何在 iSCSI 目标设备上实际修改参数。

- 1 成为超级用户。
- 2 列出 iSCSI 启动器和目标设备的当前参数。
 - a. 列出 iSCSI 启动器的当前参数。例如：

```
# iscsiadm list initiator-node
```

```
Initiator node name: iqn.1986-03.com.sun:01:0003ba4d233b.425c293c
```

```
Initiator node alias: zzr1200
```

```
Login Parameters (Default/Configured):
```

```
Header Digest: NONE/-
```

```
Data Digest: NONE/-
```

```
Authentication Type: NONE
```

```
RADIUS Server: NONE
```

```
RADIUS access: unknown
```

```
Configured Sessions: 1
```

- b. 列出 iSCSI 目标设备的当前参数。例如：

```
# iscsiadm list target-param -v iqn.1992-08.com.abcstorage:sn.84186266
```

```
Target: iqn.1992-08.com.abcstorage:sn.84186266
```

```
Alias: -
```

```
Bi-directional Authentication: disabled
```

```

Authentication Type: NONE

Login Parameters (Default/Configured):

    Data Sequence In Order: yes/-

    Data PDU In Order: yes/-

    Default Time To Retain: 20/-

    Default Time To Wait: 2/-

    Error Recovery Level: 0/-

    First Burst Length: 65536/-

    Immediate Data: yes/-

    Initial Ready To Transfer (R2T): yes/-

    Max Burst Length: 262144/-

    Max Outstanding R2T: 1/-

    Max Receive Data Segment Length: 65536/-

    Max Connections: 1/-

    Header Digest: NONE/-

    Data Digest: NONE/-

```

```
Configured Sessions: 1
```

请注意，对于 iSCSI 启动器和 iSCSI 目标设备，当前已将头摘要和数据摘要参数设置为 NONE。

要查看 iSCSI 目标设备的缺省参数，请参见[示例 15-1](#) 中的 `iscsiadm list target-param` 输出。

3 修改 iSCSI 启动器的参数。

例如，将头摘要设置为 CRC32。

```
# iscsiadm modify initiator-node -h CRC32
```

如果更改了启动器节点名，则系统可能会注销 iSNS 搜索到的目标，并将其从启动器的目标列表中删除，前提是新名称与目标名称不属于同一搜索域。但是，如果当前正在使用目标并且目标繁忙，则不会将其删除。例如，如果某个文件已打开或已在这些目标上挂载了文件系统。

如果目标和新启动器节点名属于同一搜索域，则在名称更改后还可以看到新目标。

4 确认已修改了参数。

a. 显示 iSCSI 启动器的更新后的参数信息。例如：

```
# iscsiadm list initiator-node
```

```
Initiator node name: iqn.1986-03.com.sun:01:0003ba4d233b.425c293c
```

```
Initiator node alias: zzr1200
```

```
Login Parameters (Default/Configured):
```

```
Header Digest: NONE/CRC32
```

```
Data Digest: NONE/-
```

```
Authentication Type: NONE
```

```
RADIUS Server: NONE
```

```
RADIUS access: unknown
```

```
Configured Sessions: 1
```

请注意，现在已将头摘要设置为 CRC32。

b. 显示 iSCSI 目标设备的更新后的参数信息。例如：

```
# iscsiadm list target-param -v iqn.1992-08.com.abcstorage:sn.84186266
```

```
Target: iqn.1992-08.com.abcstorage:sn.84186266
```

```
Alias: -
```

```
Bi-directional Authentication: disabled
```

```
Authentication Type: NONE
```

```
Login Parameters (Default/Configured):
```

```
Data Sequence In Order: yes/-
```

```
Data PDU In Order: yes/-
```

```
Default Time To Retain: 20/-
```

```
Default Time To Wait: 2/-
```

Error Recovery Level: 0/-
First Burst Length: 65536/-
Immediate Data: yes/-
Initial Ready To Transfer (R2T): yes/-
Max Burst Length: 262144/-
Max Outstanding R2T: 1/-
Max Receive Data Segment Length: 65536/-
Max Connections: 1/-
Header Digest: CRC32/-
Data Digest: NONE/-

Configured Sessions: 1

请注意，现在已将头摘要设置为 CRC32。

5 确认 iSCSI 启动器已重新连接至 iSCSI 目标。例如：

```
# iscsiadm list target -v iqn.1992-08.com.abcstorage:sn.84186266
```

Target: iqn.1992-08.com.abcstorage:sn.84186266

TPGT: 2

ISID: 4000002a0000

Connections: 1

CID: 0

IP address (Local): nnn.nn.nn.nnn:64369

IP address (Peer): nnn.nn.nn.nnn:3260

Discovery Method: SendTargets

Login Parameters (Negotiated):

.

.

```
Header Digest: CRC32
```

```
Data Digest: NONE
```

6 取消设置 iSCSI 启动器参数或 iSCSI 目标设备参数。

通过使用 `iscsiadm modify` 命令将参数设回其缺省设置，可以取消设置参数。或者，可以使用 `iscsiadm remove` 命令将所有目标属性重置为缺省设置。

`iscsiadm modify target-param` 命令仅用于更改在命令行中指定的参数。

以下示例说明如何将头摘要重置为 None：

```
# iscsiadm modify target-param -h none iqn.1992-08.com.abcstorage:sn...
```

有关使用 `iscsiadm remove target-param` 命令的信息，请参见 `iscsiadm.1m`。

▼ 如何为目标启用多个 iSCSI 会话

可以使用此过程创建与单个目标连接的多个 iSCSI 会话。此方案对于支持登录重定向或在同一个目标门户组中具有多个目标门户的 iSCSI 目标设备非常有用。应该将每个目标多个 iSCSI 会话支持与 Solaris SCSI 多路径 (MPxIO) 组合使用。

- 1 成为超级用户。
- 2 列出 iSCSI 启动器和目标的当前参数。

a. 列出 iSCSI 启动器的当前参数。例如：

```
# iscsiadm list initiator-node
```

```
Initiator node name: iqn.1986-03.com.sun:01:0003ba4d233b.425c293c
```

```
Initiator node alias: zzr1200
```

```
Configured Sessions: 1
```

b. 列出 iSCSI 目标设备的当前参数。例如：

```
# iscsiadm list target-param -v iqn.1992-08.com.abcstorage:sn.84186266
```

```
Target: iqn.1992-08.com.abcstorage:sn.84186266
```

```
Alias: -
```

```
.
```

```
.
```

```
.
```

```
Configured Sessions: 1
```

已配置的会话值是为目标门户组中的每个目标名称创建的已配置 iSCSI 会话的数目。

- 3 选择以下命令之一，修改启动器节点处的已配置会话的数目，以便应用于所有目标；或修改某个目标级别的已配置会话的数目，以便应用于特定目标。

目标的会话数目必须介于 1 到 4 之间。

- 将参数应用于 iSCSI 启动器节点。

例如：

```
# iscsiadm modify initiator-node -c 2
```

- 将参数应用于 iSCSI 目标。

例如：

```
# iscsiadm modify target-param -c 2 iqn.1992-08.com.abcstorage:sn.84186266
```

还可以将已配置的会话绑定到特定的本地 IP 地址。使用此方法时，在以逗号分隔的列表中将提供一个或多个本地 IP 地址。每个 IP 地址都表示一个 iSCSI 会话。此方法也可以在启动器节点或目标参数级别使用。例如：

```
# iscsiadm modify initiator-node -c 10.0.0.1,10.0.0.2
```

注 - 如果指定的 IP 地址无法路由，则将忽略该地址，并对此会话使用缺省的 Solaris 路由和 IP 地址。

- 4 确认已修改了参数。
 - a. 显示更新后的启动器节点信息。例如：

```
# iscsiadm list initiator-node
```

```
Initiator node name: iqn.1986-03.com.sun:01:0003ba4d233b.425c293c
```

```
Initiator node alias: zzr1200
```

```

      .
      .
      .

Configured Sessions: 2

```

b. 显示更新后的目标节点信息。例如：

```
# iscsiadm list target-param -v iqn.1992-08.com.abcstorage:sn.84186266
```

```
Target: iqn.1992-08.com.abcstorage:sn.84186266
```

```
Alias: -
```

```

      .
      .
      .

Configured Sessions: 2

```

iSCSI 配置问题的疑难解答

以下工具可用于解决一般的 iSCSI 配置问题：

- snoop — 此工具已更新为支持 iSCSI 数据包。
- ethereal — 可以从 <http://www.ethereal.com> 获取此免费软件产品。

这两个工具都可以筛选端口 3260 上的 iSCSI 数据包。

以下各节介绍了各种 iSCSI 疑难解答和错误消息解析方案。

从本地系统到 iSCSI 目标无连接

▼ 如何解决 iSCSI 连接问题

- 1 成为超级用户。

- 2 列出 iSCSI 目标信息。

例如：

```
# iscsiadm list target
```

```
Target: iqn.2001-05.com.abcstorage:6-8a0900-37ad70401-bcfff02df8a421df-zzr1200-01
```

```
TPGT: default
```

```
ISID: 4000002a0000
```

```
Connections: 0
```

- 3 如果 `iscsiadm list target` 输出中未列出任何连接，请检查 `/var/adm/messages` 文件，以找出连接失败的可能原因。

通过使用 `ping` 命令（或使用 `telnet` 命令与存储设备的 iSCSI 端口连接），还可验证连接是否可访问，从而可确保 iSCSI 服务可用。缺省端口为 3260。

此外，请检查存储设备的日志文件中是否有错误。

- 4 如果目标未列在 `iscsiadm list target` 输出中，请检查 `/var/adm/messages` 文件，以找出可能的原因。

如果当前使用 `SendTargets` 作为搜索方法，请尝试使用 `-v` 选项列出 *discovery-address*，以确保预期目标对于主机是可见的。例如：

```
# iscsiadm list discovery-address -v 10.0.0.1
```

```
Discovery Address: 10.0.0.1:3260
```

```
Target name: eui.210000203787dfc0
```

```
Target address:      10.0.0.1:11824
```

```
Target name: eui.210000203787e07b
```

```
Target address:      10.0.0.1:11824
```

如果当前使用 iSNS 作为搜索方法，请尝试启用 iSNS 搜索模式，并使用 `-v` 选项列出 *isns-server*，以确保预期目标对主机是可见的。例如：

```
# iscsiadm list isns-server -v

iSNS Server IP Address: 10.20.56.56:3205

    Target name: iqn.1992-08.com.xyz:sn.1234566

        Target address:    10.20.57.161:3260, 1

    Target name: iqn.2003-10.com.abc:group-0:154:abc-65-01

        Target address:    10.20.56.206:3260, 1

    Target name: iqn.2003-10.com.abc:group-0:154:abc-65-02

        Target address:    10.20.56.206:3260, 1

.

.

.
```

iSCSI 设备或磁盘在本地系统上不可用

▼ 如何解决 iSCSI 设备或磁盘的不可用性问题

- 1 成为超级用户。
- 2 确定在枚举过程中在该目标上搜索到的 LUN。

例如：

```
# iscsiadm list target -S

Target: iqn.2001-05.com.abcstorage:6-8a0900-37ad70401-bcfff02df8a421df-zzr1200-01

    TPGT: default

    ISID: 4000002a0000

    Connections: 1

    LUN: 0
```

Vendor: ABCSTOR

Product: 0010

OS Device Name: /dev/rdisk/c3t34d0s2

-s 选项可显示在枚举过程中在该目标上搜索到的 LUN。如果您认为某个 LUN 应该已列出，但却没有列出，请查看 /var/adm/messages 文件，以查看是否报告了错误。请检查存储设备的日志文件中是否有错误。另外外，请确保任何存储设备 LUN 屏蔽均已正确配置。

在使用 iSNS 搜索方法时使用 LUN 屏蔽

请避免将 iSNS 搜索域用作控制特定启动器的存储验证的方式。如果要确保只有获得授权的启动器才能访问 LUN，请改用 LUN 屏蔽。

如果在目标正在使用时从搜索域中删除该目标，则 iSCSI 启动器将不会从该目标注销。如果不需要该启动器访问该目标（以及关联的 LUN），则必须使用 LUN 屏蔽。仅从搜索域中删除目标是不够的。

一般 iSCSI 错误消息

本节介绍可能在 /var/adm/messages 文件中找到的 iSCSI 消息以及可能的恢复解决方案。

消息格式如下：

<code>iscsi TYPE (OID) STRING (STATUS-CLASS#/STATUS-DETAIL#)</code>	
<code>TYPE</code>	连接或会话。
<code>OID</code>	连接或会话的对象 ID。此 ID 对于 OS 实例是唯一的。
<code>STRING</code>	条件的说明。
<code><STATUS-CLASS#>/<STATUS-DETAIL#></code>	这些值将在 iSCSI 登录响应中返回，如 RFC 3720 所定义的那样。

`iscsi connection(OID) login failed - Miscellaneous iSCSI initiator errors.`
原因: 设备登录因某种形式的启动器错误而失败。

iscsi connection(*OID*) login failed - Initiator could not be successfully authenticated.

原因:设备不能成功验证启动器。

解决方法:如果适用，请验证 CHAP 名称、CHAP 口令或 RADIUS 的设置是否正确。

iscsi connection(*OID*) login failed - Initiator is not allowed access to the given target.

原因:该设备不允许启动器访问 iSCSI 目标设备。

解决方法:检查启动器名称，确认已正确屏蔽该名称，或该名称是由存储设备提供。

iscsi connection(*OID*) login failed - Requested ITN does not exist at this address.

原因:该设备不提供对正在请求的 iSCSI 目标名称 (iSCSI target name, ITN) 的访问权限。

解决方法:确认正确输入了启动器搜索信息，并且正确配置了存储设备。

iscsi connection(*OID*) login failed - Requested ITN has been removed and no forwarding address is provided.

原因:该设备无法再提供对正在请求的 iSCSI 目标名称 (ITN) 的访问权限。

解决方法:确认正确指定了启动器搜索信息，并且正确配置了存储设备。

iscsi connection(*OID*) login failed - Requested iSCSI version range is not supported by the target.

原因:存储设备不支持启动器的 iSCSI 版本。

iscsi connection(*OID*) login failed - No more connections can be accepted on this Session ID (SSID).

原因:存储设备无法接受此启动器节点到 iSCSI 目标设备的更多连接。

iscsi connection(*OID*) login failed - Missing parameters (e.g., iSCSI initiator and/or target name).

原因:存储设备正在报告未正确指定启动器或目标名称。

解决方法:请正确指定 iSCSI 启动器或目标名称。

iscsi connection(*OID*) login failed - Target hardware or software error.

原因:存储设备遇到硬件或软件错误。

解决方法:请查阅存储文档或与存储设备供应商联系以获得进一步的帮助。

iscsi connection(*OID*) login failed - iSCSI service or target is not currently operational.

原因: 存储设备当前无法运行。

解决方法: 请查阅存储文档或与存储设备供应商联系以获得进一步的帮助。

iscsi connection(*OID*) login failed - Target has insufficient session, connection or other resources.

原因: 存储设备具有的资源不足。

解决方法: 请查阅存储文档或与存储设备供应商联系以获得进一步的帮助。

iscsi connection(*OID*) login failed - unable to initialize authentication

iscsi connection(*OID*) login failed - unable to set authentication

iscsi connection(*OID*) login failed - unable to set username

iscsi connection(*OID*) login failed - unable to set password

iscsi connection(*OID*) login failed - unable to set ipsec

iscsi connection(*OID*) login failed - unable to set remote authentication

原因: 该启动器无法正确初始化或设置验证。

解决方法: 确认已正确配置启动器的验证设置。

iscsi connection(*OID*) login failed - unable to make login pdu

原因: 启动器无法根据启动器或存储设备的设置生成登录有效负荷数据单元 (payload data unit, PDU)。

解决方法: 请尝试重置所有目标登录参数或其他非缺省设置。

iscsi connection(*OID*) login failed - failed to transfer login

iscsi connection(*OID*) login failed - failed to receive login response

原因: 启动器无法通过网络连接传输或接收登录有效负荷数据单元 (payload data unit, PDU)。

解决方法: 检验网络连接是否可以访问。

iscsi connection(*OID*) login failed - received invalid login response (*OP CODE*)

原因: 存储设备采用非预期响应来响应登录。

iscsi connection(*OID*) login failed - login failed to authenticate with target

原因: 启动器无法验证存储设备。

解决方法: 确认已正确配置启动器的验证设置。

iscsi connection(*OID*) login failed - initiator name is required

原因:必须配置启动器名称才能执行所有操作。

解决方法:确认已配置启动器名称。

iscsi connection(*OID*) login failed - authentication receive failed

iscsi connection(*OID*) login failed - authentication transmit failed

原因:该启动器无法传输或接收验证信息。

解决方法:根据情况验证存储设备或 RADIUS 服务器的网络连接。

iscsi connection(*OID*) login failed - login redirection invalid

原因:存储设备尝试将启动器重定向到无效目标。

解决方法:请查阅存储文档或与存储设备供应商联系以获得进一步的帮助。

iscsi connection(*OID*) login failed - target protocol group tag mismatch,
expected <TPGT>, received <TPGT>

原因:启动器和目标的 TPGT (target portal group tag, 目标门户组标记) 不匹配。

解决方法:请验证启动器或存储设备上的 TPGT 搜索设置。

iscsi connection(*OID*) login failed - can't accept *PARAMETER* in security stage

原因:设备已在登录安全阶段使用不支持的登录参数进行了响应。

解决方法:请记录参数名称以供参考。请查阅存储文档或与存储设备供应商联系以获得进一步的帮助。

iscsi connection(*OID*) login failed - HeaderDigest=CRC32 is required, can't
accept *VALUE*

iscsi connection(*OID*) login failed - DataDigest=CRC32 is required, can't accept
VALUE

原因:对于此目标, 该启动器已配置为仅接受将 HeaderDigest 或 DataDigest 设置为 CRC32 的情况。设备返回值 *VALUE*。

解决方法:确认启动器和设备摘要设置兼容。

iscsi connection(*OID*) login failed - HeaderDigest=None is required, can't
accept *VALUE*

iscsi connection(*OID*) login failed - DataDigest=None is required, can't accept
VALUE

原因:对于此目标, 该启动器已配置为只接受将 HeaderDigest 或 DataDigest 设置为 None 的情况。设备返回值 *VALUE*。

解决方法: 确认启动器和设备摘要设置兼容。

`iscsi connection(OID) login failed - can't accept PARAMETER`

原因: 启动器不支持此参数。

`iscsi connection(OID) login failed - can't accept MaxOutstandingR2T VALUE`

原因: 该启动器不接受 MaxOutstandingR2T 值为所记录 *VALUE* 的情况。

`iscsi connection(OID) login failed - can't accept MaxConnections VALUE`

原因: 该启动器不接受最大连接数为所记录 *VALUE* 的情况。

`iscsi connection(OID) login failed - can't accept ErrorRecoveryLevel VALUE`

原因: 该启动器不接受错误恢复级别值为所记录 *VALUE* 的情况。

`iscsi session(OID) NAME offline`

原因: 此目标 *NAME* 的所有连接都已删除或失败。

`iscsi connection(OID) failure - unable to schedule enumeration`

原因: 启动器无法枚举此目标上的 LUN。

解决方法: 可以通过运行 `devfsadm -i iscsi` 命令来强制执行 LUN 枚举。有关更多信息，请参见 `devfsadm(1M)`。

`iscsi connection(OID) unable to connect to target NAME (errno:ERRNO)`

原因: 启动器无法建立网络连接。

解决方法: 有关连接故障的特定 *ERRNO* 的信息，请参见 `/usr/include/sys/errno.h` 文件。

format 实用程序（参考）

本章介绍 `format` 实用程序的菜单和命令。

以下是本章中参考信息的列表。

- 第 309 页中的“有关使用 `format` 实用程序的建议和要求”
- 第 310 页中的“`format` 菜单和命令说明”
- 第 317 页中的“`format.dat` 文件”
- 第 322 页中的“`format` 命令的输入规则”
- 第 325 页中的“获取有关 `format` 实用程序的帮助”

有关何时使用 `format` 实用程序的概述，请参见第 211 页中的“`format` 实用程序”。

有关使用 `format` 实用程序的建议和要求

您必须是超级用户或承担等效角色才能使用 `format` 实用程序。否则，尝试使用 `format` 实用程序时，将显示以下错误消息：

```
$ format
```

```
Searching for disks...done
```

```
No permission (or no disks found)!
```

使用 `format` 实用程序并需要保留现有数据时，请牢记以下原则：

- 备份磁盘驱动器上的所有文件。
- 使用 `format` 实用程序的 `dump` 命令将所有的缺陷列表都保存到文件中。文件名应包括驱动器类型、型号和序列号。
- 保存驱动器附带的制造商缺陷列表的书面副本。

format 菜单和命令说明

format 主菜单的显示与以下内容类似：

FORMAT MENU:

disk	- select a disk
type	- select (define) a disk type
partition	- select (define) a partition table
current	- describe the current disk
format	- format and analyze the disk
fdisk	- run the fdisk program (x86 only)
repair	- repair a defective sector
label	- write label to the disk
analyze	- surface analysis
defect	- defect list management
backup	- search for backup labels
verify	- read and display labels
save	- save new disk/partition definitions
inquiry	- show vendor, product and revision
volname	- set 8-character volume name
!<cmd>	- execute <cmd>, then return
quit	

format>

下表描述了 format 实用程序的主菜单项。

表 16-1 format 实用程序的主菜单项说明

菜单项	命令或菜单？	说明
disk	命令	列出系统的所有驱动器。此外，还可用于选择要在后续操作中使用的磁盘。该磁盘称为当前磁盘。
type	命令	标识当前磁盘的制造商和型号。另外，还可显示已知驱动器类型列表。可用于为所有 SCSI-2 磁盘驱动器选择 Auto configure 选项。
partition	菜单	创建和修改片。有关更多信息，请参见第 312 页中的“partition 菜单”。
current	命令	显示以下有关当前磁盘的信息： <ul style="list-style-type: none"> ■ 设备名称和设备类型 ■ 柱面、备用柱面、磁头和扇区的数目 ■ 物理设备名称
format	命令	按照一定顺序使用以下信息源之一格式化当前磁盘： <ol style="list-style-type: none"> 1. 在 format.dat 文件中找到的信息 2. 来自自动配置过程的信息 3. 不存在 format.dat 项时在提示符下键入的信息 <p>此命令不适用于 IDE 磁盘。IDE 磁盘由制造商进行格式化。</p>
fdisk	菜单	仅限于 x86 平台：运行 fdisk 程序可创建 Solaris fdisk 分区。 <p>在容量超过 1 TB 且带有 EFI 标号的磁盘上，不能使用 fdisk 命令。</p>
repair	命令	修复当前磁盘上的特定块。
label	命令	将新标号写入当前磁盘。
analyze	菜单	运行读取、写入和比较测试。有关更多信息，请参见第 314 页中的“analyze 菜单”。
defect	菜单	检索并显示缺陷列表。有关更多信息，请参见第 316 页中的“defect 菜单”。此功能不适用于 IDE 磁盘。IDE 磁盘会自动管理缺陷。
backup	命令	VTOC —搜索备份标号。 EFI —不支持。
verify	命令	显示以下有关当前磁盘的信息： <ul style="list-style-type: none"> ■ 设备名称和设备类型 ■ 柱面、备用柱面、磁头和扇区的数目 ■ 分区表

表 16-1 format 实用程序的主菜单项说明 (续)

菜单项	命令或菜单?	说明
save	命令	VTOC—保存新的磁盘和分区信息。 EFI—不适用。
inquiry	命令	仅限于 SCSI—显示当前磁盘的供应商、产品名称和修订版级别。
volname	命令	使用指定的由八个字符组成的新卷名标记磁盘。
quit	命令	退出 format 菜单。

partition 菜单

partition 菜单的显示与以下内容类似：

```
format> partition

PARTITION MENU:

      0      - change '0' partition

      1      - change '1' partition

      2      - change '2' partition

      3      - change '3' partition

      4      - change '4' partition

      5      - change '5' partition

      6      - change '6' partition

      7      - change '7' partition

select - select a predefined table

modify - modify a predefined partition table

name   - name the current table

print  - display the current table

label  - write partition map and label to the disk

quit
```


partition>

下表描述了 partition 菜单项。

表 16-2 partition 菜单项的说明

子命令	说明
change 'n' partition	用于指定新分区的以下信息： <ul style="list-style-type: none">■ 标识标记■ 权限标志■ 起始柱面■ 大小
select	用于选择预定义的分區表。
modify	用于更改分区表中的所有片。此命令优先于各个 change 'x' partition 命令。
name	用于为当前分区表指定名称。
print	显示当前分区表。
label	将分区映射和标号写入当前磁盘。
quit	退出 partition 菜单。

x86: fdisk 菜单

fdisk 菜单仅出现在基于 x86 的系统上，其显示与以下内容类似。

format> fdisk

Total disk size is 14169 cylinders

Cylinder size is 2510 (512 byte) blocks

Cylinders						
Partition	Status	Type	Start	End	Length	%
=====	=====	=====	=====	=====	=====	=====
1	Active	x86 Boot	1	9	9	0
2		Solaris2	10	14168	14159	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:

下表描述了 fdisk 菜单项。

表 16-3 x86: fdisk 菜单项的说明

菜单项	说明
Create a partition	创建 fdisk 分区。必须为每个 OS（如 Solaris 或 DOS）创建单独的分区。每个磁盘最多包含四个分区。系统会提示以磁盘百分比形式输入 fdisk 分区的大小。
Specify the active partition	用于指定要用于引导的分区。此菜单项可标识第一阶段引导程序查找第二阶段引导程序的位置。
Delete a partition	删除以前创建的分区。此命令将破坏分区中的所有数据。
Change between Solaris and Solaris2 Partition IDs	将分区 ID 从 130 (0x82) 更改为 191 (0xbf) 或执行相反操作。
Exit (update disk configuration and exit)	写入新版本的分区表并退出 fdisk 菜单。
Cancel (exit without updating disk configuration)	退出 fdisk 菜单，而不修改分区表。

analyze 菜单

analyze 菜单的显示与以下内容类似。

format> **analyze**

ANALYZE MENU:

- read - read only test (doesn't harm SunOS)
- refresh - read then write (doesn't harm data)
- test - pattern testing (doesn't harm data)
- write - write then read (corrupts data)
- compare - write, read, compare (corrupts data)
- purge - write, read, write (corrupts data)
- verify - write entire disk, then verify (corrupts data)
- print - display data buffer
- setup - set analysis parameters
- config - show analysis parameters
- quit

analyze>

下表描述了 analyze 菜单项。

表 16-4 analyze 菜单项的说明

子命令	说明
read	读取当前磁盘上的每个扇区。缺省情况下会修复有缺陷的块。
refresh	读取当前磁盘上的数据然后再将数据写入磁盘，而不损害数据。缺省情况下会修复有缺陷的块。
test	将一组模式写入磁盘，而不损害数据。缺省情况下会修复有缺陷的块。
write	将一组模式写入磁盘，然后读回磁盘上的数据。破坏磁盘上的现有数据。缺省情况下会修复有缺陷的块。
compare	将一组模式写入磁盘，读回数据，然后将其与写缓冲区中的数据进行比较。破坏磁盘上的现有数据。缺省情况下会修复有缺陷的块。

表 16-4 analyze 菜单项的说明 (续)

子命令	说明
purge	从磁盘中删除所有数据，以便不能通过任何方式检索数据。通过在整个磁盘（或磁盘的一部分）范围写入三种不同的模式来删除数据。如果验证通过，则将在整个磁盘（或磁盘的一部分）范围内写入十六进制位模式。 缺省情况下会修复有缺陷的块。
verify	第一遍将特殊数据写入整个磁盘上的每个块。第二遍读取并验证这些数据。破坏磁盘上的现有数据。缺省情况下会修复有缺陷的块。
print	显示读/写缓冲区中的数据。
setup	用于指定以下分析参数： Analyze entire disk? yes Starting block number: <i>depends on drive</i> Ending block number: <i>depends on drive</i> Loop continuously? no Number of passes: 2 Repair defective blocks? yes Stop after first error? no Use random bit patterns? no Number of blocks per transfer: 126 (0/n/nn) Verify media after formatting? yes Enable extended messages? no Restore defect list? yes Restore disk label? yes
config	显示当前分析参数。
quit	退出 analyze 菜单。

defect 菜单

defect 菜单的显示与以下内容类似：

```
format> defect
```

DEFECT MENU:

- primary - extract manufacturer's defect list
- grown - extract manufacturer's and repaired defects lists
- both - extract both primary and grown defects lists
- print - display working list
- dump - dump working list to file
- quit

```
defect>
```

下表描述了 defect 菜单项。

表 16-5 defect 菜单项说明

子命令	说明
primary	从磁盘驱动器读取制造商的缺陷列表并更新内存中的缺陷列表。
grown	读取产生的缺陷列表，然后更新内存中的缺陷列表。产生的缺陷是指在分析过程中检测到的缺陷。
both	读取制造商的缺陷列表和产生的缺陷列表。然后更新内存中的缺陷列表。
print	显示内存中的缺陷列表。
dump	将内存中的缺陷列表保存到文件中。
quit	退出 defect 菜单。

format.dat 文件

Solaris OS 附带的 format.dat 文件支持许多标准磁盘。如果 format.dat 文件中未列出您的磁盘驱动器，则可以执行以下操作：

- 向磁盘的 format.dat 文件中添加项。
- 通过选择 type 命令和 other 选项来添加 format 实用程序的各项。

如果将在整个站点中使用磁盘驱动器，则向 format.dat 文件添加项可以节省时间。要使用其他系统上的 format.dat 文件，请将该文件复制到每个将使用特定磁盘驱动器（已添加到 format.dat 文件中）的系统。

如果具有以下设备之一，则可能需要修改系统的 `/etc/format.dat` 文件：

- Solaris OS 不支持的磁盘
- 分区表与 Solaris OS 的缺省配置不同的磁盘

注 - 请勿修改 `/etc/format.dat` 文件中的缺省项。如果要修改缺省项，请复制该项，并为该项指定不同的名称，再进行相应更改以避免混淆。

`/etc/format.dat` 不适用于带有 EFI 标号的磁盘。

format.dat 文件的内容

`format.dat` 包含 `format` 实用程序使用的磁盘驱动器信息。`format.dat` 文件中定义了以下三项：

- 搜索路径
- 磁盘类型
- 片表

format.dat 文件的语法

以下语法规则适用于 `/etc/format.dat` 文件：

- 井号 (#) 是注释字符。`format` 实用程序不会解释一行中在井号后面的任何文本。
- `format.dat` 文件中的每个定义都出现在单独的逻辑行中。如果定义长度超过一行，则定义的所在行（最后一行除外）都必须以反斜杠 (\) 结束。
- 定义包含一系列赋值，这些赋值的左侧是标识符，右侧是一个或多个值。赋值运算符是等号 (=)。定义中的多个赋值必须以冒号 (:) 分隔。
- `format` 实用程序会忽略空格。如果要所赋的值包含空格，请将整个值用双引号 (") 引起来。此语法将导致引号中的空格作为赋值的一部分保留下来。
- 有些赋值的右侧可能有多个值。请使用逗号分隔这些值。

format.dat 文件中的关键字

`format.dat` 文件包含启动 `format` 实用程序时该程序读取的磁盘定义。每个定义都以以下关键字之一开头：`disk_type` 或 `partition`。下表描述了这些关键字。

表 16-6 format.dat 文件的关键字说明

关键字	说明
disk_type	定义控制器和磁盘型号。每个 disk_type 定义都包含有关磁盘物理几何参数的信息。缺省数据文件包含 Solaris OS 支持的控制器和磁盘的定义。 仅当使用不支持的磁盘时，才需要添加新的 disk_type 定义。可以向数据文件中添加任意数目的 disk_type 定义。
partition	为特定磁盘类型定义分区表。分区表包含分区信息以及可在 format 实用程序中引用的名称。缺省的 format.dat 文件包含多种磁盘驱动器的缺省分区定义。如果在系统中的任何磁盘上重新创建了分区，请添加分区定义。可以向数据文件中添加任意数目的分区定义。

磁盘类型 (format.dat)

format.dat 文件中的 disk_type 关键字定义了控制器和磁盘型号。每个 disk_type 定义都包含有关磁盘物理几何参数的信息。缺省的 format.dat 文件包含 Solaris OS 支持的控制器和磁盘的定义。仅当使用不支持的磁盘时，才需要添加新的 disk_type。可以向数据文件中添加任意数目的 disk_type 定义。

可将磁盘类型名称指定给关键字本身。该名称出现在磁盘的标号中，并且在运行 format 实用程序时用于标识磁盘类型。使用双引号将该名称引起来可保留名称中的所有空格。下表描述了在所有的 disk_type 定义中还必须被赋值的标识符。

表 16-7 所需的 disk_type 标识符 (format.dat)

标识符	说明
ctlr	标识磁盘类型的控制器类型。当前，支持的值为 SCSI 和 ATA。
ncyl	指定磁盘类型中的数据柱面的数目。这将确定允许系统访问的逻辑磁盘柱面数目。
acyl	指定磁盘类型中的备用柱面的数目。这些柱面供 format 实用程序用于存储信息（如驱动器的缺陷列表）。应始终保留至少两个柱面进行备用。
pcyl	指定磁盘类型中的物理柱面的数目。此数字用于计算磁盘介质的范围。此数字通常等于 ncyl 加上 acyl。
nhead	指定磁盘类型中的磁头的数目。此数字用于计算磁盘介质的范围。
nsect	指定磁盘类型中每个磁道的数据扇区的数目。此数字用于计算磁盘介质的范围。请注意，此数字仅包括数据扇区。任何备件都不会反映在每个磁道的数据区段数目中。
rpm	指定磁盘类型的每分钟旋转次数。此信息将置于标签中，随后由文件系统用于计算文件数据的最佳保存位置。

其他标识符可能是必需的，具体取决于控制器。下表描述了 SCSI 控制器所需的标识符。

表 16-8 SCSI 控制器 format.dat 所需的 disk_type 标识符

标识符	说明
fmt_time	指定一个数字，用于指明格式化给定驱动器所需的时间。有关更多信息，请参见控制器手册。
cache	指定一个数字，用于控制 format 实用程序运行时执行的板载高速缓存操作。有关更多信息，请参见控制器手册。
trks_zone	指定一个数字，用于标识每个缺陷区域中存在的要在备用扇区映射中使用的磁道数。有关更多信息，请参见控制器手册。
asect	指定一个数字，用于标识给定缺陷区域中可用于备用映射的扇区。有关更多信息，请参见控制器手册。

示例 16-1 SCSI 控制器 (format.dat) 所需的 disk_type 标识符

以下是 disk_type 定义的示例：

```
disk_type = "SUN1.3G" \  
  
    : ctlr = SCSI : fmt_time = 4 \  
  
    : trks_zone = 17 : asect = 6 : atrks = 17 \  
  
    : ncyl = 1965 : acyl = 2 : pcyl = 3500 : nhead = 17 : nsect = 80 \  
  
    : rpm = 5400 : bpt = 44823  
  
disk_type = "SUN2.1G" \  
  
    : ctlr = SCSI : fmt_time = 4 \  
  
    : ncyl = 2733 : acyl = 2 : pcyl = 3500 : nhead = 19 : nsect = 80 \  
  
    : rpm = 5400 : bpt = 44823  
  
disk_type = "SUN2.9G" \  
  
    : ctlr = SCSI : fmt_time = 4 \
```


示例 16-1 SCSI 控制器 (format.dat) 所需的 disk_type 标识符 (续)

```
: ncyl = 2734 : acyl = 2 : pcyl = 3500 : nhead = 21 : nsect = 99 \

: rpm = 5400
```

分区表 (format.dat)

format.dat 文件中的分区表为特定磁盘类型定义了片表。

将分区表的名称指定给 format.dat 文件中的 partition 关键字。使用双引号将该名称引起来可保留名称中的所有空格。下表描述了所有分区表中必须赋值的标识符。

表 16-9 分区表 (format.dat) 所需的标识符

标识符	说明
disk	为此分区表定义的 disk_type 的名称。此名称必须与 disk_type 定义中显示的完全相同。
ctlr	可与此分区表连接的磁盘控制器的类型。当前，对于 ATA 控制器，支持的值为 ATA；对于 SCSI 控制器，支持的值为 SCSI。另外，还必须为在 disk_type 定义中指定的 disk_type 定义此处指定的控制器类型。

片定义中的其他标识符描述了实际的分区信息。标识符是 0 到 7 之间的数字。这些标识符是可选的。未显式指定的任何分区长度都设置为 0。其中每个标识符的值都是以逗号分隔的一对数字。第一个数字是分区的起始柱面。第二个数字是片中的扇区数目。

示例 16-2 分区表 (format.dat) 所需的标识符

以下是一些片信息示例：

```
partition = "SUN1.3G" \

: disk = "SUN1.3G" : ctlr = SCSI \

: 0 = 0, 34000 : 1 = 25, 133280 : 2 = 0, 2672400 : 6 = 123, 2505120

partition = "SUN2.1G" \

: disk = "SUN2.1G" : ctlr = SCSI \

: 0 = 0, 62320 : 1 = 41, 197600 : 2 = 0, 4154160 : 6 = 171, 3894240
```

示例 16-2 分区表 (format.dat) 所需的标识符 (续)

```
partition = "SUN2.9G" \  
  
: disk = "SUN2.9G" : ctrlr = SCSI \  
  
: 0 = 0, 195426 : 1 = 94, 390852 : 2 = 0, 5683986 : 6 = 282, 5097708
```

为 format 实用程序指定备用数据文件

format 实用程序按一定顺序通过以下方法确定备用文件的位置：

1. 如果使用 `format -x` 选项给定文件名称，则始终将该文件用作数据文件。
2. 如果未指定 `-x` 选项，则 `format` 实用程序将搜索当前目录中名为 `format.dat` 的文件。如果该文件存在，则将其用作数据文件。
3. 如果上述任一方法都无法产生数据文件，则 `format` 实用程序将使用 `/etc/format.dat` 文件作为数据文件。此文件是 Solaris OS 附带的，并且会始终提供此文件。

format 命令的输入规则

使用 `format` 实用程序时，需要提供各种信息。本节介绍这类信息的规则。有关在指定数据时使用 `format` 的帮助功能的信息，请参见第 325 页中的“[获取有关 format 实用程序的帮助](#)”。

为 format 命令指定数字

`format` 实用程序中的多个位置都要求使用数字作为输入。必须指定相应数据，或从选项列表表中选择一个数字。在任一情况下，帮助功能都会导致 `format` 显示预期的数字上限和下限。只需输入合适的数字即可。如果未将基数显式指定为数字的一部分（例如，`0x` 表示十六进制），则假定该数字采用十进制格式。

以下是整数输入的示例：

```
Enter number of passes [2]: 34
```

```
Enter number of passes [34] 0xf
```

为 format 命令指定块编号

每次要求指定磁盘块编号时，都可以使用两种方式来执行此操作：

- 将块编号指定为整数
- 采用柱面/磁头/扇区格式指定块编号

可以将该信息指定为表示逻辑块编号的整数。可以指定任何基数的编号，但缺省值为十进制。另外，还可以在此处使用最大值运算符（美元符号 \$），以便 format 实用程序可以选择合适的值。SunOS 磁盘驱动器会在错误消息中使用逻辑块格式。

另一种指定块编号的方法是使用柱面/磁头/扇区格式。使用此方法时，必须显式指定块编号的三个逻辑组成部分：柱面、磁头和扇区值。这些值仍然为逻辑值。但是，可以使用这些值来定义与介质布局相关的磁盘区域。

如果未指定任何柱面/磁头/扇区编号，则假定该值为零。另外，还可以使用最大值运算符来代替任何编号。这样，format 实用程序即可选择合适的值。以下是一些柱面、磁头和扇区值的示例：

```
Enter defective block number: 34/2/3
```

```
Enter defective block number: 23/1/
```

```
Enter defective block number: 457//
```

```
Enter defective block number: 12345
```

```
Enter defective block number: 0xabcd
```

```
Enter defective block number: 334/$/2
```

```
Enter defective block number: 892//
```

format 实用程序始终采用这两种格式显示块编号。另外，帮助功能也会采用这两种格式显示预期块编号的上限和下限。

指定 format 命令名称

只要 format 实用程序显示菜单提示，就需要将命令名称作为输入。可以缩写命令名称，只要键入的内容可充分唯一标识所需命令即可。

例如，使用 p 访问 format 菜单中的 partition 菜单。那么，键入 p 显示当前的片表。

```
format> p
```

```
PARTITION MENU:
```

```
0      - change '0' partition

1      - change '1' partition

2      - change '2' partition

3      - change '3' partition

4      - change '4' partition

5      - change '5' partition

6      - change '6' partition

7      - change '7' partition

select - select a predefined table

modify - modify a predefined partition table

name   - name the current table

print  - display the current table

label  - write partition map and label to the disk

quit

partition> p
```

为 format 命令指定磁盘名称

在 `format` 实用程序的某些位置，必须指定一些名称。在这些情况下，可以自由指定名称所需的任何字符串。如果名称中包含空格，则必须使用双引号 (") 将整个名称引起来。否则，仅使用名称的第一个单词。

例如，如果要标识磁盘的特定分区表，则可以使用 `partition` 菜单中可用的 `name` 子命令：

```
partition> name

Enter table name (remember quotes): "new disk3"
```

获取有关 format 实用程序的帮助

format 实用程序提供了 format 实用程序每次要求输入时可以使用的帮助功能。通过键入问号 (?), 可以请求有关要求的输入的 help。format 实用程序可显示所需输入类型的简短说明。

如果在菜单提示下键入 ?, 则将显示可用命令的列表。

与 format 实用程序关联的手册页包括以下信息：

- `format(1M)` — 介绍基本的 format 实用程序功能并提供所有命令行变量的说明。
- `format.dat(4)` — 介绍 format 实用程序的磁盘驱动器配置信息。

管理文件系统（概述）

管理文件系统是最重要的系统管理任务之一。

以下是本章中概述信息的列表。

- 第 327 页中的 “Solaris 10 6/06 发行版中文件系统方面的新增功能”
- 第 338 页中的 “Solaris 10 发行版中文件系统方面的新增功能”
- 第 343 页中的 “有关文件系统管理任务的参考信息”
- 第 343 页中的 “文件系统概述”
- 第 344 页中的 “文件系统的类型”
- 第 349 页中的 “用于文件系统管理的命令”
- 第 350 页中的 “缺省的 Solaris 文件系统”
- 第 348 页中的 “交换空间”
- 第 351 页中的 “UFS 文件系统”
- 第 353 页中的 “挂载和取消挂载文件系统”
- 第 358 页中的 “确定文件系统的类型”

Solaris 10 6/06 发行版中文件系统方面的新增功能

本节介绍 Solaris 10 6/06 发行版中文件系统方面的新增功能。

- 第 327 页中的 “ZFS 文件系统”
- 第 328 页中的 “UFS 文件系统实用程序（`fsck`、`mkfs` 和 `newfs`）的增强功能”

ZFS 文件系统

Solaris 10 6/06：ZFS 是一种全新的文件系统，可提供简单管理、事务性语义、端到端数据完整性以及极好的可伸缩性。此外，ZFS 还可提供以下管理功能：

- 备份和恢复功能
- 设备管理支持

- GUI 管理工具
- 持久性快照和克隆功能
- 可以为文件系统设置的配额
- 基于 RBAC 的访问控制
- 为文件系统预留存储池空间
- 支持安装了区域的 Solaris 系统

可以在同一 Solaris 系统上同时设置 ZFS 和 UFS 文件系统。但是，不能将 ZFS 用作根文件系统。有关使用 ZFS 时的其他限制的信息，请参见《Solaris 10 新增功能》中的“Solaris ZFS 文件系统”。

有关使用 ZFS 的更多信息，请参见《Solaris ZFS 管理指南》。

UFS 文件系统实用程序（`fsck`、`mkfs` 和 `newfs`）的增强功能

Solaris 10 6/06：文件系统检查实用程序 `fsck` 已经增强，包括 FreeBSD 4.9 版本的 `fsck` 程序的功能以及其他增强功能。

此 Solaris 发行版中的 `fsck` 实用程序包括以下改进：

- 更彻底地检查和修复文件系统，并且提供改进的错误消息。例如，在某些情况下，`fsck` 可确定缺少的结构并对其进行适当替换。
- 自动搜索备份超级块。
- 报告 `fsck` 何时需要重新运行。
- 清除目录时，`fsck` 现在会尝试立即恢复目录内容，因此可减少重新运行此实用程序所用的时间。
- 如果 `fsck` 发现重复块，并且并非所有引用重复块的文件都已清除，则 `fsck` 会在 `fsck` 运行结束时报告 inode 的数目。然后，可以使用 `find` 命令查看已损坏的 inode。
- 包括与扩展属性和其他特殊文件（如设备文件和 ACL 项）的状态有关的改进错误消息。
- 包括用于启用更详细消息的 `-v` 选项。

此外，还更新了 `newfs` 和 `mkfs` 命令，其中包括用于以文本格式显示文件系统超级块信息或以二进制格式转储超级块信息的新选项。

```
newfs [ -S or -B ] /dev/rdisk/...
```

-S 以文本格式显示文件系统的超级块

-B 以二进制格式转储文件系统的超级块

```
mkfs [ -o calcsb or -o calcbinsb ] /dev/rdisk/... size
```


- o calcsb 以文本格式显示文件系统的超级块
- o calcbinsb 以二进制格式转储文件系统的超级块

`fsck` 实用程序使用此超级块信息搜索备份超级块。

以下各节介绍了特定的 `fsck` 增强功能及其对应的错误消息。有关使用 `fsck` 实用程序修复损坏的超级块的逐步说明，请参见第 446 页中的“[如何恢复坏的超级块（仅限 Solaris 10 6/06 发行版）](#)”。

自动搜索备份超级块

以下 `fsck` 错误消息示例说明了自动搜索备份超级块的功能。



注意 – 如果文件系统包含损坏的超级块，并且该超级块是使用 `newfs` 或 `mkfs` 自定义参数（如 `ntrack` 或 `nsect`）创建的，则使用 `fsck` 自动搜索的超级块执行修复过程可能会损坏文件系统。

如果文件系统是使用自定义参数创建的，并且包含错误的超级块，则 `fsck` 会提供取消 `fsck` 会话的提示：

CANCEL FILESYSTEM CHECK?

- 如果文件系统是使用 `newfs` 命令创建的，并且 `fsck` 响应表明只是主超级块已损坏，请考虑使用 `fsck` 恢复超级块。

```
# fsck /dev/dsk/c1t2d0s0
```

```
** /dev/rdisk/c1t2d0s0
```

```
BAD SUPERBLOCK AT BLOCK 16: BLOCK SIZE LARGER THAN MAXIMUM SUPPORTED
```

```
LOOK FOR ALTERNATE SUPERBLOCKS WITH MKFS? no
```

```
LOOK FOR ALTERNATE SUPERBLOCKS WITH NEWFS? yes
```

```
FOUND ALTERNATE SUPERBLOCK 32 WITH NEWFS
```

USE ALTERNATE SUPERBLOCK? **yes**

FOUND ALTERNATE SUPERBLOCK AT 32 USING NEWFS

If filesystem was created with manually-specified geometry, using
auto-discovered superblock may result in irrecoverable damage to
filesystem and user data.

CANCEL FILESYSTEM CHECK? **no**

** Last Mounted on

** Phase 1 - Check Blocks and Sizes

** Phase 2 - Check Pathnames

** Phase 3a - Check Connectivity

** Phase 3b - Verify Shadows/ACLs

** Phase 4 - Check Reference Counts

** Phase 5 - Check Cylinder Groups

CORRECT GLOBAL SUMMARY

SALVAGE? **y**

UPDATE STANDARD SUPERBLOCK? **y**

81 files, 3609 used, 244678 free (6 frags, 30584 blocks, 0.0%

```
fragmentation)
```

```
***** FILE SYSTEM WAS MODIFIED *****
```

- 如果文件系统是使用 `mkfs` 命令创建的，并且 `fsck` 响应表明只是主超级块已损坏，请考虑使用 `fsck` 恢复超级块。

```
# fsck /dev/dsk/c1t2d0s0
```

```
** /dev/rdisk/c1t2d0s0
```

```
BAD SUPERBLOCK AT BLOCK 16: BLOCK SIZE LARGER THAN MAXIMUM SUPPORTED
```

```
LOOK FOR ALTERNATE SUPERBLOCKS WITH MKFS? yes
```

```
FOUND ALTERNATE SUPERBLOCK 32 WITH MKFS
```

```
USE ALTERNATE SUPERBLOCK? yes
```

```
FOUND ALTERNATE SUPERBLOCK AT 32 USING MKFS
```

If filesystem was created with manually-specified geometry, using auto-discovered superblock may result in irrecoverable damage to filesystem and user data.

```
CANCEL FILESYSTEM CHECK? no
```

```
** Last Mounted on
```

```
** Phase 1 - Check Blocks and Sizes
```

```
** Phase 2 - Check Pathnames

** Phase 3a - Check Connectivity

** Phase 3b - Verify Shadows/ACLs

** Phase 4 - Check Reference Counts

** Phase 5 - Check Cylinder Groups
```

```
CORRECT GLOBAL SUMMARY
```

```
SALVAGE? y
```

```
UPDATE STANDARD SUPERBLOCK? y
```

```
81 files, 3609 used, 243605 free (117 frags, 30436 blocks, 0.0%
```

```
fragmentation)
```

```
***** FILE SYSTEM WAS MODIFIED *****
```

以下示例说明在超级块已损坏的情况下指定 `fsck` 的 `-y` 选项时将发生的情况。您将自动从 `fsck` 会话退出。此时会显示一条消息，指示使用替代超级块重新运行该命令。

```
# fsck -y /dev/dsk/c1t2d0s0
```

```
#
```

```
** /dev/rdisk/c1t2d0s0
```

```
BAD SUPERBLOCK AT BLOCK 16: BLOCK SIZE LARGER THAN MAXIMUM SUPPORTED
```

```
LOOK FOR ALTERNATE SUPERBLOCKS WITH MKFS? yes
```

LOOK FOR ALTERNATE SUPERBLOCKS WITH NEWFS? yes

SEARCH FOR ALTERNATE SUPERBLOCKS FAILED.

USE GENERIC SUPERBLOCK FROM MKFS? yes

CALCULATED GENERIC SUPERBLOCK WITH MKFS

If filesystem was created with manually-specified geometry, using auto-discovered superblock may result in irrecoverable damage to filesystem and user data.

CANCEL FILESYSTEM CHECK? yes

Please verify that the indicated block contains a proper superblock for the filesystem (see fsdb(1M)).

FSCK was running in YES mode. If you wish to run in that mode using the alternate superblock, run 'fsck -y -o b=453920 /dev/rdisk/clt2d0s0'.

以下 fsck 错误消息情况说明备份超级块的新提示，但在此示例中并未取消运行 fsck。如果此文件系统是使用自定义参数创建的，或者在此文件系统上运行 fsck 可能会带来其他问题，则应取消 fsck 会话。

以斜体提供的各种超级块错误情况如下所示：

```
# fsck /dev/rdisk/c0t1d0s0
```

**** /dev/rdisk/c0t1d0s0**

BAD SUPERBLOCK AT BLOCK 16: BLOCK SIZE LARGER THAN MAXIMUM SUPPORTED

BAD SUPERBLOCK AT BLOCK 16: NUMBER OF DATA BLOCKS OUT OF RANGE

BAD SUPERBLOCK AT BLOCK 16: INODES PER GROUP OUT OF RANGE

BAD SUPERBLOCK AT BLOCK 16: MAGIC NUMBER WRONG

BAD SUPERBLOCK AT BLOCK 16: BAD VALUES IN SUPER BLOCK

BAD SUPERBLOCK AT BLOCK 16: NCG OUT OF RANGE

BAD SUPERBLOCK AT BLOCK 16: CPG OUT OF RANGE

*BAD SUPERBLOCK AT BLOCK 16: NCYL IS INCONSISTENT WITH NCG*CPG*

BAD SUPERBLOCK AT BLOCK 16: SIZE OUT OF RANGE

BAD SUPERBLOCK AT BLOCK 16: NUMBER OF DIRECTORIES OUT OF RANGE

BAD SUPERBLOCK AT BLOCK 16: ROTATIONAL POSITION TABLE SIZE OUT OF RANGE

BAD SUPERBLOCK AT BLOCK 16: SIZE OF CYLINDER GROUP SUMMARY AREA WRONG

BAD SUPERBLOCK AT BLOCK 16: INOPB NONSENSICAL RELATIVE TO BSIZE

LOOK FOR ALTERNATE SUPERBLOCKS WITH MKFS? **yes**

FOUND ALTERNATE SUPERBLOCK 32 WITH MKFS

USE ALTERNATE SUPERBLOCK? **yes**

FOUND ALTERNATE SUPERBLOCK AT 32 USING MKFS

If filesystem was created with manually-specified geometry, using
auto-discovered superblock may result in irrecoverable damage to
filesystem and user data.

CANCEL FILESYSTEM CHECK? **no**

**** Last Mounted on**

**** Phase 1 - Check Blocks and Sizes**

**** Phase 2a - Check Duplicated Names**

**** Phase 2b - Check Pathnames**

**** Phase 3a - Check Connectivity**

**** Phase 3b - Verify Shadows/ACLs**

**** Phase 4 - Check Reference Counts**

**** Phase 5 - Check Cylinder Groups**

SALVAGE? **yes**

UPDATE STANDARD SUPERBLOCK? **yes**

82 files, 3649 used, 244894 free (6 frags, 30611 blocks, 0.0%
fragmentation)

***** FILE SYSTEM WAS MODIFIED *****

fsck 报告它何时需要重新运行

如果 **fsck** 更好地报告它何时需要重新运行，则会减少多次运行所需的时间和必要，在大文件系统上多次运行可能特别耗时。

以下新消息提示您在错误情况结束时重新运行 **fsck** 实用程序：

```
***** PLEASE RERUN FSCK *****
```

或：

```
Please rerun fsck(1M) to correct this.
```

这些新提示解决了以前在确定是否应该重新运行 **fsck** 方面的困难。

除非出现上述类似消息提示重新运行 **fsck**，否则不需要运行 **fsck**，即使显示以下消息后也是如此：

```
***** FILE SYSTEM WAS MODIFIED *****
```

但是，在出现此消息后重新运行 **fsck** 并不会损害文件系统。此消息只是有关 **fsck** 的更正操作的信息。

有关扩展属性的 fsck 新消息

提供新的 **fsck** 消息，可报告和修复具有扩展属性的文件。例如：

```
BAD ATTRIBUTE REFERENCE TO I=1 FROM I=96
```

```
Attribute directory I=97 not attached to file I=96
```

```
    I=96  OWNER=root MODE=40755
```

```
SIZE=512 MTIME=Jul 21 16:23 2005
```

```
DIR= <xattr>
```

```
FIX?  yes
```

```
ZERO LENGTH ATTR DIR I=12  OWNER=root MODE=160755
```

```
SIZE=0 MTIME=Jul 21 16:25 2005
```

```
CLEAR?  yes
```


File should BE marked as extended attribute

I=22 OWNER=root MODE=100644

SIZE=0 MTIME=Jul 21 16:36 2005

FILE= <xattr>

FIX? **yes**

UNREF ATTR DIR I=106 OWNER=root MODE=160755

SIZE=512 MTIME=Jul 21 16:36 2005

RECONNECT? **yes**

File I=107 should NOT be marked as extended attribute

I=107 OWNER=root MODE=100644

SIZE=0 MTIME=Jul 21 16:36 2005

FILE=?/attfsdir-7-att

FIX? **yes**

DIR I=106 CONNECTED.

更好地处理重复块或段

fsck 错误消息现在会报告有关块、段或 LFN（从文件开头算起的逻辑段编号）的信息。例如，可能会显示以下类似输出：

**** Phase 1 - Check Blocks and Sizes**

FRAGMENT 784 DUP I=38 LFN 0

FRAGMENT 785 DUP I=38 LFN 1

FRAGMENT 786 DUP I=38 LFN 2

.

.
.

fsck 将对象作为段处理，但是在以前的 Solaris 发行版中，该命令只是将对象作为块来报告其信息。现在，它将对象作为段来正确报告其信息。

如果 fsck 发现错误情况涉及重复块或段，则 fsck 会在 fsck 输出的结尾显示未清除的文件。例如，可能会显示以下类似输出：

LIST REMAINING DUPS? **yes**

Some blocks that were found to be in multiple files are still
assigned to file(s).

Fragments sorted by inode and logical offsets:

Inode 38:

Logical Offset	0x00000000	Physical Fragment	784
Logical Offset	0x00000800	Physical Fragment	786
Logical Offset	0x00001000	Physical Fragment	788
Logical Offset	0x00001800	Physical Fragment	790

然后，可以使用 `find -i inode-number` 命令标识此示例中 inode 38 的名称。

Solaris 10 发行版中文件系统方面的新增功能

本节介绍 Solaris 10 发行版中文件系统方面的新增功能。

- 第 339 页中的“缺省情况下启用 UFS 日志记录”
- 第 339 页中的“NFS 版本 4”
- 第 340 页中的“64 位: 对多 TB UFS 文件系统的支持”
- 第 343 页中的“lib_hwcap”

缺省情况下启用 UFS 日志记录

Solaris 10 – 除了以下情况，所有 UFS 文件系统均缺省启用日志记录：

- 明确禁用了日志记录。
- 用于日志的文件系统空间不足。

在以前的 Solaris 发行版中，必须手动启用 UFS 日志记录。有关 UFS 日志记录的更多信息，请参见第 352 页中的“UFS 日志记录”。

在此发行版中使用 UFS 日志记录时，请牢记以下问题：

- 确保具有足够的磁盘空间可用于一般系统需要，如用于用户和应用程序以及用于 UFS 日志记录。
- 如果没有足够的磁盘空间用于记录数据，则会显示以下类似消息：

```
# mount /dev/dsk/c0t4d0s0 /mnt

/mnt: No space left on device

Could not enable logging for /mnt on /dev/dsk/c0t4d0s0.

#
```

但是，文件系统仍然会挂载。例如：

```
# df -h /mnt

Filesystem              size  used  avail capacity  Mounted on

/dev/dsk/c0t4d0s0       142M   142M    0K    100%      /mnt

#
```

- 通常为空的已启用日志记录的 UFS 文件系统会占用一些磁盘空间用于日志。
- 如果从以前的 Solaris 发行版升级到此 Solaris 发行版，即使 `/etc/vfstab` 文件中未指定 `logging` 选项，UFS 文件系统仍会启用日志记录。要禁用日志记录，请向 `/etc/vfstab` 文件中的 UFS 文件系统各项添加 `nologging` 选项。

NFS 版本 4

此 Solaris 发行版包括 Sun 实现的 NFS 版本 4 分布式文件访问协议。

NFS 版本 4 将文件访问、文件锁定和挂载协议集成到一个统一的协议中，从而使穿越防火墙进行遍历更为容易，并提高了安全性。Solaris 实现的 NFS 版本 4 与 Kerberos V5 完

全集成（也称作 SEAM），因此可以提供验证机制、完整性和保密性。使用 NFS 版本 4，还可在客户机和服务器之间协商要使用的安全特性。使用 NFS 4.0 版，服务器可以为不同的文件系统提供不同的安全特性。

有关 NFS 版本 4 功能的更多信息，请参见《系统管理指南：网络服务》中的“NFS 服务的新增功能”。

NFS 版本 4 和 CacheFS 的兼容性问题

如果 CacheFS 客户机和 CacheFS 服务器运行的都是 NFS 版本 4，则文件不会再在后台文件系统中高速缓存。后台文件系统将提供所有文件访问权。另外，由于前台文件系统中没有高速缓存任何文件，因此特定于 CacheFS 的挂载选项（该选项旨在影响前台文件系统）会被忽略。特定于 CacheFS 的挂载选项不适用于后台文件系统。

注 - 第一次在系统上配置 NFS 版本 4 时，控制台上将出现一条警告，指出高速缓存不再起作用。

如果要按照以前 Solaris 发行版中的方式实现 CacheFS 挂载，可在 CacheFS mount 命令中指定 NFS 版本 3。例如：

```
mount -F cachefs -o backfstype=nfs,cachedir=/local/mycache,vers=3
```

```
starbug:/docs /docs
```

64 位: 对多 TB UFS 文件系统的支持

此 Solaris 发行版在运行 64 位 Solaris 内核的系统上提供对多 TB UFS 文件系统的支持。

以前，UFS 文件系统在 64 位系统和 32 位系统上的大小都仅限于 1 TB 左右。所有 UFS 文件系统命令和实用程序都已更新，可支持多 TB UFS 文件系统。

例如，`ufsdump` 命令已经更新，可以使用更大的块大小来转储大 UFS 文件系统：

```
# ufsdump 0f /dev/md/rdisk/d97 /dev/md/rdisk/d98

DUMP: Date of this level 0 dump: Tue Jan 07 14:23:36 2003

DUMP: Date of last level 0 dump: the epoch

DUMP: Dumping /dev/md/rdisk/d98 to /dev/md/rdisk/d97.

DUMP: Mapping (Pass I) [regular files]
```

```
DUMP: Mapping (Pass II) [directories]

DUMP: Forcing larger tape block size (2048).

DUMP: Writing 32 Kilobyte records

DUMP: Estimated 4390629500 blocks (2143862.06MB).

DUMP: Dumping (Pass III) [directories]

DUMP: Dumping (Pass IV) [regular files]
```

管理小于 1 TB 的 UFS 文件系统保持不变。小于 1 TB 的 UFS 文件系统和大于 1 TB 的文件系统之间不存在管理差异。

可以首先创建小于 1 TB 的 UFS 文件系统，然后使用 `newfs -T` 选项指定最终可以将其扩展为多 TB 文件系统。此选项可设置 inode 和段的密度针对多 TB 文件系统相应地进行伸缩。

如果在运行 32 位内核的系统上创建小于 1 TB 的 UFS 文件系统时使用 `newfs -T` 选项，则在 64 位内核下引导此文件系统时使用 `growfs` 命令可最终扩展此系统。有关更多信息，请参见 `newfs(1M)`。

可以使用 `fstyp -v` 命令，通过检查 `magic` 列中的以下值确定 UFS 文件系统是否具有多 TB 支持：

```
# /usr/sbin/fstyp -v /dev/md/rdisk/d3 | head -5

ufs

magic    decade    format    dynamic time    Fri May 21 11:10:10 2004
```

没有多 TB 支持的 UFS 文件系统会生成以下 `fstyp` 输出：

```
# /usr/sbin/fstyp -v /dev/md/rdisk/d0 | head -5

ufs

magic    11954     format    dynamic time    Fri May 21 11:46:40 2004
```

可以使用 `growfs` 命令将 UFS 文件系统扩展为片或卷的大小，而不会丢失服务或数据。有关更多信息，请参见 `growfs(1M)`。

新增的两项相关功能是：EFI 磁盘标号的多 TB 卷支持和 Solaris Volume Manager 的多 TB 卷支持。有关更多信息，请参见第 202 页中的“带有 EFI 磁盘标号的多 TB 磁盘支持”和《Solaris Volume Manager 管理指南》。

多 TB UFS 文件系统的功能

多 TB UFS 文件系统包括以下功能：

- 提供创建最大大小为 16 TB 的 UFS 文件系统的能力。
- 提供创建小于 16 TB 的文件系统并且以后可以将其大小增至高达 16 TB 的能力。
- 可以在物理磁盘、Solaris Volume Manager 的逻辑卷和 Veritas 的 VxVM 逻辑卷上创建多 TB 文件系统。
- 多 TB 文件系统可从启用 UFS 日志记录而提高的性能中受益。多 TB 文件系统还可从日志记录的可用性中受益，因为启用日志记录时可能不必运行 `fsck` 命令。
- 为多 TB UFS 文件系统创建分区时，将用 EFI 磁盘标号自动标记磁盘。有关 EFI 磁盘标号的更多信息，请参见第 202 页中的“带有 EFI 磁盘标号的多 TB 磁盘支持”。
- 通过在文件系统超过 512 GB 时创建多个后备存储文件，提供为多 TB 文件系统拍摄快照的能力。

多 TB UFS 文件系统的限制

多 TB UFS 文件的限制如下：

- 在 32 位系统上不支持此功能。
- 不能在运行 32 位 Solaris 内核的系统上挂载大于 1 TB 的文件系统。
- 不能在运行 64 位 Solaris 内核的系统上从大于 1 TB 的文件系统进行引导。此限制意味着不能将根 (/) 文件系统放置在多 TB 文件系统上。
- 不支持大于 1 TB 的单独文件。
- 每 TB 的 UFS 文件系统的最大文件数为 1 百万个文件。例如，4 TB 文件系统可以包含 4 百万个文件。

此限制旨在减少使用 `fsck` 命令检查文件系统所需的时间。

- 可以对多 TB UFS 文件系统设置的最高配额是 2 TB 的 1024 字节块。

有关多 TB UFS 任务的参考信息

请参见以下内容，以了解有关使用多 TB UFS 文件系统的逐步说明。

多 TB UFS 任务	更多信息
创建多 TB UFS 文件系统	第 363 页中的“如何创建多 TB UFS 文件系统” 第 365 页中的“如何扩展多 TB UFS 文件系统” 第 367 页中的“如何将 UFS 文件系统扩展为多 TB UFS 文件系统”
创建多 TB UFS 快照	示例 26-2

多 TB UFS 任务	更多信息
对多 TB UFS 问题进行疑难解答	第 369 页中的“对多 TB UFS 文件系统问题进行疑难解答”

libc_hwcap

x86 系统上的挂载输出可能包括 libc_hwcap 库的回送挂载，即 libc 的硬件优化实现。此 libc 实现旨在优化 32 位应用程序的性能。

此回送挂载不需要进行管理并且不会占用磁盘空间。

有关文件系统管理任务的参考信息

请参见以下内容，以了解管理文件系统的逐步说明。

文件系统管理任务	更多信息
创建新的文件系统。	第 18 章 和 第 20 章
使本地文件和远程文件可供用户使用。	第 19 章
连接和配置新的磁盘设备。	第 11 章
制定和实现备份计划，并根据需要恢复文件和文件系统。	第 24 章
检查并更正文件系统不一致的地方。	第 22 章

文件系统概述

文件系统是用于组织和存储文件的目录结构。术语**文件系统**可用于描述以下各项：

- 特定类型的文件系统：基于磁盘、基于网络或虚拟
- 整个文件树，从根 (/) 目录开始
- 磁盘片或其他介质存储设备的数据结构
- 文件树结构的一部分，已连接到主文件树上的挂载点，以便可以访问文件

通常，可根据上下文知道具体的含义。

Solaris OS 使用**虚拟文件系统** (virtual file system, VFS) 体系结构，该体系结构可为不同的文件系统类型提供标准接口。使用 VFS 体系结构，内核可处理读取、写入和列出文件等基本操作。VFS 体系结构还使添加新文件系统变得更容易。

文件系统的类型

Solaris OS 支持三种类型的文件系统：

- 基于磁盘
- 基于网络
- 虚拟

要确定文件系统的类型，请参见第 358 页中的“确定文件系统的类型”。

基于磁盘的文件系统

基于磁盘的文件系统存储在硬盘、CD-ROM 和软盘等物理介质上。可以按不同的格式向基于磁盘的文件系统中进行写入。下表介绍了可用的格式。

基于磁盘的文件系统	格式说明
UFS	UNIX 文件系统（基于 4.3 Tahoe 发行版中提供的 BSD Fat Fast File 系统）。UFS 是 Solaris OS 缺省的基于磁盘的文件系统。 必须首先对磁盘格式化并将其分成片，然后才能在磁盘上创建 UFS 文件系统。有关格式化磁盘和将磁盘分片的信息，请参见第 11 章。
ZFS	ZFS 文件系统是 Solaris 10 6/06 发行版中新增的文件系统。有关更多信息，请参见《Solaris ZFS 管理指南》。
HSFS	High Sierra、Rock Ridge 和 ISO 9660 文件系统。High Sierra 是第一个 CD-ROM 文件系统。ISO 9660 是 High Sierra 文件系统的官方标准版本。HSFS 文件系统在 CD-ROM 上使用，它是只读文件系统。Solaris HSFS 支持 ISO 9660 的 Rock Ridge 扩展。这些扩展存在于 CD-ROM 上时可提供所有 UFS 文件系统功能和文件类型，但可写性和硬链接除外。
PCFS	PC 文件系统，该系统可用于对 DOS 格式磁盘上（针对基于 DOS 的个人计算机上写入）的数据和程序进行读写访问。
UDF	通用磁盘格式 (Universal Disk Format, UDF) 文件系统，这是使用称为 DVD（Digital Versatile Disc 或 Digital Video Disc，数字通用光盘或数字视频光盘）的光介质技术来存储信息的行业标准格式。

每种类型的基于磁盘的文件系统通常都与以下特定的介质设备关联，如下所示：

- UFS 与硬盘
- HSFS 与 CD-ROM
- PCFS 与软盘
- UDF 与 DVD

但是，这些关联并不具有限制性。例如，可以在 CD-ROM 和软盘上创建 UFS 文件系统。

通用磁盘格式 (Universal Disk Format, UDF) 文件系统

UDF 文件系统是用于在 DVD（Digital Versatile Disc 或 Digital Video Disc，数字通用光盘或数字视频光盘）光介质上存储信息的行业标准格式。

UDF 文件系统是作为可动态装入的 32 位和 64 位模块提供的，并且使用系统管理实用程序同时在 SPARC 和 x86 平台上创建、挂载和检查该文件系统。Solaris UDF 文件系统适用于支持的 ATAPI 和 SCSI DVD 驱动器、CD-ROM 设备以及磁盘和软盘驱动器。此外，Solaris UDF 文件系统完全符合 UDF 1.50 规范。

UDF 文件系统可提供以下功能：

- 可以访问行业标准的 CD-ROM 和 DVD-ROM 介质，前提是这些介质包含 UDF 文件系统
- 跨平台和操作系统灵活交换信息
- 使用基于 UDF 格式的 DVD 视频规范，实现具有丰富的广播质量视频、高质量声音和交互性的新应用程序的机制

UDF 文件系统中不包括以下功能：

- 对使用连续一次写入记录方式和增量记录方式的一次写入介质（CD-RW 和 DVD-RAM）的支持
- 不是 UDF 1.50 规范一部分的 UFS 组件，如配额、ACL、事务日志记录、文件系统锁定和文件系统线程

UDF 文件系统需要以下内容：

- 最低 Solaris 7 11/99 发行版
- 支持的 SPARC 或 x86 平台
- 支持的 CD-ROM 或 DVD-ROM 设备

Solaris UDF 文件系统实现提供了以下内容：

- 对行业标准读/写 UDF 版本 1.50 的支持
- 完全国际化的文件系统实用程序

基于网络的文件系统

可以从网络对**基于网络的文件系统**进行访问。通常，基于网络的文件系统驻留在一个系统上（通常是服务器），并由其他系统通过网络进行访问。

通过 NFS，可以管理分布式**资源**（文件或目录），方法是将其从服务器导出，然后在单独的客户机上进行挂载。有关更多信息，请参见第 357 页中的“NFS 环境”。

虚拟文件系统

虚拟文件系统是基于内存的文件系统，该文件系统提供对特殊内核信息和工具的访问。大多数虚拟文件系统不占用文件系统磁盘空间。但是，CacheFS 文件系统使用磁盘

上的文件系统以包含高速缓存。另外，一些虚拟文件系统（如临时文件系统 (temporary file system, TMPFS)）还会使用磁盘上的交换空间。

CacheFS 文件系统

CacheFS™ 文件系统可以用于提高远程文件系统或慢速设备（如 CD-ROM 驱动器）的性能。高速缓存文件系统时，会将从远程文件系统或 CD-ROM 中读取的数据存储在本地系统上的高速缓存中。

如果要提高 NFS 或 CD-ROM 文件系统的性能和可伸缩性，应该使用 CacheFS 文件系统。CacheFS 软件是一种通用的文件系统高速缓存机制，该软件通过减少服务器和网络的负载来提高 NFS 服务器的性能和可伸缩性。

由于 CacheFS 软件设计为分层文件系统，因此该软件可将一个文件系统高速缓存到另一个文件系统上。在 NFS 环境中，CacheFS 软件可增加客户机与服务器数目的比例，减少服务器和网络负载，并提高客户机在点对点协议 (Point-to-Point Protocol, PPP) 等慢速链路上的性能。另外，也可以将 CacheFS 文件系统与 AutoFS 服务结合使用，以帮助提高性能和可伸缩性。

有关 CacheFS 文件的详细信息，请参见第 20 章。

临时文件系统

临时文件系统 (temporary file system, TMPFS) 使用本地内存进行文件系统的读取和写入。通常，使用内存进行文件系统的读取和写入比使用 UFS 文件系统要快得多。使用 TMPFS 文件系统可以节省利用本地磁盘或网络读取和写入临时文件的成本，从而提高系统性能。例如，在编译程序时即会创建临时文件。OS 在处理这些文件的同时会产生大量的磁盘活动或网络活动。使用 TMPFS 保存这些临时文件可以显著加快对其进行创建、处理和删除的速度。

TMPFS 文件系统文件不是永久性的。在取消挂载文件系统以及关闭或重新引导系统时，将删除这些文件。

TMPFS 是 Solaris OS 中 /tmp 目录的缺省文件系统类型。可以将文件复制或移动到 /tmp 目录中或从中复制或移动文件，就像在 UFS 文件系统中进行的操作一样。

TMPFS 文件系统使用交换空间作为临时后备存储。如果包含 TMPFS 文件系统的系统没有充足的交换空间，则可能会出现以下两种问题：

- TMPFS 文件系统的空间可能不足，就像常规文件系统一样。
- 由于 TMPFS 会分配交换空间以保存文件数据（如有必要），因此一些程序可能因交换空间不足而无法执行。

有关创建 TMPFS 文件系统的信息，请参见第 18 章。有关增加交换空间的信息，请参见第 21 章。

回送文件系统

使用回送文件系统 (loopback file system, LOFS)，可以创建新的虚拟文件系统，以便使用替代的路径名访问文件。例如，可以在 `/tmp/newroot` 上创建根 (`/`) 目录的回送挂载。此回送挂载使整个文件系统分层结构看起来就像在 `/tmp/newroot` 下对其进行复制一样，其中包括从 NFS 服务器挂载的任何文件系统。路径名以根 (`/`) 目录或 `/tmp/newroot` 开头的所有文件都将是可访问的。

有关如何创建 LOFS 文件系统的信息，请参见第 18 章。

进程文件系统

进程文件系统 (process file system, PROCFS) 驻留在内存中，并且包含 `/proc` 目录中按进程号排序的活动进程的列表。`/proc` 目录中的信息由 `ps` 等命令使用。调试器和其他开发工具也可以使用文件系统调用来访问进程的地址空间。



注意 – 请勿删除 `/proc` 目录中的文件。从 `/proc` 目录中删除进程不会中止这些进程。`/proc` 文件不占用磁盘空间，因此不必从此目录中删除文件。

`/proc` 目录无需管理。

其他虚拟文件系统

以下列出的其他类型的虚拟文件系统用于参考。这些文件系统无需管理。

虚拟文件系统	说明
CTFS	CTFS (contract file system, 合同文件系统) 是用于创建、控制和查看合同的界面。合同可通过提供更丰富的错误报告，以及采用延迟资源删除时间的方法 (可选)，增强进程与其所依赖的系统资源之间的关系。 服务管理工具 (service management facility, SMF) 使用进程合同 (一种合同类型) 来跟踪构成服务的进程，以便可以将多进程服务某部分中出现的故障标识为该服务的故障。
FIFOFS (先入先出)	为进程提供对数据的公共访问的指定管道文件
EDFS (文件描述符)	使用文件描述符提供用于打开文件的显式名称
MNTFS	为本地系统提供对已挂载文件系统表的只读访问
NAMEFS	主要由 STREAMS 用来在文件顶部动态挂载文件描述符

虚拟文件系统	说明
OBJFS	OBJFS（对象）文件系统可描述当前由内核装入的所有模块的状态。此文件系统由调试器用来访问有关内核符号的信息，而不必直接访问内核。
SPECFS（特殊）	提供对字符特殊设备和块设备的访问
SWAPFS	由内核使用以进行交换

扩展的文件属性

UFS、NFS 和 TMPFS 文件系统均已增强，包括扩展的文件属性。通过扩展的文件属性，应用程序开发者可以将特定属性与文件关联。例如，开发用于管理窗口系统的应用程序的开发者可能会选择将显示图标与文件关联。扩展的文件属性在逻辑上表示为与目标文件关联的隐藏目录中的文件。

可以使用 `runat` 命令在扩展属性名称空间中添加属性和执行 `shell` 命令。此名称空间是与指定文件关联的隐藏属性目录。

要使用 `runat` 命令向文件添加属性，必须首先创建属性文件。

```
$ runat filea cp /tmp/attrdata attr.1
```

然后，使用 `runat` 命令列出文件的属性。

```
$ runat filea ls -l
```

有关更多信息，请参见 `runat(1)` 手册页。

许多 Solaris 文件系统命令已经修改，可以通过提供可识别属性的选项来支持文件系统属性。使用此选项可查询、复制或查找文件属性。有关更多信息，请参见每个文件系统命令的特定手册页。

交换空间

Solaris OS 将一些磁盘片用于临时存储而不是用于文件系统。这些片称为**交换片**或**交换空间**。如果系统没有足够的物理内存来处理当前进程，则可将交换空间用于虚拟内存存储区域。

由于许多应用程序依赖于交换空间，因此您应该知道如何规划、监视和添加更多交换空间（如果需要）。有关交换空间的概述和添加交换空间的说明，请参见[第 21 章](#)。

用于文件系统管理的命令

用于文件系统管理的大多数命令同时包含通用组成部分和文件系统特定的组成部分。应尽可能使用通用命令，这些命令可调用文件系统特定的组成部分。下表列出了用于文件系统管理的通用命令。这些命令位于 `/usr/sbin` 目录中。

表 17-1 用于文件系统管理的通用命令

命令	说明	手册页
<code>clri</code>	清除 inode	<code>clri(1M)</code>
<code>df</code>	报告空闲磁盘块和文件的数量	<code>df(1M)</code>
<code>ff</code>	列出文件系统的文件名和统计信息	<code>ff(1M)</code>
<code>fsck</code>	检查文件系统的完整性并修复找到的任何损坏	<code>fsck(1M)</code>
<code>fsdb</code>	调试文件系统	<code>fsdb(1M)</code>
<code>fstyp</code>	确定文件系统的类型	<code>fstyp(1M)</code>
<code>labelit</code>	将文件系统复制到磁带（仅供 <code>volcopy</code> 命令使用）时，列出或提供文件系统的标签	<code>labelit(1M)</code>
<code>mkfs</code>	创建新文件系统	<code>mkfs(1M)</code>
<code>mount</code>	挂载本地文件系统和远程文件系统	<code>mount(1M)</code>
<code>mountall</code>	挂载虚拟文件系统表 (<code>/etc/vfstab</code>) 中指定的所有文件系统	<code>mountall(1M)</code>
<code>ncheck</code>	生成路径名及其 inode 编号的列表	<code>ncheck(1M)</code>
<code>umount</code>	取消挂载本地文件系统和远程文件系统	<code>mount(1M)</code>
<code>umountall</code>	取消挂载虚拟文件系统表 (<code>/etc/vfstab</code>) 中指定的所有文件系统	<code>mountall(1M)</code>
<code>volcopy</code>	创建文件系统的映像副本	<code>volcopy(1M)</code>

文件系统命令如何确定文件系统的类型

通用文件系统命令按以下顺序确定文件系统的类型：

1. 通过 `-F` 选项（如果提供）。
2. 通过将特殊设备与 `/etc/vfstab` 文件中的项匹配（如果提供了 *special* 设备）。例如，`fsck` 会首先根据 `fsck device` 字段查找匹配项。如果未找到匹配项，则该命令会检查 *special* 设备字段。
3. 通过使用本地文件系统的 `/etc/default/fs` 文件和远程文件系统的 `/etc/dfs/fstypes` 文件中指定的缺省值。

通用和特定的文件系统命令的手册页

在《man pages section 1M: System Administration Commands》中可找到通用命令和特定命令的手册页。通用文件系统命令的手册页仅提供有关通用命令选项的信息。特定文件系统命令的手册页包含有关该文件系统的选项的信息。要查看特定文件系统的手册页，请将下划线和文件系统类型的缩写附加到通用命令名称的后面。例如，要查看有关挂载 UFS 文件系统的特定手册页，请键入以下内容：

```
$ man mount_ufs
```

缺省的 Solaris 文件系统

Solaris UFS 文件系统是一个分层文件系统，从根目录 (/) 开始，向下延伸出许多目录。在 Solaris 安装过程中，可以安装一组缺省目录，并使用一组约定将类似的文件类型组合在一起。

有关 Solaris 文件系统和目录的内容说明，请参见 `filesystem(5)`。

下表汇总了缺省的 Solaris 文件系统。

表 17-2 缺省的 Solaris 文件系统

文件系统或目录	文件系统类型	说明
根目录 (/)	UFS	分层文件树的顶层。根 (/) 目录包含对系统操作至关重要的目录和文件，如内核、设备驱动程序和用于引导系统的程序。根 (/) 目录还包含挂载点目录，可以在这些目录中将本地文件系统和远程文件系统附加到文件树。
/usr	UFS	可以与其他用户共享的系统文件和目录。仅在某些类型的系统上运行的文件（例如 SPARC 可执行文件）包含在 /usr 文件系统中。可以在所有类型的系统上使用的文件（如手册页）包含在 /usr/share 目录中。
/export/home 或 /home	NFS、UFS	用户起始目录（用于存储用户的工作文件）的挂载点。缺省情况下，/home 目录是自动挂载的文件系统。在独立系统上，/home 目录可能是本地磁盘片上的 UFS 文件系统。
/var	UFS	可能随本地系统的生命周期而更改或增长的系统文件和目录。这些系统文件和目录包括系统日志、vi 和 ex 备份文件以及 uucp 文件。
/opt	NFS、UFS	第三方软件的可选挂载点。在一些系统上，/opt 目录可能是本地磁盘片上的 UFS 文件系统。
/tmp	TMPFS	临时文件，每次引导系统或取消挂载 /tmp 将会删除这些文件。

表 17-2 缺省的 Solaris 文件系统 (续)

文件系统或目录	文件系统类型	说明
/proc	PROCFS	按进程号排序的活动进程的列表。
/etc/mnttab	MNTFS	一种虚拟文件系统，可为本地系统提供对已挂载文件系统表的只读访问。
/var/run	TMPFS	基于内存的文件系统，用于存储引导系统后不需要的临时文件。
/system/contract	CTFS	一种虚拟文件系统，用于维护合同信息。
/system/object	OBJFS	一种虚拟文件系统，由调试器用来访问有关内核符号的信息，而不必直接访问内核。

运行系统时需要使用根 (/) 文件系统和 /usr 文件系统。/usr 文件系统中的一些最基本的命令（如 mount）也包含在根 (/) 文件系统中。因此，在系统引导期间（或处于单用户模式下）未挂载 /usr 时，也可以使用这些命令。有关根 (/) 文件系统和 /usr 文件系统的缺省目录的更多详细信息，请参见第 23 章。

UFS 文件系统

UFS 是 Solaris OS 中缺省的基于磁盘的文件系统。管理基于磁盘的文件系统时，大多数情况下管理的是 UFS 文件系统。UFS 可提供以下功能。

UFS 功能	说明
扩展的基本类型 (extended fundamental type, EFT)	提供 32 位用户 ID (user ID, UID)、组 ID (group ID, GID) 和设备编号。
大文件系统	在最大大小可以为 16 TB 的文件系统中，允许大小约为 1 TB 的文件。可以在具有 EFI 磁盘标号的磁盘上创建多 TB UFS 文件系统。
日志记录	UFS 日志记录会将组成一个完整 UFS 操作的多个元数据更改打包成一个事务。事务集记录在盘上日志中，然后会应用于实际 UFS 文件系统的元数据。
多 TB 文件系统	通过多 TB 文件系统，可以创建最大可用空间约为 16 TB 的 UFS 文件系统，其中会减去约 1% 的开销。
状态标志	显示文件系统的状态：clean、stable、active、logging 或 unknown。这些标志可避免不必要的文件系统检查。如果文件系统的状态为“clean”、“stable”或“logging”，则不运行文件系统检查。

有关 UFS 文件系统结构的详细信息，请参见第 23 章。

规划 UFS 文件系统

对文件系统进行布局时，需要考虑可能相冲突的需求。以下是一些建议：

- 请尽可能在不同的 I/O 系统和磁盘驱动器之间均匀地分布工作负荷。请将 `/export/home` 文件系统和交换空间均匀地分布在磁盘之间。
- 请将项目的各部分或组的各成员放在同一文件系统中。
- 请在每个磁盘上使用尽可能少的文件系统。在系统（或引导）磁盘上，应该具有三个文件系统：根（`/`）、`/usr` 和交换空间。在其他磁盘上，请创建一个或最多两个文件系统，其中一个文件系统最好是附加的交换空间。与许多过度拥挤的小文件系统相比，空间较大的少量文件系统产生的文件段较少。由于 `ufsdump` 命令可以处理多个卷，因此采用较高容量的磁带机可以更轻松地备份较大的文件系统。
- 如果某些用户一直创建非常小的文件，请考虑创建包含更多 `inode` 的单独文件系统。但是，大多数站点都无需在同一文件系统中保存相似类型的用户文件。

有关缺省文件系统参数以及创建新 UFS 文件系统的过程的信息，请参见第 18 章。

UFS 日志记录

UFS 日志记录会将组成一个完整 UFS 操作的多个元数据更改打包成一个事务。事务集记录在盘上日志中，然后会应用于实际文件系统的元数据。

重新引导时，系统会废弃未完成的事务，但是会对已完成的操作应用事务。文件系统将保持一致，因为仅应用了已完成的事务。即使在系统崩溃时，也仍会保持此一致性。系统崩溃可能会中断系统调用，并导致 UFS 文件系统出现不一致。

UFS 日志记录功能有两个好处：

- 如果文件系统已经通过事务日志达到一致，则在系统崩溃或异常关机后可能不必运行 `fsck` 命令。有关异常关机的更多信息，请参见第 433 页中的“[fsck 命令检查和尝试修复的内容](#)”。
- 从 Solaris 9 12/02 发行版开始，UFS 日志记录的性能已经提高甚至超过了无日志记录功能的文件系统的性能级别。这一改进之所以能够实现是由于启用日志记录功能的文件系统可以将对相同数据的多重更新转换为单一更新。因此，减少了磁盘操作所需的开销。

UFS 事务日志具有以下特征：

- 从文件系统上的空闲块分配而来
- 对于每 1 GB 的文件系统，其大小约为 1 MB，最大为 64 MB
- 填满时会不断刷新
- 取消挂载文件系统或使用任何 `lockfs` 命令之后也会刷新。

所有 UFS 文件系统均缺省启用 UFS 日志记录。

如果需要禁用 UFS 日志记录，请在 `/etc/vfstab` 文件中或手动挂载文件系统时，向文件系统的项添加 `nologging` 选项。

如果需要启用 UFS 日志记录，请在 `/etc/vfstab` 文件中或手动挂载文件系统时，在 `mount` 命令中指定 `-o logging` 选项。可以在包括根 (`/`) 文件系统的任何 UFS 文件系统上启用日志记录。另外，`fsdb` 命令还包含支持 UFS 日志记录的新调试命令。

在一些操作系统中，启用了日志记录的文件系统称为**日记记录**文件系统。

UFS 快照

可以使用 `fssnap` 命令创建文件系统的只读快照。**快照**是旨在用于备份操作的文件系统临时映像。

有关更多信息，请参见第 26 章。

UFS 直接输入/输出 (Input/Output, I/O)

直接 I/O 旨在用于改进批量 I/O 操作。批量 I/O 操作使用大的缓冲区来传输大文件（大于 256 KB）。

使用 UFS 直接 I/O 可能会使执行各自内部缓冲的应用程序（如数据库引擎）受益。从 Solaris 8 1/01 发行版开始，UFS 直接 I/O 已经增强，允许在访问原始设备时发生的同一种类 I/O 并行。现在，可以从文件系统命名和灵活性中受益，并且仅会降低很少性能。请咨询数据库供应商，以了解它是否可以在其产品配置选项中启用 UFS 直接 I/O。

通过在 `mount` 命令中使用 `forcedirectio` 选项，也可以在文件系统上启用直接 I/O。仅当文件系统传输大量连续数据时，启用直接 I/O 才会提高性能。

使用此选项挂载文件系统时，将直接在用户的地址空间和磁盘之间传输数据。如果未对文件系统启用强制直接 I/O，则在用户的地址空间和磁盘之间传输的数据会首先在内核地址空间中进行缓冲。

缺省行为是不对 UFS 文件系统启用强制直接 I/O。有关更多信息，请参见 `mount_ufs(1M)`。

挂载和取消挂载文件系统

需要首先挂载文件系统，然后才能访问该文件系统上的文件。挂载文件系统时，需要将该文件系统附加到目录（**挂载点**）并使其对系统可用。始终会挂载根 (`/`) 文件系统。可以将其他任何文件系统与根 (`/`) 文件系统连接或断开。

挂载文件系统时，只要挂载了文件系统，基础挂载点目录中的任何文件或目录都将不可用。挂载过程不会永久影响这些文件。取消挂载文件系统时，这些文件即重新变为可用。但是，由于通常不希望隐藏现有文件，因此挂载目录经常为空。

例如，下图显示了一个从根 (/) 文件系统以及 `sbin`、`etc` 和 `opt` 子目录开始的本地文件系统。

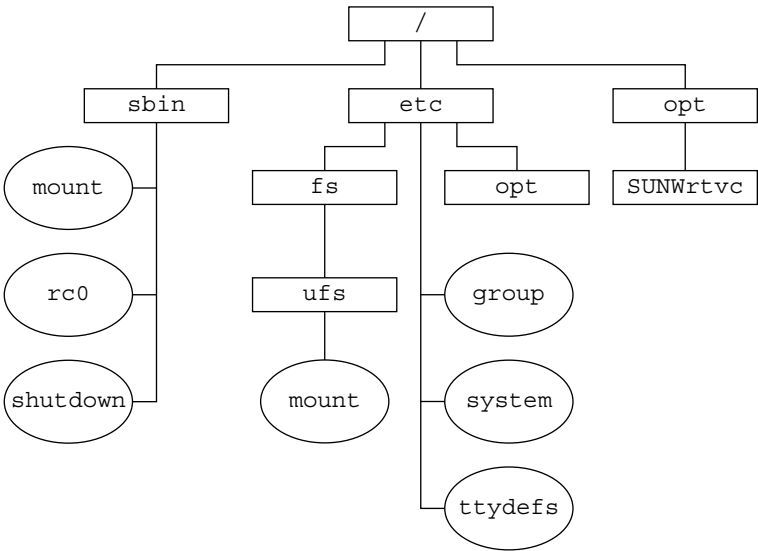
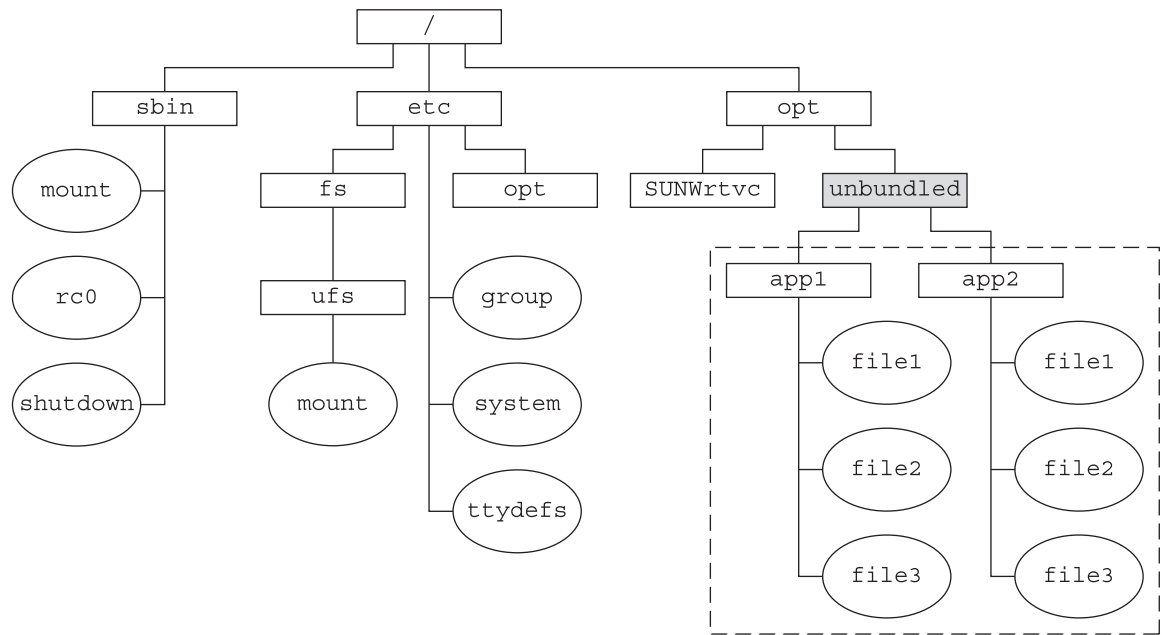


图 17-1 根 (/) 文件系统样例

要从包含一组非随附产品的 `/opt` 文件系统访问本地文件系统，必须执行以下操作：

- 首先，必须创建一个目录（例如 `/opt/unbundled`），将其用作要挂载的文件系统的挂载点。
- 创建挂载点后，可以使用 `mount` 命令挂载文件系统。此命令使 `/opt/unbundled` 中的所有文件和目录可用，如下图所示。



■ 挂载点
⌈⌋ 文件系统
图 17-2 挂载文件系统

有关如何挂载文件系统的逐步说明，请参见第 19 章。

已挂载文件系统表

每次挂载或取消挂载文件系统时，都会使用当前挂载的文件系统的列表修改 `/etc/mnttab`（挂载表）文件。可以使用 `cat` 或 `more` 命令显示此文件的内容。但是，不能编辑此文件。以下是 `/etc/mnttab` 文件的示例：

```
$ more /etc/mnttab

/dev/dsk/c0t0d0s0      /          ufs      rw,intr,largefiles,logging,xattr,onerror
=panic,dev=2200008    1093882623

/devices              /devices   devfs    dev=4340000    1093882603

ctfs                  /system/contract  ctfs     dev=4380001    1093882603
```

```
proc      /proc      proc      dev=43c0000      1093882603

mnttab    /etc/mnttab  mntfs    dev=4400001      1093882603

swap      /etc/svc/volatile      tmpfs    xattr,dev=4440001      1093882603

/dev/dsk/c0t0d0s6      /usr      ufs      rw,intr,largefiles,logging,xattr,onerror
=panic,dev=220000e      1093882623

objfs     /system/object  objfs    dev=44c0001      1094150403

fd        /dev/fd fd      rw,dev=45c0001  1093882624

swap      /var/run      tmpfs    xattr,dev=4440002      1093882625

swap      /tmp      tmpfs    xattr,dev=4440003      1093882625

/dev/dsk/c0t0d0s7      /export/home      ufs      rw,intr,largefiles,logging,xattr
,onerror=panic,dev=220000f      1093882637

$
```

虚拟文件系统表

每次要访问文件系统时都将其手动挂载会非常耗时并且容易出错。为避免这些问题，虚拟文件系统表（`/etc/vfstab` 文件）提供了文件系统的列表以及如何将其挂载的信息。

`/etc/vfstab` 文件提供了以下两项重要功能：

- 可以指定文件系统在系统引导时自动挂载。
- 可以通过仅使用挂载点名称来挂载文件系统。`/etc/vfstab` 文件包含挂载点与实际设备片名称之间的映射。

安装系统时会创建缺省的 `/etc/vfstab` 文件，具体取决于在安装过程中进行的选择。但是，只要需要，即可在系统上编辑 `/etc/vfstab` 文件。要添加项，需要指定如下信息：

- 文件系统所驻留的设备
- 文件系统的挂载点
- 文件系统类型
- 是否希望文件系统在系统引导时自动挂载（通过使用 `mountall` 命令）
- 任何挂载选项

以下是 `/etc/vfstab` 文件的示例。注释行以 `#` 开头。本示例显示了具有两个磁盘（`c0t0d0` 和 `c0t3d0`）的系统的 `/etc/vfstab` 文件。

```
$ more /etc/vfstab

#device          device          mount          FS      fsck    mount    mount
#to mount        to fsck          point          type    pass    at boot  options
#
fd               -               /dev/fd        fd      -       no       -
/proc            -               /proc          proc    -       no       -
/dev/dsk/c0t0d0s1 -               -              swap    -       no       -
/dev/dsk/c0t0d0s0 /dev/rdisk/c0t0d0s0 /              ufs     1       no       -
/dev/dsk/c0t0d0s6 /dev/rdisk/c0t0d0s6 /usr           ufs     1       no       -
/dev/dsk/c0t0d0s7 /dev/rdisk/c0t0d0s7 /export/home   ufs     2       yes      -
/dev/dsk/c0t0d0s5 /dev/rdisk/c0t0d0s5 /opt           ufs     2       yes      -
/devices         -               /devices       devfs   -       no       -
ctfs             -               /system/contract ctfs    -       no       -
objfs            -               /system/object  objfs   -       no       -
swap             -               /tmp           tmpfs   -       yes      -

$
```

在本示例中，`/dev/dsk/c0t0d0s7` 片上 `/export/home` 的 UFS 文件系统项在系统引导时会自动挂载到 `/test` 挂载点上。请注意，对于根 (`/`) 和 `/usr`，`mount at boot` 字段值被指定为 `no`。这些文件系统是在运行 `mountall` 命令之前，由内核在顺序引导过程中挂载的。

有关每个 `/etc/vfstab` 字段的说明以及如何编辑和使用该文件的信息，请参见 [第 19 章](#)。

NFS 环境

NFS 是一种分布式文件系统服务，可以用于将一个系统（通常为服务器）中的资源（文件或目录）与网络中的其他系统共享。例如，您可能希望与其他系统上的用户共享第三方应用程序或源文件。

使用 NFS 时资源的实际物理位置与用户无关。使用 NFS 服务可以在一个系统的磁盘上放置一个副本并使所有其他系统通过网络对其进行访问，而不是在每个系统上都放置常用文件的副本。使用 NFS 时，远程文件和本地文件实际没有分别。

有关更多信息，请参见《系统管理指南：网络服务》中的第4章“管理网络文件系统（概述）”。

如果系统包含要在网络中共享的资源，则该系统将成为 NFS 服务器。服务器保存了当前共享的资源及其访问限制（如读/写访问或只读访问）的列表。

共享某个资源时，即可使该资源供远程系统进行挂载。

可以使用以下方法共享资源：

- 使用 `share` 或 `shareall` 命令
- 向 `/etc/dfs/dfstab`（分布式文件系统表）文件添加一个项并重新引导系统

有关如何共享资源的信息，请参见第19章。有关 NFS 的完整说明，请参见《系统管理指南：网络服务》中的第4章“管理网络文件系统（概述）”。

自动挂载或 AutoFS

可以使用称为**自动挂载**（或 *AutoFS*）的客户端服务来挂载 NFS 文件系统资源。每次访问 NFS 资源时，AutoFS 都允许系统对其进行自动挂载和取消挂载。只要仍在目录中并且使用该目录中的文件，资源就会保持挂载状态。如果在某个时间段内未访问资源，则会自动取消挂载该资源。

AutoFS 提供了以下功能：

- 系统引导时无需挂载 NFS 资源，从而节省引导时间。
- 用户无需知道用于挂载和取消挂载 NFS 资源的超级用户口令。
- 可能会减少网络通信流量，因为仅在使用 NFS 资源时才会将其挂载。

AutoFS 服务通过 `automount` 实用程序进行初始化，该实用程序在引导系统时自动运行。`automountd` 守护进程会持续不断地运行，并且负责按需挂载和取消挂载 NFS 文件系统。缺省情况下，`/home` 文件系统由 `automount` 守护进程挂载。

使用 AutoFS，可以指定多台服务器提供同一文件系统。这样，如果其中一台服务器关闭，则 AutoFS 可以尝试从其他计算机挂载该文件系统。

有关如何设置和管理 AutoFS 的完整信息，请参见《系统管理指南：IP 服务》。

确定文件系统的类型

可以使用以下方法之一确定文件系统的类型：

- 虚拟文件系统表（`/etc/vfstab` 文件）中的 `FS type` 字段
- 本地文件系统的 `/etc/default/fs` 文件
- NFS 文件系统的 `/etc/dfs/fstypes` 文件

如何确定文件系统的类型

无论是否挂载了文件系统，此过程都适用。

请使用 `grep` 命令确定文件系统的类型。

```
$ grep mount-point fs-table
```

mount-point 指定希望知道其文件系统类型的文件系统的挂载点名称。例如 `/var` 目录。

fs-table 指定要在其中搜索文件系统类型的文件系统表的绝对路径。如果已挂载该文件系统，则 *fs-table* 应该是 `/etc/mnttab`。如果未挂载该文件系统，则 *fs-table* 应该是 `/etc/vfstab`。

此时会显示挂载点的信息。

注 – 如果已知磁盘片的原始设备名称，则可以使用 `fstyp` 命令确定文件系统的类型（如果磁盘片包含文件系统）。有关更多信息，请参见 `fstyp(1M)`。

示例 17-1 确定文件系统的类型

以下示例使用 `/etc/vfstab` 文件确定 `/export` 文件系统的文件系统类型。

```
$ grep /export /etc/vfstab
```

```
/dev/dsk/c0t3d0s6    /dev/rdisk/c0t3d0s6  /export ufs      2          yes      -
```

```
$
```

以下示例使用 `/etc/mnttab` 文件确定当前挂载的软盘的文件系统类型。软盘以前通过 `vold` 进行挂载。

```
$ grep /floppy /etc/mnttab
```

```
/vol/dev/diskette0/unnamed_floppy  /floppy/unnamed_floppy  pcfs rw,
```

```
nohidden,nofoldcase,dev=16c0009      89103376
```

```
$
```


创建 UFS、TMPFS 和 LOFS 文件系统（任务）

本章介绍如何创建 UFS 文件系统、临时 (TMPFS) 文件系统和回送 (LOFS) 文件系统。对于 UFS 文件系统，本章介绍如何使用 `newfs` 命令创建文件系统。由于 TMPFS 和 LOFS 是虚拟的文件系统，因此您实际上是通过挂载这两种文件系统来对其进行“访问”的。

以下是本章中的逐步说明列表。

- 第 362 页中的 “如何创建 UFS 文件系统”
- 第 363 页中的 “如何创建多 TB UFS 文件系统”
- 第 365 页中的 “如何扩展多 TB UFS 文件系统”
- 第 367 页中的 “如何将 UFS 文件系统扩展为多 TB UFS 文件系统”
- 第 370 页中的 “如何创建和挂载 TMPFS 文件系统”
- 第 371 页中的 “如何创建和挂载 LOFS 文件系统”

注 - 有关如何在可移除介质上创建 UFS 和 DOS 文件系统的说明，请参见第 1 章。

创建 UFS 文件系统

必须格式化磁盘并将其分片之后，才能在其中创建 UFS 文件系统。**磁盘片**是磁盘的物理子集，由单一范围的连续块组成。片可以用作提供交换空间等的原始设备，或者用于保存基于磁盘的文件系统。有关格式化磁盘和将磁盘分片的完整信息，请参见第 11 章。

磁盘和存储管理产品（如 Solaris™ 卷管理器）用于创建更复杂的**卷**。卷的范围不局限于单个片或单个磁盘。有关使用卷的更多信息，请参见《Solaris Volume Manager 管理指南》。

注 - Solaris 设备名称使用术语“片”（设备名称中使用字母 s）表示片编号。片也称为分区。

由于 Solaris OS 会在安装过程中自动创建 UFS 文件系统，因此只是偶尔需要创建这些文件系统。如果希望执行以下操作，则需要创建（或重新创建）UFS 文件系统：

- 添加或更换磁盘
- 更改磁盘的现有分区结构
- 完全恢复文件系统

使用 `newfs` 命令是创建 UFS 文件系统的标准方法。`newfs` 命令是 `mkfs` 命令的便利前端，前者实际上用于创建新文件系统。`newfs` 命令可从将包含新文件系统的磁盘的标号中读取参数缺省值，如每个柱面的磁道数和每个磁道的扇区数。所选的选项会传递给 `mkfs` 命令以生成文件系统。

有关 `newfs` 命令所用的缺省参数的信息，请参见 `newfs(1M)`。

▼ 如何创建 UFS 文件系统

开始之前 请确保已满足以下先决条件：

- 磁盘必须已格式化并分片。
- 如果要重新创建现有的 UFS 文件系统，请取消挂载该文件系统。
- 需要知道将包含文件系统的片的设备名称。

有关查找磁盘和磁盘片编号的信息，请参见第 12 章。

有关格式化磁盘和将磁盘分片的信息，请参见第 11 章。

1 您必须是超级用户或承担等效角色。

2 创建 UFS 文件系统。

```
# newfs [-N] [-b size] [-i bytes] /dev/rdisk/device-name
```

-N 显示 `newfs` 命令会传递给 `mkfs` 命令的参数，而不实际创建文件系统。此选项是用于测试 `newfs` 命令的一种好方法。

-b size 指定文件系统的块大小，每块为 4096 或 8192 字节。缺省值为 8192。

-i bytes 指定每个 inode 的字节数。缺省值根据磁盘大小而异。有关更多信息，请参见 `newfs(1M)`。

device-name 指定在其中创建新文件系统的磁盘设备的名称。

系统会请求进行确认。



注意—执行此步骤之前，请确保为片指定了正确的设备名称。如果指定错误的片，则在创建新文件系统时将删除其内容。此错误可能会导致系统出现紧急情况。

3 要验证是否创建了 UFS 文件系统，请检查新文件系统。

```
# fsck /dev/rdisk/device-name
```

其中 *device-name* 参数指定包含新文件系统的磁盘设备的名称。

`fsck` 命令可检查新文件系统的一致性，报告出现的任何问题，并在修复问题之前进行提示。有关 `fsck` 命令的更多信息，请参见[第 22 章](#)或 `fsck(1M)`。

示例 18-1 创建 UFS 文件系统

以下示例说明如何在 `/dev/rdisk/c0t1d0s7` 中创建 UFS 文件系统。

```
# newfs /dev/rdisk/c0t1d0s7
```

```
/dev/rdisk/c0t1d0s7: 725760 sectors in 720 cylinders of 14 tracks, 72 sectors
```

```
354.4MB in 45 cyl groups (16 c/g, 7.88MB/g, 3776 i/g)
```

```
super-block backups (for fsck -F ufs -o b=#) at:
```

```
32, 16240, 32448, 48656, 64864, 81072, 97280, 113488, 129696, 145904, 162112,
```

```
178320, 194528, 210736, 226944, 243152, 258080, 274288, 290496, 306704,
```

```
322912, 339120, 355328, 371536, 387744, 403952, 420160, 436368, 452576,
```

```
468784, 484992, 501200, 516128, 532336, 548544, 564752, 580960, 597168,
```

```
613376, 629584, 645792, 662000, 678208, 694416, 710624,
```

```
fsck /dev/rdisk/c0t1d0s7
```

```
#
```

更多信息 创建 UFS 文件系统后...

要挂载 UFS 文件系统并使其可用，请转到[第 19 章](#)。

▼ 如何创建多 TB UFS 文件系统

对多 TB UFS 文件系统的支持假定作为 Solaris 卷管理器、VxVM 卷或大于 1 TB 的物理磁盘提供的多 TB LUN 可用。

创建多 TB UFS 文件系统之前，请首先验证是否执行了以下任一操作：

- 使用 `format` 实用程序或 Solaris 安装实用程序创建了多 TB 磁盘分区
- 使用 Solaris 卷管理器设置了多 TB 卷

有关多 TB UFS 文件系统的更多信息，请参见第 340 页中的“64 位: 对多 TB UFS 文件系统的支持”。

- 1 成为超级用户。
- 2 在逻辑卷上创建多 TB UFS 文件系统。

例如，此命令可为 1.8 TB 的卷创建 UFS 文件系统：

```
# newfs /dev/md/rdisk/d99
```

```
newfs: construct a new file system /dev/md/rdisk/d99: (y/n)? y
```

```
/dev/md/rdisk/d99: 3859402752 sectors in 628158 cylinders of 48 tracks,  
128 sectors
```

```
1884474.0MB in 4393 cyl groups (143 c/g, 429.00MB/g, 448 i/g)
```

```
super-block backups (for fsck -F ufs -o b=#) at:
```

```
32, 878752, 1757472, 2636192, 3514912, 4393632, 5272352, 6151072, 702...
```

```
Initializing cylinder groups:
```

```
.....
```

```
super-block backups for last 10 cylinder groups at:
```

```
3850872736, 3851751456, 3852630176, 3853508896, 3854387616, 3855266336,
```

```
3856145056, 3857023776, 3857902496, 3858781216,
```

```
#
```

- 3 验证新创建的文件系统的完整性。

例如：

```
# fsck /dev/md/rdisk/d99
```

```
** /dev/md/rdisk/d99
```

```
** Last 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

2 files, 2 used, 241173122 free (0 frags, 241173122 blocks, 0.0%
fragmentation)

#
```

4 挂载新创建的文件系统并对其进行验证。

例如：

```
# mount /dev/md/dsk/d99 /bigdir

# df -h /bigdir

Filesystem              size  used  avail capacity  Mounted on
/dev/md/dsk/d99         1.8T   64M   1.8T     1%    /bigdir
```

▼ 如何扩展多TB UFS 文件系统

创建多TB UFS 文件系统后，可以使用 `growfs` 命令扩展该文件系统。例如，使用在前面的过程中为卷创建的文件系统，可以将其他磁盘添加到此卷。然后，扩展文件系统。

- 1 成为超级用户。
- 2 将其他磁盘添加到卷。

例如：

```
# metattach d99 c4t5d0s4

d99: component is attached

# metastat

d99: Concat/Stripe

      Size: 5145882624 blocks (2.4 TB)
```

Stripe 0:				
Device	Start Block	Dbase	Reloc	
c0t1d0s4	36864	Yes	Yes	
Stripe 1:				
Device	Start Block	Dbase	Reloc	
c3t7d0s4	0	No	Yes	
Stripe 2:				
Device	Start Block	Dbase	Reloc	
c1t1d0s4	0	No	Yes	
Stripe 3:				
Device	Start Block	Dbase	Reloc	
c4t5d0s4	0	No	Yes	

3 扩展文件系统。

```
例如：
# growfs -v /dev/md/rdisk/d99

/usr/lib/fs/ufs/mkfs -G /dev/md/rdisk/d99 5145882624

/dev/md/rdisk/d99: 5145882624 sectors in 837546 cylinders of 48 tracks,
128 sectors

2512638.0MB in 5857 cyl groups (143 c/g, 429.00MB/g, 448 i/g)

super-block backups (for fsck -F ufs -o b=#) at:

32, 878752, 1757472, 2636192, 3514912, 4393632, 5272352, 6151072, 702...

Initializing cylinder groups:

.....

super-block backups for last 10 cylinder groups at:

5137130400, 5138009120, 5138887840, 5139766560, 5140645280, 5141524000,
```

```
5142402720, 5143281440, 5144160160, 5145038880,  
#
```

- 4 挂载已扩展的文件系统并对其进行验证。
例如：

```
# mount /dev/md/dsk/d99 /bigdir  
  
# df -h /bigdir  
  
Filesystem              size  used  avail capacity  Mounted on  
/dev/md/dsk/d99         2.4T   64M   2.4T      1%    /bigdir
```

▼ 如何将 UFS 文件系统扩展为多TB UFS 文件系统

使用以下过程可将 UFS 文件系统的大小扩展为大于 1 TB。此过程假定最初使用 `newfs -T` 选项创建 UFS 文件系统。

- 1 成为超级用户。
2 确定当前磁盘或卷的大小。
例如，以下卷的大小为 800 GB：

```
# metastat d98  
  
d98: Concat/Stripe  
  
Size: 1677754368 blocks (800 GB)  
  
Stripe 0:  
  
Device      Start Block  Dbase  Reloc  
c0t1d0s4      0         No     Yes  
  
Stripe 1:  
  
Device      Start Block  Dbase  Reloc  
c3t7d0s4      0         No     Yes
```

3 将卷增至大于 1TB。

例如：

```
# metattach d98 c1t1d0s4
```

d98: component is attached

```
# metastat d98
```

d98: Concat/Stripe

Size: 2516631552 blocks (1.2 TB)

Stripe 0:

Device	Start Block	Dbase	Reloc
c0t1d0s4	0	No	Yes

Stripe 1:

Device	Start Block	Dbase	Reloc
c3t7d0s4	0	No	Yes

Stripe 2:

Device	Start Block	Dbase	Reloc
c1t1d0s4	0	No	Yes

4 将磁盘或卷的 UFS 文件系统扩展为大于 1TB。

例如：

```
growfs -v /dev/md/rdisk/d98
```

```
/usr/lib/fs/ufs/mkfs -G /dev/md/rdisk/d98 2516631552
```

/dev/md/rdisk/d98: 2516631552 sectors in 68268 cylinders of 144 tracks,
256 sectors

1228824.0MB in 2731 cyl groups (25 c/g, 450.00MB/g, 448 i/g)

super-block backups (for fsck -F ufs -o b=#) at:

32, 921888, 1843744, 2765600, 3687456, 4609312, 5531168, 6453024, 737...


```
8296736,  
  
Initializing cylinder groups:  
  
.....  
  
super-block backups for last 10 cylinder groups at:  
  
2507714848, 2508636704, 2509558560, 2510480416, 2511402272, 2512324128,  
  
2513245984, 2514167840, 2515089696, 2516011552,
```

5 挂载已扩展的文件系统并对其进行验证。

例如：

```
# mount /dev/md/dsk/d98 /datadir  
  
# df -h /datadir  
  
Filesystem              size  used  avail capacity  Mounted on  
/dev/md/dsk/d98         1.2T   64M   1.2T      1%    /datadir
```

对多TB UFS 文件系统问题进行疑难解答

使用以下错误消息和解决方案可以对多 TB UFS 文件系统问题进行疑难解答。

错误消息（与以下内容类似）：

```
mount: /dev/rdisk/c0t0d0s0 is not this fstype.
```

原因

尝试在运行早于 Solaris 10 发行版的 Solaris 发行版的系统上挂载大于 1 TB 的 UFS 文件系统。

解决方案

在运行 Solaris 10 或更高发行版的系统上挂载大于 1 TB 的 UFS 文件系统。

错误消息

```
"File system was not set up with the multi-terabyte format." "Its size  
  
cannot be increased to a terabyte or more."
```

原因

尝试扩展不是使用 newfs -T 命令创建的文件系统。

解决方案

1. 备份要扩展为大于 1 TB 的文件系统的数据。
2. 使用 `newfs` 命令重新创建文件系统，以创建多 TB 文件系统。
3. 将备份数据恢复到新创建的文件系统中。

创建临时文件系统 (Temporary File System, TMPFS)

临时文件系统 (*temporary file system, TMPFS*) 使用本地内存进行文件系统的读取和写入，这通常比在 UFS 文件系统中读取和写入快得多。TMPFS 文件系统通过节省利用本地磁盘或网络读取和写入临时文件的成本，可以提高系统性能。重新引导或取消挂载后，TMPFS 文件系统中的文件即不存在。

如果创建多个 TMPFS 文件系统，请注意这些文件系统都使用相同的系统资源。除非使用 `mount` 命令的 `-o size` 选项限制 TMPFS 的大小，否则在 TMPFS 文件系统下创建的文件将用尽可用于其他任何 TMPFS 文件系统的空间。

有关更多信息，请参见 `tmpfs(7FS)`。

▼ 如何创建和挂载 TMPFS 文件系统

- 1 成为超级用户或承担等效角色。
- 2 如有必要，请创建要作为 TMPFS 文件系统挂载的目录。

```
# mkdir /mount-point
```

其中 *mount-point* 是挂载 TMPFS 文件系统的目录。

- 3 挂载 TMPFS 文件系统。

```
# mount -F tmpfs [-o size=number] swap mount-point
```

`-o size=number` 指定 TMPFS 文件系统的大小限制（以 MB 为单位）。

mount-point 指定在其中挂载 TMPFS 文件系统的目录。

要将系统设置为在引导时自动挂载 TMPFS 文件系统，请参见[示例 18-3](#)。

- 4 验证是否已创建 TMPFS 文件系统。

```
# mount -v
```

示例 18-2 创建和挂载 TMPFS 文件系统

以下示例说明如何创建、挂载 TMPFS 文件系统 `/export/reports` 并将其大小限制为 50 MB。

```
# mkdir /export/reports

# chmod 777 /export/reports

# mount -F tmpfs -o size=50m swap /export/reports

# mount -v
```

示例 18-3 在引导时挂载 TMPFS 文件系统

通过添加 `/etc/vfstab` 项，可以将系统设置为在引导时自动挂载 TMPFS 文件系统。以下示例说明了 `/etc/vfstab` 文件中的某一项，该项在引导时将 `/export/test` 作为 TMPFS 文件系统挂载。由于未指定 `size=number` 选项，因此 `/export/test` 中 TMPFS 文件系统的大小仅受可用系统资源的限制。

```
swap - /export/test tmpfs - yes -
```

有关 `/etc/vfstab` 文件的更多信息，请参见第 376 页中的“`/etc/vfstab` 文件的字段说明”。

创建和挂载回送文件系统 (Loopback File System, LOFS)

LOFS 文件系统是一种虚拟文件系统，可提供现有文件系统的替代路径。将其他文件系统挂载到 LOFS 文件系统上时，原始文件系统不会更改。

有关更多信息，请参见 `lofs(7FS)`。

注 - 创建 LOFS 文件系统时请务必谨慎。由于 LOFS 文件系统是虚拟文件系统，因此极有可能混淆用户和应用程序。

▼ 如何创建和挂载 LOFS 文件系统

- 1 成为超级用户或承担等效角色。
- 2 如有必要，请创建要作为 LOFS 文件系统挂载的目录。

```
# mkdir loopback-directory
```
- 3 授予新创建的目录适当的权限和拥有权。

- 4 如有必要，请创建要在其中挂载 LOFS 文件系统的挂载点。

```
# mkdir /mount-point
```

- 5 挂载 LOFS 文件系统。

```
# mount -F lofs loopback-directory /mount-point
```

loopback-directory 指定要在回送挂载点上挂载的文件系统。

/mount-point 指定要在其中挂载 LOFS 文件系统的目录。

- 6 验证是否已挂载 LOFS 文件系统。

```
# mount -v
```

示例 18-4 创建和挂载 LOFS 文件系统

以下示例说明如何创建、挂载作为回送文件系统的 */new/dist* 目录并测试其中的新软件，而不必实际安装该软件。

```
# mkdir /tmp/newroot
```

```
# mount -F lofs /new/dist /tmp/newroot
```

```
# chroot /tmp/newroot newcommand
```

示例 18-5 在引导时挂载 LOFS 文件系统

通过向 */etc/vfstab* 文件的结尾添加一项，可以将系统设置为在引导时自动挂载 LOFS 文件系统。以下示例说明了 */etc/vfstab* 文件中的某一项，该项为 */tmp/newroot* 中的根 (/) 文件系统挂载 LOFS 文件系统。

```
/ - /tmp/newroot lofs - yes -
```

请确保回送项是 */etc/vfstab* 文件中的最后几项。否则，如果回送文件系统的 */etc/vfstab* 项位于要包括在该项中的文件系统的前面，则无法挂载回送文件系统。

另请参见 有关 */etc/vfstab* 文件的更多信息，请参见第 376 页中的“[/etc/vfstab 文件的字段说明](#)”。

挂载和取消挂载文件系统（任务）

本章介绍如何在 Solaris OS 中挂载和取消挂载文件系统。

以下是本章中的逐步说明列表。

- 第 377 页中的 “如何确定挂载的文件系统”
- 第 378 页中的 “如何向 `/etc/vfstab` 文件添加项”
- 第 380 页中的 “如何挂载文件系统（`/etc/vfstab` 文件）”
- 第 382 页中的 “如何挂载 UFS 文件系统（`mount` 命令）”
- 第 382 页中的 “如何挂载不包含大文件的 UFS 文件系统（`mount` 命令）”
- 第 384 页中的 “如何挂载 NFS 文件系统（`mount` 命令）”
- 第 384 页中的 “x86: 如何从硬盘挂载 PCFS (DOS) 文件系统（`mount` 命令）”
- 第 386 页中的 “如何验证文件系统是否已取消挂载”
- 第 386 页中的 “如何停止访问文件系统的所有进程”
- 第 387 页中的 “如何取消挂载文件系统”

挂载文件系统概述

创建文件系统后，需要使其对系统可用，以便您可以使用它。通过挂载文件系统可使该文件系统可用，这会将文件系统连接到指定挂载点上的系统目录树。始终会挂载根（`/`）文件系统。

下表提供了根据文件系统的使用方法对其进行挂载的指南。

所需挂载类型	建议挂载方法
无需频繁挂载的本地或远程文件系统	从命令行手动键入的 <code>mount</code> 命令。
需要频繁挂载的本地文件系统	<code>/etc/vfstab</code> 文件，用于在多用户状态下引导系统时自动挂载文件系统。

所需挂载类型	建议挂载方法
需要频繁挂载的远程文件系统，如起始目录	<div><div><div>■ /etc/vfstab 文件，用于在多用户状态下引导系统时自动挂载文件系统。</div><div>■ AutoFS，用于对其进行访问时自动挂载文件系统，或在转到其他目录时取消挂载文件系统。</div></div><div>要提高性能，还可以使用 CacheFS 文件系统高速缓存远程文件系统。</div></div>

如有必要，可以通过将包含文件系统的可移除介质插入到驱动器中并运行 `volcheck` 命令，挂载该介质。有关挂载可移除介质的更多信息，请参见第 1 章。

用于挂载和取消挂载文件系统的命令

下表列出了 `/usr/sbin` 目录中用于挂载和取消挂载文件系统的命令。

表 19-1 用于挂载和取消挂载文件系统的命令

命令	说明	手册页
<code>mount</code>	挂载文件和远程资源。	<code>mount(1M)</code>
<code>mountall</code>	挂载 <code>/etc/vfstab</code> 文件中指定的所有文件系统。当系统进入多用户模式时， <code>mountall</code> 命令会自动运行。	<code>mountall(1M)</code>
<code>umount</code>	取消挂载文件和远程资源。	<code>mount(1M)</code>
<code>umountall</code>	取消挂载 <code>/etc/vfstab</code> 文件中指定的所有文件系统。	<code>mountall(1M)</code>

在使用 `mount` 和 `mountall` 命令时，请牢记以下要点：

- `mount` 和 `mountall` 命令不能挂载具有明确不一致地方的读/写文件系统。如果从 `mount` 或 `mountall` 命令收到错误消息，则可能需要检查文件系统。有关如何检查文件系统的信息，请参见第 22 章。
- `umount` 和 `umountall` 命令不会取消挂载繁忙的文件系统。如果以下条件之一成立，则认为文件系统繁忙：
 - 用户正在访问文件系统上的文件或目录。
 - 程序打开了文件系统上的文件。
 - 文件系统被共享。
- 仅当从只读访问重新挂载为读写访问时，才可以使用 `remount` 选项。不能从读写访问重新挂载为只读访问。

常用挂载选项

下表介绍了可以使用 `mount -o` 选项指定的常用选项。如果指定多个选项，请使用逗号（无空格）将其分隔。例如，`-o ro,nosuid`。

有关每种文件系统类型的挂载选项的完整列表，请参阅特定的挂载手册页（例如 `mount_ufs(1M)`）。

表 19-2 常用的 -o 挂载选项

挂载选项	文件系统	说明
<code>bg fg</code>	NFS	如果第一个挂载尝试失败，请在后台 (<code>bg</code>) 或前台 (<code>fg</code>) 重试其他挂载。此选项对于非关键的 <code>vfstab</code> 项是安全的。缺省情况下使用 <code>fg</code> 。
<code>hard soft</code>	NFS	如果服务器没有响应，则指定过程。 <code>soft</code> 选项表示返回了错误。 <code>hard</code> 选项表示继续重试请求，直到服务器响应为止。缺省情况下使用 <code>hard</code> 。
<code>intr nointr</code>	NFS	指定在硬挂载的文件系统上等待响应时，是否将键盘中断传送到挂起的进程。缺省情况下使用 <code>intr</code> （允许中断）。
<code>largefiles nolargefiles</code>	UFS	可用于创建大于 2 GB 的文件。 <code>largefiles</code> 选项意味着使用此选项挂载的文件系统可能包含大于 2 GB 的文件。如果指定了 <code>nolargefiles</code> 选项，则无法在运行 Solaris 2.6 或兼容版本的系统上挂载文件系统。缺省情况下使用 <code>largefiles</code> 。
<code>logging nologging</code>	UFS	<p>用于启用或禁用文件系统的日志记录。UFS 日志记录是在将事务（构成完整 UFS 操作的更改）应用于 UFS 文件系统之前将事务存储到日志中的过程。日志记录有助于防止 UFS 文件系统变得不一致，这意味着可以跳过 <code>fsck</code>。跳过 <code>fsck</code> 可减少系统崩溃或异常关闭系统后重新引导系统的时间。</p> <p>日志是通过文件系统上的空闲块分配的，对于每 1 GB 文件系统其大小约为 1 MB，最大为 64 MB。缺省情况下使用 <code>logging</code>。</p>
<code>atime noatime</code>	UFS	限制文件的访问时间更新，但是如果更新与上次更改文件状态的时间或上次修改文件的时间的更新一致则除外。有关更多信息，请参见 <code>stat(2)</code> 。此选项可减少访问时间不重要的文件系统上的磁盘活动（例如，Usenet 新闻假脱机）。缺省情况下会记录常规访问时间 (<code>atime</code>)。

表 19-2 常用的 -o 挂载选项（续）

挂载选项	文件系统	说明
remount	所有	更改与已挂载的文件系统关联的挂载选项。通常，此选项可以与除 ro 之外的任何选项结合使用。但是，使用此选项可进行的更改取决于文件系统的类型。
retry= <i>n</i>	NFS	在挂载操作失败时重试该操作。 <i>n</i> 是重试的次数。
ro rw	CacheFS、NFS、PCFS、UFS、HSFS	指定读/写 (rw) 或只读 (ro)。如果不指定此选项，则缺省情况下使用 rw。HSFS 的缺省选项是 ro。
suid nosuid	CacheFS、HSFS、NFS、UFS	允许或禁止执行 setuid。缺省情况下允许执行 setuid。

/etc/vfstab 文件的字段说明

/etc/vfstab 文件中的项包含七个字段，下表介绍了这些字段。

表 19-3 /etc/vfstab 文件的字段说明

字段名	说明
device to mount	此字段可标识以下内容之一： <ul style="list-style-type: none">本地 UFS 文件系统的块设备名称（例如 /dev/dsk/c0t0d0s0）。远程文件系统的资源名称（例如 myserver:/export/home）。有关 NFS 的更多信息，请参见《系统管理指南：IP 服务》。要交换的片的块设备名称（例如 /dev/dsk/c0t3d0s1）。虚拟文件系统的目录。
device to fsck	与 device to mount 字段标识的 UFS 文件系统对应的原始（字符）设备名称（例如 /dev/rdsk/c0t0d0s0）。此字段可确定 fsck 命令使用的原始接口。如果没有适用设备，请使用连字符 (-)，如对于只读文件系统或远程文件系统。
mount point	标识挂载文件系统的位置（例如 /usr）。
FS type	标识文件系统的类型。

表 19-3 /etc/vfstab 文件的字段说明 (续)

字段名	说明
fsck pass	<p>此字段为 fsck 命令用来确定是否检查文件系统的传送号。如果此字段包含短划线 (-)，则不会检查文件系统。</p> <p>如果此字段包含零，则不会检查 UFS 文件系统。但是，会检查非 UFS 文件系统。如果此字段包含的值大于零，则会始终检查文件系统。</p> <p>对于此字段中的值为 1 的所有文件系统，将按它们在 vfstab 文件中出现的顺序进行检查（每次检查一个）。如果在 fsck pass 值大于 1 的多个 UFS 文件系统上运行 fsck 命令并且使用了整理选项 (-o p)，则 fsck 命令将自动并行检查不同磁盘上的文件系统，以最大程度地提高效率。否则，传送号的值没有任何作用。</p>
mount at boot	<p>设置为 yes 或 no，以指示在引导系统时 mountall 命令是否应该自动挂载文件系统。请注意，此字段与 AutoFS 无关。根 (/)、/usr 和 /var 文件系统最初不是从 vfstab 文件挂载的。对于这些文件系统以及诸如 /proc 和 /dev/fd 之类的虚拟文件系统，此字段应该始终设置为 no。</p>
mount options	<p>挂载文件系统时使用的一系列以逗号分隔的选项（中间无空格）。使用短划线 (-) 表示没有选项。有关常用挂载选项的列表，请参见表 19-2。</p>

注 - 必须为 /etc/vfstab 文件中的每个字段输入一个值。如果某个字段没有值，请务必指定一个短划线 (-)。否则，系统可能无法成功引导。同样，字段值也不能为空白。

挂载文件系统

以下各节介绍如何通过向 /etc/vfstab 文件中添加项或从命令行使用 mount 命令来挂载文件系统。

如何确定挂载的文件系统

可以使用 mount 命令确定已经挂载的文件系统。

```
$ mount [ -v ]
```

-v 可以详细模式显示已挂载文件系统的列表。

示例 19-1 确定挂载的文件系统

此示例说明如何使用 mount 命令显示有关当前挂载的文件系统的信息。

```
$ mount
```

```
/ on /dev/dsk/c0t0d0s0 read/write/setuid/intr/largefiles/xattr/onerror=...
```

示例 19-1 确定挂载的文件系统 (续)

```

/devices on /devices read/write/setuid/dev=46c0000 on Thu Sep ...
/system/contract on ctfs read/write/setuid/devices/dev=43c0001 ...
/usr on /dev/dsk/c0t0d0s6 read/write/setuid/intr/largefiles/xattr/...
/proc on /proc read/write/setuid/dev=4700000 on Thu Sep 2 ...
/etc/mnttab on mnttab read/write/setuid/dev=47c0000 on Thu Sep 2 ...
/etc/svc/volatile on swap read/write/setuid/devices/xattr/dev=4480001 ...
/system/object on objfs read/write/setuid/devices/dev=44c0001 ...
/dev/fd on fd read/write/setuid/dev=4800000 on Thu Sep 2 ...
/var/run on swap read/write/setuid/xattr/dev=1 on Thu Sep 2 ...
/tmp on swap read/write/setuid/xattr/dev=2 on Thu Sep 2 ...
/stuff on /dev/dsk/c0t0d0s5 read/write/setuid/intr/largefiles/xattr...
/export/home on /dev/dsk/c0t0d0s7 read/write/setuid/intr/largefiles/...
/home/rimmer on pluto:/export/home/rimmer remote/read/write/setuid/xattr/...
$

```

▼ 如何向 /etc/vfstab 文件添加项

- 1 成为超级用户或承担等效角色。
- 2 如有必要，请为要挂载的文件系统创建挂载点。

```
# mkdir /mount-point
```

本地系统上必须存在挂载点才能挂载文件系统。**挂载点**是要向其附加已挂载文件系统的目录。

- 3 编辑 /etc/vfstab 文件并添加项。请确保执行以下操作：

- a. 使用空白（空格或制表符）分隔各个字段。
- b. 如果字段没有内容，请指定连字符(-)。

c. 保存更改。

有关 `/etc/vfstab` 字段各项的详细信息，请参见表 19-3。

注 - 由于在引导过程中内核会将根 (`/`) 文件系统挂载为只读，因此只有 `remount` 选项（以及可以与 `remount` 结合使用的选项）会影响 `/etc/vfstab` 文件中的根 (`/`) 项。

示例 19-2 向 `/etc/vfstab` 文件添加项

以下示例说明如何将磁盘片 `/dev/dsk/c0t3d0s7` 作为 UFS 文件系统挂载到挂载点 `/files1`。原始字符设备 `/dev/rdisk/c0t3d0s7` 会指定为 `device to fsck`。`fsck pass` 值为 2 意味着将检查文件系统，但不是按顺序检查。

#device	device	mount	FS	fsck	mount	mount
#to mount	to fsck	point	type	pass	at boot	options
#						
/dev/dsk/c0t3d0s7	/dev/rdisk/c0t3d0s7	/files1	ufs	2	yes	-

以下示例说明如何将 `/export/man` 目录作为 NFS 文件系统从系统 `pluto` 挂载到挂载点 `/usr/man` 上。由于该目录是 NFS 文件系统，因此既不会指定 `device to fsck` 也不会指定 `fsck pass`。在本示例中，`mount options` 为 `ro`（只读）和 `soft`。

#device	device	mount	FS	fsck	mount	mount
#to mount	to fsck	point	type	pass	at boot	options
pluto:/export/man	-	/usr/man	nfs	-	yes	ro,soft

以下示例说明如何将根 (`/`) 文件系统挂载到回送挂载点 `/tmp/newroot` 上。必须始终在 LOFS 文件系统种的文件系统之后挂载 LOFS 文件系统。

#device	device	mount	FS	fsck	mount	mount
#to mount	to fsck	point	type	pass	at boot	options
#						
/	-	/tmp/newroot	lofs	-	yes	-

▼ 如何挂载文件系统（ /etc/vfstab 文件 ）

- 1 成为超级用户或承担等效角色。
- 2 挂载 /etc/vfstab 文件中列出的文件系统。

```
# mount /mount-point
```

其中 */mount-point* 指定了 /etc/vfstab 文件的 mount point 或 device to mount 字段中的项。通常，指定挂载点较为容易。

示例 19-3 挂载文件系统（ /etc/vfstab 文件 ）

以下示例说明如何挂载 /etc/vfstab 文件中列出的 /usr/dist 文件系统。

```
# mount /usr/dist
```

示例 19-4 挂载所有文件系统（ /etc/vfstab 文件 ）

以下示例说明在已挂载文件系统的情况下使用 mountall 命令时显示的消息。

```
# mountall
```

```
/dev/rdisk/c0t0d0s7 already mounted
```

```
mount: /tmp already mounted
```

```
mount: /dev/dsk/c0t0d0s7 is already mounted, /export/home is busy,
```

```
or the allowable number of mount points has been exceeded
```

使用 mountall 命令时，在挂载具有 device to fsck 项的所有文件系统之前，将对其进行检查和修复（如有必要）。

以下示例说明如何挂载 /etc/vfstab 文件中列出的所有本地系统。

```
# mountall -l
```

```
# mount
```

```
/ on /dev/dsk/c0t0d0s0 read/write/setuid/intr/largefiles/xattr/onerror=...
```

```
/devices on /devices read/write/setuid/dev=46c0000 on Thu Sep ...
```

```
/system/contract on ctfs read/write/setuid/devices/dev=43c0001 ...
```

```
/usr on /dev/dsk/c0t0d0s6 read/write/setuid/intr/largefiles/xattr/...
```

```

/proc on /proc read/write/setuid/dev=4700000 on Thu Sep  2 ...
/etc/mnttab on mnttab read/write/setuid/dev=47c0000 on Thu Sep  2 ...
/etc/svc/volatile on swap read/write/setuid/devices/xattr/dev=4480001 ...
/system/object on objfs read/write/setuid/devices/dev=44c0001 ...
/dev/fd on fd read/write/setuid/dev=4800000 on Thu Sep  2 ...
/var/run on swap read/write/setuid/xattr/dev=1 on Thu Sep  2 ...
/tmp on swap read/write/setuid/xattr/dev=2 on Thu Sep  2 ...
/stuff on /dev/dsk/c0t0d0s5 read/write/setuid/intr/largefiles/xattr...
/export/home on /dev/dsk/c0t0d0s7 read/write/setuid/intr/largefiles/...

```

以下示例说明如何挂载 `/etc/vfstab` 文件中列出的所有远程文件系统。

```

# mountall -r

# mount

/ on /dev/dsk/c0t0d0s0 read/write/setuid/intr/largefiles/xattr/onerror=...
/devices on /devices read/write/setuid/dev=46c0000 on Thu Sep  ...
/system/contract on ctfs read/write/setuid/devices/dev=43c0001 ...
/usr on /dev/dsk/c0t0d0s6 read/write/setuid/intr/largefiles/xattr/...
/proc on /proc read/write/setuid/dev=4700000 on Thu Sep  2 ...
/etc/mnttab on mnttab read/write/setuid/dev=47c0000 on Thu Sep  2 ...
/etc/svc/volatile on swap read/write/setuid/devices/xattr/dev=4480001 ...
/system/object on objfs read/write/setuid/devices/dev=44c0001 ...
/dev/fd on fd read/write/setuid/dev=4800000 on Thu Sep  2 ...
/var/run on swap read/write/setuid/xattr/dev=1 on Thu Sep  2 ...
/tmp on swap read/write/setuid/xattr/dev=2 on Thu Sep  2 ...
/stuff on /dev/dsk/c0t0d0s5 read/write/setuid/intr/largefiles/xattr...

```

```
/stuff on /dev/dsk/c0t0d0s5 read/write/setuid/intr/largefiles/xattr...

/export/home on /dev/dsk/c0t0d0s7 read/write/setuid/intr/largefiles/...

/home/rimmer on pluto:/export/home/rimmer remote/read/write/setuid/xattr/...
```

▼ 如何挂载 UFS 文件系统（mount 命令）

- 1 成为超级用户或承担等效角色。
- 2 如有必要，请为要挂载的文件系统创建挂载点。

```
# mkdir /mount-point
```

本地系统上必须存在挂载点才能挂载文件系统。**挂载点**是要向其附加已挂载文件系统的目录。

- 3 挂载 UFS 文件系统。

```
# mount [-o mount-options] /dev/dsk/device-name /mount-point
```

-o mount-options 指定可以用来挂载 UFS 文件系统的挂载选项。有关选项的列表，请参见表 19-2 或 `mount_ufs(1M)`。

/dev/dsk/device-name 指定包含文件系统的片的磁盘设备名称（例如 `/dev/dsk/c0t3d0s7`）。要查看磁盘的片信息，请参见第 227 页中的“如何显示磁盘片信息”。

/mount-point 指定要挂载文件系统的目录。

示例 19-5 挂载 UFS 文件系统（mount 命令）

以下示例说明如何将 `/dev/dsk/c0t3d0s7` 挂载到 `/files1` 目录上。

```
# mount /dev/dsk/c0t3d0s7 /files1
```

▼ 如何挂载不包含大文件的 UFS 文件系统（mount 命令）

挂载文件系统时，缺省情况下会选用 `largefiles` 选项。使用此选项可以创建大于 2 GB 的文件。一旦文件系统包含大文件，就无法使用 `nolargefiles` 选项重新挂载文件系统或将其挂载到运行 Solaris 2.6 或兼容版本的系统上，直到删除任何大文件并运行 `fsck` 命令将状态重置为 `nolargefiles` 为止。

此过程假定文件系统包含在 `/etc/vfstab` 文件中。

- 1 成为超级用户或承担等效角色。
- 2 如有必要，请为要挂载的文件系统创建挂载点。

```
# mkdir /mount-point
```

本地系统上必须存在挂载点才能挂载文件系统。**挂载点**是要向其附加已挂载文件系统的目录。

- 3 确保文件系统中不存在大文件。

```
# cd /mount-point
```

```
# find . -xdev -size +20000000 -exec ls -l {} \;
```

其中 `/mount-point` 用于标识要检查是否包含大文件的文件系统的挂载点。

- 4 如有必要，请删除此文件系统中的任何大文件或将其移动到其他文件系统。

- 5 取消挂载文件系统。

```
# umount /mount-point
```

- 6 重置文件系统状态。

```
# fsck /mount-point
```

- 7 使用 `nolargefiles` 选项重新挂载文件系统。

```
# mount -o nolargefiles /mount-point
```

示例 19-6 挂载不包含大文件的文件系统（`mount` 命令）

以下示例说明如何检查 `/datab` 文件系统并使用 `nolargefiles` 选项将其重新挂载。

```
# cd /datab
```

```
# find . -xdev -size +20000000 -exec ls -l {} \;
```

```
# umount /datab
```

```
# fsck /datab
```

```
# mount -o nolargefiles /datab
```

▼ 如何挂载 NFS 文件系统（mount 命令）

- 1 成为超级用户或承担等效角色。

- 2 如有必要，请为要挂载的文件系统创建挂载点。

```
# mkdir /mount-point
```

本地系统上必须存在挂载点才能挂载文件系统。挂载点是要向其附加已挂载文件系统的目录。

- 3 确保资源（文件或目录）在服务器中可用。

要挂载 NFS 文件系统，必须使用 share 命令使资源在服务器中可用。有关如何共享资源的信息，请参见《系统管理指南：网络服务》中的“关于 NFS 服务”。

- 4 挂载 NFS 文件系统。

```
# mount -F nfs [-o mount-options] server:/directory /mount-point
```

-o mount-options 指定可以用来挂载 NFS 文件系统的挂载选项。有关常用的 mount 选项的列表，请参见表 19-2；有关选项的完整列表，请参见 mount_nfs(1M)。

server:/directory 指定包含共享资源的服务器主机名，以及要挂载的文件或目录的路径。

/mount-point 指定要挂载文件系统的目录。

示例 19-7 挂载 NFS 文件系统（mount 命令）

以下示例说明如何将 /export/packages 目录从服务器 pluto 挂载到 /mnt 上。

```
# mount -F nfs pluto:/export/packages /mnt
```

▼ x86: 如何从硬盘挂载 PCFS (DOS) 文件系统（mount 命令）

使用以下过程可从硬盘挂载 PCFS (DOS) 文件系统。

- 1 成为超级用户或承担等效角色。

- 2 如有必要，请为要挂载的文件系统创建挂载点。

```
# mkdir /mount-point
```


本地系统上必须存在挂载点才能挂载文件系统。**挂载点**是要向其附加已挂载文件系统的目录。

3 挂载 PCFS 文件系统。

```
# mount -F pcfs [-o rw | ro] /dev/dsk/device-name:logical-drive /mount-point
```

`-o rw | ro` 指定可以将 PCFS 文件系统挂载为读/写 (rw) 还是只读 (ro)。如果不指定此选项，则缺省情况下使用 rw。

`/dev/dsk/device-name` 指定整个磁盘的设备名称（例如 `/dev/dsk/c0t0d0p0`）。

`logical-drive` 指定 DOS 逻辑驱动器名（c 到 z）或驱动器号（1 到 24）。驱动器 c 相当于驱动器 1，表示驱动器上的主 DOS 片。所有其他字母或数字均表示扩展的 DOS 片内的 DOS 逻辑驱动器。

`/mount-point` 指定要挂载文件系统的目录。

请注意，必须使用冒号分隔 `device-name` 和 `logical-drive`。

示例 19-8 x86: 从硬盘挂载 PCFS (DOS) 文件系统（mount 命令）

以下示例说明如何将主 DOS 片中的逻辑驱动器挂载到 `/pcfs/c` 目录上。

```
# mount -F pcfs /dev/dsk/c0t0d0p0:c /pcfs/c
```

以下示例说明如何在 `/mnt` 目录上将扩展的 DOS 片中的第一个逻辑驱动器挂载为只读。

```
# mount -F pcfs -o ro /dev/dsk/c0t0d0p0:2 /mnt
```

取消挂载文件系统

取消挂载文件系统会将该文件系统从文件系统挂载点删除，并从 `/etc/mnttab` 文件中删除相应项。部分文件系统管理任务无法在已挂载的文件系统上执行。出现以下情况时，应取消挂载文件系统：

- 文件系统不再需要，或者已被包含更新的软件的文件系统所替换。
- 需要使用 `fsck` 命令检查和修复文件系统。有关 `fsck` 命令的更多信息，请参见第 22 章。

进行完整备份之前，应取消挂载文件系统。有关进行备份的更多信息，请参见第 25 章。

注 – 在系统关闭过程中，会自动取消挂载文件系统。

在紧急情况下，可以使用 `umount -f` 选项强制取消挂载繁忙的文件系统。建议在正常情况下不要使用此做法，因为取消挂载包含打开文件的文件系统可能会导致数据丢失。此选项只能用于 UFS 和 NFS 文件系统。

取消挂载文件系统的先决条件

取消挂载文件系统的先决条件包括：

- 您必须是超级用户或承担等效角色。
- 必须有一个文件系统可用于取消挂载。不能取消挂载繁忙的文件系统。如果出现以下任一情况，则认为文件系统繁忙：用户正在访问文件系统上的目录，程序打开了该文件系统中的文件，文件系统正在被共享。通过执行以下操作，可以使文件系统可用于取消挂载：
 - 转到其他文件系统上的目录。
 - 从系统中注销。
 - 使用 `fuser` 命令列出正在访问文件系统的所有进程并将其停止（如有必要）。有关更多详细信息，请参见第 386 页中的“如何停止访问文件系统的所有进程”。如果需要取消挂载用户正在使用的文件系统，请通知他们。
- 取消共享文件系统。有关取消共享文件系统的信息，请参见 `unshare(1M)`。

如何验证文件系统是否已取消挂载

要验证是否已取消挂载一个或多个系统，请检查 `mount` 命令的输出。

```
$ mount | grep unmounted-file-system
```

```
$
```

▼ 如何停止访问文件系统的所有进程

- 1 成为超级用户或承担等效角色。
- 2 列出正在访问文件系统的所有进程，以便确定要停止的进程。

```
# fuser -c [ -u ] /mount-point
```

`-c` 针对文件系统挂载点的文件以及那些已挂载文件系统内的任何文件进行报告。

`-u` 显示每个进程 ID 的用户登录名。

/mount-point 指定要为其停止进程的文件系统的名称。

3 停止正在访问文件系统的所有进程。

```
# fuser -c -k /mount-point
```

SIGKILL 会发送到正在使用文件系统的每个进程。

注 – 在没有首先警告用户的情况下，不应停止用户进程。

4 验证是否没有进程正在访问文件系统。

```
# fuser -c /mount-point
```

示例 19-9 停止访问文件系统的所有进程

以下示例说明如何停止正在使用 */export/home* 文件系统的进程 4006c。

```
# fuser -c /export/home
```

```
/export/home:      4006c
```

```
# fuser -c -k /export/home
```

```
/export/home:      4006c
```

```
# fuser -c /export/home
```

```
/export/home:
```

▼ 如何取消挂载文件系统

使用以下过程可取消挂载除根 (/)、*/usr* 或 */var* 文件系统之外的文件系统。

注 – 根 (/)、*/usr* 和 */var* 文件系统只能在关机过程中取消挂载。系统需要有这些文件系统才能正常工作。

1 确保已满足第 386 页中的“取消挂载文件系统的先决条件”中列出的先决条件。

2 取消挂载文件系统。

```
# umount /mount-point
```

其中 */mount-point* 是要取消挂载的文件系统的名称。这可以是以下各项之一：

- 挂载文件系统的目录的名称
- 文件系统的设备名称路径
- NFS 文件系统的资源
- LOFS 文件系统的回送目录

示例 19-10 取消挂载文件系统

以下示例说明如何取消挂载本地起始文件系统。

```
# umount /export/home
```

以下示例说明如何取消挂载片 7 上的文件系统。

```
# umount /dev/dsk/c0t0d0s7
```

以下示例说明如何强制取消挂载 /export 文件系统。

```
# umount -f /export
```

```
#
```

以下示例说明如何取消挂载 /etc/vfstab 文件中除根 (/)、/proc、/var 和 /usr 文件系统之外的所有文件系统。

```
# umountall
```

除繁忙的那些文件系统以外，将取消挂载其他所有文件系统。

使用 CacheFS 文件系统（任务）

本章介绍如何设置和维护 CacheFS™ 文件系统。

以下列出本章中的任务列表。

- [第 389 页中的“使用 CacheFS 文件系统的高级视图（任务列表）”](#)
- [第 392 页中的“创建并挂载 CacheFS 文件系统（任务列表）”](#)
- [第 397 页中的“维护 CacheFS 文件系统（任务列表）”](#)
- [第 403 页中的“压缩已高速缓存的文件系统（任务列表）”](#)
- [第 413 页中的“收集 CacheFS 统计信息（任务列表）”](#)

有关对 CacheFS 错误进行疑难解答的信息，请参见[第 410 页中的“对 cachefspack 错误进行疑难解答”](#)。

注 - 有关 NFS 版本 4 和 CacheFS 软件的重要信息，请参见[第 340 页中的“NFS 版本 4 和 CacheFS 的兼容性问题”](#)。

使用 CacheFS 文件系统的高级视图（任务列表）

使用此任务列表可确定有关使用 CacheFS 文件系统的所有任务。每个任务指向一系列其他任务，如创建并挂载 CacheFS 文件系统，以及压缩和维护高速缓存。

任务	说明	参考
1. 创建并挂载 CacheFS 文件系统。	创建高速缓存并在该高速缓存中挂载文件系统。	第 392 页中的“创建并挂载 CacheFS 文件系统（任务列表）”

任务	说明	参考
2. 维护 CacheFS 文件系统。	通过取消挂载、删除或重新创建高速缓存，显示和修改 CacheFS 文件系统。	第 397 页中的 “维护 CacheFS 文件系统（任务列表）”
3. （可选）压缩和解压缩 CacheFS 文件系统。	确定是否要压缩高速缓存和使用压缩列表。压缩高速缓存可确保某些文件和目录始终在高速缓存中进行更新。	第 403 页中的 “压缩已高速缓存的文件系统（任务列表）”
4. 收集 CacheFS 统计信息。	确定高速缓存性能和适当的高速缓存大小。	第 413 页中的 “收集 CacheFS 统计信息（任务列表）”

CacheFS 文件系统概述

CacheFS 文件系统是一种通用的高速缓存机制，它通过减少服务器和网络负载来提高 NFS 服务器的性能和可伸缩性。由于设计为分层文件系统，CacheFS 文件系统可提供将一个文件系统高速缓存在另一个文件系统上的功能。在 NFS 环境中，CacheFS 文件系统增加客户机与服务器之比，减少服务器和网络负载，并提高客户机在点对点协议 (Point-to-Point Protocol, PPP) 等慢速链路上的性能。

CacheFS 文件系统的工作原理

在客户机系统上创建 CacheFS 文件系统，以便您高速缓存的文件系统可以由客户机在本地访问，而不是通过网络访问。下图说明使用 CacheFS 文件系统时所涉及的各组件的关系。

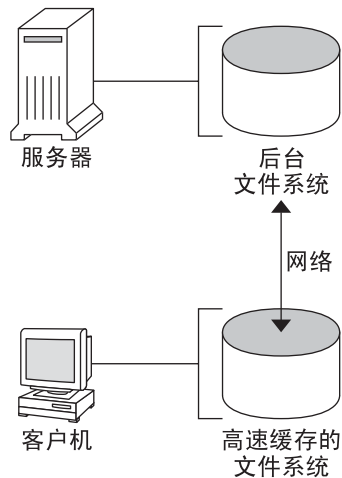


图 20-1 CacheFS 文件系统的工作原理

后台文件系统是指定要在高速缓存中挂载的文件系统。后台文件系统可以是 NFS 或 HSFS（High Sierra File System，高层 Sierra 文件系统）。当用户尝试访问属于后台文件系统的文件时，这些文件放置在高速缓存中。**前台文件系统**是在高速缓存中挂载并从本地挂载点访问的文件系统。前台文件系统的类型必须是 UFS。

对用户而言，如果是首次请求使用 CacheFS 文件系统中的文件，响应可能会显得较慢。但是，如果以后再次使用同一文件，则响应会变快。

CacheFS 文件系统的结构和行为

每个高速缓存都具有一组确定其结构和行为的参数。这些参数设置为下表中列出的缺省值。缺省值指定将整个前台文件系统用于高速缓存，这是对文件系统进行高速缓存时推荐使用的方法。

表 20-1 CacheFS 文件的参数及其缺省值

CacheFS 文件的参数	缺省值	定义
maxblocks	90 %	设置允许 CacheFS 文件系统在后台文件系统中请求的最大块数。
minblocks	0 %	设置允许 CacheFS 文件系统在后台文件系统中请求的最小块数。
threshblocks	85 %	设置在 CacheFS 文件系统可以请求的块数大于 minblocks 指定的块数之前必须在后台文件系统中可用的块数。

表 20-1 CacheFS 文件系统的参数及其缺省值（续）

CacheFS 文件系统的参数	缺省值	定义
maxfiles	90 %	设置允许 CacheFS 文件系统在前台文件系统中请求的最大可用 inode 数（文件数）。
minfiles	0 %	设置允许 CacheFS 文件系统在前台文件系统中请求的最小可用 inode 数。
threshfiles	85 %	设置在 CacheFS 文件系统可以请求大于 minfiles 中指定的文件数之前，前台文件系统中必须提供的 inode 数。

通常，不应更改上述任何参数值。它们设置为缺省值是为了实现最佳的高速缓存行为。但是，如果在前台文件系统中有一些高速缓存未使用的空间，且您希望将它用于其他某个文件系统，则可能需要修改 maxblocks 和 maxfiles 值。使用 cfsadmin 命令可以实现此操作。例如：

```
$ cfsadmin -o maxblocks=60
```

创建并挂载 CacheFS 文件系统（任务列表）

使用此任务列表中的过程可创建并挂载 CacheFS 文件系统。

任务	说明	参考
1. 共享要高速缓存的文件系统。	验证是否已共享要高速缓存的文件系统。	share(1M)
2. 创建高速缓存。	使用 cfsadmin 命令创建高速缓存。	第 393 页中的“如何创建高速缓存”
3. 在高速缓存中挂载文件系统。	使用以下方法之一在高速缓存中挂载文件系统：	
	使用 mount 命令挂载 CacheFS 文件系统。	第 394 页中的“如何挂载 CacheFS 文件系统 (mount)”
	通过编辑 /etc/vfstab 文件挂载 CacheFS 文件系统。	第 396 页中的“如何挂载 CacheFS 文件系统 (/etc/vfstab)”
	使用 AutoFS 挂载已高速缓存的文件系统。	第 397 页中的“如何挂载 CacheFS 文件系统 (AutoFS)”

▼ 如何创建高速缓存

- 1 成为客户机系统上的超级用户。
- 2 创建高速缓存。

```
# cfsadmin -c /cache-directory
```

其中 *cache-directory* 指示高速缓存所驻留的目录的名称。

有关更多信息，请参见 *cfsadmin(1M)*。

注 – 在创建高速缓存后，不要在高速缓存目录本身内执行任何操作。这样做可能在 CacheFS 软件内导致冲突。

示例 20-1 创建高速缓存

以下示例说明如何使用高速缓存参数的缺省值在 */local/mycache* 目录中创建高速缓存。

```
# mkdir /local
```

```
# cfsadmin -c /local/mycache
```

在高速缓存中挂载文件系统

在高速缓存中指定要挂载的文件系统，以便用户可以在本地访问该文件系统中的文件。在用户访问这些文件之前，它们实际上并未放置在高速缓存中。

下表介绍挂载 CacheFS 文件系统的三种方法。

CacheFS 文件系统的挂载类型	CacheFS 挂载类型的使用频率
使用 <i>mount</i> 命令	每次系统重新引导以便访问同一文件系统时都使用。
编辑 <i>/etc/vfstab</i> 文件	仅一次。 <i>/etc/vfstab</i> 文件在系统重新引导后保持不变。
使用 AutoFS	仅一次。AutoFS 映射在系统重新引导后保持不变。

选择最适合您环境的文件系统挂载方法。

只能挂载共享的文件系统。有关共享文件系统的信息，请参见 *share(1M)*。

注 - 在 CacheFS 文件系统中不支持对根 (/) 文件系统和 /usr 文件系统进行高速缓存。

▼ 如何挂载 CacheFS 文件系统 (mount)

- 1 成为客户机系统上的超级用户。
- 2 如有必要，请创建挂载点。

```
# mkdir /mount-point
```

可以从任何位置创建挂载点，但是它必须是 UFS 文件系统所在的位置。与 mount 命令一起使用的 CacheFS 选项（如下一步所示），确定将创建的挂载点高速缓存在指定的高速缓存目录中。

- 3 在高速缓存中挂载文件系统。

```
# mount -F cachefs -o backfstype=fstype,cachedir=/cache-directory[,options]
```

<i>/back-filesystem</i>	<i>/mount-point</i>
<i>fstype</i>	指示后台文件系统的文件系统类型，其类型可以是 NFS 或 HSFS。
<i>/cache-directory</i>	指示高速缓存所驻留的 UFS 目录的名称。此名称与在 第 393 页中的“如何创建高速缓存” 中创建高速缓存时指定的名称相同。
<i>options</i>	指定在高速缓存中挂载文件系统时可以包括的其他挂载选项。有关 CacheFS 挂载选项的列表，请参见 mount_cachefs(1M)。
<i>/back-filesystem</i>	指定要高速缓存的后台文件系统的挂载点。如果后台文件系统是 NFS 文件系统，则必须指定从其挂载文件系统的服务器的主机名和要高速缓存的文件系统的名称，并用冒号分隔这两个名称。例如 <i>merlin:/data/abc</i> 。
<i>/mount-point</i>	指示在其中挂载文件系统的目录。

- 4 验证实际上是否已挂载所创建的高速缓存。

```
# cachefsstat /mount-point
```

/mount-point 是所创建的 CacheFS 文件系统。

例如：

```
# cachefsstat /docs
```

```
/docs
```

```
cache hit rate: 100% (0 hits, 0 misses)
```

```
consistency checks:      1 (1 pass, 0 fail)

modifies:                0

garbage collection:      0
```

如果没有在高速缓存中挂载文件系统，则显示与以下内容类似的错误消息：

```
# cacheostat /mount-point
```

```
cacheostat: mount-point: not a cache file system mountpoint
```

有关 `cacheostat` 命令的更多信息，请参见第 414 页中的“收集 CacheFS 统计信息”。

示例 20-2 挂载 CacheFS 文件系统 (mount)

以下示例说明如何在名为 `/local/mycache` 的高速缓存中将名为 `merlin:/docs` 的 NFS 文件系统作为 CacheFS 文件系统 `/docs` 挂载。

```
# mkdir /docs
```

```
# mount -F cache -o backfstype=nfs,cachedir=/local/mycache merlin:/docs /docs
```

以下示例说明如何将 Solaris 9 SPARC™ CD（HSFS 文件系统）作为 CacheFS 文件系统 `/cfssrc` 使用。由于无法写入 CD，因此指定了 `ro` 参数以便使 CacheFS 文件系统成为只读文件系统。此示例假定 `vol` 守护进程未运行。

```
# mount -F hsfs -o ro /dev/dsk/c0t6d0s0 /sol9
```

```
# mount -F cache -o backfstype=hsfs,cachedir=/cfs/cache,ro,noconst,
```

```
backpath=/sol9 /dev/dsk/c0t6d0s0 /cfssrc
```

```
# ls /cfssrc
```

```
Copyright Solaris_9
```

以下示例说明如何在 `vol` 运行时将 Solaris 9 SPARC CD 作为 CacheFS 文件系统挂载。

```
# mount -F cache -o backfstype=hsfs,cachedir=/cfs/cache,ro,noconst,
```

```
backpath=/cdrom/sol_9_sparc/s0 /vol/dev/dsk/c0t2d0/sol_9_sparc/s0 /cfssrc
```

以下示例说明如何在 `vol` 运行时将 CD 作为 CacheFS 文件系统挂载。

```
# mount -F cachefs -o backfstype=hsfs,cachedir=/cfs/cache,ro,noconst,  
  
backpath=/cdrom/epson /vol/dev/dsk/c0t2d0/epson /drvrs  
  
以下示例使用 demandconst 选项指定对 NFS CacheFS 文件系统 /docs（其后台文件系统  
为 merlin:/docs）根据需要进行一致性检查。有关更多信息，请参见第 400 页中的  
“CacheFS 文件系统的一致性检查”。  
  
# mount -F cachefs -o backfstype=nfs,cachedir=/local/mycache,demandconst merlin:/docs /docs
```

▼ 如何挂载 CacheFS 文件系统 (/etc/vfstab)

- 1 成为客户机系统上的超级用户。
- 2 使用编辑器，在 /etc/vfstab 文件中指定要挂载的文件系统。
请参见下面的示例。
有关 /etc/vfstab 文件的更多信息，请参见第 376 页中的 “/etc/vfstab 文件的字段说明”。
- 3 挂载 CacheFS 文件系统。

```
# mount /mount-point
```


或者，重新引导系统。

示例 20-3 挂载 CacheFS 文件系统 (/etc/vfstab)

以下示例说明来自远程系统 starbug 的 /data/abc 目录的 /etc/vfstab 项，它是在高速缓存目录 /opt/cache 中挂载的。

#device	device	mount	FS	fsck	mount	mount
#to mount	to fsck	point	type	pass	at boot	options
#						
starbug:/data/abc	/local/abc	/opt/cache	cachefs	7	yes	local-access,bg, nosuid,demandconst,backfstype=nfs,cachedir=/opt/cache

▼ 如何挂载 CacheFS 文件系统 (AutoFS)

通过在自动挂载映射中指定 `-fstype=cachefs` 挂载选项，可以使用 AutoFS 在高速缓存中挂载文件系统。请注意，CacheFS 挂载选项（例如 `backfstype` 和 `cachedir`）也是在自动挂载映射中指定的。

有关自动挂载映射的详细信息，请参见《系统管理指南：网络服务》中的“Autofs 管理的任务概述”或 `automount(1M)`。

- 1 成为客户机系统上的超级用户。

- 2 使用编辑器，将以下行添加到 `auto_direct` 映射：

```
/mount-point -fstype=cachefs,cachedir=/directory,backfstype=nfs
server:/file-system
```

- 3 使用编辑器，将以下行添加到 `auto_master` 映射：

```
/-
/- 项是一个指针，用于检查 auto_direct 映射。
```

- 4 重新引导系统。

- 5 通过转到在高速缓存中挂载的文件系统再列出其内容，验证是否正确创建了该项。

```
# cd /filesystem
# ls
```

示例 20-4 挂载 CacheFS 文件系统 (AutoFS)

以下 `auto_direct` 项将在 `/docs` 目录中自动挂载 CacheFS 文件系统。

```
/docs -fstype=cachefs,cachedir=/local/mycache,backfstype=nfs merlin:/docs
```

维护 CacheFS 文件系统（任务列表）

设置 CacheFS 文件系统后，几乎不需要维护它。如果需要对 CacheFS 文件系统执行维护任务，请使用此任务列表中的可选过程。

任务	说明	参考
修改 CacheFS 文件系统。	通过取消挂载、删除或重新创建高速缓存，修改 CacheFS 文件系统的行为。	第 398 页中的“修改 CacheFS 文件系统”
显示 CacheFS 文件系统信息。	使用 <code>cfsadmin</code> 命令显示有关 CacheFS 文件系统的信息。	第 399 页中的“如何显示有关 CacheFS 文件系统的信息”
执行一致性检查。	使用 <code>cfsadmin</code> 命令按需执行一致性检查。	第 401 页中的“如何根据需要指定高速缓存一致性检查”
删除 CacheFS 文件系统。	使用 <code>umount</code> 命令和 <code>cfsadmin</code> 命令删除 CacheFS 文件系统。	第 401 页中的“如何删除 CacheFS 文件系统”
检查 CacheFS 文件系统的完整性。	使用 <code>fsck_cachefs</code> 命令检查 CacheFS 文件系统的完整性。	第 403 页中的“如何检查 CacheFS 文件系统的完整性”

维护 CacheFS 文件系统

本节介绍如何维护 CacheFS 文件系统。

如果要使用 `/etc/vfstab` 文件挂载文件系统，请通过在 `/etc/vfstab` 文件中编辑文件系统选项来修改高速缓存。如果要使用 AutoFS，请通过在 AutoFS 映射中编辑文件系统选项来修改高速缓存。

修改 CacheFS 文件系统

在高速缓存中修改文件系统时，需要先删除高速缓存，再重新创建它。还可能需要在单用户模式下重新引导计算机，具体取决于共享和访问文件系统的方式。

在以下示例中，使用为 `/docs` 文件系统指定的 `demandconst` 选项，删除、重新创建并重新挂载高速缓存。

```
# shutdown -g30 -y

.

.

.

Root password for system maintenance (control-d to bypass):

single-user privilege assigned to /dev/console.
```

```

.

.

.

Here is where you might be prompted to run fsck on the
file system where the cache is located.


# fsck /local

# mount /local

# cfsadmin -d all /local/mycache

# cfsadmin -c /local/mycache

# init 6

.

.

.

console login:

password:

# mount -F cachefs -o backfstype=nfs,cachedir=/local/cache1,demandconst

merlin:/docs /docs

#

```

▼ 如何显示有关 CacheFS 文件系统的信息

- 1 成为客户机系统上的超级用户。
- 2 显示有关在指定高速缓存下高速缓存的所有文件系统的信息。

```
# cfsadmin -l /cache-directory
```

其中 */cache-directory* 是高速缓存所驻留的目录的名称。

示例 20-5 显示有关 CacheFS 文件系统的信息

以下示例说明有关 `/local/mycache` 高速缓存目录的信息。在此示例中，`/docs` 文件系统高速缓存在 `/local/mycache` 中。最后一行显示 CacheFS 文件系统的名称。

```
# cfsadmin -l /local/mycache

cfsadmin: list cache FS information

maxblocks      90%

minblocks      0%

threshblocks   85%

maxfiles       90%

minfiles       0%

threshfiles    85%

maxfilesize    3MB

merlin:_docs:_docs

#
```

CacheFS 文件系统的一致性检查

为确保已高速缓存的目录和文件保持最新，CacheFS 软件定期检查在高速缓存中存储的文件的一致性。为检查一致性，CacheFS 软件将最新的修改时间与以前的修改时间进行比较。如果这两个修改时间不同，则从高速缓存中清除目录或文件的所有数据和属性。然后，从后台文件系统检索新的数据和属性。

按需执行的一致性检查

仅当显式请求对使用 `-o demandconst` 选项挂载的文件系统进行检查时，才执行一致性检查。如果使用此选项在高速缓存中挂载文件系统，之后就可以使用带有 `-s` 选项的 `cfsadmin` 命令来请求一致性检查。缺省情况下，在访问文件时逐个对文件执行一致性检查。如果未访问文件，则不执行检查。使用 `-o demandconst` 选项可避免出现在网络中执行大量一致性检查的情况。

有关更多信息，请参见 `mount_cachefs(1M)`。

▼ 如何根据需要指定高速缓存一致性检查

- 1 成为客户机系统上的超级用户。

- 2 在高速缓存中挂载文件系统，然后指定高速缓存一致性检查。

```
# mount -F cachefs -o backfstype=nfs,cachedir=/directory,demandconst
server:/file-system /mount-point
```

- 3 启动对特定 CacheFS 文件系统的一致性检查。

```
# cfsadmin -s /mount-point
```

▼ 如何删除 CacheFS 文件系统

- 1 成为客户机系统上的超级用户。

- 2 取消挂载 CacheFS 文件系统。

```
# umount /mount-point
```

其中 */mount-point* 指定要删除的 CacheFS 文件系统。

- 3 确定 CacheFS 文件系统的名称（高速缓存 ID）。

```
# cfsadmin -l /cache-directory
```

```
cfsadmin: list cache FS information
```

```
maxblocks      90%
```

```
minblocks      0%
```

```
threshblocks   85%
```

```
maxfiles       90%
```

```
minfiles       0%
```

```
threshfiles    85%
```

```
maxfilesize    3MB
```

```
cache-ID
```

```
#
```

4 从指定的高速缓存中删除 CacheFS 文件系统。

```
# cfsadmin -d cache-ID /cache-directory
```

cache-ID 指示 CacheFS 文件系统的名称，即 `cfsadmin -l` 输出的最后一行。有关更多信息，请参见第 399 页中的[“如何显示有关 CacheFS 文件系统的信息”](#)。通过将 *cache-ID* 指定为 `all`，可以删除特定高速缓存中的所有 CacheFS 文件系统。

/cache-directory 指定高速缓存所驻留的目录。

5 验证是否已删除 CacheFS 文件系统。

刚删除的文件系统的高速缓存 ID 应该不会出现在 `cfsadmin -l` 输出。

```
# cfsadmin -l /cache-directory
```

```
cfsadmin: list cache FS information
```

```
maxblocks      90%
```

```
minblocks      0%
```

```
threshblocks   85%
```

```
maxfiles       90%
```

```
minfiles       0%
```

```
threshfiles    85%
```

```
maxfilesize    3MB
```

```
#
```

有关在命令输出中指定的字段的更多信息，请参阅 `cfsadmin(1M)`。

6 更新高速缓存的资源计数。

```
# fsck -F cacheFs /cache-directory
```

有关更多信息，请参见第 403 页中的[“如何检查 CacheFS 文件系统的完整性”](#)。

示例 20-6 删除 CacheFS 文件系统

以下示例说明如何从高速缓存中删除文件系统。

```
# umount /cfssrc
```

```
# cfsadmin -l /cfssrc
```

```
# cfsadmin -d _dev_dsk_c0t6d0s0:_cfssrc

# cfsadmin -l

# fsck -F cacheefs /cache-directory
```

▼ 如何检查 CacheFS 文件系统的完整性

使用 `fsck` 命令可检查 CacheFS 文件系统的完整性。CacheFS 版本的 `fsck` 命令可自动更正问题，而不要求用户进行交互。应该无需对 CacheFS 文件系统手动运行 `fsck` 命令，因为 `fsck` 命令在引导时或挂载文件系统时自动运行。如果要手动检查完整性，可以使用以下过程。

有关更多信息，请参见 `fsck_cacheefs(1M)`。

- 1 成为客户机系统上的超级用户。
- 2 检查指定高速缓存中的文件系统。

```
# fsck -F cacheefs [-m -o noclean] /cache-directory
```

<code>-m</code>	导致 <code>fsck</code> 命令检查 CacheFS 文件系统，而不进行任何修复。
<code>-o noclean</code>	仅对 CacheFS 文件系统强制检查。不进行任何修复。
<code>/cache-directory</code>	指示高速缓存所驻留的目录的名称。

示例 20-7 检查 CacheFS 文件系统的完整性

以下示例说明如何检查在 `/local/mycache` 高速缓存中高速缓存的文件系统。

```
# fsck -F cacheefs /local/mycache

#
```

压缩已高速缓存的文件系统（任务列表）

以下任务列表介绍与压缩 CacheFS 文件系统关联的过程。其中的所有过程都是可选的。

任务	说明	参考
在高速缓存中压缩文件。	确定要在高速缓存中装入的文件和目录，然后对它们进行压缩。压缩可确保这些文件的最新副本在高速缓存中是可用的。	第 405 页中的 “如何在高速缓存中压缩文件”
创建压缩列表。	如果不希望指定要在高速缓存中压缩的每个单独文件，请创建压缩列表。	第 407 页中的 “如何创建压缩列表”
使用压缩列表在高速缓存中压缩文件。	指定要在高速缓存中压缩的文件的压缩列表的名称。	第 408 页中的 “如何使用压缩列表在高速缓存中压缩文件”
从高速缓存中解压缩文件或压缩列表。	从高速缓存中删除不再需要的文件。	第 409 页中的 “如何从高速缓存中解压缩文件或压缩列表” 第 409 页中的 “如何从高速缓存中解压缩文件或压缩列表”
显示已压缩文件的信息。	查看有关已压缩文件的信息，包括其压缩状态。	第 405 页中的 “如何显示已压缩文件的信息”

压缩 CacheFS 文件系统

对于常规用途，CacheFS 软件在设置后自动运行，而不要求用户执行任何操作。仅对最新使用的文件进行高速缓存。利用**压缩**功能，通过确保始终在高速缓存中更新某些文件或目录，可以在管理高速缓存方面承担更主动的角色。

使用 `cachefspack` 命令可以指定要在高速缓存中装入的文件和目录。此命令确保这些文件的最新副本在高速缓存中是可用的。

压缩列表包含特定文件和目录的名称。压缩列表也可以包含其他压缩列表。需要在高速缓存中压缩多个项时，利用此功能就不必指定单独的文件和目录。

使用 `-h` 选项可以输出所有 `cachefspack` 选项的简短帮助摘要，如下所示：

```
$ cachefspack -h

Must select 1 and only 1 of the following 5 options

-d Display selected filenames

-i Display selected filenames packing status

-p Pack selected filenames

-u Unpack selected filenames
```

```
-U Unpack all files in directory 'dir'

-f Specify input file containing rules

-h Print usage information

-r Interpret strings in LIST rules as regular expressions

-s Strip './' from the beginning of a pattern name

-v Verbose option

files - a list of filenames to be packed/unpacked
```

▼ 如何在高速缓存中压缩文件

- 在高速缓存中压缩文件。

```
$ cachebspack -p filename
```

-p 指定要压缩一个或多个文件。此选项也是缺省选项。

filename 指定要在高速缓存中压缩的文件或目录的名称。指定目录时，也将压缩该目录的所有子目录。有关更多信息，请参见 `cachebspack(1M)`。

示例 20-8 示例一 在高速缓存中压缩文件

以下示例说明在高速缓存中压缩 `projects` 文件。

```
$ cachebspack -p projects
```

以下示例说明在高速缓存中压缩三个文件。

```
$ cachebspack -p projects updates master_plan
```

以下示例说明在高速缓存中压缩一个目录。

```
$ cachebspack -p /data/abc/bin
```

▼ 如何显示已压缩文件的信息

- 显示已压缩文件的信息。

```
$ cachebspack -i[v] cached-filename-or-directory
```

-i 指定希望查看有关已压缩文件的信息。

`-v` 是详细选项。
`cached-filename-or-directory` 指定要显示其信息的文件或目录的名称。

示例 20-9 显示已压缩文件的信息

以下示例说明已成功压缩 `doc_file` 文件。

```
$ cachebspack -i doc_file
```

```
cachebspack: file doc_file marked packed YES, packed YES
```

在以下示例中，`/data/abc` 目录包含 `bin` 子目录。`bin` 子目录包含以下三个文件：`big`、`medium` 和 `small`。虽然指定要压缩 `big` 和 `small` 文件，但是未压缩它们。成功压缩了 `medium` 文件。

```
$ cd /data/abc
```

```
$ cachebspack -i bin
```

```
.  
.  
.
```

```
cachebspack: file /bin/big marked packed YES, packed NO
```

```
cachebspack: file /bin/medium marked packed YES,  
packed YES
```

```
cachebspack: file /bin/small marked packed YES,  
packed NO
```

```
.  
.  
.
```

如果一起使用 `-iv` 选项，则可以获取有关是否已从高速缓存中清除指定的文件或目录的其他信息。例如：

```
$ cd /data/bin
```

```
FSCACHEPACK-4$ cachefspack -iv bin
.
.
.
cachefspack: file /bin/big marked packed YES, packed NO,
nocache YES
cachefspack: file /bin/medium marked packed YES,
packed YES, nocache NO
cachefspack: file /bin/small marked packed YES,
packed NO
nocache NO
.
.
.
```

此示例的最后一行说明，未从高速缓存中清除该目录的内容。

使用压缩列表

`cachefspack` 命令的一项功能是能够创建压缩列表。

压缩列表包含要在高速缓存中压缩的文件或目录。如果一个目录包含在压缩列表中，则也将压缩它的所有子目录和文件。

利用此功能，就可以节省必须指定要在高速缓存中压缩的每个单独文件的时间。

▼ 如何创建压缩列表

► 使用 vi 创建压缩列表文件。

压缩列表文件使用的格式与 `filesync` 命令的格式相同。有关更多信息，请参见 `filesync(1)`。

以下是压缩列表的两项功能：

- 可以将压缩列表中的文件标识为正则表达式而不是字面上的文件名，便可不必指定每个单独的文件名。
- 通过确保仅压缩您所拥有的那些文件，可以压缩共享目录中的文件。

有关使用这些功能的更多信息，请参见 `cachefspack(1M)`。

示例 20-10 创建压缩列表

以下示例说明压缩列表文件的内容：

```
BASE /home/ignatz
```

```
LIST plans
```

```
LIST docs
```

```
IGNORE *.ps
```

- 使用 `BASE` 语句标识的路径是包含要压缩的项的目录。
- 两条 `LIST` 语句标识该目录中要压缩的特定文件。
- `IGNORE` 语句标识文件类型 `.ps`（不希望压缩）。

▼ 如何使用压缩列表在高速缓存中压缩文件

- 对压缩列表中的文件进行压缩。

```
$ cachefspack -f packing-list
```

`-f` 指定希望使用压缩列表。

`packing-list` 指定压缩列表的名称。

示例 20-11 使用压缩列表在高速缓存中压缩文件

此示例将 `list.pkg` 文件用作 `cachefspack` 命令的压缩列表。

```
$ cachefspack -f list.pkg
```


从高速缓存中解压缩文件或压缩列表

可能需要从高速缓存中删除或解压缩文件。也许一些文件或目录的优先级高于其他文件或目录，因此需要解压缩不太关键的文件。例如，已完成一个项目，并归档了与该项目关联的文件。现在正从事一个新项目，因此使用一组新文件。

▼ 如何从高速缓存中解压缩文件或压缩列表

- ▮ 从高速缓存中解压缩文件或压缩列表。

```
$ cachebspack -u filename | -U cache-directory
```

-u 指定要解压缩的一个或多个文件。必须使用此选项指定一个文件名。

filename 指定要在高速缓存中解压缩的文件或压缩列表的名称。

-U 指定希望解压缩高速缓存中的所有文件。

有关更多信息，请参见 `cachebspack(1M)`。

示例 20-12 从高速缓存中解压缩文件或压缩列表

以下示例说明从高速缓存中解压缩文件 `/data/abc/bin/big`。

```
$ cachebspack -u /data/abc/bin/big
```

以下示例说明从高速缓存中解压缩三个文件。

```
$ cd /data/abc/bin/big
```

```
$ cachebspack -u big small medium
```

以下示例说明如何对压缩列表进行解压缩。压缩列表是一个包含目录（包含若干文件）路径的文件：

```
$ cachebspack -uf list.pkg
```

以下示例使用 -U 选项指定对高速缓存目录中的所有文件进行解压缩。

```
$ cachebspack -U /local/mycache
```

无法解压缩未挂载任何文件系统的高速缓存。使用 -U 选项时，如果指定的高速缓存不包含已挂载的文件系统，则显示与以下内容类似的输出：

```
$ cachebspack -U /local/mycache
```

```
cachefspack: Could not unpack cache /local/mycache, no mounted  
filesystems in the cache.
```

对 cachefspack 错误进行疑难解答

使用 cachefspack 命令时，可能会看到以下错误消息。

```
cachefspack: pathname - can't open directory: permission denied
```

原因

可能没有访问文件或目录的正确权限。

操作

设置正确的权限。

```
cachefspack: pathname - can't open directory: no such file or  
directory
```

原因

指定的文件或目录可能不正确。

操作

检查是否存在可能的键入错误。

```
cachefspack: pathname - can't open directory: stale NFS file handle
```

原因

在尝试访问文件或目录时，该文件或目录可能已从服务器中移走或删除。

操作

验证服务器上的该文件或目录是否仍然可访问。

```
cachefspack: pathname - can't open directory: interrupted system  
call
```

原因

在发出命令时，可能无意中按了 Ctrl-C 组合键。

操作

重新发出命令。

```
cachefspack: pathname - can't open directory: I/O error
```

原因

可能存在硬件问题。

操作

检查硬件连接。

```
cachefspack: error opening dir
```

原因

指定的文件或目录可能不正确。在文件格式中 **BASE** 命令之后标识的路径可能是文件，而不是目录。指定的路径必须是目录。

操作

检查是否存在可能的键入错误。检查在文件格式中 **BASE** 命令之后标识的路径。确保该路径标识的是目录而不是文件。

```
cachefspack: unable to get shared objects
```

原因

可执行文件可能已损坏，或者其格式无法识别。

操作

替换可执行文件。

```
cachefspack: filename - can't pack file: permission denied
```

原因

可能没有访问文件或目录的正确权限。

操作

设置正确的权限。

```
cachefspack: filename - can't pack file: no such file or directory
```

原因

指定的文件或目录可能不正确。

操作

检查是否存在可能的键入错误。

```
cachefspack: filename- can't pack file: stale NFS file handle
```

原因

在尝试访问文件或目录时，该文件或目录可能已从服务器中移走或删除。

操作

验证服务器上的该文件或目录是否仍然可访问。

```
cachefspack: filename- can't pack file: interrupted system call
```

原因

在发出命令时，可能无意中按了 Ctrl-C 组合键。

操作

重新发出命令。

`cachefspack: filename- can't pack file: I/O error`

原因

可能存在硬件问题。

操作

检查硬件连接。

`cachefspack: filename- can't pack file: no space left on device.`

原因

高速缓存的磁盘空间不足。

操作

需要通过增加磁盘空间来增加高速缓存的大小。

`cachefspack: filename - can't unpack file: permission denied`

原因

可能没有访问文件或目录的正确权限。

操作

设置正确的权限。

`cachefspack: filename - can't unpack file: no such file or directory`

原因

指定的文件或目录可能不正确。

操作

检查是否存在可能的键入错误。

`cachefspack: filename- can't unpack file: stale NFS file handle`

原因

在尝试访问文件或目录时，该文件或目录可能已从服务器中移走或删除。

操作

验证服务器上的该文件或目录是否仍然可访问。

`cachefspack: filename- can't unpack file: interrupted system call`

原因

在发出命令时，可能无意中按了 Ctrl-C 组合键。

操作

重新发出命令。

cachefspack: *filename*- can't unpack file I/O error

原因
可能存在硬件问题。

操作
检查硬件连接。

cachefspack: only one 'd', 'i', 'p', or 'u' option allowed

原因
在一个命令会话中指定了其中的多个选项。

操作
为命令会话选择一个选项。

cachefspack: can't find environment variable.

原因
忘记设置对应的环境变量，以使其与配置文件中的 \$ 匹配。

操作
在正确的位置定义该环境变量。

cachefspack: skipping LIST command - no active base

原因
在配置文件中存在 LIST 命令，但该命令没有对应的 BASE 命令。

操作
定义 BASE 命令。

收集 CacheFS 统计信息（任务列表）

以下任务列表说明收集 CacheFS 统计信息所涉及的步骤。其中的所有过程都是可选的。

任务	说明	参考
设置日志记录。	使用 cachefslog 命令设置对 CacheFS 文件系统的日志记录。	第 415 页中的“如何设置 CacheFS 日志记录”
查找日志文件。	使用 cachefslog 命令查找日志文件。	第 415 页中的“如何查找 CacheFS 日志文件”
停止日志记录。	使用 cachefslog 命令停止日志记录。	第 416 页中的“如何停止 CacheFS 日志记录”

任务	说明	参考
查看高速缓存大小。	使用 <code>cachefswssize</code> 命令查看高速缓存大小。	第 416 页中的“如何查看工作集（高速缓存）大小”
查看高速缓存统计信息。	使用 <code>cacheofsstat</code> 命令查看高速缓存统计信息。	第 418 页中的“如何查看 CacheFS 统计信息”

收集 CacheFS 统计信息

通过收集 CacheFS 统计信息，可以执行以下操作：

- 确定适当的高速缓存大小。
- 监视高速缓存的性能。

这些统计信息有助于确定如何平衡高速缓存大小和所需的高速缓存性能。

下表介绍 CacheFS 统计命令。

命令	说明	手册页
<code>cachefslog</code>	指定日志文件的位置。此命令还显示当前记录统计信息的位置，并允许您停止日志记录。	<code>cachefslog(1M)</code>
<code>cachefswssize</code>	解释日志文件，给出建议的高速缓存大小。	<code>cachefswssize(1M)</code>
<code>cacheofsstat</code>	显示有关特定 CacheFS 文件系统或所有 CacheFS 文件系统的统计信息。命令输出中提供的信息是直接 from 高速缓存中提取的。	<code>cacheofsstat(1M)</code>

注 – 可以从任何目录发出 CacheFS 统计命令。必须是超级用户才能发出 `cachefswssize` 命令。

在您创建日志文件时，CacheFS 统计信息将开始累积。工作会话结束时，使用 `cachefslog -h` 命令停止日志记录，如[第 416 页中的“如何停止 CacheFS 日志记录”](#)所述。

使用 CacheFS 统计命令之前，必须执行以下操作：

- 使用 `cfsadmin` 命令设置高速缓存。
- 确定一段适当的时间，允许在此期间将统计信息收集在所创建的日志文件中。这段时间应该相当于一个典型工作会话的时间。例如，一天、一周或一个月。

- 选择日志文件的位置或路径。确保有足够的空间可以满足日志文件增长的需要。允许在日志文件中收集统计信息的预期时间越长，所需的空间就越多。

注- 以下过程是按建议的顺序介绍的。您可以不按照此顺序来执行操作。

▼ 如何设置 CacheFS 日志记录

- 1 设置日志记录。

```
$ cachefslog -f log-file-path /mount-point
```

-f 设置日志记录。

log-file-path 指定日志文件的位置。日志文件是使用编辑器创建的标准文件，如 vi。

/mount-point 指定要为其收集统计信息的挂载点（CacheFS 文件系统）。

- 2 验证是否正确设置了日志文件。

```
$ cachefslog /mount-point
```

示例 20-13 设置 CacheFS 日志记录

以下示例说明如何设置 /var/tmp/samlog 日志文件，以收集有关 /home/sam 目录的统计信息。

```
$ cachefslog -f /var/tmp/samlog /home/sam
```

```
/var/tmp/samlog: /home/sam
```

▼ 如何查找 CacheFS 日志文件

- 1 显示记录 CacheFS 统计信息的位置。

```
$ cachefslog /mount-point
```

其中 /mount-point 指定要查看其统计信息的 CacheFS 文件系统。

也可以使用不带选项的 cachefslog 命令，查找特定挂载点的日志文件。

示例 20-14 查找 CacheFS 日志文件

以下示例说明设置日志文件后将看到的内容。日志文件的位置是 /var/tmp/stufflog。

```
$ cachefslog /home/stuff
```

```
    /var/tmp/stufflog: /home/stuff
```

以下示例说明没有为指定的文件系统设置日志文件。

```
$ cachefslog /home/zap
```

```
not logged: /home/zap
```

如何停止 CacheFS 日志记录

使用 `cachefslog -h` 选项可停止日志记录。

```
$ cachefslog -h /mount-point
```

以下示例说明如何停止对 `/home/stuff` 的日志记录。

```
$ cachefslog -h /home/stuff
```

```
not logged: /home/stuff
```

如果得到的系统响应与此处指定的响应不同，则说明未成功停止日志记录。确定使用的日志文件名和挂载点是否正确。

▼ 如何查看工作集（高速缓存）大小

可能希望检查是否需要增大高速缓存的大小。或者，可能希望根据上次将 `cachefslog` 命令用于特定挂载点以来的活动，确定理想的高速缓存大小。

- 1 成为客户机系统上的超级用户。
- 2 查看当前的高速缓存大小和已记录的最大高速缓存大小。

```
# cachefssize log-file-path
```

有关更多信息，请参见 `cachefssize(1M)`。

示例 20-15 查看工作集（高速缓存）大小

在以下示例中，`end size` 是发出 `cachefssize` 命令时高速缓存的大小。`high water size` 是发生日志记录期间高速缓存的最大大小。


```
# cache swsize /var/tmp/samlog
```

```
/home/sam
```

```
end size: 10688k
```

```
high water size: 10704k
```

```
/
```

```
end size: 1736k
```

```
high water size: 1736k
```

```
/opt
```

```
end size: 128k
```

```
high water size: 128k
```

```
/nfs/saturn.dist
```

```
end size: 1472k
```

```
high water size: 1472k
```

```
/data/abc
```

```
end size: 7168k
```

```
high water size: 7168k
```

```
/nfs/venus.svr4
```

```
end size: 4688k
```

high water size: 5000k

/data

end size: 4992k

high water size: 4992k

total for cache

initial size: 110960k

end size: 30872k

high water size: 30872k

查看 CacheFS 统计信息

下表解释了在 CacheFS 文件系统的统计信息输出中显示的术语。

表 20-2 CacheFS 统计信息术语

输出术语	说明
cache hit rate	高速缓存命中率与高速缓存未命中率之比，后跟实际的命中次数和未命中次数。当用户希望对一个或多个文件执行操作，而且这个或这些文件实际上位于高速缓存中时，将发生 高速缓存命中 。当文件不在高速缓存中时，将发生 高速缓存未命中 。服务器上的负载是高速缓存未命中、一致性检查和修改(modifies)的总和。
consistency checks	执行的一致性检查数，后跟通过次数和失败次数。
modifies	修改操作数。例如，写入或创建。

▼ 如何查看 CacheFS 统计信息

使用 `cachefsstat` 命令查看统计信息。可以随时查看统计信息。例如，不必设置日志记录，即可查看统计信息。

► 查看 CacheFS 统计信息。

```
$ cachefsstat /mount-point
```

其中 */mount-point* 指定要查看其统计信息的 CacheFS 文件系统。

如果不指定挂载点，将显示所有已挂载的 CacheFS 文件系统的统计信息。

示例 20-16 查看 CacheFS 统计信息

此示例说明如何查看有关已高速缓存的文件系统 `/home/sam` 的统计信息。

```
$ cachefsstat /home/sam

    cache hit rate: 73% (1234 hits, 450 misses)

consistency checks: 700 (650 pass, 50 fail)

      modifies: 321

garbage collection: 0
```


配置其他交换空间（任务）

本章提供安装 Solaris OS 后配置其他交换空间的指导和逐步说明。

以下是本章中的逐步说明列表。

- 第 427 页中的 “如何创建交换文件并使其可用”
- 第 429 页中的 “如何删除不需要的交换空间”

以下是本章中概述信息的列表。

- 第 421 页中的 “关于交换空间”
- 第 423 页中的 “如何了解是否需要更多交换空间？”
- 第 424 页中的 “如何分配交换空间”
- 第 425 页中的 “规划交换空间”
- 第 425 页中的 “监视交换资源”
- 第 426 页中的 “添加更多交换空间”

关于交换空间

要确定以下内容，应了解 SunOS™ 交换机制的功能：

- 交换空间要求
- 交换空间与 TMPFS 文件系统之间的关系
- 如何从与交换空间有关的错误消息恢复

交换空间与虚拟内存

Solaris 软件将一些磁盘片用于临时存储而不是用于文件系统。这些片称为**交换片**。当系统没有足够的物理内存来处理当前进程时，会将交换片用作虚拟内存存储区域。

虚拟内存系统将磁盘上文件的物理副本映射到内存中的虚拟地址。文件系统中的普通文件或交换空间可以备份包含这些映射的数据的物理内存页面。如果交换空间对内存进行了备份，则该内存称为**匿名内存**，因为没有为备份内存的磁盘空间指定标识。

Solaris OS 使用的**虚拟交换空间**概念是指，匿名内存页面和实际备份这些页面的物理存储（或磁盘备份交换空间）之间的层。系统的虚拟交换空间等于其所有物理（磁盘备份）交换空间与当前部分可用物理内存的和。

虚拟交换空间具有以下优点：

- 由于虚拟交换空间不必对应于物理（磁盘）存储，因此减少了对大量物理交换空间的需求。
- 称为 SWAPFS 的伪文件系统为匿名内存页面提供地址。由于 SWAPFS 可以控制内存页面的分配，因此它在确定要对页面执行的操作方面具有更大的灵活性。例如，SWAPFS 可以更改页面对磁盘备份交换存储的要求。

交换空间与 TMPFS 文件系统

在 Solaris 环境中，TMPFS 文件系统由 `/etc/vfstab` 文件中的项自动激活。TMPFS 文件系统将文件及其关联信息存储在内存中（存储在 `/tmp` 目录中）而不是磁盘上，从而加快了访问这些文件的速度。对于大量使用 `/tmp` 的应用程序（例如编译器和 DBMS 产品），此功能可显著增强这些应用程序的性能。

TMPFS 文件系统从系统的交换资源分配 `/tmp` 目录中的空间。此功能意味着，用尽 `/tmp` 目录中的空间时，也就用尽了交换空间。因此，如果应用程序大量使用 `/tmp` 目录，而且您未监视交换空间使用情况，则系统可能会用尽交换空间。

如果要使用 TMPFS，但交换资源有限，请采用以下措施：

- 使用大小选项 (`-o size`) 挂载 TMPFS 文件系统，以便控制 TMPFS 可以使用的交换资源量。
- 使用编译器的 `TMPDIR` 环境变量以指向其他较大的目录。
使用编译器的 `TMPDIR` 变量仅可以控制编译器是否使用 `/tmp` 目录。此变量对其他程序是否使用 `/tmp` 目录没有影响。

将交换空间作为转储设备

转储设备通常是存储系统崩溃转储信息而保留的磁盘空间。缺省情况下，会将系统的转储设备配置为交换片。如有可能，应将备用磁盘分区配置为**专用转储设备**，以便为崩溃转储提供增强的可靠性，并缩短系统出现故障后重新引导的时间。可以使用 `dumpadm` 命令配置专用转储设备。有关更多信息，请参见《系统管理指南：高级管理》中的第 24 章“管理系统崩溃转储信息（任务）”。

如果使用卷管理器（例如 Solaris Volume Manager）管理磁盘，请勿将专用转储设备配置为受其控制。建议由 Solaris Volume Manager 控制交换区域。但是，出于可访问性和性能原因，应将另一个磁盘配置为不受 Solaris Volume Manager 控制的专用转储设备。

交换空间和动态重新配置

建议分配足够的交换空间，以便在动态重新配置期间支持出现故障的 CPU 或系统板。否则，CPU 或系统板故障可能会导致主机或域重新引导时内存不足。

如果此其他交换空间不可用，则一个或多个应用程序可能会因内存不足而无法启动。此问题需要进行手动干预，以便添加其他交换空间或重新配置这些应用程序的内存使用。

如果已分配其他交换空间来处理重新引导时内存的潜在不足，则所有需要大量内存的应用程序可照常启动。这意味着，用户可以使用系统，但系统速度可能会因一些其他交换而变慢。

有关更多信息，请参见硬件动态重新配置指南。

如何了解是否需要更多交换空间？

使用 `swap -l` 命令可确定系统是否需要更多交换空间。

例如，以下 `swap -l` 输出说明此系统的交换空间几乎完全占用或者已经 100% 分配。

```
% swap -l

swapfile              dev    swaplo blocks    free
/dev/dsk/c0t0d0s1    136,1      16 1638608      88
```

当系统的交换空间为 100% 分配时，会暂时锁定应用程序的内存页面。虽然不会出现应用程序错误，但可能会降低系统性能。

有关将更多交换空间添加到系统的信息，请参见第 427 页中的[“如何创建交换文件并使其可用”](#)。

与交换相关的错误消息

这些消息指示应用程序尝试获取更多的匿名内存。但是，没有剩余的交换空间对其进行备份。

```
application is out of memory
```

```
malloc error 0
```

```
messages.1:Sep 21 20:52:11 mars genunix: [ID 470503 kern.warning]
WARNING: Sorry, no swap space to grow stack for pid 100295 (myprog)
```

与TMPFS相关的错误消息

如果写入文件时无法分配页面，则会显示以下消息。如果TMPFS尝试写入的量超出其允许的量，或当前执行的程序使用了大量内存，则可能会出现此问题。

```
directory: File system full, swap space limit exceeded
```

以下消息表明，TMPFS尝试创建新的文件或目录时物理内存不足：

```
directory: File system full, memory allocation failed
```

有关从与TMPFS相关的错误消息恢复的信息，请参见 `tmpfs(7FS)`。

如何分配交换空间

最初，交换空间是在 Solaris 安装过程中分配的。如果使用安装程序的磁盘片自动布局，且不手动更改交换片的大小，则 Solaris 安装程序会分配 512 MB 的缺省交换区域。

从 Solaris 9 发行版开始，安装程序从第一个可用磁盘柱面（通常为柱面 0）开始分配交换空间。在缺省磁盘布局中，这种放置方法为根 (/) 文件系统提供了最大的空间，并允许在升级时扩展根 (/) 文件系统。

有关分配交换空间的一般规则，请参见第 425 页中的“规划交换空间”。

通过创建交换文件，可以将其他交换空间分配给系统。有关创建交换文件的信息，请参见第 426 页中的“添加更多交换空间”。

交换区域和 /etc/vfstab 文件

安装系统后，交换片和交换文件将在 `/etc/vfstab` 文件中列出。在引导系统时，它们将由 `/sbin/swapadd` 脚本激活。

`/etc/vfstab` 文件中交换设备的项包含以下内容：

- 交换片或交换文件的全路径名
- 交换片或交换文件的文件系统类型

激活交换文件之前，必须挂载包含交换文件的文件系统。因此，在 `/etc/vfstab` 文件中，应确保挂载文件系统的项位于激活交换文件的项之前。

规划交换空间

确定交换空间大小的最重要因素是系统的软件应用程序的要求。例如，大型应用程序（例如计算机辅助设计模拟器、数据库管理产品、事务监视器和地质分析系统）占用的交换空间可高达 200–1000 MB。

请咨询应用程序供应商，以了解其应用程序的交换空间要求。

如果无法通过咨询应用程序供应商确定交换空间要求，请根据系统类型使用以下一般规则来分配交换空间。

系统类型	交换空间的大小	专用转储设备的大小
物理内存约为 4 GB 的工作站	1 GB	1 GB
物理内存约为 8 GB 的中型服务器	2 GB	2 GB
物理内存约为 16 到 128 GB 的高端服务器	4 GB	4 GB

除这些一般规则外，还应考虑为以下项分配交换空间或磁盘空间：

- 专用转储设备。
- 确定大型应用程序（例如编译器）是否将使用 /tmp 目录。然后，分配 TMPFS 要使用的其他交换空间。有关 TMPFS 的信息，请参见第 422 页中的“[交换空间与 TMPFS 文件系统](#)”。

监视交换资源

/usr/sbin/swap 命令用于管理交换区域。-l 和 -s 这两个选项显示有关交换资源的信息。

使用 swap -l 命令可识别系统的交换区域。在 swapfile 列下会列出已激活的交换设备或文件。

```
# swap -l

swapfile          dev  swaplo blocks   free

/dev/dsk/c0t0d0s1 136,1      16 1638608 1600528
```

使用 swap -s 命令可监视交换资源。

```
# swap -s
```

```
total: 57416k bytes allocated + 10480k reserved = 67896k used,  
  
833128k available
```

`used` 值与 `available` 值的和等于系统上的总交换空间，其中包括部分物理内存和交换设备（或文件）。

可以使用可用交换空间量和已用交换空间量（在 `swap -s` 输出中显示）监视在一段时间内交换空间使用情况的变化。如果系统性能良好，请使用 `swap -s` 确定可用交换空间量。如果系统性能下降，请检查可用交换空间量以确定其是否已下降。然后，可以确定对系统所做的哪些更改导致交换空间使用量增加。

使用此命令时，请记住可用于交换的物理内存量会因内核和用户进程的锁定和物理内存的释放而动态变化。

注 - `swap -l` 命令以 512 字节块为单位显示交换空间。`swap -s` 命令以 1024 字节块为单位显示交换空间。如果将 `swap -l` 输出中的块累加，然后将其转换为 KB，则结果小于 `used + available`（在 `swap -s` 输出中显示）。原因是 `swap -l` 在计算交换空间时不包括物理内存。

下表概述了 `swap -s` 命令的输出。

表 21-1 `swap -s` 命令的输出

关键字	说明
<code>bytes allocated</code>	当前已分配作为备份存储（磁盘备份交换空间）的总交换空间量（以 1024 字节块为单位）。
<code>reserved</code>	当前未分配、但内存已请求供将来使用的总交换空间量（以 1024 字节块为单位）。
<code>used</code>	已分配或保留的总交换空间量（以 1024 字节块为单位）。
<code>available</code>	当前可用于将来进行预留和分配的总交换空间量（以 1024 字节块为单位）。

添加更多交换空间

随着系统配置的更改和新软件包的安装，可能需要添加更多交换空间。添加更多交换空间的最简单方法是，使用 `mkfile` 和 `swap` 命令将部分现有 UFS 或 NFS 文件系统指定为补充交换区域。使用这些命令（在以下各节中介绍）可以添加更多交换空间而不必对磁盘重新分区。

添加更多交换空间的其他方法是，对现有磁盘重新分区或添加其他磁盘。有关如何对磁盘重新分区的信息，请参见第 11 章。

创建交换文件

创建交换文件涉及以下常规步骤：

- 使用 `mkfile` 命令创建交换文件。
- 使用 `swap` 命令激活交换文件。
- 在 `/etc/vfstab` 文件中添加交换文件的项，以便在引导系统时自动激活交换文件。

mkfile 命令

`mkfile` 命令创建一个适合用作已挂载 NFS 交换区域或本地交换区域的文件。设置粘滞位，并用零填充文件。可以字节为单位（缺省设置）指定交换文件的大小，或者分别使用 `k`、`b` 或 `m` 后缀以 KB、块或 MB 为单位指定交换文件的大小。

下表说明 `mkfile` 命令选项。

表 21-2 `mkfile` 命令的选项

选项	说明
<code>-n</code>	创建空文件。标注大小。但是，在将数据写入磁盘块之前，不会分配磁盘块。
<code>-v</code>	报告已创建文件的名称和大小。

注 - 应仅在创建 NFS 交换文件时使用 `-n` 选项。

▼ 如何创建交换文件并使其可用

- 1 成为超级用户。
无需超级用户权限，即可创建交换文件。但是，为避免意外覆写，超级用户应该是交换文件的属主。
- 2 如果需要，请为交换文件创建一个目录。
- 3 创建交换文件。

```
# mkfile nnn[k|b|m] filename
```

创建了具有指定 `filename` 的交换文件，且其大小为 `nnn`（以 KB、字节或 MB 为单位）。
- 4 激活交换文件。

```
# /usr/sbin/swap -a /path/filename
```

必须使用绝对路径名指定交换文件。添加了交换文件，该交换文件在取消挂载文件系统、重新引导系统或删除交换文件之前可用。请记住，将某个进程或程序交换至交换文件时，不能取消挂载文件系统。

- 5 将交换文件的项添加到指定该文件全路径名的 `/etc/vfstab` 文件，并将 `swap` 指定为文件系统类型。

```
/path/filename - - swap - no -
```

- 6 检验是否添加了交换文件。

```
$ /usr/sbin/swap -l
```

示例 21-1 创建交换文件并使其可用

以下示例说明如何创建名为 `/files/swapfile` 的 100 MB 交换文件。

```
# mkdir /files

# mkfile 100m /files/swapfile

# swap -a /files/swapfile

# vi /etc/vfstab

(An entry is added for the swap file):

/files/swapfile - - swap - no -

# swap -l

swapfile          dev  swaplo blocks   free
/dev/dsk/c0t0d0s1  136,1      16 1638608 1600528
/files/swapfile    -          16 204784  204784
```

删除不用的交换文件

如果具有不需要的交换空间，则可以删除它。

▼ 如何删除不需要的交换空间

1 成为超级用户。

2 删除该交换空间。

```
# /usr/sbin/swap -d /path/filename
```

删除了交换文件名称，以便它不再可用于交换。文件本身并没有删除。

3 编辑 `/etc/vfstab` 文件，并删除交换文件的项。

4 恢复磁盘空间以便可以将它用于其他目的。

```
# rm /path/filename
```

如果交换空间是一个文件，则删除它。或者，如果交换空间位于单独的片上，并确信不再需要它，则建立新的文件系统并挂载该文件系统。

有关挂载文件系统的信息，请参见第 19 章。

5 检验交换文件是否不再可用。

```
# swap -l
```

示例 21-2 删除不需要的交换空间

以下示例说明如何删除 `/files/swapfile` 交换文件。

```
# swap -d /files/swapfile
```

```
# (Remove the swap entry from the /etc/vfstab file)
```

```
# rm /files/swapfile
```

```
# swap -l
```

```
swapfile          dev  swaplo  blocks  free
/dev/dsk/c0t0d0s1  136,1    16 1638608 1600528
```


检查 UFS 文件系统一致性（任务）

本章提供有关检查 UFS 文件系统一致性的概述信息和逐步说明。

以下是本章中的逐步说明列表。

- 第 440 页中的 “如何从备用引导设备检查根 (/)、/usr 或 /var 文件系统”
- 第 442 页中的 “如何检查其他文件系统（不是根 (/)、/usr 或 /var）”
- 第 445 页中的 “如何整理 UFS 文件系统”
- 第 451 页中的 “如何恢复坏的超级块（Solaris 8、9 和 10 发行版）”
- 第 446 页中的 “如何恢复坏的超级块（仅限 Solaris 10 6/06 发行版）”

以下是本章中概述信息的列表。

- 第 432 页中的 “文件系统一致性”
- 第 432 页中的 “如何记录文件系统的状态”
- 第 433 页中的 “fsck 命令检查和尝试修复的内容”
- 第 439 页中的 “以交互方式检查和修复 UFS 文件系统”
- 第 446 页中的 “恢复坏的超级块”
- 第 453 页中的 “fsck 命令的语法和选项”

有关 Solaris 10 6/06 发行版中 fsck 的新信息，请参见第 328 页中的 “UFS 文件系统实用程序（fsck、mkfs 和 newfs）的增强功能”。

有关 fsck 错误消息的信息，请参见《系统管理指南：高级管理》中的第 28 章“解决 UFS 文件系统不一致问题（任务）”。

有关本章中引用的 UFS 文件系统结构的背景信息，请参见第 23 章。

文件系统一致性

UFS 文件系统依赖于一组内部表来跟踪已用的 `inode` 和可用的块。当这些内部表与磁盘上的数据未正确同步时，会导致不一致性问题，从而需要修复文件系统。

由于以下情况导致操作系统突然终止，文件系统可能会不一致：

- 电源故障
- 意外关闭系统电源
- 没有按正确的关闭过程关闭系统
- 内核中出现软件错误

文件系统不一致性问题尽管严重，但并不常见。引导系统时，会自动执行对文件系统一致性的检查（使用 `fsck` 命令）。通常，此文件系统检查会修复它遇到的问题。

`fsck` 命令将已分配但未引用的文件和目录放置在 `lost+found` 目录中。将 `inode` 编号指定为未引用的文件和目录的名称。如果 `lost+found` 目录不存在，则 `fsck` 命令将创建它。如果 `lost+found` 目录中没有足够的空间，则 `fsck` 命令会增加其大小。

有关 `inode` 的说明，请参见第 456 页中的“`Inode`”。

如何记录文件系统的状态

`fsck` 命令使用存储在超级块中的状态标志来记录文件系统的状态。`fsck` 命令使用此标志来确定是否需要检查文件系统的一致性。`/sbin/rcS` 脚本在引导期间会使用此标志，此外 `fsck -m` 命令也会使用此标志。如果忽略 `fsck -m` 命令的结果，则可以检查所有文件系统，无论状态标志是何设置。

有关超级块的说明，请参见第 456 页中的“`超级块`”。

下表给出了可能的状态标志值。

表 22-1 文件系统状态标志的值

状态标志值	说明
FSACTIVE	指示已挂载的文件系统在内存中具有已修改的数据。如果已挂载文件系统具有此状态标志，则表明当系统断电时，用户数据或元数据将会丢失。
FSBAD	指示文件系统包含不一致的文件系统数据。
FSCLEAN	指示未损坏的、干净取消挂载的文件系统。

表 22-1 文件系统状态标志的值（续）

状态标志值	说明
FSLOG	指示文件系统已启用日志记录功能。设置了此标志的文件系统要么已挂载，要么已取消挂载。如果文件系统已启用日志记录功能，则只能具有标志 FSLOG 或 FSBAD。禁用日志记录功能的文件系统可以具有 FSACTIVE、FSSTABLE 或 FSCLEAN 标志。
FSSTABLE	指示空闲的已挂载文件系统。如果已挂载文件系统具有此状态标志，则表明当系统断电时，用户数据或元数据都不会丢失。

fsck 命令检查和尝试修复的内容

本节介绍文件系统正常运行时发生的情况、可能会出现的问题、fsck 命令（检查和修复实用程序）可查找的问题，以及此命令如何更正它找到的不一致性问题。

为什么可能出现 UFS 文件系统的不一致性问题

在每个工作日中，可能会创建、修改和删除数百个文件。每次修改文件时，操作系统都会执行一系列文件系统更新。如果这些更新被可靠地写入磁盘，便会产生一致的文件系统。

用户程序执行更改文件系统的操作（例如写入操作）时，会先将要写入的数据复制到内核中的核心缓冲区。通常，以异步方式处理磁盘更新。尽管在写入系统调用返回很长时间之后才会写入数据，但是允许用户进程继续执行。这样，在任何给定时间，由于文件系统驻留在磁盘上，因此它将滞后于核心信息所表示的文件系统状态。

当缓冲区需要用于其他用途时，或者内核自动运行 fsflush 守护进程（时间间隔为 30 秒）时，将更新磁盘信息以反映核心信息。如果在未写出核心信息的情况下停止系统，则磁盘上的文件系统可能会处于不一致状态。

文件系统可能由于以下几种原因出现不一致性问题。最常见的原因是操作错误和硬件故障。

异常关机可能会导致此类问题，例如未正确关闭系统，或者未采用正确的方式使已挂载的文件系统脱机。为防止异常关机，在关闭系统、从驱动器中物理移除磁盘组或使磁盘脱机之前，必须将文件系统的当前状态写入磁盘（即“进行同步”）。

不一致性问题也可能是由硬件缺陷或者磁盘或控制器固件的问题导致的。在磁盘驱动器上，块随时都可能会损坏。此外，磁盘控制器可能无法正常工作。

接受一致性检查的 UFS 组件

本节介绍 fsck 命令将针对以下 UFS 文件系统组件执行的各种一致性检查：超级块、柱面组块、inode、间接块和数据块。

有关 UFS 文件系统结构的信息，请参见第 455 页中的“UFS 文件系统的柱面组结构”。

超级块检查

超级块存储摘要信息，它是 UFS 文件系统中最常损坏的组件。对文件系统 `inode` 或数据块进行的每个更改也会修改超级块。如果 CPU 暂停工作且最后一个命令不是 `sync` 命令，则几乎可以肯定超级块会被损坏。

将根据以下内容检查超级块是否存在不一致性问题：

- 文件系统大小
- `inode` 数
- 空闲块计数
- 空闲 `inode` 计数

文件系统大小和 Inode 列表大小检查

文件系统大小必须大于超级块和 `inode` 列表所用的块数。`inode` 数必须小于文件系统所允许的最大数。一个 `inode` 表示一个文件的所有信息。对于 `fsck` 命令，文件系统大小和布局信息是最关键的信息。无法实际检查这些大小，因为它们是在创建文件系统时静态确定的。但是，`fsck` 命令可以检查这些大小是否在合理的范围内。所有其他文件系统检查都要求这些大小正确。如果 `fsck` 命令检测到主超级块的静态参数已损坏，则会要求操作员指定备用超级块的位置。

有关 UFS 文件系统结构的更多信息，请参见第 455 页中的“UFS 文件系统的柱面组结构”。

空闲块检查

空闲块存储在柱面组块图中。`fsck` 命令检查标记为空闲的所有块是否未被任何文件请求。将所有块统计在内后，`fsck` 命令将检查空闲块数与 `inode` 所请求的块数之和是否等于文件系统中的总块数。如果块图出现任何错误，则 `fsck` 命令将重新构建它们，不考虑已分配的块。

超级块中的摘要信息包括文件系统中空闲块总数的计数。`fsck` 命令将此计数与它在文件系统中找到的空闲块数进行比较。如果这两个计数不一致，则 `fsck` 命令会将超级块中的计数替换为实际的空闲块计数。

空闲 Inode 检查

超级块中的摘要信息包含文件系统中空闲 `inode` 的计数。`fsck` 命令将此计数与它在文件系统中找到的空闲 `inode` 数进行比较。如果这两个计数不一致，则 `fsck` 命令会将超级块中的计数替换为实际的空闲 `inode` 计数。

Inode

将按顺序从 inode 2 开始检查 inode 列表（Inode 0 和 inode 1 是保留的）。将根据以下内容检查每个 inode 是否存在不一致性问题：

- 格式和类型
- 链接计数
- 重复块
- 坏块编号
- Inode 大小

Inode 的格式和类型

每个 inode 都包含**模式字**，用于说明 inode 的类型和状态。Inode 可以是以下九种类型之一：

- 常规
- 目录
- 块特殊
- 字符特殊
- FIFO（命名管道）
- 符号链接
- 阴影（用于 ACL）
- 属性目录
- 套接字

Inode 可能处于以下三种状态之一：

- 已分配
- 未分配
- 部分分配

创建文件系统时，会保留固定数目的 inode。但是，仅在需要这些 inode 时才分配它们。**已分配 inode**是指向文件的 inode。**未分配 inode**不指向文件，因此它应该为空。**部分分配**状态表示 inode 未正确格式化。例如，如果因硬件故障而将错误数据写入 inode 列表，则 inode 将变为此状态。fsck 命令可以执行的唯一更正操作是清除 inode。

链接计数检查

每个 inode 都包含与其链接的目录项数的计数。fsck 命令通过从根 (/) 目录开始检查整个目录结构并计算每个 inode 的实际链接计数，验证每个 inode 的链接计数。

存储在 inode 中的链接计数和由 fsck 命令确定的实际链接计数之间的差异可能是以下三种类型之一：

- 存储的计数**不为 0**，而实际计数为 0。
如果 inode 的目录项不存在，则可能出现此情况。在这种情况下，fsck 命令会将已断开的文件放置在 lost+found 目录中。

- 存储的计数不为 0 且实际计数不为 0。但是，这两个计数不相等。

如果已添加或删除目录项，但尚未更新 inode，则可能会出现此情况。在这种情况下，fsck 命令会将存储的链接计数替换为实际的链接计数。

- 存储的计数为 0，而实际计数不为 0。

在这种情况下，fsck 命令会将 inode 的链接计数更改为实际计数。

重复块检查

每个 inode 都包含由 inode 请求的所有块的列表或指向列表的指针（间接块）。由于间接块由 inode 拥有，因此间接块的不一致性问题会直接影响拥有间接块的 inode。

fsck 命令将 inode 请求的每个块编号与已分配块的列表进行比较。如果另一个 inode 已请求某个块编号，则将该块编号放置在重复块列表中。否则，会将已分配块的列表更新为包括该块编号。

如果发现重复块，则 fsck 命令再次遍历 inode 列表，以查找请求每个重复块的其他 inode。fsck 命令不能肯定哪个 inode 出现错误。因此，fsck 命令会提示您选择应保留和应清除的 inode。请注意，inode 中的大量重复块可能是由于未将间接块写入文件系统而导致的。

坏块编号检查

fsck 命令检查 inode 请求的每个块编号，以确定其值是否大于文件系统中第一个数据块的值并小于最后一个数据块的值。如果块编号超出了此范围，则认为它是坏块编号。

inode 中的坏块编号可能是由于未将间接块写入文件系统而导致的。fsck 命令会提示您清除 inode。

Inode 大小检查

每个 inode 都包含它所引用的数据块数的计数。实际数据块的数目等于已分配数据块与间接块的和。fsck 命令计算数据块数，并将该块计数与 inode 请求的块数进行比较。如果 inode 包含的计数不正确，则 fsck 命令会提示您修复它。

每个 inode 都包含一个 64 位大小的字段。此字段说明与 inode 关联的文件中的字符数（数据字节）。粗略检查 inode 的大小字段是否一致时，会使用大小字段中所示的字符数，以计算应与 inode 关联的块数，然后将该块数与 inode 请求的实际块数进行比较。

间接块

间接块由 inode 拥有。因此，间接块的不一致性问题会影响拥有它的 inode。可以检查的不一致性问题如下：

- 已由另一 inode 请求的块
- 超出文件系统范围的块编号

对直接块也执行以上列出的一致性检查。

数据块

inode 可以直接或间接引用三种数据块。引用的所有块必须属于同一种类。这三种类型的数据块如下：

- 纯文本数据块
- 符号链接数据块
- 目录数据块

纯文本数据块包含存储在文件中的信息。**符号链接数据块**包含存储在符号链接中的路径名。**目录数据块**包含目录项。fsck 命令只能检查目录数据块的有效性。

通过 inode 的 mode 字段中的项，可以将目录与常规文件区分开。与目录关联的数据块包含目录项。检查目录数据块是否存在一致性问题涉及以下内容：

- 指向未分配 inode 的目录 inode 编号
- 大于文件系统中 inode 数的目录 inode 编号
- “.”和“..”目录的错误目录 inode 编号
- 从文件系统断开的目录

未分配目录检查

如果目录数据块中的 inode 编号指向未分配的 inode，则 fsck 命令删除该目录项。如果修改并写出包含新目录项的数据块，但未写出 inode，则可能出现此情况。如果突然关闭 CPU，则可能出现此情况。

错误 Inode 编号检查

如果目录项 inode 编号超出了 inode 列表的范围，则 fsck 命令删除该目录项。将错误数据写入目录数据块时，可能出现此情况。

错误“.”和“..”项检查

“.”的目录 inode 编号项必须是目录数据块中的第一项。目录 inode 编号必须引用自身。即，其值必须等于目录数据块的 inode 编号。

“..”的目录 inode 编号项必须是目录数据块中的第二项。目录的 inode 编号值必须等于父目录的 inode 编号或它自己的 inode 编号（如果该目录是根 (/) 目录）。

如果“.”和“..”的目录 inode 编号不正确，则 fsck 命令将它们替换为正确值。如果有多个硬链接指向一个目录，则将找到的第一个硬链接视为“..”应指向的实际父级。在这种情况下，fsck 命令建议您删除其他名称。

断开的目录

fsck 命令检查文件系统的常规连通性。如果找到未链接到文件系统的目录，则 fsck 命令将该目录链接到文件系统的 `lost+found` 目录。当 `inode` 已写入文件系统，但是对应的目录数据块未写入时，可能出现此情况。

常规数据块

与常规文件关联的数据块包含文件的内容。fsck 命令不会尝试检查常规文件的数据块内容的有效性。

fsck 摘要消息

以交互方式运行 fsck 命令且成功完成时，将显示与以下内容类似的消息：

```
# fsck /dev/rdisk/c0t0d0s7

** /dev/rdisk/c0t0d0s7

** Last Mounted on /export/home

** Phase 1 - Check Blocks and Sizes

** Phase 2 - Check Pathnames

** Phase 3 - Check Connectivity

** Phase 4 - Check Reference Counts

** Phase 5 - Check Cyl groups

2 files, 9 used, 2833540 free (20 frags, 354190 blocks, 0.0% fragmentation)

#

# fsck /dev/rdisk/c0t0d0s7

** /dev/rdisk/c0t0d0s7

** Last Mounted on /export/home

** Phase 1 - Check Blocks and Sizes

** Phase 2 - Check Pathnames

** Phase 3a - Check Connectivity
```

```

** Phase 3b - Verify Shadows/ACLs

** Phase 4 - Check Reference Counts

** Phase 5 - Check Cylinder Groups

2 files, 9 used, 2833540 free (20 frags, 354190 blocks, 0.0% fragmentation)

#

```

fsck 输出的最后一行说明有关文件系统的以下信息：

# files	正在使用的 inode 数
# used	正在使用的段数
# free	未使用的段数
# frags	未使用的非块段数
# blocks	未使用的完整块数
% fragmentation	段百分比，其中：空闲段数 x 100 / 文件系统中的总段数

有关段的信息，请参见第 459 页中的“段大小”。

以交互方式检查和修复 UFS 文件系统

在以下情况下可能需要以交互方式检查文件系统：

- 无法挂载文件系统
- 使用文件系统时出现不一致性问题

正在使用的文件系统出现不一致性问题时，可能会在控制台窗口或系统消息文件中显示错误消息。或者，系统可能会崩溃。例如，系统消息文件 `/var/adm/messages` 可能包括与以下内容类似的消息：

```

Sep  5 13:42:40 hostname ufs: [ID 879645 kern.notice] NOTICE: /: unexpected
free inode 630916, run fsck(1M)

```

`hostname` 是报告该错误的系统。

使用 `fsck` 命令之前，您可能希望参阅以下内容以了解有关解决 `fsck` 错误消息的信息：

- 第 453 页中的“`fsck` 命令的语法和选项”

- 《系统管理指南：高级管理》中的第 28 章“解决 UFS 文件系统不一致问题（任务）”

运行 `fsck` 命令检查 UFS 文件系统时，请牢记以下要点：

- 使用 `fsck` 检查文件系统时，文件系统**应该**处于非活动状态。等待刷新到磁盘的文件系统更改或在 `fsck` 检查过程中发生的文件系统更改，可能被解释为文件系统损坏。这些问题不能可靠地表明存在问题。
- 使用 `fsck` 修复文件系统时，文件系统**必须**处于非活动状态。等待刷新到磁盘的文件系统更改或在 `fsck` 修复过程中发生的文件系统更改，可能导致文件系统损坏。或者，它们可能导致系统崩溃。
- 在文件系统上使用 `fsck` 之前，取消挂载文件系统。这样做可确保文件系统的数据结构尽可能一致。活动根 (`/`)、`/usr` 和 `/var` 文件系统除外，因为必须挂载它们才能运行 `fsck`。
- 如果需要修复根 (`/`)、`/usr` 和 `/var` 文件系统，应尽可能从备用设备引导系统，以便这些文件系统处于非挂载和非活动状态。

有关在根 (`/`)、`/usr` 或 `/var` 文件系统上运行 `fsck` 的逐步说明，请参见第 440 页中的“[如何从备用引导设备检查根 \(`/`\)、`/usr` 或 `/var` 文件系统](#)”。

▼ 如何从备用引导设备检查根 (`/`)、`/usr` 或 `/var` 文件系统

有关 Solaris 10 6/06 发行版中 `fsck` 的新信息，请参见第 328 页中的“[UFS 文件系统实用程序（`fsck`、`mkfs` 和 `newfs`）的增强功能](#)”。如果看到以下消息，则无需重新运行 `fsck`：

```
***** FILE SYSTEM WAS MODIFIED *****
```

但是，在出现此消息后重新运行 `fsck` 并不会损害文件系统。此消息只是有关 `fsck` 的更正操作的信息。

此过程假定本地 CD 或网络引导服务器可用，从而可以从备用设备引导系统。

有关恢复坏的超级块的信息，请参见第 446 页中的“[如何恢复坏的超级块（仅限 Solaris 10 6/06 发行版）](#)”或第 451 页中的“[如何恢复坏的超级块（Solaris 8、9 和 10 发行版）](#)”。

- 1 成为超级用户或承担等效角色。

- 2 仅适用于具有镜像根 (/) 文件系统的系统：在从备用设备进行引导之前分离根 (/) 镜像，否则会有损坏文件系统的风险。
有关分离根 (/) 镜像的信息，请参见《Solaris Volume Manager 管理指南》中的“处理子镜像”。
- 3 识别需要检查的根 (/)、/usr 或 /var 文件系统的设备，例如 /dev/dsk/c0t0d0s0。
从备用设备进行引导时，将需要提供此设备名称。已从备用设备引导后识别此设备会更困难。
- 4 在单用户模式下，从备用设备（如本地 CD 或网络）引导具有需要检查的根 (/)、/usr 或 /var 文件系统的系统。
这样做可确保在这些文件系统上没有任何活动。

例如：

```
# init 0
```

```
ok boot net -s
```

```
.
```

```
.
```

```
.
```

```
#
```

- 5 检查包含步骤 3 中识别的根 (/)、/usr 或 /var 文件系统的设备。
如果要检查或修复的文件系统的硬件已更改，则设备名称可能已更改。检查 fsck -n 消息 Last Mounted on ... 是否指示文件系统的预期设备。

在此示例中，要检查的根 (/) 文件系统是 /dev/dsk/c0t0d0s0。

```
# fsck -n /dev/rdisk/c0t0d0s0
```

```
** /dev/rdisk/c0t0d0s0 (NO WRITE)
```

```
** Last Mounted on /
```

```
.
```

```
.
```

```
.
```

```
fsck /dev/rdisk/c0t0d0s0
```

```

** /dev/rdisk/c0t0d0s0

** Last Mounted on /

** Phase 1 - Check Blocks and Sizes

** Phase 2 - Check Pathnames

.

.

.

```

6 更正任何报告的 fsck 错误。

有关如何响应以交互方式检查一个或多个 UFS 文件系统时出现的错误消息，请参见《系统管理指南：高级管理》中的第 28 章“解决 UFS 文件系统不一致问题（任务）”。

7 如果在运行 fsck 后无法修复所有问题，请参见第 445 页中的“修复 fsck 命令无法修复的 UFS 文件系统。”。

8 挂载已修复的文件系统，以确定 lost+found 目录中是否存在任何文件。

由 fsck 命令放置在 lost+found 目录中的各个文件是使用其 inode 编号重命名的。如有可能，请重命名这些文件，并将它们移动到所属的位置。请尝试使用 grep 命令匹配各个文件中的短语，并尝试使用 file 命令确定文件类型。

最后，删除遗留在 lost+found 目录中的无法识别的文件或目录，以免该目录不必要地被填满。

9 使系统返回到多用户模式。

```
# init 6
```

10 仅适用于具有镜像根 (/) 文件系统的系统：重新连接根 (/) 镜像。

▼ 如何检查其他文件系统（不是根 (/)、/usr 或 /var）

有关 Solaris 10 6/06 发行版中 fsck 的新信息，请参见第 328 页中的“UFS 文件系统实用程序（fsck、mkfs 和 newfs）的增强功能”。如果看到以下消息，则无需重新运行 fsck：

```
***** FILE SYSTEM WAS MODIFIED *****
```

但是，在出现此消息后重新运行 `fsck` 并不会损害文件系统。此消息只是有关 `fsck` 的更正操作的信息。

此过程假定已取消挂载要检查的文件系统。

有关恢复坏的超级块的信息，请参见第 446 页中的“如何恢复坏的超级块（仅限 Solaris 10 6/06 发行版）”或第 451 页中的“如何恢复坏的超级块（Solaris 8、9 和 10 发行版）”。

1 成为超级用户或承担等效角色。

2 取消挂载本地文件系统以确保文件系统上没有任何活动。

将挂载点目录或 `/dev/dsk/device-name` 指定为 `fsck` 命令的参数。将显示有关不一致性问题的所有消息。

例如：

```
# umount /export/home

# fsck /dev/rdisk/c0t0d0s7

** /dev/dsk/c0t0d0s7

** Last Mounted on /export/home

.

.

.
```

3 更正任何报告的 `fsck` 错误。

有关如何响应以交互方式检查一个或多个 UFS 文件系统时出现的错误消息，请参见《系统管理指南：高级管理》中的第 28 章“解决 UFS 文件系统不一致问题（任务）”。

4 如果在运行 `fsck` 后无法修复所有问题，请参见第 445 页中的“修复 `fsck` 命令无法修复的 UFS 文件系统。”。

5 挂载已修复的文件系统，以确定 `lost+found` 目录中是否存在任何文件。

由 `fsck` 命令放置在 `lost+found` 目录中的单独文件是使用其 `inode` 编号重命名的。

6 重命名并移动放置在 `lost+found` 目录中的任何文件。

如有可能，请重命名这些文件，并将它们移动到所属的位置。请尝试使用 `grep` 命令匹配各个文件中的短语，并尝试使用 `file` 命令确定文件类型。

最后，删除遗留在 `lost+found` 目录中的无法识别的文件或目录，以免该目录不必要地被填满。

示例 22-1 以交互方式检查非根 (/) 或非 /usr 文件系统

以下示例说明如何检查 `/dev/rdisk/c0t0d0s6` 文件系统并更正不正确的块计数。此示例假定已取消挂载文件系统。

```
# fsck /dev/rdisk/c0t0d0s6

** Phase 1 - Check Block and Sizes

INCORRECT BLOCK COUNT I=2529 (6 should be 2)

CORRECT? y


** Phase 2 - Check Pathnames

** Phase 3 - Check Connectivity

** Phase 4 - Check Reference Counts

** Phase 5 - Cylinder Groups

929 files, 8928 used, 2851 free (75 frags, 347 blocks, 0.6%
fragmentation)

***** FILE SYSTEM WAS MODIFIED *****

#
```

整理 UFS 文件系统

`fsck -op` 命令（`p` 表示整理）检查 UFS 文件系统，并自动解决通常因系统意外关闭而导致的问题。如果此命令遇到要求操作员干预的问题，则它会立即退出。此命令还允许并行检查文件系统。

在异常关机后，可以运行 `fsck -op` 命令以整理文件系统。在此模式下，`fsck` 命令不查看“干净”标志，而是执行完整检查。这些操作是 `fsck` 命令以交互方式运行时所执行操作的子集。

▼ 如何整理 UFS 文件系统

此过程假定文件系统已取消挂载或处于非活动状态。

1 成为超级用户或承担等效角色。

2 取消挂载 UFS 文件系统。

```
# umount /mount-point
```

3 用整理选项检查 UFS 文件系统。

```
# fsck -o p /dev/rdisk/device-name
```

通过将 */mount-point* 或 */dev/rdisk/device-name* 用作 `fsck` 命令的参数，可以整理单独的文件系统。

示例 22-2 整理 UFS 文件系统

以下示例说明如何整理 */export/home* 文件系统。

```
# fsck -o p /export/home
```

修复 `fsck` 命令无法修复的 UFS 文件系统。

`fsck` 命令运行若干遍，在稍后的一遍中更正的问题可能会暴露仅在前几遍中检测到的其他问题。因此，有时需要一直运行 `fsck`，直到它不再报告任何问题。这样做可确保找出并修复所有错误。

请注意 `fsck` 命令所显示的信息。此信息可能有助于您解决问题。例如，消息可能会指出损坏的目录。如果删除该目录，则可能发现 `fsck` 命令不再报告任何错误。

如果 `fsck` 命令仍无法修复文件系统，请尝试使用 `ff`、`clri` 和 `ncheck` 命令找出并修复问题。有关如何使用这些命令的信息，请参见以下内容：

- `fsdb(1M)`
- `ff(1M)`
- `clri(1M)`
- `ncheck(1M)`

最后，可能需要重新创建文件系统，然后从备份介质恢复其内容。

有关恢复完整文件系统的信息，请参见[第 27 章](#)。

如果无法完全修复文件系统，但可以将它挂载为只读，请尝试使用 `cp`、`tar` 或 `cpio` 命令从文件系统检索所有数据或部分数据。

如果问题是由硬件磁盘错误导致的，则在重新创建和恢复文件系统之前，可能需要再次重新格式化磁盘并对其重新分区。在更换磁盘设备之前，请检查设备电缆和连接器是否正常工作。硬件错误通常会在使用不同的命令时一再显示同一错误。`format` 命令尝试修复磁盘上的坏块。但是，如果磁盘损坏得太严重，则问题可能会一直存在，即使重新格式化后也是如此。有关使用 `format` 命令的信息，请参见 `format(1M)`。有关安装新磁盘的信息，请参见第 13 章或第 14 章。

恢复坏的超级块

当文件系统的超级块损坏时，必须恢复它。`fsck` 命令会在超级块已损坏的时候告知您。好在文件系统中存储有超级块的多个副本。

可以使用 `fsck -o b` 命令将超级块替换为其中一个副本，或使用 `fsck` 的自动搜索备份超级块功能（Solaris 10 6/06 发行版中的新增功能）。有关此功能的更多信息，请参见第 329 页中的“自动搜索备份超级块”。

有关超级块的更多信息，请参见第 456 页中的“超级块”。

如果根 (/) 文件系统中的超级块损坏，而且您无法恢复它，则您有以下两种选择：

- 重新安装系统。
- 从网络或本地 CD 进行引导，然后尝试执行以下步骤。如果这些步骤失败，请使用 `newfs` 命令重新创建根 (/) 文件系统，然后从备份副本恢复它。

▼ 如何恢复坏的超级块（仅限 Solaris 10 6/06 发行版）

此过程是 Solaris 10 6/06 发行版的新增内容。如果文件系统具有坏的超级块，则 `fsck` 会自动计算备用超级块，如以下消息所示：

```
BAD SUPERBLOCK AT ...
```

```
LOOK FOR ALTERNATE SUPERBLOCKS WITH MKFS?
```

```
LOOK FOR ALTERNATE SUPERBLOCKS WITH NEWFS?
```



注意 – 如果具有损坏的超级块的文件系统是使用 `newfs` 或 `mkfs` 自定义参数（例如 `ntrack` 或 `nsect`）创建的，则使用 `fsck` 自动计算的超级块执行修复过程可能会对文件系统造成无法恢复的损坏。

如果文件系统是使用自定义参数创建的，并且它具有坏的超级块，则 `fsck` 提供以下提示以取消 `fsck` 会话：

```
CANCEL FILESYSTEM CHECK?
```

如果此文件系统是使用自定义参数创建的，或者在此文件系统上运行 `fsck` 可能会带来其他问题，则应取消 `fsck` 会话。

- 1 成为超级用户或承担等效角色。
- 2 检查怀疑有坏的超级块的文件系统。

```
# fsck /dev/rdisk/c0t1d0s0
```

```
** /dev/rdisk/c0t1d0s0
```

```
BAD SUPERBLOCK at ...
```

- 3 确定文件系统是如何创建的，然后选择以下项之一：
 - 文件系统是使用 `newfs` 命令创建的。
 - `fsck` 响应所有超级块都已损坏，而且必须使用通用超级块。按以下示例所示应答 `fsck` 提示。



注意 – 如果文件系统是使用自定义参数创建的，则不要使用此选项。仅应在没有其他方法时使用此选项。为从备份副本恢复文件系统做好准备。

```
# fsck /dev/dsk/c1t2d0s0
```

```
** /dev/rdisk/c1t2d0s0
```

```
BAD SUPERBLOCK AT BLOCK 16: BLOCK SIZE LARGER THAN MAXIMUM SUPPORTED
```

```
LOOK FOR ALTERNATE SUPERBLOCKS WITH MKFS? no
```

LOOK FOR ALTERNATE SUPERBLOCKS WITH NEWFS? **yes**

SEARCH FOR ALTERNATE SUPERBLOCKS FAILED.

USE GENERIC SUPERBLOCK FROM MKFS? **no**

USE GENERIC SUPERBLOCK FROM NEWFS? **yes**

CALCULATED GENERIC SUPERBLOCK WITH NEWFS

If filesystem was created with manually-specified geometry, using
auto-discovered superblock may result in irrecoverable damage to
filesystem and user data.

CANCEL FILESYSTEM CHECK? **no**

** Last Mounted on

** Phase 1 - Check Blocks and Sizes

** Phase 2 - Check Pathnames

** Phase 3a - Check Connectivity

** Phase 3b - Verify Shadows/ACLs


```
** Phase 4 - Check Reference Counts
```

```
** Phase 5 - Check Cylinder Groups
```

```
CORRECT GLOBAL SUMMARY
```

```
SALVAGE? y
```

```
UPDATE STANDARD SUPERBLOCK? y
```

```
81 files, 3609 used, 244678 free (6 frags, 30584 blocks, 0.0% fragmentation)
```

```
***** FILE SYSTEM WAS MODIFIED *****
```

- fsck 响应它找到备用超级块，并显示与以下内容类似的消息：

```
FOUND ALTERNATE SUPERBLOCK 32 WITH NEWFS
```

在此 fsck 方案中，按照第 329 页中的“自动搜索备份超级块”所示的提示操作。

- 文件系统是使用 **mkfs** 命令创建的。
- fsck 响应所有超级块都已损坏，而且必须使用通用超级块。按以下示例所示应答 fsck 提示。



注意 – 如果文件系统是使用自定义参数创建的，则不要使用此选项。仅应在没有其他方法时使用此选项。为从备份副本恢复文件系统做好准备。

```
# fsck /dev/dsk/c1t2d0s0
```

```
** /dev/rdisk/c1t2d0s0
```

```
BAD SUPERBLOCK AT BLOCK 16: BLOCK SIZE LARGER THAN MAXIMUM SUPPORTED
```

```
LOOK FOR ALTERNATE SUPERBLOCKS WITH MKFS? yes
```

LOOK FOR ALTERNATE SUPERBLOCKS WITH NEWFS? **no**

SEARCH FOR ALTERNATE SUPERBLOCKS FAILED.

USE GENERIC SUPERBLOCK FROM MKFS? **yes**

CALCULATED GENERIC SUPERBLOCK WITH MKFS

If filesystem was created with manually-specified geometry, using
auto-discovered superblock may result in irrecoverable damage to
filesystem and user data.

CANCEL FILESYSTEM CHECK? **no**

**** Last Mounted on**

**** Phase 1 - Check Blocks and Sizes**

**** Phase 2 - Check Pathnames**

**** Phase 3a - Check Connectivity**

**** Phase 3b - Verify Shadows/ACLs**

**** Phase 4 - Check Reference Counts**

**** Phase 5 - Check Cylinder Groups**

CORRECT GLOBAL SUMMARY

SALVAGE? **y**

```
UPDATE STANDARD SUPERBLOCK? y
```

```
81 files, 3609 used, 243605 free (117 frags, 30436 blocks, 0.0% fragmentation)
```

- fsck 响应它找到备用超级块，并显示与以下内容类似的消息：

```
FOUND ALTERNATE SUPERBLOCK 32 WITH MKFS
```

在此 fsck 方案中，按照第 329 页中的“自动搜索备份超级块”所示的提示操作。

4 应答提示以挽救和恢复超级块。

在看到以下消息时，无需重新运行 fsck：

```
***** FILE SYSTEM WAS MODIFIED *****
```

但是，在出现此消息后重新运行 fsck 并不会损害文件系统。此消息只是有关 fsck 的更正操作的信息。

▼ 如何恢复坏的超级块（Solaris 8、9 和 10 发行版）

1 成为超级用户或承担等效角色。

2 确定根 (/)、/usr 或 /var 文件系统中是否存在坏的超级块，然后选择以下项之一：

- 如果根 (/)、/usr 或 /var 文件系统中存在坏的超级块，则从网络或本地连接的 CD 进行引导。

从本地连接的 CD 进行引导时使用以下命令：

```
ok boot cdrom -s
```

从已设置引导服务器或安装服务器的网络进行引导时使用以下命令：

```
ok boot net -s
```

如果需要有关如何停止系统的帮助，请参见《系统管理指南：基本管理》中的第 10 章“引导系统（任务）”或《系统管理指南：基本管理》中的第 11 章“基于 GRUB 的引导（任务）”。

- 如果根 (/)、/usr、/var 文件系统中不存在坏的超级块，则转到损坏的文件系统之外的目录，并取消挂载文件系统。

```
# umount /mount-point
```



注意 – 务必在下一步中使用 `newfs -N`。如果省略 `-N` 选项，则会损坏文件系统的所有数据，并将它替换为空的文件系统。

3 使用 `newfs -N` 命令显示超级块的值。

```
# newfs -N /dev/rdisk/device-name
```

除非文件系统是使用特殊参数创建的，否则命令输出将显示 `newfs` 命令创建文件系统时用于超级块副本的块编号。有关创建自定义文件系统的信息，请参见第 458 页中的“自定义 UFS 文件系统参数”。

4 使用 `fsck` 命令提供备用超级块。

```
# fsck -F ufs -o b=block-number /dev/rdisk/device-name
```

`fsck` 命令使用指定的备用超级块恢复主超级块。可以始终尝试将 32 作为备用块。或者，使用 `newfs -N` 命令所示的任何备用块。

示例 22-3 恢复坏的超级块（Solaris 8、9 和 10 发行版）

以下示例说明如何恢复超级块副本 5264。

```
# newfs -N /dev/rdisk/c0t3d0s7
```

```
/dev/rdisk/c0t3d0s7: 163944 sectors in 506 cylinders of 9 tracks, 36 sectors
```

```
83.9MB in 32 cyl groups (16 c/g, 2.65MB/g, 1216 i/g)
```

```
super-block backups (for fsck -b #) at:
```

```
32, 5264, 10496, 15728, 20960, 26192, 31424, 36656, 41888,
```

```
47120, 52352, 57584, 62816, 68048, 73280, 78512, 82976, 88208,
```

```
93440, 98672, 103904, 109136, 114368, 119600, 124832, 130064, 135296,
```

```
140528, 145760, 150992, 156224, 161456,
```

```
# fsck -F ufs -o b=5264 /dev/rdisk/c0t3d0s7
```

```
Alternate superblock location: 5264.
```

```
** /dev/rdisk/c0t3d0s7

** Last 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

36 files, 867 used, 75712 free (16 frags, 9462 blocks, 0.0% fragmentation)

***** FILE SYSTEM WAS MODIFIED *****

#
```

fsck 命令的语法和选项

fsck 命令检查和修复文件系统中的不一致性问题。如果运行不带任何选项的 fsck 命令，则该命令会在进行修复之前以交互方式要求进行确认。此命令有四个选项。

命令和选项	说明
fsck -m	检查是否可以挂载文件系统
fsck -y	接受所有修复
fsck -n	拒绝所有修复
fsck -o p	以非交互方式整理文件系统，解决所有预期的（无害的）不一致性问题，但是在遇到严重问题时退出

UFS 文件系统（参考）

以下是本章中参考信息的列表。

- [第 455 页中的“UFS 文件系统的柱面组结构”](#)
- [第 458 页中的“自定义 UFS 文件系统参数”](#)

UFS 文件系统的柱面组结构

创建 UFS 文件系统时，磁盘片被分成若干个柱面组。**柱面组**由一个或多个连续的磁盘柱面组成。柱面组又进一步分成若干个可寻址的块，以控制和组织柱面组中文件的结构。在文件系统中，每种类型的块都具有特定的功能。UFS 文件系统具有以下四种类型的块。

块类型	所存储信息的类型
引导块	在引导系统时使用的信息
超级块	有关文件的详细信息
Inode	有关文件的所有信息
存储块或数据块	每个文件的数据

以下各节提供有关这些块的组织和功能的其他信息。

引导块

引导块存储在引导系统时使用的对象。如果文件系统不用于引导，则将引导块保留为空。引导块仅出现在第一个柱面组（柱面组 0）中，它是片中的前 8 KB。

超级块

超级块存储有关文件系统的大多数信息，其中包括：

- 文件系统的大小和状态
- 标号，包括文件系统名称和卷名称
- 文件系统逻辑块的大小
- 上次更新的日期和时间
- 柱面组的大小
- 柱面组中的数据块数
- 摘要数据块
- 文件系统状态
- 最后一个挂载点的路径名

由于超级块包含关键数据，因此在创建文件系统时建立了多个超级块。

摘要信息块保留在超级块内。不复制摘要信息块，而是将其与主超级块组合在一起（通常在柱面组 0 中）。摘要块记录在使用文件系统时发生的更改。此外，摘要块列出文件系统中的 inode、目录、段和存储块的数目。

Inode

inode 包含有关文件的所有信息，但文件的名称（保存在目录中）除外。一个 inode 为 128 字节。inode 信息保存在柱面信息块中，它包含以下内容：

- 文件类型：
 - 常规
 - 目录
 - 块特殊
 - 字符特殊
 - FIFO，也称为命名管道
 - 符号链接
 - 套接字
 - 其他 inode—属性目录和阴影（用于 ACL）
- 文件的模式（读-写-执行权限集）
- 指向文件的硬链接数
- 文件属主的用户 ID
- 文件所属的组 ID
- 文件中的字节数
- 包含 15 个磁盘块地址的数组
- 上次访问文件的日期和时间
- 上次修改文件的日期和时间

■ 更改 inode 的日期和时间

包含 15 个磁盘块地址（0 到 14）的数组指向存储文件内容的数据块。前 12 个地址是直接地址。即，它们直接指向文件内容的前 12 个逻辑存储块。如果文件大于 12 个逻辑块，则第 13 个地址指向间接块，该块包含直接块地址而不是文件内容。第 14 个地址指向双重间接块，该块包含间接块的地址。第 15 个地址用于三重间接地址。下图描述从 inode 开始这些地址块之间的关系链。

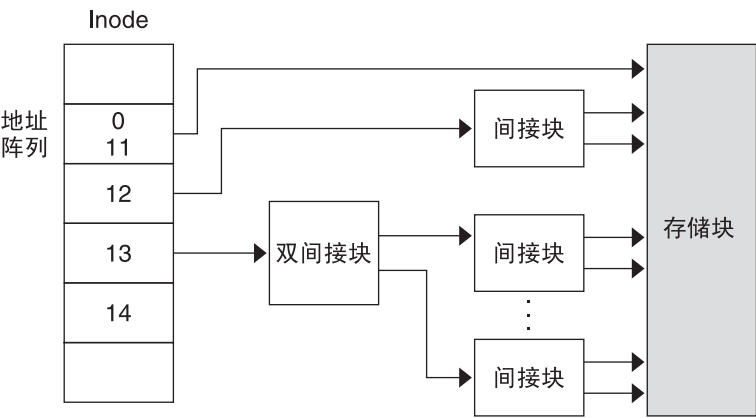


图 23-1 UFS 文件系统的地址链

数据块

数据块也称为**存储块**，它包含为文件系统分配的其余空间。这些数据块的大小是在创建文件系统时确定的。缺省情况下，为数据块分配以下两种大小：8 KB 的逻辑块大小和 1 KB 的段大小 (fragment size)。

对于常规文件，数据块包含文件的内容。对于目录，数据块包含提供目录中文件的 inode 编号和文件名的项。

空闲块

在柱面组图中，当前未用作 inode、间接地址块或存储块的块被标记为空闲。此图还跟踪段以防止段化降低磁盘的性能。

为使您了解典型的 UFS 文件系统结构，下图说明普通 UFS 文件系统的一系列柱面组。



图 23-2 典型的 UFS 文件系统

自定义 UFS 文件系统参数

在更改由 `newfs` 命令指定的缺省文件系统参数之前，需要了解这些参数。本节介绍这些参数：

- 第 458 页中的“逻辑块大小”
- 第 459 页中的“段大小”
- 第 459 页中的“最小空闲空间”
- 第 460 页中的“旋转延迟”（过时）
- 第 460 页中的“优化类型”
- 第 460 页中的“Inode 数（文件）”

有关自定义这些参数的命令选项的说明，请参见 `newfs(1M)` 和 `mkfs_ufs(1M)`。

逻辑块大小

逻辑块大小是 UNIX® 内核用来读取或写入文件的块的大小。逻辑块大小通常与物理块大小不同。物理块大小通常为 512 字节，这是磁盘控制器可以读取或写入的最小块的大小。

缺省情况下，逻辑块大小设置为系统的页面大小。对于 UFS 文件系统，缺省的逻辑块大小为 8192 字节 (8 KB)。UFS 文件系统支持 4096 或 8192 字节（4 或 8 KB）的块大小。建议的逻辑块大小为 8 KB。

仅适用于 SPARC – 在 sun-4u™ 平台上，只能指定 8192 字节的块大小。

要选择最适合您系统的逻辑块大小，应同时考虑所需性能和可用空间。对于大多数 UFS 系统，8 KB 的文件系统可提供最佳的性能，在磁盘性能与主内存和磁盘的空间使用之间实现很好的平衡。

通常，当大多数文件都非常大时，要提高效率，应为文件系统使用较大的逻辑块大小。当大多数文件都非常小时，应为文件系统使用较小的逻辑块大小。可以对文件系统使用 `quot -c filesystem` 命令，显示有关按块大小分布文件的完整报告。

但是，在大多数情况下，在创建文件系统时设置的页面大小很可能是最佳大小。

段大小

在创建或扩展文件时，会按完整逻辑块或部分逻辑块（称为**段**）为其分配磁盘空间。当文件需要磁盘空间时，首先分配完整块，然后为剩余部分分配某个块的一个或多个段。对于小文件，首先分配段。

能够为文件分配块中段而不是仅分配完整块，就减少了块中未使用的孔洞所导致的磁盘空间**段化**，从而节省了空间。

在创建 UFS 文件系统时，可定义**段大小**。缺省的段大小为 1 KB。每个块都可以分成 1 个、2 个、4 个或 8 个段，因此段大小从 8192 字节到 512 字节（仅适用于 4 KB 文件系统）不等。下限实际上为磁盘扇区大小（通常是 512 字节）。

对于多 TB 文件系统，段大小必须等于文件系统的块大小。

注 - 段的上限为逻辑块大小，在这种情况下段根本不是段。如果相对于空间而言，您更关心速度，则此配置对于具有非常大的文件的文件系统可能是最佳的。

选择段大小时，请考虑时间和空间之间的平衡：小的段大小可节省空间，但是需要更多的时间进行分配。通常，当大多数文件都很大时，要提高存储效率，应为文件系统使用较大的段大小。当大多数文件都很小时，应为文件系统使用较小的段大小。

最小空闲空间

最小空闲空间是在创建文件系统时保留的磁盘空间占总磁盘空间的百分比。缺省的保留空间为 $((64 \text{ MB}/\text{分区大小}) * 100)$ ，向下舍入到最接近的整数，且介于 1% 与 10%（包括 1% 和 10%）之间。

空闲空间是很重要的，因为随着文件系统变满，文件访问的效率会越来越低。只要存在足够的空闲空间，UFS 文件系统就可以有效地运行。当文件系统变满，用尽可用的用户空间时，只有超级用户才能访问保留的空闲空间。

诸如 `df` 之类的命令报告用户可用空间的百分比，不包括作为最小空闲空间分配的百分比。当命令报告文件系统中正使用的磁盘空间超过 100% 时，说明一些保留空间已被超级用户使用。

如果将配额强加于用户，则可供他们使用的空间不包括保留的空闲空间。可以使用 `tunefs` 命令更改现有文件系统的最小空闲空间值。

旋转延迟

此参数已过时。不管指定什么值，该值始终设置为 0。

优化类型

优化类型参数设置为空间或时间。

- **空间**—选择空间优化时，会以最大程度地减小段化的方式分配磁盘块，同时会使磁盘的使用得到优化。
- **时间**—选择时间优化时，将尽可能快速地分配磁盘块，而不太注重其位置。如果存在足够的空闲空间，则分配磁盘块相对容易，而不会导致段化程度过高。缺省值是时间。

可以使用 `tunefs` 命令为现有文件系统更改优化类型参数的值。

有关更多信息，请参见 `tunefs(1M)`。

Inode 数（文件）

每个 `inode` 的字节数指定文件系统中的 `inode` 密度。用文件系统的总大小除以该数，就可以确定要创建的 `inode` 数。分配 `inode` 后，不重新创建文件系统，就无法更改该数。

如果文件系统小于 1 GB，则每 `inode` 的缺省字节数为 2048 字节 (2 KB)。如果文件系统大于 1 GB，则使用以下公式：

文件系统大小	每 Inode 的字节数
小于或等于 1 GB	2048
小于 2 GB	4096
小于 3 GB	6144
3 GB 至 1 TB	8192
大于 1 TB 或是使用 <code>-T</code> 选项创建的	1048576

如果使用的文件系统包含许多符号链接，则这些符号链接可以降低平均文件大小。如果文件系统将要包含许多小文件，则可以为此参数指定一个较小的值。但是，请注意具有太多的 `inode` 比 `inode` 不足要好得多。如果具有太少的 `inode`，则可能会在磁盘片还几乎为空时达到最大文件数。

UFS 文件和文件系统的最大大小

UFS 文件系统的最大大小约为 16 TB 的可用空间减去约 1% 的开销。**稀疏**文件可以具有 1 TB 的逻辑大小。但是，由于文件系统开销所致，可以存储在文件中的实际数据量比 1 TB 大约小 1%。

最大 UFS 子目录数

UFS 文件系统中每目录的最大子目录数为 32,767。此限制是预定义的，不能进行更改。

备份和恢复文件系统（概述）

本章提供有关使用 `ufsdump` 和 `ufsrestore` 命令备份和恢复文件系统的指南和规划信息。

以下是本章中概述信息的列表。

- 第 463 页中的 “有关备份和恢复任务的参考信息”
- 第 464 页中的 “备份和恢复文件系统介绍”
- 第 464 页中的 “备份文件系统的原因”
- 第 465 页中的 “应该备份的文件系统”
- 第 466 页中的 “选择备份类型”
- 第 467 页中的 “选择磁带设备”
- 第 467 页中的 “备份和恢复文件系统的高级视图（任务列表）”
- 第 468 页中的 “计划备份注意事项”
- 第 471 页中的 “样例备份计划”

有关备份和恢复任务的参考信息

备份或恢复任务	更多信息
使用 <code>ufsdump</code> 命令备份文件系统。	第 25 章
使用 <code>fssnap</code> 命令创建 UFS 快照。	第 26 章
使用 <code>ufsrestore</code> 命令恢复文件系统。	第 27 章
使用 <code>cpio</code> 、 <code>dd</code> 、 <code>pax</code> 和 <code>cpio</code> 命令复制文件和目录。	第 29 章

备份和恢复文件系统介绍

备份文件系统是指将文件系统复制到可移除介质（如磁带），以防止其丢失、损坏或破坏。**恢复**文件系统是指采用适当的方式将当前的备份文件从可移除介质复制到工作目录。

本章介绍了用于备份和恢复 UFS 文件系统的 `ufsdump` 和 `ufsrestore` 命令。其他命令可用于复制文件和文件系统，以实现文件共享或传输。下表提供了用于将各文件和文件系统复制到其他介质的所有命令的链接。

表 24-1 用于备份和恢复文件和文件系统的命令

任务	命令	更多信息
将一个或多个文件系统备份到本地磁带设备或远程磁带设备。	<code>ufsdump</code>	第 25 章 或 第 28 章
创建文件系统的只读副本。	<code>fssnap</code>	第 26 章
通过备份服务器备份网络上系统的所有文件系统。	Solstice Backup 软件	《Solstice Backup 6.1 Administration Guide》
备份和恢复 NIS+ 主服务器。	<code>nisbackup</code> 和 <code>nisrestore</code>	《System Administration Guide: Naming and Directory Services (NIS+)》
复制、列出和检索磁带或软盘上的文件。	<code>tar</code> , <code>cpio</code> 或 <code>pax</code>	第 29 章
从主磁盘复制到克隆磁盘。	<code>dd</code>	第 29 章
将完整的文件系统或各个文件从可移除介质恢复到工作目录。	<code>ufsrestore</code>	第 27 章

备份文件系统的原因

备份文件是最重要的系统管理功能之一。应该定期按计划执行备份，以防止因以下类型的问题造成数据丢失：

- 系统崩溃
- 无意中删除文件
- 硬件故障
- 自然灾害，如火灾、飓风或地震
- 重新安装或升级系统时出现问题

应该备份的文件系统

应该备份对用户非常重要的所有文件系统，包括频繁更改的文件系统。下表提供了有关为独立系统和服务器进行文件系统备份的一般指南。

表 24-2 为独立系统备份的文件系统

要备份的文件系统	说明	备份间隔
根目录 (/) – 片 0	此文件系统包含内核，并可能包含 /var 目录。/var 目录包含临时文件、日志文件或状态文件，而且可能包含频繁更新的系统记帐和邮件文件。	定期，如每周或每天
/usr – 片 6， /opt	/usr 和 /opt 文件系统包含软件和可执行文件。/opt 目录或者是根目录 (/) 的一部分，或者是其自己的文件系统。	偶尔
/export/home – 片 7	此文件系统可以包含独立系统上的所有用户的目录和子目录。	比根目录 (/) 或 /usr 的备份更频繁，可以每天备份一次，具体取决于站点的需要
/export, /var 或其他文件系统	/export 文件系统可以包含无盘客户机的内核和可执行文件。/var 目录包含临时文件、日志文件或状态文件。	根据站点要求

表 24-3 要为服务器备份的文件系统

要备份的文件系统	说明	备份间隔
根目录 (/) – 片 0	此文件系统包含内核和可执行文件。	每天一次到每月一次，具体取决于站点的需要。 如果在网络上频繁地添加及删除用户和系统，则必须更改此文件系统中的配置文件。在这种情况下，应该以介于每周备份一次和每月备份一次之间的频率对根 (/) 文件系统执行完整备份。 如果站点将用户邮件存放在邮件服务器的 /var/mail 目录（即客户机随后挂载的目录）中，则可能需要每天都备份根目录 (/)。或者，如果它是单独的文件系统，则需要备份 /var 目录。

表 24-3 要为服务器备份的文件系统（续）

要备份的文件系统	说明	备份间隔
/export – 片 3	此文件系统可以包含无盘客户机的内核和可执行文件。	每天一次到每月一次，具体取决于站点的需要。 由于此文件系统与信息片 0 中的服务器根目录类似，因此此文件系统不会频繁发生更改。如果站点不向客户机系统发送邮件，则只需偶尔备份此文件系统即可。否则，应该更频繁地备份 /export。
/usr – 片 6， /opt	/usr 和 /opt 文件系统包含软件和可执行文件。 /opt 目录或者是根目录 (/) 的一部分，或者是其自己的文件系统。	每天一次到每月一次，具体取决于站点的需要。 除非频繁地添加或删除软件，否则这些文件系统通常是静态的。
/export/home – 片 7	此文件系统可以包含系统中所有用户的起始目录。 此文件系统中的文件是可变文件。	每天一次到每周一次。

选择备份类型

可以使用 `ufsdump` 命令执行完整备份或增量备份。可以使用 `fssnap` 命令创建文件系统的临时映像。下表列出了这些备份过程类型之间的差异。

表 24-4 各种备份类型之间的差异

备份类型	结果	优点	缺点
完整	复制完整的文件系统或目录	所有数据都处于一个位置中	需要大量备份磁带，写入时间很长。由于驱动器必须按顺序移至磁带上文件所在的点，因此检索各个文件需较长时间。可能必须搜索多个磁带。
快照	创建文件系统的临时映像	系统可以处于多用户模式下	创建快照时，系统性能可能会降低。
增量	仅复制指定文件系统中自以前某一次备份以来发生更改的那些文件	更容易检索文件系统的小的更改	查找包含文件的增量磁带需要较长时间。可能必须返回上一次的完整备份。

选择磁带设备

下表显示了用于在备份过程期间存储文件系统的典型磁带设备。存储容量取决于驱动器类型和当前写入磁带的数量。有关磁带设备的更多信息，请参见[第 30 章](#)。

表 24-5 用于备份文件系统的典型介质

备份介质	存储容量
1/2 英寸盘式磁带	140 MB (6250 bpi)
2.5 GB 1/4 英寸盒式 (QIC) 磁带	2.5 GB
DDS3 4 毫米盒式磁带 (DAT)	12–24 GB
14 GB 8 毫米盒式磁带	14 GB
DLT 7000 1/2 英寸盒式磁带	35–70 GB

备份和恢复文件系统的高级视图（任务列表）

使用此任务列表可确定备份和恢复文件系统的所有任务。每个任务都指向一系列其他任务，如确定要执行的备份类型。

任务	说明	参考
1. 确定要备份的文件系统。	确定需要每天、每周或每月备份的文件系统。	第 465 页中的“应该备份的文件系统”
2. 确定备份类型。	确定需要对站点上的文件系统执行的备份类型。	第 466 页中的“选择备份类型”
3. 创建备份。	请使用以下方法之一：	
	如果要同时具备文件系统的完整备份和增量备份，请使用 <code>ufsdump</code> 命令。	第 25 章
	如果要在文件系统处于活动状态且已挂载时创建此文件系统的快照，请考虑使用 <code>fssnap</code> 命令。	第 26 章
	如果只是要拥有个人起始目录或较小且不太重要文件系统的完整备份，请使用 <code>tar</code> 、 <code>cpio</code> 或 <code>pax</code> 命令。	第 29 章
4. （可选）恢复文件系统。	选择基于用于备份文件或文件系统的命令的恢复方法：	

任务	说明	参考
	恢复使用 <code>ufsdump</code> 命令创建的文件系统备份。	第 27 章
	恢复使用 <code>tar</code> 、 <code>cpio</code> 或 <code>pax</code> 命令创建的文件系统。	第 29 章
5.（可选）恢复根 (/) 或 /usr 文件系统。	恢复根 (/) 或 /usr 文件系统比恢复非关键文件系统更为复杂。需要在恢复这些文件系统的同时从本地 CD 或从网络进行引导。	第 516 页中的“如何恢复根 (/) 和 /usr 文件系统”

计划备份注意事项

备份计划是为运行 `ufsdump` 命令而建立的计划。本节确定了在创建备份计划时要考虑的注意事项。本节还包括样例备份计划。

创建的备份计划取决于以下内容：

- 需要将用于备份的磁带数目减至最少。
- 可用于执行备份的时间
- 可用于对损坏的文件系统执行完整恢复的时间
- 可用于检索无意中删除的各个文件的时间

执行备份的频率

如果无需将时间要求和用于备份的介质数目降到最低，则可以每天执行完整备份。但是，此备份方法对于大多数站点并不实际，因此增量备份是最为常用的。在这种情况下，应该适当地备份站点，以便可以恢复最近四周的文件。此计划需要至少四组磁带，每周一组。这样便可以每个月重复使用这些磁带。此外，还应该将每月的备份保留至少一年。并将每年的备份保留若干年。

备份间隔术语和定义

下表介绍了备份间隔术语和定义。

术语	定义
快照	创建文件系统的临时映像。
完整备份	复制完整的文件系统或目录。

术语	定义
增量备份	<p>仅复制指定文件系统中自以前某一次备份以来发生更改的那些文件。增量备份类型包括以下各项：</p> <ul style="list-style-type: none">■ 每天累积－复制星期一一天的文件更改内容。然后，使用星期二的文件更改覆写星期一的备份，接着使用星期三的文件更改覆写星期二的备份，依此类推。■ 每天增量－复制一天的文件更改内容，以便针对星期一的更改、星期二的更改等保留有不同的磁带。■ 每周累积－复制所有在这一周内发生更改的文件，并且包括上一周的文件更改。■ 每周增量－复制自上周备份以来在这一周内发生更改的文件。

备份计划指南

下表提供了备份计划指南。有关其他备份计划注意事项，请参见第 468 页中的“计划备份注意事项”。

表 24-6 备份计划指南

文件恢复需求	备份间隔	注释
恢复不同版本的文件（例如，用于进行文字处理的文件系统）	<p>每个工作日都执行每日增量备份。</p> <p>请勿重复使用同一磁盘进行每日增量备份。</p>	<p>此计划将保存当天修改的所有文件，以及自上次较低级别备份以来已修改且仍然存放在磁盘上的那些文件。但是，对于此计划，每天都应该使用不同的磁带，否则可能无法恢复所需版本的文件。</p> <p>例如，在星期二更改某文件后，在星期四再次对该文件进行了更改，这样，进入星期五的较低级别备份的将是星期四晚上更改的文件，而不会是星期二晚上更改的文件。如果用户需要星期二的版本，您无法进行恢复，除非您具有星期二的备份磁带（或星期三的备份磁带）。同样，星期二和星期三出现的文件，如果在星期四被删除，则不会出现在星期五的较低级别备份中。</p>
快速恢复完整的文件系统	更频繁地执行较低级别备份。	—

表 24-6 备份计划指南（续）

文件恢复需求	备份间隔	注释
在同一服务器上备份大量文件系统	考虑为不同的文件系统交错制定计划。	这样就不会在同一天执行所有 0 级备份。
将使用的磁盘数降到最少	增加一周内执行的增量备份的级别	在每个每日磁带上仅保存一天天的更改。
	增加周末完成的备份级别。将每天和每周的增量备份置于同一磁带上。	在每周磁带上仅保存一周周（而不是整月）的更改。
	将每天和每周的增量备份置于同一磁带上。	为此，请使用 <code>ufsdump</code> 命令的不反绕选项，如指定 <code>/dev/rmt/0n</code> 。

使用转储级别创建增量备份

在 `ufsdump` 命令中指定的转储级别 (0-9) 可确定要备份的文件。转储级别 0 可创建完整备份。级别 1-9 用于计划增量备份，但**没有明确的含义**。级别 1-9 只是用于计划累积备份或独立备份的一系列编号。级别 1-9 所具有的唯一含义就是表示这些编号彼此之间的相互关系，即指示较大编号或较小编号。较小的转储编号会始终重新启动完整备份或累积备份。以下示例说明了使用级别 1-9 的增量转储过程的灵活性。

示例—每日累积备份的转储级别

执行每日累积增量备份是最常用的备份计划，建议在大多数情况下使用该计划。以下示例显示的计划在星期一到星期四使用级别 9 转储，在星期五则使用级别 5 转储来重新启动此过程。

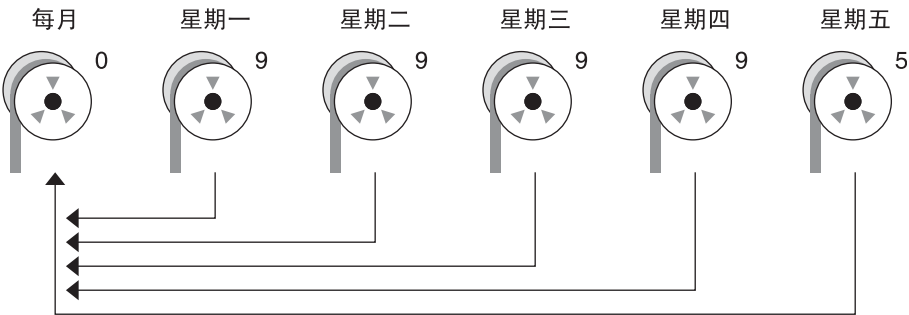


图 24-1 增量备份：每日累积

在以上示例中，可以使用 1-9 范围内的其他编号产生相同的结果。关键是从星期一到星期四使用相同的编号，而在星期五使用任何**较小**编号。例如，您可以指定级别 4、4、4、4、2 或 7、7、7、7、5。

示例—每日增量备份的转储级别

在以下示例说明的计划中，将仅备份一天内不同磁带上上的工作结果。这种类型的备份称为每日增量备份。在这种情况下，在一周内会使用连续的转储级别编号（3、4、5、6），在星期五使用较小的编号（2）。星期五的较小编号会重新启动备份的处理。

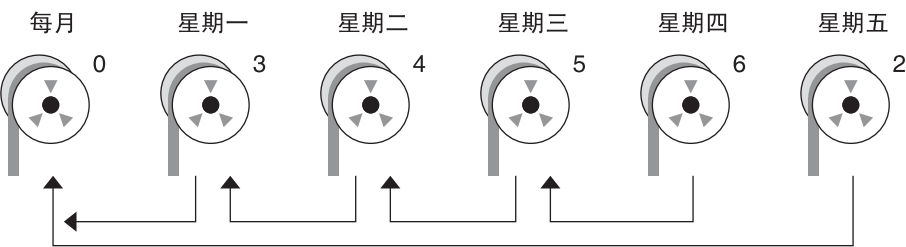


图 24-2 增量备份：每日增量

在以上示例中，您可以顺序使用6、7、8、9，然后再使用2；或使用5、6、7、8，然后再使用3。请记住，编号本身并没有明确的含义。通过以指定顺序排列这些编号，可以使其具有意义，如下示例所示。

样例备份计划

本节提供样例备份计划。所有计划都假定从完整备份（转储级别0）开始，并且使用 -u 选项在 /etc/dumpdates 文件中记录每个备份。

示例—每日累积，每周累积备份计划

表 24-7 显示了最常用的增量备份计划。建议在大多数情况下使用此计划。采用此计划时，会出现以下情况：

- 每天都保存自上周末较低级别备份以来发生更改的所有文件。
- 对于一周中的各工作日级别 9 备份，上一次级别 0 或级别 5 备份是最近的较低级别备份。因此，每盒工作日磁带都包含自上周末或第一周初始级别 0 备份以来发生更改的所有文件。
- 对于每个星期五的级别 5 备份，最近的较低级别备份是在月初执行的级别 0 备份。因此，每个星期五的磁带都包含截止到本月此时这段时间发生更改的所有文件。

表 24-7 每日累积/每周累积备份计划

	浮动	周一	周二	周三	周四	周五
每月的第一天	0					
第 1 周		9	9	9	9	5

表 24-7 每日累积/每周累积备份计划（续）

浮动	周一	周二	周三	周四	周五
第 2 周	9	9	9	9	5
第 3 周	9	9	9	9	5
第 4 周	9	9	9	9	5

下表显示了在采用每日累积、每周累积计划的两周内磁带内容发生更改的方式。每个字母都代表一个不同的文件。

表 24-8 每日累积/每周累积备份计划的磁带内容

	周一	周二	周三	周四	周五
第 1 周	ab	abc	abcd	abcde	abcdef
第 2 周	g	gh	ghi	ghij	abcdefghijk

每日累积、每周累积计划的磁带需求

采用此计划时，如果要重复使用每日磁带，则需要六盒磁带。但是，如果要使用四盒不同的每日磁带，则需要九盒磁带：

- 一盒磁带用于级别 0 备份
- 四盒磁带用于星期五的备份
- 一盒或四盒每日磁带

如果需要恢复完整的文件系统，则需要以下磁带：

- 级别 0 磁带
- 最近的星期五的磁带
- 自上星期五磁带以来最近的每日磁带（如果存在）

示例—每日累积、每周增量备份计划

下表显示的计划中每盒工作日磁带累积了自本周开始或第一周初始级别 0 备份以来发生更改的所有文件。此外，每个星期五的磁带都包含本周发生更改的所有文件。

表 24-9 每日累积，每周增量备份计划

浮动	周一	周二	周三	周四	周五
每月的第一天	0				
第 1 周	9	9	9	9	3
第 2 周	9	9	9	9	4

表 24-9 每日累积，每周增量备份计划（续）

	浮动	周一	周二	周三	周四	周五
第 3 周		9	9	9	9	5
第 4 周		9	9	9	9	6

下表显示了在采用每日累积，每周增量备份计划的两周内磁带内容发生更改的方式。每个字母都代表一个不同的文件。

表 24-10 每日累积，每周增量备份计划的磁带内容

	周一	周二	周三	周四	周五
第 1 周	a b	a b c	a b c d	a b c d e	a b c d e f
第 2 周	g	g h	g h i	g h i j	g h i j k

每日累积，每周增量备份计划的磁带需求

采用此计划时，如果要重复使用每日磁带，则需要六盒磁带。但是，如果要使用四盒不同的每日磁带，则需要九盒磁带：

- 一盒磁带用于级别 0 备份
- 四盒磁带用于星期五的备份
- 一盒或四盒每日磁带

如果需要恢复完整的文件系统，则需要以下磁带：

- 级别 0 磁带
- 所有的星期五磁带
- 自上星期五磁带以来最近的每日磁带（如果存在）

示例—每日增量、每周累积备份计划

下表显示的计划中每盒工作日磁带仅包含自前一天以来发生更改的文件。此外，每盒星期五磁带都包含自月初的初始级别 0 备份以来发生更改的所有文件。

表 24-11 每日增量、每周累积备份计划

	浮动	周一	周二	周三	周四	周五
每月的第一天	0					
第 1 周		3	4	5	6	2
第 2 周		3	4	5	6	2
第 3 周		3	4	5	6	2

表 24-11 每日增量、每周累积备份计划（续）

浮动	周一	周二	周三	周四	周五
第 4 周	3	4	5	6	2

下表显示了在采用每日增量、每周累积计划的两周内磁带内容发生更改的方式。每个字母都代表一个不同的文件。

表 24-12 每日增量，每周累积备份计划的磁带内容

	周一	周二	周三	周四	周五
第 1 周	a b	c d	e f g	h i	a b c d e f g h i
第 2 周	j k l	m	n o	p q	a b c d e f g h i j k l m n o p q r s

每日增量，每周累积计划的磁带需求

采用此计划时，如果要重复使用每日磁带（不推荐），则至少需要 9 盒磁带。如果将每周磁带保存一个月，则需要 21 盒磁带（推荐）：一盒磁带用于级别 0 备份、4 盒磁带用于星期五的备份，以及 4 或 16 盒每日磁带。

- 1 盒磁带用于级别 0 备份
- 4 盒磁带用于所有的星期五备份
- 4 或 16 盒每日磁带

如果需要恢复完整的文件系统，则需要以下磁带：

- 级别 0 磁带
- 最近的星期五的磁带
- 自上星期五磁带以来所有的每日磁带（如果存在）

示例－服务器的每月备份计划

下表显示了针对用户用于执行大量文件工作的小型网络（如程序开发或文档生成）上频繁使用的文件服务器的示例备份策略。此示例假定备份期从星期日开始并且包含四周（每周七天）。

表 24-13 服务器每月备份计划的示例

目录	日期	转储级别	磁带名称
根目录 (/)	第 1 个星期日	0	n 盒磁带
/usr	第 1 个星期日	0	n 盒磁带

表 24-13 服务器每月备份计划的示例（续）

目录	日期	转储级别	磁带名称
/export	第 1 个星期日	0	<i>n</i> 盒磁带
/export/home	第 1 个星期日	0	<i>n</i> 盒磁带
	第 1 个星期一	9	A
	第 1 个星期二	9	B
	第 1 个星期三	5	C
	第 1 个星期四	9	D
	第 1 个星期五	9	E
	第 1 个星期六	5	F
根目录 (/)	第 2 个星期日	0	<i>n</i> 盒磁带
/usr	第 2 个星期日	0	<i>n</i> 盒磁带
/export	第 2 个星期日	0	<i>n</i> 盒磁带
/export/home	第 2 个星期日	0	<i>n</i> 盒磁带
	第 2 个星期一	9	G
	第 2 个星期二	9	H
	第 2 个星期三	5	I
	第 2 个星期四	9	J
	第 2 个星期五	9	K
	第 2 个星期六	5	L
根目录 (/)	第 3 个星期日	0	<i>n</i> 盒磁带
/usr	第 3 个星期日	0	<i>n</i> 盒磁带
/export	第 3 个星期日	0	<i>n</i> 盒磁带
/export/home	第 3 个星期日	0	<i>n</i> 盒磁带
	第 3 个星期一	9	M
	第 3 个星期二	9	N
	第 3 个星期三	5	O
	第 3 个星期四	9	P
	第 3 个星期五	9	Q
	第 3 个星期六	5	R

表 24-13 服务器每月备份计划的示例（续）

目录	日期	转储级别	磁带名称
根目录 (/)	第 4 个星期日	0	<i>n</i> 盒磁带
/usr	第 4 个星期日	0	<i>n</i> 盒磁带
/export	第 4 个星期日	0	<i>n</i> 盒磁带
/export/home	第 4 个星期日	0	<i>n</i> 盒磁带
	第 4 个星期一	9	S
	第 4 个星期二	9	T
	第 4 个星期三	5	U
	第 4 个星期四	9	V
	第 4 个星期五	9	W
	第 4 个星期六	5	X

采用此计划时，需要使用 $4n$ 盒磁带，即根目录 (/)、/usr、/export 和 /export/home 文件系统的 4 个完整备份所需的磁带数。此外，还需要 24 盒磁带，用于 /export/home 文件系统的增量备份。此计划假定每次增量备份使用一盒磁带，并且将磁带保存一个月。

以下是此计划的工作原理：

1. 在每个星期日，执行根目录 (/)、/usr、/export 和 /export/home 文件系统的完整备份（级别 0）。将级别 0 磁带保存至少三个月。
2. 在本月的第一个星期一，使用磁带 A 执行 /export/home 文件系统的级别 9 备份。ufsdump 命令用于复制自上一次较低级别备份以来更改的所有文件。在这种情况下，前一次较低级别备份是在星期日执行的级别 0 备份。
3. 在本月的第一个星期二，使用磁带 B 执行 /export/home 文件系统的级别 9 备份。同样，再使用 ufsdump 命令复制自上一次较低级别备份（星期日的级别 0 备份）以来更改的所有文件。
4. 在本月的第一个星期三，使用磁带 C 执行 /export/home 文件系统的级别 5 备份。ufsdump 命令用于复制自星期日以来更改的所有文件。
5. 在磁带 D 和 E 上执行 /export/home 文件系统的星期四和星期五级别 9 备份。ufsdump 命令用于复制自上一次较低级别备份（星期三的级别 5 备份）以来更改的所有文件。
6. 在本月的第一个星期六，使用磁带 F 执行 /export/home 的级别 5 备份。ufsdump 命令用于复制自前一次较低级别备份（在本示例中是指在星期日执行的级别 0 备份）以来更改的所有文件。再次使用磁带 A-F 时，请存储这些磁带，直到下一个四周期间的第一个星期一为止。

7. 对于接下来的三周，请使用磁带 G-L 和 4n 盒磁带（用于执行星期日的级别 0 备份）重复步骤 1-6，依此类推。
8. 对于每个四周期间，请使用一组新磁带执行级别 0 备份并重复使用磁带 A-X 执行增量备份来重复步骤 1-7。三个月后，即可重新使用级别 0 磁带。

通过此计划，可以将各种状态的文件保存一个月。此计划需要许多磁带，但可以确保您具有一个可供提取的磁带库。要减少磁带数量，可以每周重新使用磁带 A-F。

备份文件和文件系统（任务）

本章介绍使用 `ufsdump` 命令备份文件系统的过程。

有关这些过程的信息，请参见第 479 页中的“备份文件和文件系统（任务列表）”。

有关执行备份的概述信息，请参见第 24 章。

有关将各个文件备份到软盘的信息，请参见第 29 章。

有关 `ufsdump` 命令的其他信息，请参见第 28 章。

备份文件和文件系统（任务列表）

任务	说明	参考
1. 准备文件系统备份。	确定文件系统、备份类型以及用于备份的磁带设备。	第 480 页中的“准备文件系统备份”
2. 确定备份文件系统所需的磁带数目。	确定文件系统的完整备份所需的磁带数目。	第 481 页中的“如何确定进行完整备份所需的磁带数目”
3. 备份文件系统。	执行文件系统的完整备份，以获取所有文件的基线副本。 根据保留每天更改的文件副本对于站点是否重要，执行文件系统的增量备份。	第 482 页中的“如何将文件系统备份到磁带”

准备文件系统备份

备份文件系统的准备工作从规划开始，规划在[第 24 章](#)中介绍并且包括选择以下内容：

- 要备份的文件系统
- 要执行的备份类型（完整或增量）
- 备份计划
- 磁带机

有关更多信息，请参见[第 24 章](#)。

本节介绍备份文件系统之前可能需要执行的其他两项任务：

- 查找要备份的文件系统的名称
- 确定进行完整备份所需的磁带数目

▼ 如何查找文件系统名称

- 1 显示 `/etc/vfstab` 文件的内容。
`$ more /etc/vfstab`
- 2 查看与文件系统名称对应的 `mount point` 列。
- 3 备份文件系统时，请使用 `mount point` 列中列出的目录名称。

示例 25-1 查找文件系统名称

在本示例中，要备份的文件系统为根 (`/`)、`/usr`、`/datab` 和 `/export/home`。

```
$ more /etc/vfstab

#device          device          mount          FS    fsck mount    mount

#to mount        to fsck         point          type  pass at boot options

/devices         -               /devices       devfs -    no    -

.

.

.

/proc            -               /proc          proc  -    no    -
```


/dev/dsk/c0t0d0s1	-	-	swap	-	no	-
/dev/dsk/c0t0d0s0	/dev/rdisk/c0t0d0s0	/	ufs	1	no	-
/dev/dsk/c0t0d0s6	/dev/rdisk/c0t0d0s6	/usr	ufs	1	no	-
/dev/dsk/c0t0d0s5	/dev/rdisk/c0t0d0s5	/datab	ufs	2	yes	-
/dev/dsk/c0t0d0s7	/dev/rdisk/c0t0d0s7	/export/home	ufs	2	yes	-
swap	-	/tmp	tmpfs	-	yes	-

▼ 如何确定进行完整备份所需的磁带数目

- 1 成为超级用户或承担等效角色。
- 2 估计备份大小（以字节为单位）。
ufsdump [0]S file-system
如果是文件系统的第一份备份，请使用 S 选项显示进行该备份估计所需的字节数。
如果不是文件系统的第一份备份，请使用 0S 选项显示进行该备份估计所需的字节数。
- 3 按照磁带容量划分估计大小，以确定所需的磁带数目。
有关磁带容量的列表，请参见表 24-5。

示例 25-2 确定磁带数目

在本示例中，一个 150 MB 的磁带足以满足 489,472 字节的文件系统的需求。

```
# ufsdump S /export/home  
  
489472
```

备份文件系统

以下是执行备份的一般原则：

- 除非正在创建文件系统的快照，否则请使用单用户模式或取消挂载文件系统。有关 UFS 快照的信息，请参见第 26 章。
- 请注意，在目录级操作（如创建、删除和重命名文件）和文件级活动同时进行时备份文件系统，即表示某些数据不会包括在备份中。

- 可以从单个系统运行 `ufsdump` 命令，并通过远程 `shell` 或远程登录在网络中远程备份多组系统。此外，还可以将输出定向到磁带设备所在的系统。通常，磁带设备位于运行 `ufsdump` 命令的系统上，但这并不是必须的。

将文件备份到远程设备的另一种方式是将 `ufsdump` 命令的输出传输到 `dd` 命令。有关使用 `dd` 命令的信息，请参见第 29 章。

- 如果是通过网络执行远程备份，则配备了磁带设备的系统在其 `/.rhosts` 文件中必须具有与使用该设备的每台客户机对应的项。另外，启动备份的系统必须包括在将要备份的每个系统的 `/.rhosts` 文件中。

▼ 如何将文件系统备份到磁带

以下是使用 `ufsdump` 命令备份文件系统的常规步骤。这些示例说明了选项和参数的特定用途。

- 1 成为超级用户或承担等效角色。

- 2 使系统进入运行级 S（单用户模式）。

例如：

```
# shutdown -g30 -y
```

- 3 （可选）检查文件系统的一致性。

例如：

```
# fsck -m /dev/rdisk/c0t0d0s7
```

`fsck -m` 命令用于检查文件系统的一致性。例如，电源故障可能会使文件处于不一致的状态。有关 `fsck` 命令的更多信息，请参见第 22 章。

- 4 如果需要将文件系统备份到远程磁带机，请执行以下步骤：

- a. 在连接磁带机的系统（磁带服务器）上，将以下项添加到该系统的 `/.rhosts` 文件中：

```
host root
```

`host` 项指定了要在其中运行 `ufsdump` 命令来执行备份的系统的名称。

- b. 在磁带服务器上，检验添加到 `/.rhosts` 文件中的主机是否可通过名称服务进行访问。

- 5 确定磁带机的设备名称。

缺省的磁带机为 `/dev/rmt/0` 设备。

- 6 将可写的磁带插入磁带机。

7 备份文件系统。

```
# ufsdump options arguments filenames
```

可以备份文件系统或目录，也可以备份文件系统中的文件。有关备份各个文件的信息，请参见 `tar(1)` 或 `cpio(1)`。

以下示例说明如何使用最常见的 `ufsdump` 选项和参数：

- 示例 25-3
- 示例 25-4
- 示例 25-5
- 示例 25-6

有关其他 `ufsdump` 选项和参数的信息，请参见第 28 章。

8 如果出现提示，请取出磁带并插入下一个磁带卷。

9 使用卷编号、转储级别、日期、系统名称、磁盘片和文件系统标记每个磁带。

10 通过按 Control-D 组合键使系统回到运行级 3。

11 检验备份是否成功。

```
# ufsrestore tf device-name
```

示例 25-3 执行根 (/) 文件系统的完整备份

以下示例说明如何执行根 (/) 文件系统的完整备份。本示例中的系统将在备份之前进入单用户模式。包括以下 `ufsdump` 选项：

- 0 指定 0 级转储（或完整备份）。
- u 指定将 `/etc/dumpdates` 文件更新为此备份的日期。
- c 标识盒式磁带设备。
- f `/dev/rmt/0` 标识磁带设备。
- / 表示正在备份的文件系统。

例如：

```
# init 0
```

```
ok boot -s
```

```
# ufsdump 0ucf /dev/rmt/0 /
```

```
DUMP: Date of this level 0 dump: Wed Jul 28 16:13:52 2004
```

```
DUMP: Date of last level 0 dump: the epoch
```

DUMP: Dumping /dev/rdisk/c0t0d0s0 (starbug:/) to /dev/rmt/0.

DUMP: Mapping (Pass I) [regular files]

DUMP: Mapping (Pass II) [directories]

DUMP: Writing 63 Kilobyte records

DUMP: Estimated 363468 blocks (177.47MB).

DUMP: Dumping (Pass III) [directories]

DUMP: Dumping (Pass IV) [regular files]

DUMP: Tape rewinding

DUMP: 369934 blocks (180.63MB) on 1 volume at 432 KB/sec

DUMP: DUMP IS DONE

DUMP: Level 0 dump on Wed Jul 28 16:13:52 2004

ufsrestore tf /dev/rmt/0

2 .

3 ./lost+found

4 ./usr

5 ./export

6 ./export/home

7 ./var

8 ./var/sadm

9 ./var/sadm/install

10 ./var/sadm/install/admin

823 ./var/sadm/install/admin/default

11 ./var/sadm/install/logs

697 ./var/sadm/install/logs/SUNWmpatchmgr

```

905      ./var/sadm/install/logs/Additional_Software_install...
906      ./var/sadm/install/logs/Additional_Software_install...
13      ./var/sadm/install/.lockfile
14      ./var/sadm/install/install.db
824      ./var/sadm/install/special_contents
838      ./var/sadm/install/contents
.
.
.

# (Press Control-D to bring system to run level 3)

```

示例 25-4 执行根 (/) 文件系统的增量备份

以下示例说明如何在单用户模式下执行根 (/) 文件系统的增量备份。包括以下 `ufsdump` 选项：

- `9` 指定 9 级转储（或增量备份）。
- `u` 指定将 `/etc/dumpdates` 文件更新为此备份的日期。
- `c` 标识盒式磁带设备。
- `f /dev/rmt/0` 标识磁带设备。
- `/` 表示正在备份的文件系统。

```

# init 0

ok boot -s

# ufsdump 9ucf /dev/rmt/0 /

DUMP: Date of this level 9 dump: Wed Jul 28 14:26:50 2004

DUMP: Date of last level 0 dump: Wed Jul 28 11:15:41 2004

DUMP: Dumping /dev/rdisk/c0t0d0s0 (starbug:/) to /dev/rmt/0.

DUMP: Mapping (Pass I) [regular files]

DUMP: Mapping (Pass II) [directories]

DUMP: Writing 63 Kilobyte records

```

DUMP: Estimated 335844 blocks (163.99MB).

DUMP: Dumping (Pass III) [directories]

DUMP: Dumping (Pass IV) [regular files]

DUMP: 335410 blocks (163.77MB) on 1 volume at 893 KB/sec

DUMP: DUMP IS DONE

DUMP: Level 9 dump on Wed Jul 28 14:30:50 2004

ufsrestore tf /dev/rmt/0

2 .

3 ./lost+found

5696 ./usr

11392 ./var

17088 ./export

22784 ./export/home

28480 ./opt

5697 ./etc

11393 ./etc/default

11394 ./etc/default/sys-suspend

11429 ./etc/default/cron

11430 ./etc/default/devfsadm

11431 ./etc/default/dhcpagent

11432 ./etc/default/fs

11433 ./etc/default/inetinit

11434 ./etc/default/kbd

11435 ./etc/default/nfslogd

```
11436      ./etc/default/passwd
```

```
11437      ./etc/default/tar
```

```
.
```

```
.
```

```
.
```

示例 25-5 执行起始目录的完整备份

以下示例说明如何执行 `/export/home/kryten` 起始目录的完整备份。包括以下 `ufsdump` 选项：

- `0` 指定这是 `0` 级转储（或完整备份）。
- `u` 指定将 `/etc/dumpdates` 文件更新为此备份的日期。
- `c` 标识盒式磁带设备。
- `f /dev/rmt/0` 标识磁带设备。
- `/export/home/kryten` 表示正在备份的目录。

```
# ufsdump 0ucf /dev/rmt/0 /export/home/kryten
```

```
DUMP: Date of this level 0 dump: Wed Jul 28 15:02:48 2004
```

```
DUMP: Date of last level 0 dump: the epoch
```

```
DUMP: Dumping /dev/rdisk/c0t0d0s7 (starbug:/export/home) to /dev/rmt/0.
```

```
DUMP: Mapping (Pass I) [regular files]
```

```
DUMP: Mapping (Pass II) [directories]
```

```
DUMP: Writing 63 Kilobyte records
```

```
DUMP: Estimated 2412 blocks (1.18MB).
```

```
DUMP: Dumping (Pass III) [directories]
```

```
DUMP: Dumping (Pass IV) [regular files]
```

```
DUMP: 2392 blocks (1.17MB) on 1 volume at 4241 KB/sec
```

```
DUMP: DUMP IS DONE
```

```
# ufsrestore tf /dev/rmt/0
```

```

232      ./kryten
233      ./kryten/filea
234      ./kryten/fileb
235      ./kryten/filec
236      ./kryten/letters
237      ./kryten/letters/letter1
238      ./kryten/letters/letter2
239      ./kryten/letters/letter3
240      ./kryten/reports
241      ./kryten/reports/reportA
242      ./kryten/reports/reportB
243      ./kryten/reports/reportC
#

```

示例 25-6 执行到远程系统的完整备份（Solaris 10 数据到 Solaris 10 系统）

以下示例说明如何在单用户模式下将 Solaris 10 系统 (mars) 上的本地 /export/home 文件系统完整备份到远程 Solaris 10 系统 (earth) 上的磁带设备。包括以下 ufsdump 选项：

- 0 指定 0 级转储（或完整备份）。
- u 指定将 /etc/dumpdates 文件更新为此备份的日期。
- c 标识盒式磁带设备。
- f earth:/dev/rmt/0 标识远程系统名称和磁带设备。
- /export/home 表示正在备份的文件系统。

```
# ufsdump 0ucf earth:/dev/rmt/0 /export/home
```

```
DUMP: Date of this level 0 dump: Wed Jul 28 15:52:59 2004
```

```
DUMP: Date of last level 0 dump: the epoch
```

```
DUMP: Dumping /dev/rdisk/c0t0d0s7 (mars:/export/home) to earth:/dev/rmt/0.
```

```
DUMP: Mapping (Pass I) [regular files]
```

```
DUMP: Mapping (Pass II) [directories]
```



```
DUMP: Writing 63 Kilobyte records

DUMP: Estimated 8282 blocks (4.04MB).

DUMP: Dumping (Pass III) [directories]

DUMP: Dumping (Pass IV) [regular files]

DUMP: Tape rewinding

DUMP: 8188 blocks (4.00MB) on 1 volume at 67 KB/sec

DUMP: DUMP IS DONE

DUMP: Level 0 dump on Wed Jul 28 15:52:59 2004
```

```
# ufsrestore tf earth:/dev/rmt/0
```

```
2      .

3      ./lost+found

4      ./kryten

5      ./kryten/filea

6      ./kryten/fileb

7      ./kryten/filec

8      ./kryten/letters

9      ./kryten/letters/letter1

10     ./kryten/letters/letter2

11     ./kryten/letters/letter3

12     ./kryten/reports
```

```
.
```

```
.
```

```
.
```

```
#
```


使用 UFS 快照（任务）

本章介绍如何创建和备份 UFS 快照。

有关创建 UFS 快照的过程信息，请参见第 491 页中的 “使用 UFS 快照（任务列表）”。

有关执行备份的概述信息，请参见第 24 章。

使用 UFS 快照（任务列表）

任务	说明	参考
1. 创建 UFS 快照。	使用 <code>fssnap</code> 命令创建文件系统的只读副本。	第 494 页中的 “如何创建 UFS 快照”
2. 显示 UFS 快照信息。	确定 UFS 快照信息，如原始快照设备。	第 496 页中的 “如何显示 UFS 快照信息”
3. （可选）删除 UFS 快照。	删除已备份或不再需要的快照。	第 497 页中的 “如何删除 UFS 快照”
4. （可选）备份 UFS 快照。	请选择以下备份方法之一：	
	使用 <code>ufsdump</code> 命令创建 UFS 快照的完整备份。	第 499 页中的 “如何创建 UFS 快照的完整备份 (ufsdump)”
	使用 <code>ufsdump</code> 命令创建 UFS 快照的增量备份。	第 500 页中的 “如何创建 UFS 快照的增量备份 (ufsdump)”
	使用 <code>tar</code> 命令备份 UFS 快照。	第 500 页中的 “如何备份 UFS 快照 (tar)”

任务	说明	参考
5.（可选）从 UFS 快照中恢复数据。	使用 <code>ufsrestore</code> 命令恢复 UFS 快照，其方式与恢复数据的方式相同。	第 512 页中的“如何恢复完整的文件系统”

UFS 快照概述

可以使用 `fssnap` 命令在挂载文件系统的同时备份文件系统。此命令可用于创建文件系统的只读快照。**快照**是旨在用于备份操作的文件系统临时映像。

运行 `fssnap` 命令时，此命令将创建一个虚拟设备和一个后备存储文件。可以使用现有的任何 Solaris 备份命令备份**虚拟设备**，此设备的外观和操作均与实际设备相同。**后备存储文件**是一个位图文件，其中包含自捕获快照以来已修改的预备快照数据副本。

指定后备存储文件时，请牢记以下要点：

- 后备存储文件的目标路径必须具有足够的空闲空间来保存文件系统数据。后备存储文件的大小会随文件系统上的活动量而异。
- 后备存储文件位置必须不同于快照中捕获的文件系统的位置。
- 后备存储文件可以驻留在任何类型的文件系统中，包括其他 UFS 文件系统或 NFS 文件系统。
- 创建大于 512 GB 的 UFS 文件系统快照时，将会创建多个后备存储文件。
- 后备存储文件是稀疏文件。`ls` 命令报告的稀疏文件的逻辑大小与 `du` 命令报告的已分配给稀疏文件的空间量不同。

有关为大于 512 GB 的 UFS 文件系统创建快照的更多信息，请参见[第 494 页中的“创建多 TB UFS 快照”](#)。

使用 UFS 快照的原因

UFS 快照功能为文件系统的备份提供了更高的可实现性及便利，这是因为文件系统始终保持挂载状态，并且该系统在备份过程中始终处于多用户模式下。然后，即可使用 `tar` 或 `cpio` 命令将 UFS 快照备份到磁带中进行更长久的存储。如果使用 `ufsdump` 命令执行备份，则系统应该处于单用户模式下，以便在执行备份时使文件系统保持不活动状态。

`fssnap` 命令向非企业级系统的管理员提供了无大量存储需求的企业级工具（如 Sun StorEdge™ Instant Image）的强大功能。

UFS 快照与 Instant Image 产品的功能类似。尽管 UFS 快照可以生成大型文件系统的副本，但是 Instant Image 更适合于企业级系统。UFS 快照更适合于较小的系统。Instant Image 可分配与所捕获的整个文件系统大小相同的空间。但是，UFS 快照创建的后备存储文件仅会占用所需大小的磁盘空间。

下表介绍了 UFS 快照与 Instant Image 之间的具体差异。

UFS 快照	Sun StorEdge Instant Image
后备存储文件的大小取决于自捕获快照以来已更改的数据量	后备存储文件的大小与正在复制的整个文件系统的大小相等
在系统重新引导期间会发生变化	在系统重新引导期间始终保持不变
在 UFS 文件系统上工作	不能用于根 (/) 或 /usr 文件系统
从 Solaris 8 1/01 发行版开始可用	是 Sun StorEdge 产品的一部分

UFS 快照性能问题

首次创建 UFS 快照时，文件系统的用户可能会注意到很短的暂停。暂停的时间长度随要捕获的文件系统大小的增加而增加。当快照处于活动状态时，文件系统的用户可能会发现在向文件系统写入数据时，会对其造成非常小的性能影响。但是，读取文件系统时，用户不会发现任何影响。

创建和删除 UFS 快照

使用 `fssnap` 命令创建 UFS 快照时，请观察后备存储文件占用的磁盘空间量。后备存储文件最初不使用任何空间，随后该文件会快速增大，特别是在使用频繁的系统中。请确保后备存储文件有足够的空间可以进行扩展。或者，请使用 `-o maxsize=n [k,m,g]` 选项限制该文件的大小，其中 `n [k,m,g]` 是后备存储文件的最大大小。



注意 – 如果后备存储文件空间不足，则快照可能会自行删除，从而导致备份失败。请检查 `/var/adm/messages` 文件中可能的快照错误。

还可以为后备存储路径指定目录，这表示后备存储文件将在该指定目录中创建。例如，如果为后备存储路径指定 `/var/tmp`，则将会创建以下后备存储文件。

```
/var/tmp/snapshot0
```

如果已为 `/export/home`、`/usr` 等创建了一个大的根 (/) 文件系统而不是单独的文件系统，则将无法创建这些单独文件系统的快照。例如，对于在 `Mounted on` 列下指示的 `/usr`，此系统没有单独的文件系统：

```
# df -k /usr
```

Filesystem	kbytes	used	avail	capacity	Mounted on
/dev/dsk/c0t0d0s0	3618177	2190002	1391994	62%	/

如果尝试为 `/usr` 文件系统创建快照，则将显示以下类似消息：

```
# fssnap -F ufs -o bs=/snaps/usr.back.file /usr
```

```
snapshot error: Invalid backing file path
```

此消息表明，不能在与正在抓取的文件系统相同的系统上包含后备存储文件，例如，本示例中的 `/usr` 文件系统。

有关更多信息，请参见 `fssnap_ufs(1M)` 手册页。

创建多 TB UFS 快照

创建多 TB UFS 快照与为较小的 UFS 文件系统创建快照相同。唯一的区别在于为每个 512 GB 的文件系统空间创建了多个后备存储文件。

为大于 512 GB 的文件系统创建快照时，请牢记以下要点：

- 将创建多个后备存储文件。
 - 如果在创建快照时指定了后备存储文件名，则后续的后备存储文件名将基于指定的文件名进行迭代。后续的后备存储文件将具有相同的名称，但后缀分别为 `.2`、`.3`，依此类推。
 - 如果仅指定了后备存储文件目标（或目录）而未指定后备存储文件名，则将创建多个后备存储文件名并分别以后缀 `.2`、`.3` 进行迭代，依此类推。
- 即使创建了多个后备存储文件，`fssnap -i` 命令也只报告第一个后备存储文件名。但是，报告的后备存储长度为快照的所有后备存储文件的大小之和。

注 - 后备存储文件是稀疏文件。`ls` 命令报告的稀疏文件的逻辑大小与 `du` 命令报告的已分配给稀疏文件的空间量不同。

- 备份快照后或只希望删除快照时，必须手动删除后备存储文件，前提是创建快照时未使用 `unlink` 选项。

有关为大于 512 GB 的文件系统创建快照的示例，请参见[示例 26-2](#)。

有关更多信息，请参见 `fssnap_ufs(1M)`。

▼ 如何创建 UFS 快照

- 1 成为超级用户或承担等效角色。
- 2 确保文件系统有足够的磁盘空间可用于后备存储文件。

```
# df -k
```

- 3 确保名称和位置相同的后备存储文件尚不存在。

```
# ls /backing-store-file
```

- 4 创建 UFS 快照。

```
# fssnap -F ufs -o bs=/backing-store-file /file-system
```

注 - 后备存储文件必须驻留在与使用 UFS 快照捕获的文件系统不同的文件系统上。

- 5 检验是否已创建快照。

```
# /usr/lib/fs/ufs/fssnap -i /file-system
```

示例 26-1 创建 UFS 快照

以下示例说明如何创建 /usr 文件系统的快照。后备存储文件为 /scratch/usr.back.file。虚拟设备为 /dev/fssnap/1。

```
# fssnap -F ufs -o bs=/scratch/usr.back.file /usr
```

```
/dev/fssnap/1
```

以下示例说明如何将后备存储文件限制到 500 MB。

```
# fssnap -F ufs -o maxsize=500m,bs=/scratch/usr.back.file /usr
```

```
/dev/fssnap/1
```

示例 26-2 创建多 TB UFS 快照

以下示例说明如何创建 1.6 TB UFS 文件系统的快照。

```
# fssnap -F ufs -o bs=/var/tmp /data2
```

```
/dev/fssnap/0
```

```
# /usr/lib/fs/ufs/fssnap -i
```

```
Snapshot number           : 0
Block Device               : /dev/fssnap/0
Raw Device                 : /dev/rfssnap/0
Mount point                : /data2
Device state                : idle
```

```
Backing store path           : /var/tmp/snapshot0

Backing store size           : 0 KB

Maximum backing store size   : Unlimited

Snapshot create time         : Fri Sep 10 13:13:02 2004

Copy-on-write granularity    : 32 KB

# ls /var/tmp

snapshot0    snapshot0.2  snapshot0.3  snapshot0.4
```

▼ 如何显示 UFS 快照信息

可以使用 `fssnap -i` 选项显示系统上的当前快照。如果指定了文件系统，则会显示有关该文件系统快照的详细信息。如果未指定文件系统，则会显示有关所有当前 UFS 快照及其对应虚拟设备的信息。

注 – 使用 UFS 文件系统特定的 `fssnap` 命令可以查看以下示例中显示的扩展快照信息。

- 1 成为超级用户或承担等效角色。
- 2 列出所有当前快照。

```
例如：

# /usr/lib/fs/ufs/fssnap -i

Snapshot number             : 0

Block Device                 : /dev/fssnap/0

Raw Device                   : /dev/rfssnap/0

Mount point                  : /export/home

Device state                 : idle

Backing store path           : /var/tmp/home.snap0

Backing store size           : 0 KB

Maximum backing store size   : Unlimited
```


Snapshot create time : Thu Jul 01 14:50:38 2004

Copy-on-write granularity : 32 KB

3 显示有关特定快照的详细信息。

例如：

/usr/lib/fs/ufs/fssnap -i /export

Snapshot number : 1

Block Device : /dev/fssnap/1

Raw Device : /dev/rfssnap/1

Mount point : /export

Device state : idle

Backing store path : /var/tmp/export.snap0

Backing store size : 0 KB

Maximum backing store size : Unlimited

Snapshot create time : Thu Jul 01 15:03:22 2004

Copy-on-write granularity : 32 KB

删除 UFS 快照

创建 UFS 快照时，可以指定解除链接后备存储文件。删除快照后，将会删除解除链接的后备存储文件。如果创建 UFS 快照时不指定 `-o unlink` 选项，则必须手动删除后备存储文件。

无论是使用 `-o unlink` 选项删除后备存储文件，还是手动删除该文件，删除快照之前后备存储文件会一直占用磁盘空间。

▼ 如何删除 UFS 快照

可以通过重新引导系统来删除快照，也可以通过使用 `fssnap -d` 命令来删除快照。使用此命令时，必须指定包含 UFS 快照的文件系统的路径。

1 成为超级用户或承担等效角色。

2 确定要删除的快照。

```
# /usr/lib/fs/ufs/fssnap -i
```

3 删除快照。

```
# fssnap -d /file-system
```

```
Deleted snapshot 1.
```

4 如果在创建快照时未使用 -o unlink 选项，请手动删除后备存储文件。

```
# rm /file-system/backing-store-file
```

示例 26-3 删除 UFS 快照

以下示例说明如何删除快照并假定未使用 -o unlink 选项。

```
# fssnap -i
```

```
0    /export/home
```

```
1    /export
```

```
# fssnap -d /usr
```

```
Deleted snapshot 1.
```

```
# rm /var/tmp/export.snap0
```

备份 UFS 快照

您可以创建 UFS 快照的完整备份或增量备份，并可以使用标准的 Solaris 备份命令来备份 UFS 快照。

包含 UFS 快照的虚拟设备可用作标准的只读设备。因此，可以按照备份文件系统设备一样，备份该虚拟设备。

如果要使用 `ufsdump` 命令备份 UFS 快照，则可以在备份过程中指定快照名称。有关更多信息，请参见以下过程。

▼ 如何创建 UFS 快照的完整备份 (ufsdump)

1 成为超级用户或承担等效角色。

2 确定要备份的 UFS 快照。

```
# /usr/lib/fs/ufs/fssnap -i /file-system
```

例如：

```
# /usr/lib/fs/ufs/fssnap -i /usr
```

```
Snapshot number           : 1
Block Device              : /dev/fssnap/1
Raw Device                : /dev/rfssnap/1
Mount point               : /usr
Device state              : idle
Backing store path        : /var/tmp/usr.snap0
Backing store size        : 0 KB
Maximum backing store size : Unlimited
Snapshot create time      : Thu Jul 01 15:17:33 2004
Copy-on-write granularity : 32 KB
```

3 备份 UFS 快照。

```
# ufsdump 0ucf /dev/rmt/0 /snapshot-name
```

例如：

```
# ufsdump 0ucf /dev/rmt/0 /dev/rfssnap/1
```

4 检验是否已备份快照。

例如：

```
# ufsrestore tf /dev/rmt/0
```

▼ 如何创建 UFS 快照的增量备份 (ufsdump)

以增量方式备份 UFS 快照意味着仅备份自上一个快照以来修改的文件。使用带有 **N** 选项的 `ufsdump` 命令。此选项可指定为跟踪增量转储而插入 `/etc/dumpdates` 文件中的文件系统的设备名称。

以下 `ufsdump` 命令可指定用于创建文件系统增量备份的嵌入式 `fssnap` 命令。

- 1 成为超级用户或承担等效角色。

- 2 创建 UFS 快照的增量备份。

例如：

```
# ufsdump 1uF /dev/rmt/0 /dev/rdisk/c0t1d0s0 'fssnap -F ufs -o raw,bs=
/export/scratch,unlink /dev/rdisk/c0t1d0s0'
```

在本示例中，`-o raw` 选项用于显示原始设备而不是块设备的名称。通过使用此选项，可以更轻易地在要求使用原始设备的命令（如 `ufsdump` 命令）中嵌入 `fssnap` 命令。

- 3 检验是否已备份快照。

```
# ufsrestore ta /dev/rmt/0
```

▼ 如何备份 UFS 快照 (tar)

如果要使用 `tar` 命令备份快照，请在备份之前先挂载该快照。

- 1 成为超级用户或承担等效角色。

- 2 为快照创建挂载点。

例如：

```
# mkdir /backups/home.bkup
```

- 3 挂载快照。

```
# mount -F ufs -o ro /dev/fssnap/1 /backups/home.bkup
```

- 4 转到已挂载快照的目录。

```
# cd /backups/home.bkup
```

- 5 使用 `tar` 命令备份快照。

```
# tar cvf /dev/rmt/0 .
```

从 UFS 快照备份中恢复数据

实际上，从虚拟设备中创建的备份只是捕获快照时的原始文件系统的备份。从备份中恢复文件系统时，就像从原始文件系统中直接提取备份那样进行恢复。这类恢复使用 `ufsrestore` 命令。有关使用 `ufsrestore` 命令恢复文件或文件系统的信息，请参见 [第 27 章](#)。

恢复文件和文件系统（任务）

本章介绍如何使用 `ufsrestore` 命令恢复使用 `ufsdump` 命令备份的文件和文件系统。

有关恢复文件和文件系统的过程信息，请参见第 503 页中的“恢复文件和文件系统备份（任务列表）”。

有关可用于归档、恢复、复制或移动文件和文件系统的其他命令的信息，请参见第 29 章。

有关备份和恢复文件系统的信息，请参见第 24 章。

恢复文件和文件系统备份（任务列表）

以下任务列表介绍了恢复文件和文件系统的过程。

任务	说明	参考
准备恢复文件和文件系统。	确定要恢复的文件系统或文件、磁带设备以及恢复它们的方式。	第 504 页中的“准备恢复文件和文件系统”
确定要使用的磁带。	查看备份磁带，以了解包含需要恢复的文件或文件系统的上一次备份的日期。	第 505 页中的“如何确定要使用的磁带”
恢复文件。	请选择以下恢复方法之一： 以交互方式恢复文件—如果不确定文件名，请使用此方法，这是因为您可以浏览介质内容并选择各个文件和目录。 以非交互方式恢复文件—如果已知要恢复的少数几个文件名，请使用此方法。	第 506 页中的“如何以交互方式恢复文件” 第 509 页中的“如何以非交互方式恢复特定文件”

任务	说明	参考
	恢复文件系统—如果获取了新的磁盘驱动器或要在恢复过程中执行此任务，请使用此方法。	第 512 页中的“如何恢复完整的文件系统”
恢复根 (/) 文件系统或 /usr 文件系统。	恢复根 (/) 文件系统或 /usr 文件系统涉及从本地 CD 或网络引导系统。	第 516 页中的“如何恢复根 (/) 和 /usr 文件系统”

准备恢复文件和文件系统

`ufsrestore` 命令可以将文件从使用 `ufsdump` 命令创建的备份复制到磁盘上的当前工作目录下。可以使用 `ufsrestore` 命令从级别 0 转储和其后的增量转储重新装入整个文件系统分层结构。还可以使用此命令从任何备份磁带恢复一个或多个单独的文件。如果以超级用户身份运行 `ufsrestore` 命令，则可按原始属主、最后修改时间和模式（权限）恢复文件。

开始恢复文件或文件系统之前，需要了解以下内容：

- 需要从中恢复的磁带（或软盘）
- 要在其中恢复文件系统的原始设备名称
- 将使用的磁带设备类型
- 磁带设备的设备名称（本地或远程）

确定文件系统名称

如果已正确标记了备份磁带，应可以使用磁带标签中的文件系统名称 (`/dev/rdisk/device-name`)。有关更多信息，请参见[第 480 页中的“如何查找文件系统名称”](#)。

确定所需的磁带设备类型

必须使用与备份介质兼容的磁带设备才能恢复文件。备份介质的格式确定了恢复文件所必须使用的驱动器。例如，如果备份介质为 8 毫米磁带，则必须使用 8 毫米磁带设备来恢复文件。

确定磁带设备名称

您可能已经在备份磁带标签信息中指定了磁带设备名称 (`/dev/rmt/n`)。如果您要使用同一驱动器来恢复备份磁带，则可以使用该标签中的设备名称。有关介质设备和设备名称的更多信息，请参见[第 30 章](#)。

恢复文件和文件系统

备份文件和目录时，可以相对于它们所属的文件系统来保存它们。恢复文件和目录时，`ufsrestore` 命令将在当前工作目录中重新创建文件分层结构。

例如，从 `/export/doc/books` 目录（其中 `/export` 是文件系统）备份的文件将相对于 `/export` 进行保存。换句话说，`books` 目录中的 `book1` 文件将在磁带上被另存为 `./doc/books/book1`。此后，如果将 `./doc/books/book1` 文件恢复到 `/var/tmp` 目录，则该文件将被恢复到 `/var/tmp/doc/books/book1`。

恢复各个文件和目录时，应该将它们恢复到临时位置，如 `/var/tmp` 目录。验证文件后，可以将它们移至其正确位置。但是，可以将各个文件和目录恢复到其原始位置。如果执行此操作，请确保不会使用备份磁带中的旧版本覆写新文件。

为避免与其他用户冲突，您可能需要创建并转至一个子目录（如 `/var/tmp/restore`），将文件恢复到该子目录中。

如果要恢复分层结构，应该将文件恢复到一个临时目录中，这个临时目录必须位于将驻留这些文件的那个文件系统上。然后，在恢复文件之后，便可以使用 `mv` 命令移动其所属的整个分层结构。

注 - 请勿在 `/tmp` 目录中恢复文件，即使是临时恢复也不允许。`/tmp` 目录通常作为 TMPFS 文件系统进行挂载。TMPFS 不支持 UFS 文件系统属性（如 ACL）。

▼ 如何确定要使用的磁带

- 1 询问用户上次修改要恢复文件的大约日期。
- 2 请参考备份计划，以了解包含对应文件或文件系统的上一次备份的日期。
要检索文件的最新版本，请按照从高到低转储级别以及从近到远日期的顺序追溯增量备份，除非用户有其他请求。

- 3 如果您具有联机归档文件，请确定正确的介质。

```
# ufsrestore ta archive-name ./path/filename ./path/filename
```

`t` 列出磁带中的每个文件。

`a` 读取联机归档文件（而不是磁带）中的目录。

`archive-name` 确定联机归档文件名。

`./path/filename` 确定要在联机归档文件中寻找的文件名。如果成功，`ufsrestore` 命令将列显 `inode` 编号和文件名。如果不成功，`ufsrestore` 将列显错误消息。

有关更多信息，请参见 `ufsrestore(1M)` 手册页。

- 4 将包含要恢复文件的介质插入驱动器中，并检验是否是正确的介质。

```
# ufsrestore tf /dev/rmt/n ./path/filename ./path/filename
```

确保对每个 *filename* 使用完整路径。如果文件在备份中，则会列出其名称和 inode 编号。否则，将出现一条消息，指明文件不在卷中。

- 5 如果同一磁带上有多多个备份文件，请定位至磁带上要使用的备份文件。

```
# ufsrestore xfs /dev/rmt/n tape-number
```

示例 27-1 确定要使用的磁带

以下示例说明如何检查 `/etc/passwd` 文件是否在联机归档文件中。

```
# ufsrestore ta /var/tmp/root.archive ./etc/passwd
```

以下示例说明如何验证 `/etc/passwd` 文件是否位于备份磁带中。

```
# ufsrestore tf /dev/rmt/0 ./etc/passwd
```

▼ 如何以交互方式恢复文件

- 1 成为超级用户或承担等效角色。
- 2 （可选）为安全起见，请对磁带进行写保护。
- 3 将卷 1 磁带插入磁带机。
- 4 转至用于临时将文件恢复到的目录。

```
# cd /var/tmp
```

- 5 启动交互式恢复。

```
# ufsrestore if /dev/rmt/n
```

此时将显示某些提示性消息和 `ufsrestore>` 提示符。

- 6 创建要恢复的文件的列表。

- a. 列出目录的内容。

```
ufsrestore> ls [directory-name]
```

- b. 转至目录。

```
ufsrestore> cd directory-name
```

- c. 创建要恢复的文件和目录的列表。

```
ufsrestore> add filenames
```

- d. (可选的) 从要恢复的文件列表中删除任何目录或文件 (如果必要) 。

```
ufsrestore> delete filename
```

- 7 (可选的) 在恢复文件时显示文件名称。

```
ufsrestore> verbose
```

- 8 恢复文件。

```
ufsrestore> extract
```

ufsrestore 命令将询问您要使用的卷编号。

- 9 键入对应的卷编号并按回车键。如果您只有一个卷，请键入 1 并按回车键。

```
Specify next volume #: 1
```

列表中的文件和目录将被提取并恢复到当前工作目录。

- 10 要维护当前目录的模式，请在 set owner/mode 提示符下输入 n。

```
set owner/mode for '.'? [yn] n
```

ufsrestore 命令执行其最后的清除时必须等待。

- 11 退出 ufsrestore 程序。

```
ufsrestore> quit
```

随后将显示 shell 提示符。

- 12 验证恢复的文件。

- a. 列出恢复的文件和目录。

```
# ls -l
```

此时将显示文件和目录的列表。

- b. 检查该列表，以确保列表中指定的所有文件和目录都已恢复。

- 13 将文件移至正确的目录。

示例 27-2 以交互方式恢复文件

以下示例说明如何从备份磁带中提取 `/etc/passwd` 和 `/etc/shadow` 文件。

```
# cd /var/tmp

# ufsrestore if /dev/rmt/0

ufsrestore> ls

.:

.:

.sunw/      export/     net/        sbin/       usr/

Sources/    etools/     opt/        scde/       var/

b/          home/       ptools/     set/        vol/

bin         kernel/     pkg/        share/

dev/        lib/        platform/   shared/

devices/    lost+found/ proc/        src/

etc/        mnt/        rtools/     tmp/

ufsrestore> cd etc

ufsrestore> add passwd shadow

ufsrestore> verbose

verbose mode on

ufsrestore> extract

Extract requested files

You have not read any volumes yet.

Unless you know which volume your file(s) are on you should start

with the last volume and work towards the first.

Specify next volume #: 1
```

```

extract file ./etc/shadow

extract file ./etc/passwd

Add links

Set directory mode, owner, and times.

set owner/mode for '.'? [yn] n

ufsrestore> quit

# cd etc

# mv passwd /etc

# mv shadow /etc

# ls -l /etc

```

▼ 如何以非交互方式恢复特定文件

- 1 成为超级用户或承担等效角色。
- 2 （可选）为安全起见，请对磁带进行写保护。
- 3 将卷 1 磁带插入磁带机。

- 4 转至用于临时将文件恢复到的目录。

```
# cd /var/tmp
```

- 5 恢复文件。

```
# ufsrestore xvf /dev/rmt/n filename
```

x 通知 ufsrestore 在 *filename* 参数中复制特定文件或目录。

v 在恢复文件时显示文件名。

f /dev/rmt/n 标识磁带设备名称。

filename 指定一个或多个文件名或目录名（以空格分隔）。例如：
./export/home/user1/mail ./export/home/user2/mail。

- 6 键入文件所在的卷编号。按回车键。

```
Specify next volume #: 1
```

文件将被恢复到当前工作目录。

- 7 要维护当前目录的模式，请在 set owner/mode 提示符下键入 n 并按回车键。

```
set owner/mode for '.'? [yn] n
```

- 8 验证恢复的文件。

- a. 列出恢复的文件和目录。

```
# ls -l
```

此时将显示文件和目录的列表。

- b. 检查该列表，以确保列表中指定的所有文件和目录都已恢复。

- 9 将文件移至正确的目录。

示例 27-3 以非交互方式恢复特定文件

以下示例说明如何以非交互方式将 passwd 和 shadow 文件恢复到 /var/tmp 目录。

```
# cd /var/tmp
```

```
# ufsrestore xvf /dev/rmt/0 ./etc/passwd ./etc/shadow
```

```
Verify volume and initialize maps
```

```
Media block size is 126
```

```
Dump   date: Wed Jul 28 16:13:52 2004
```

```
Dumped from: the epoch
```

```
Level 0 dump of / on starbug:/dev/dsk/c0t0d0s0
```

```
Label: none
```

```
Extract directories from tape
```

```
Initialize symbol table.
```

```
Extract requested files
```

```
You have not read any volumes yet.
```

```

Unless you know which volume your file(s) are on you should start
with the last volume and work towards the first.

Specify next volume #: 1

extract file ./etc/passwd

extract file ./etc/shadow

Add links

Set directory mode, owner, and times.

Specify next volume #:1

extract file ./etc/passwd

extract file ./etc/shadow

Add links

Set directory mode, owner, and times.

set owner/mode for '.'? [yn] n

# cd etc

# mv passwd /etc

# mv shadow /etc

# ls -l /etc

```

示例 27-4 从远程磁带设备恢复文件

在使用 `ufsrestore` 命令时，通过在远程磁带设备名的前面添加 `remote-host:`，可以从远程磁带机中恢复文件。

以下示例说明如何使用 `venus` 系统中的远程磁带机 `/dev/rmt/0` 来恢复文件。

```
# ufsrestore xf venus:/dev/rmt/0 ./etc/hosts
```

▼ 如何恢复完整的文件系统

有时文件系统会遭到严重损坏，因此必须进行完全恢复。通常，在出现磁盘故障后，需要恢复完整的文件系统。在恢复软件之前，可能需要更换硬件。有关如何更换磁盘的信息，请参见第 247 页中的“SPARC: 添加系统磁盘或辅助磁盘（任务列表）”或第 261 页中的“x86: 添加系统磁盘或辅助磁盘（任务列表）”。

完全恢复文件系统（如 /export/home）可能需要很长时间。如果已持续备份文件系统，则可以将其恢复到上次增量备份时的状态。

注 - 不能使用此过程来恢复根 (/) 文件系统或 /usr 文件系统。有关恢复这些文件系统的说明，请参见第 516 页中的“如何恢复根 (/) 和 /usr 文件系统”。

- 1 成为超级用户或承担等效角色。

- 2 如有必要，请取消挂载文件系统。

```
# umount /dev/rdisk/device-name
```

或：

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

- 3 创建新的文件系统。

```
# newfs /dev/rdisk/device-name
```

系统将询问您是否要在原始设备上构造新的文件系统。确认 *device-name* 正确，以便不会破坏其他文件系统。

有关更多信息，请参见 newfs(1M) 手册页。

- 4 确认是否应该创建新的文件系统。

```
newfs: construct a new file system /dev/rdisk/cwtxdysz:(y/n)? y
```

将创建新的文件系统。

- 5 在临时挂载点上挂载新的文件系统。

```
# mount /dev/dsk/device-name /mnt
```

- 6 转到对应的挂载点目录。

```
# cd /mnt
```

- 7 （可选）为安全起见，请对磁带进行写保护。

- 8 将级别 0 磁带的第一卷插入磁带机中。

9 恢复文件。

```
# ufsrestore rvf /dev/rmt/n
```

此时会恢复转储级别 0 备份。如果备份需要多个磁带，则系统会提示您按数字顺序装入各磁带。

10 取出磁带并在驱动器中装入下一级别的磁带。

请始终从转储级别 0 开始恢复磁带，并继续进行其他转储级别的恢复，直到到达最高的转储级别。

11 按从低级别到高级别的顺序，对每个转储级别重复执行步骤 8 到步骤 10。**12 验证文件系统是否已恢复。**

```
# ls
```

13 删除 restoresymtable 文件。

```
# rm restoresymtable
```

系统将删除已创建且通过 ufsrestore 命令执行检查点恢复的 restoresymtable 文件。

14 转到其他目录。

```
# cd /
```

15 取消挂载最近恢复的文件系统。

```
# umount /mnt
```

16 取出最后一个磁带并在磁带机中插入未进行写保护的新磁带。**17 对最近恢复的文件系统执行级别 0 备份。**

```
# ufsdump 0ucf /dev/rmt/n /dev/rdisk/device-name
```

此时会执行级别 0 备份。由于 ufsrestore 命令将重定位文件并更改 inode 分配，因此应始终立即执行新创建文件系统的完整备份。

18 挂载恢复的文件系统。

```
# mount /dev/dsk/device-name mount-point
```

恢复的文件系统已挂载并且可以使用。

19 验证恢复且挂载的文件系统是否可用。

```
# ls mount-point
```

示例 27-5 恢复完整的文件系统

以下示例说明如何恢复 /export/home 文件系统。

```
# newfs /dev/rdisk/c0t0d0s7

newfs: /dev/rdisk/c0t0d0s7 last mounted as /export/home

newfs: construct a new file system /dev/rdisk/c0t0d0s7: (y/n)? y

819314 sectors in 867 cylinders of 15 tracks, 63 sectors

      400.1MB in 55 cyl groups (16 c/g, 7.38MB/g, 3584 i/g)

super-block backups (for fsck -F ufs -o b=#) at:

    32, 15216, 30400, 45584, 60768, 75952, 91136, 106320, 121504, 136688,
    681264, 696448, 711632, 725792, 740976, 756160, 771344, 786528, 801712,
    816896,

# mount /dev/dsk/c0t0d0s7 /mnt

# cd /mnt

# ufsrestore rvf /dev/rmt/0

Verify volume and initialize maps

Media block size is 126

Dump   date: Thu Jul 29 10:14:00 2004

Dumped from: the epoch

Level 0 dump of /export/home on starbug:/dev/dsk/c0t0d0s7

Label: none

Begin level 0 restore

Initialize symbol table.

Extract directories from tape

Calculate extraction list.
```

```
Warning: ./lost+found: File exists
```

```
Make node ./rimmer
```

```
Make node ./rimmer/wdir
```

```
Make node ./lister
```

```
Make node ./pmorph
```

```
Make node ./inquisitor
```

```
Make node ./kryten
```

```
Make node ./kryten/letters
```

```
Make node ./kryten/reports
```

```
Extract new leaves.
```

```
Check pointing the restore
```

```
extract file ./rimmer/words
```

```
extract file ./rimmer/words1
```

```
extract file ./rimmer/words2
```

```
extract file ./rimmer/words3
```

```
extract file ./rimmer/wdir/words
```

```
extract file ./rimmer/wdir/words1
```

```
extract file ./rimmer/wdir/words2
```

```
extract file ./rimmer/wdir/words3
```

```
.
```

```
.
```

```
.
```

```
Add links
```

```
Set directory mode, owner, and times.

Check the symbol table.

Check pointing the restore

# rm restoresymtable

# cd /

# umount /mnt

# ufsdump 0ucf /dev/rmt/0 /export/home

.
.
.

# mount /dev/dsk/c0t0d0s7 /export/home

# ls /export/home
```

▼ 如何恢复根 (/) 和 /usr 文件系统

- 1 成为超级用户或承担等效角色。
- 2 将新的系统磁盘添加到将在其中恢复根 (/) 和 /usr 文件系统的系统。
有关添加系统磁盘的详细说明，请参阅第 248 页中的“SPARC: 如何连接系统磁盘并进行引导”和第 262 页中的“x86: 如何连接系统磁盘”。
- 3 在临时挂载点上挂载新的文件系统。

```
# mount /dev/dsk/device-name /mnt
```
- 4 转到 /mnt 目录。

```
# cd /mnt
```
- 5 （可选）为安全起见，请对磁带进行写保护。
- 6 创建磁带设备的链接。

```
# tapes
```

7 恢复根 (/) 文件系统。

```
# ufsrestore rvf /dev/rmt/n
```

系统将恢复转储级别 0 磁带。

8 取出磁带并在驱动器中装入下一级别的磁带。

始终从转储级别 0 开始按照从低转储级别到高转储级别的顺序持续恢复磁带。

9 根据需要进行恢复。

```
# ufsrestore rvf /dev/rmt/n
```

系统将恢复下一级别磁带。

10 对每盒其他磁带，重复步骤 8 和步骤 9。**11 验证文件系统是否已恢复。**

```
# ls
```

12 删除 restoresymtable 文件。

```
# rm restoresymtable
```

系统将删除已创建且通过 ufsrestore 命令执行检查点恢复的 restoresymtable 文件。

13 转到根目录 (/)。

```
# cd /
```

14 取消挂载新创建的文件系统。

```
# umount /mnt
```

15 检查新的文件系统。

```
# fsck /dev/rdisk/device-name
```

检查所恢复文件系统的一致性。

16 在根分区中创建引导块。

在 SPARC 系统上：

```
# installboot /usr/platform/'uname-i'/lib/fs/ufs/bootblk
```

```
/dev/rdisk/device-name
```

有关更多信息，请参见 installboot(1M) 手册页。

有关在基于 SPARC 的系统上使用 installboot 命令的示例，请参见[示例 27-6](#)。

在 x86 系统上：

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

有关更多信息，请参见 `installgrub(1M)`。

有关在基于 x86 的系统上使用 `installgrub` 命令的示例，请参见[示例 27-7](#)。

17 在磁带机中插入新的磁带。

18 备份新的文件系统。

```
# ufsdump 0uf /dev/rmt/n /dev/rdsk/device-name
```

系统将执行转储级别 0 备份。由于 `ufsrestore` 命令将重定位文件并更改 inode 分配，因此应始终立即执行新创建文件系统的完整备份。

19 对 /usr 文件系统重复步骤 5 到 16（如果必要）。

20 重新引导系统。

```
# init 6
```

将重新引导系统。

示例 27-6 SPARC: 恢复根 (/) 文件系统

本示例说明如何在 SPARC 系统上恢复根 (/) 文件系统。本示例假设系统是从本地 CD 或从网络进行引导的。

```
# mount /dev/dsk/c0t3d0s0 /mnt
```

```
# cd /mnt
```

```
# tapes
```

```
# ufsrestore rvf /dev/rmt/0
```

```
# ls
```

```
# rm restoresymtable
```

```
# cd /
```

```
# umount /mnt
```

```
# fsck /dev/rdsk/c0t3d0s0
```

```
# installboot /usr/platform/sun4u/lib/fs/ufs/bootblk
```

```
/dev/rdisk/c0t3d0s0

# ufsdump 0uf /dev/rmt/0 /dev/rdisk/c0t3d0s0

# init 6
```

示例 27-7 x86: 恢复根 (/) 文件系统

本示例说明如何在 x86 系统上恢复根 (/) 文件系统。本示例假设系统是从 GRUB 故障安全引导会话、本地 CD 或网络进行引导的。

```
# mount /dev/dsk/c0t3d0s0 /mnt

# cd /mnt

# tapes

# ufsrestore rvf /dev/rmt/0

# ls

# rm restoresymtable

# cd /

# umount /mnt

# fsck /dev/rdisk/c0t3d0s0

# /sbin/installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdisk/c0t3d0s0

stage1 written to partition 0 sector 0 (abs 2016)

stage2 written to to partition 0, 227 sectors starting 50 (abs 2066)

# ufsdump 0uf /dev/rmt/0 /dev/rdisk/c0t3d0s0

# init 6
```


UFS 备份和恢复命令（参考）

本章包含有关 `ufsdump` 和 `ufsrestore` 命令的参考信息。

以下是本章中信息的列表。

- 第 521 页中的“`ufsdump` 命令的工作原理”
- 第 525 页中的“指定 `ufsdump` 命令选项和参数”
- 第 526 页中的“`ufsdump` 命令和安全问题”
- 第 526 页中的“指定 `ufsrestore` 选项和参数”

有关执行备份的概述信息，请参见第 24 章。

有关备份任务的信息，请参见第 25 章。

`ufsdump` 命令的工作原理

备份文件系统时，`ufsdump` 命令将执行两遍操作。在第一遍操作中，此命令将扫描文件系统的原始设备文件，并在内存中生成一个由目录和文件组成的表。然后，此命令会将该表写入备份介质。在第二遍操作中，`ufsdump` 命令将按数字顺序遍历 inode，读取文件内容并将数据写入备份介质。

确定设备特征

`ufsdump` 命令只需要了解适当的磁带块大小以及检测介质结尾的方式。

检测介质结尾

`ufsdump` 命令可写入大小固定的记录序列。当 `ufsdump` 命令收到通知，得知只写入了某个记录的一部分时，即假定已经到达该介质的物理结尾。此方法适用于大多数设备。如果设备无法通知 `ufsdump` 命令仅写入了部分记录，则当 `ufsdump` 命令尝试写入其他记录时会出现介质错误。

注-DAT 设备和 8 毫米磁带设备会检测介质结尾，而盒式磁带设备和 1/2 英寸磁带设备则不会检测介质结尾。

ufsdump 命令将自动检测大多数设备的介质结尾。因此，通常不需要使用 -c、-d、-s 和 -t 选项来执行多卷备份。

如果 ufsdump 命令不了解设备检测介质结尾的方式，则需要使用介质结尾选项。

为确保与 restore 命令兼容，大小选项仍然可以强制 ufsdump 命令在到达当前磁带或软盘的结尾之前转至下一个磁带或软盘。

使用 ufsdump 命令复制数据

ufsdump 命令仅从原始磁盘片中复制数据。如果文件系统仍然处于活动状态，则很可能不会复制内存缓冲区中的任何数据。通过 ufsdump 命令执行的备份不会复制空闲块，也不会生成磁盘片的映像。如果符号链接指向其他片上的文件，则会复制该链接本身。

/etc/dumpdates 文件的作用

当 ufsdump 命令与 -u 选项结合使用时，可以维护和更新 /etc/dumpdates 文件。
/etc/dumpdates 文件中的每行都将显示以下信息：

- 备份的文件系统
- 上次备份的转储级别
- 备份的周日期、日期和时间

例如：

```
# cat /etc/dumpdates

/dev/rdisk/c0t0d0s0          0 Wed Jul 28 16:13:52 2004

/dev/rdisk/c0t0d0s7          0 Thu Jul 29 10:36:13 2004

/dev/rdisk/c0t0d0s7          9 Thu Jul 29 10:37:12 2004
```

执行增量备份时，ufsdump 命令将检查 /etc/dumpdates 文件以找到下一个较低转储级别的最近一次备份的日期。然后，此命令将从该较低级别备份日期以来修改的所有文件复制到介质。备份完成后，描述刚完成的备份的新信息行将取代描述该级别前一次备份的信息行。

请使用 /etc/dumpdates 文件来验证备份是否正在进行。如果您的设备出现了问题，则此验证特别重要。如果由于设备故障无法完成备份，则不会在 /etc/dumpdates 文件中记录备份。

如果需要恢复整个磁盘，请检查 `/etc/dumpdates` 文件中的最近的备份日期和级别列表，以便可以确定恢复整个文件系统所需的磁带。

注 - `/etc/dumpdates` 文件是可编辑的文本文件。但是，如果要编辑该文件，请自担风险。如果对文件所做的更改与归档文件磁带不匹配，则可能无法找到所需的磁带（或文件）。

备份设备 (*dump-file*) 参数

dump-file 参数（与 `-f` 选项对应）指定备份目标。该目标可以为以下设备之一：

- 本地磁带机
- 本地软盘驱动器
- 远程磁带机
- 远程软盘驱动器
- 标准输出

当目标不是缺省的本地磁带机 `/dev/rmt/0` 时，请使用此参数。如果使用 `-f` 选项，则必须为 *dump-file* 参数指定值。

注 - *dump-file* 参数还可以指向本地磁盘或远程磁盘上的文件。如果操作有误，则错误使用情况会充满整个文件系统。

本地磁带机或软盘驱动器

通常，*dump-file* 参数将为磁带设备或软盘指定原始设备文件。当 `ufsdump` 命令写入输出设备时，将创建一个可能跨多盒磁带或多个软盘的单个备份文件。

通过使用设备缩写，可以指定系统上的磁带设备或软盘。第一个设备始终为 `0`。例如，如果您具有 SCSI 磁带控制器和一个使用中密度格式的 QIC-24 磁带机，请使用以下设备名称：

```
/dev/rmt/0m
```

指定磁带设备名称时，还可以在名称的结尾处键入字母 `"n"`，以指示完成备份后磁带机不应该反绕。例如：

```
/dev/rmt/0mn
```

如果要将多个文件置于磁带上，请使用“不反绕”选项。如果备份期间空间不足，则在 `ufsdump` 命令要求新磁带之前，该磁带不会反绕。有关设备命名约定的完整说明，请参见第 552 页中的“备份设备名称”。

远程磁带机或软盘驱动器

可以使用语法 *host:device* 来指定远程磁带设备或远程软盘。如果本地系统上的超级用户有权访问远程系统，则 `ufsdump` 命令将写入远程设备。如果您通常以超级用户身份运行 `ufsdump` 命令，则远程系统上的 `/rhosts` 文件中必须包括本地系统的名称。如果以 *user@host:device* 格式指定设备，则 `ufsdump` 命令将尝试以指定用户的身份访问远程系统上的设备。在这种情况下，远程系统上的 `/rhosts` 文件中必须包括指定的用户。

对于设备所在的系统（而不是用于运行 `ufsdump` 命令的系统），请使用与操作系统匹配的设备的命名约定。如果驱动器所在的系统使用的是以前的 SunOS 发行版（例如 4.1.1），请使用 SunOS 4.1 设备名称（例如 `/dev/rst0`）。如果系统正在运行 Solaris 软件，请使用 SunOS 5.9 约定（例如 `/dev/rmt/0`）。

借助 ufsdump 命令使用标准输出

将连字符 (-) 指定为 *dump-file* 参数时，`ufsdump` 命令将写入标准输出。

注 – 当 *dump-file* 参数为标准输出时，`-v` 选项（验证）不起作用。

可以在管道中使用 `ufsdump` 和 `ufsrestore` 命令来复制文件系统，具体操作为：使用 `ufsdump` 命令写入标准输出，并使用 `ufsrestore` 命令读取标准输入。例如：

```
# ufsdump 0f - /dev/rdisk/c0t0d0s7 | (cd /home; ufsrestore xf -)
```

指定要备份的文件

在命令行中，必须始终以 *filenames* 作为最后一个参数。此参数指定备份的来源或内容。

对于文件系统，请按照以下方式指定原始设备文件：

```
/dev/rdisk/c0t0d0s7
```

只要文件系统对应的项存在于 `/etc/vfstab` 文件中，即可通过其挂载点目录（例如 `/export/home`）来指定文件系统。

有关设备命名约定的完整说明，请参见第 552 页中的“备份设备名称”。

对于各个文件或目录，请键入一个或多个名称（以空格分隔）。

注 – 使用 `ufsdump` 命令备份一个或多个目录或文件（而不是整个文件系统）时，将执行级别 0 备份。增量备份不适用。

指定磁带特征

如果未指定任何磁带特征，则 `ufsdump` 命令将使用一组缺省值。可以指定磁带盒 (c)、密度 (d)、大小 (s) 以及磁轨数 (t)。请注意，您可以按照任意顺序指定选项，只要其后的参数与选项顺序匹配即可。

ufsdump 命令的限制

`ufsdump` 命令不能执行以下操作：

- 自动计算备份文件系统所需的磁带或软盘的数目。在实际备份文件系统之前，可以使用预运行模式 (s 选项) 来确定所需的空间量。
- 提供内置错误检查，使备份活动文件系统时出现的问题降到最少。
- 备份从服务器远程挂载的文件。服务器上的文件必须在服务器本身中进行备份。系统拒绝用户享有对其所有的文件（位于服务器上）运行 `ufsdump` 命令的权限。

指定 ufsdump 命令选项和参数

本节介绍如何为 `ufsdump` 命令指定选项和参数。`ufsdump` 命令的语法如下：

```
/usr/sbin/ufsdump options arguments filenames
```

<i>options</i>	包含一个字母的选项名称的单个字符串。
<i>arguments</i>	标识选项参数，并且可能包含多个字符串。选项字母必须与其关联参数顺序相同。
<i>filenames</i>	标识要备份的文件。这些参数必须始终位于最后，每个参数以空格分隔。

缺省的 ufsdump 选项

如果运行不含任何选项的 `ufsdump` 命令，请使用此语法：

```
# ufsdump filenames
```

缺省情况下，`ufsdump` 命令将使用以下选项和参数：

```
ufsdump 9uf /dev/rmt/0 filenames
```

这些选项以其首选密度对缺省磁带机执行级别 9 增量备份。

有关 `ufsdump` 选项的说明，请参见 `ufsdump(1M)`。

ufsdump 命令和安全问题

如果您担心安全性，则应执行以下操作：

- 要求超级用户具备 `ufsdump` 命令的访问权限。
- 确保从客户机和服务器上的 `/.rhosts` 文件中删除超级用户访问项（如果您执行的是集中式备份）。

有关安全性的常规信息，请参见《系统管理指南：安全性服务》。

指定 ufsrestore 选项和参数

`ufsrestore` 命令的语法如下：

`/usr/sbin/ufsrestore options arguments filenames`

options 包含一个字母的选项名称的单个字符串。必须且只能选择以下选项之一：`i`、`r`、`R`、`t` 或 `x`。有关 `ufsrestore` 选项的说明，请参见 `ufsrestore(1M)`。

arguments 跟在选项字符串后与选项匹配的参数。选项字母必须与其关联参数顺序相同。

filenames 指定要作为 `x` 或 `t` 选项的参数进行恢复的一个或多个文件。这些参数必须始终位于最后，且以空格分隔。

复制 UFS 文件和文件系统（任务）

本章介绍如何使用各种备份命令将 UFS 文件和文件系统复制到磁盘、磁带和软盘。

以下是本章中的逐步说明列表。

- 第 530 页中的 “如何复制磁盘 (dd)”
- 第 534 页中的 “如何在文件系统之间复制目录 (cpio)”
- 第 535 页中的 “如何将文件复制到磁带 (tar)”
- 第 537 页中的 “如何列出磁带中的文件 (tar)”
- 第 537 页中的 “如何从磁带中恢复文件 (tar)”
- 第 538 页中的 “使用 pax 命令将文件复制到磁带”
- 第 539 页中的 “如何将目录中的所有文件复制到磁带 (cpio)”
- 第 540 页中的 “如何列出磁带中的文件 (cpio)”
- 第 541 页中的 “如何从磁带中恢复所有文件 (cpio)”
- 第 542 页中的 “如何从磁带中恢复特定文件 (cpio)”
- 第 543 页中的 “如何将文件复制到远程磁带设备 (tar 和 dd)”
- 第 545 页中的 “如何从远程磁带设备中提取文件”
- 第 546 页中的 “如何将文件复制到单张经过格式化的软盘 (tar)”
- 第 547 页中的 “如何列出软盘中的文件 (tar)”
- 第 548 页中的 “如何从软盘中恢复文件 (tar)”

用于复制文件系统的命令

需要备份和恢复完整的文件系统时，请使用第 28 章中介绍的 `ufsdump` 和 `ufsrestore` 命令。需要复制或移动各个文件、文件系统的各部分或完整的文件系统时，可以使用本章中介绍的过程，而不是 `ufsdump` 和 `ufsrestore` 命令。

下表介绍何时使用各种备份命令。

表 29-1 何时使用各种备份命令

任务	命令	更多信息
将文件系统备份到磁带。	ufsdump	第 482 页中的 “如何将文件系统备份到磁带”
创建文件系统快照。	fssnap	第 26 章
从磁带恢复文件系统。	ufsrestore	第 512 页中的 “如何恢复完整的文件系统”
将文件传输到其他系统。	pax、tar或cpio	第 535 页中的 “将文件和文件系统复制到磁带”
在磁盘之间复制文件或文件系统。	dd	第 530 页中的 “如何复制磁盘 (dd)”
将文件复制到软盘。	tar	第 546 页中的 “如何将文件复制到单张经过格式化的软盘 (tar)”

下表介绍了各种备份和恢复命令。

表 29-2 各种备份命令汇总

命令名称	是否可识别文件系统边界？	是否支持多卷备份？	物理复制还是逻辑复制？
volcopy	是	是	物理
tar	否	否	逻辑
cpio	否	是	逻辑
pax	是	是	逻辑
dd	是	否	物理
ufsdump/ufsrestore	是	是	逻辑
fssnap	N/A	N/A	逻辑

下表介绍了其中某些命令的优点和缺点。

表 29-3 tar、pax 和 cpio 命令的优点和缺点

命令	功能	优点	缺点
tar	用于将文件和目录子树复制到单个磁带。	<ul style="list-style-type: none">■ 可以在大多数 UNIX 操作系统中使用■ 可以轻松访问公共版本	<ul style="list-style-type: none">■ 不可识别文件系统边界■ 全路径名的长度不能超过 255 个字符■ 不能用于创建多个磁带卷
pax	用于复制需要多个磁带卷的文件、特殊文件或文件系统。或者，当要在与 POSIX 兼容的系统之间复制文件时使用。	<ul style="list-style-type: none">■ 对于与 POSIX 兼容的系统，其可移植性比 tar 或 cpio 命令高■ 多个供应商支持	与 tar 命令的缺点相同，不同的是 pax 命令可以创建多个磁带卷。
cpio	用于复制需要多个磁带卷的文件、特殊文件或文件系统。或者，当要将文件从运行当前的 Solaris 发行版的系统复制到运行 SunOS 4.0/4.1 发行版的系统时使用。	<ul style="list-style-type: none">■ 与使用 tar 命令相比，可以更有效地将数据打包到磁带■ 恢复时跳过磁带中所有的错误点■ 提供以不同的头格式编写文件的选项（如 tar、ustar、crc、odc、bar），以实现不同系统类型之间的可移植性■ 创建多个磁带卷	该命令的语法比 tar 或 pax 命令更为复杂。

以下各节介绍有关如何使用这些命令的逐步说明和示例。

在磁带之间复制文件系统

以下两个命令用于在磁带之间复制文件系统：

- volcopy
- dd

有关 volcopy 的更多信息，请参见 volcopy(1M)。

下一节介绍如何使用 dd 命令在磁带之间复制文件系统。

创建精确的文件系统副本

`dd` 命令可以将完整的 UFS 文件系统精确地复制到其他文件系统或磁带（块级别）。缺省情况下，`dd` 命令会将标准输入复制到标准输出。

注 – 如果未首先指定适当的块大小，请不要将 `dd` 命令和磁带长度可变的磁带机一同使用。

可以指定设备名称而不是指定标准输入和/或标准输出。在本示例中，软盘的内容将被复制到 `/tmp` 目录中的某个文件：

```
$ dd < /floppy/floppy0 > /tmp/output.file
```

```
2400+0 records in
```

```
2400+0 records out
```

`dd` 命令可报告它所读取和写入的块数目。+ 号后的数字是已复制的部分块的数目。缺省块大小为 512 字节。

`dd` 命令的语法与大多数其他命令不同。选项以 *keyword=value* 对的形式指定，其中 *keyword* 是要设置的选项，*value* 是该选项的参数。例如，可以使用以下语法替换标准输入和标准输出：

```
$ dd if=input-file of=output-file
```

要使用 *keyword=value* 对（而不使用重定向符号），则可以键入以下内容：

```
$ dd if=/floppy/floppy0 of=/tmp/output.file
```

▼ 如何复制磁盘 (dd)

考虑复制磁盘时，请牢记以下要点：

- 请勿使用此过程复制由卷管理器控制的磁盘。
- 将 UFS 文件系统数据从一个磁盘或系统复制到另一个磁盘或系统的主要方法是使用 `ufsdump` 和 `ufsrestore` 命令。有关使用这些命令的更多信息，请参见第 24 章。
- 通过创建 Flash 归档文件并将其复制到目标系统，可以克隆系统。有关创建 Flash 归档文件的更多信息，请参见《Solaris 10 Installation Guide: Solaris Flash Archives (Creation and Installation)》。
- 如果要复制带有 EFI 磁盘标号的磁盘，请参见示例 29-2。

如果您仍在考虑使用 `dd` 命令复制磁盘，请牢记以下注意事项：

- 确保源磁盘和目标磁盘具有相同的磁盘几何参数。
- 检查磁盘中要使用 `fsck` 实用程序复制的 UFS 文件系统。
- 确保在使用 `dd` 命令复制磁盘时系统处于单用户模式。

1 成为超级用户或承担等效角色。

2 (可选) 创建 `/reconfigure` 文件, 以使系统能够在重新引导时识别要添加的目标磁盘 (如有必要)。

```
# touch /reconfigure
```

3 关闭系统。

```
# init 0
```

4 将目标磁盘连接到系统。

5 引导系统。

```
ok boot -s
```

6 将源磁盘复制到目标磁盘。

```
# dd if=/dev/rdisk/device-name of=/dev/rdisk/device-name bs=block-size
```

`if=/dev/rdisk/device-name` 表示主磁盘设备重叠片, 通常为片 2。

`of=/dev/rdisk/device-name` 表示目标磁盘设备重叠片, 通常为片 2。

`bs=blocksize` 用于标识块大小, 如 128 KB 或 256 KB。较大的块大小可以缩短复制磁盘所用的时间。

有关更多信息, 请参见 `dd(1M)`。

7 检查新的文件系统。

```
# fsck /dev/rdisk/device-name
```

8 挂载目标磁盘的根 (/) 文件系统。

```
# mount /dev/dsk/device-name /mnt
```

9 转到 `/etc/vfstab` 文件所在的目录。

```
# cd /mnt/etc
```

10 使用文本编辑器编辑目标磁盘的 `/etc/vfstab` 文件, 以引用正确的设备名称。例如, 将 `c0t3d0` 的所有实例都更改为 `c0t1d0`。

11 转到目标磁盘的根 (/) 目录。

```
# cd /
```

- 12 取消挂载目标磁盘的根 (/) 文件系统。

```
# umount /mnt
```

- 13 关闭系统。

```
# init 0
```

- 14 从目标磁盘引导至单用户模式。

```
# boot diskn -s
```

注 - 目标磁盘不需要 `installboot` 命令，因为引导块是作为重叠片的一部分进行复制的。

- 15 取消配置目标磁盘。

```
# sys-unconfig
```

系统在取消配置后即会关闭。

- 16 再次从目标磁盘引导，并提供其系统信息，如主机名、时区等。

```
# boot diskn
```

- 17 引导系统后，以超级用户身份登录以验证系统信息。

```
hostname console login:
```

示例 29-1 复制带有 VTOC 标签的磁盘 (dd)

本示例说明如何将主磁盘（带有 VTOC 标签）`/dev/rdisk/c0t0d0s2` 复制到目标磁盘 `/dev/rdisk/c0t2d0s2`。

```
# touch /reconfigure
```

```
# init 0
```

```
ok boot
```

```
# dd if=/dev/rdisk/c0t0d0s2 of=/dev/rdisk/c0t2d0s2 bs=128k
```

```
# fsck /dev/rdisk/c0t2d0s2
```

```
# mount /dev/dsk/c0t2d0s2 /mnt
```

```
# cd /mnt/etc
```

```
# vi vfstab
```

(Modify entries for the new disk)

```
# cd /

# umount /mnt

# init 0

# boot disk2 -s

# sys-unconfig

# boot disk2
```

示例 29-2 复制带有 EFI 标签的磁盘 (dd)

在以前的 Solaris 发行版中，片 2(s2) 用于表示整张磁盘。在带有 EFI 标签的磁盘上，必须使用稍有不同的过程来克隆或复制大于 1 TB 的磁盘，以使克隆磁盘的 UUID 具有唯一性。如果没有为克隆磁盘创建新的标签，则其他软件产品在遇到重复的 UUID 时可能会损坏带有 EFI 标记的磁盘上的数据。

例如：

1. 克隆带有 EFI 标记的磁盘。例如：

```
# dd if=/dev/rdisk/c0t0d0 of=/dev/rdisk/c0t2d0 bs=128k
```

2. 传输要复制到 fmthard 命令中的磁盘的 prtvtoc 输出，以便为克隆磁盘创建新的标签。例如：

```
# prtvtoc /dev/rdisk/c0t0d0 | fmthard -s - /dev/rdisk/c0t2d0
```

有关 EFI 磁盘标签的更多信息，请参见第 202 页中的“带有 EFI 磁盘标号的多 TB 磁盘支持”。

在文件系统之间复制目录（cpio 命令）

可以使用 cpio（复制入和复制出）命令复制各个文件、多组文件或完整的文件系统。本节介绍如何使用 cpio 命令复制完整的文件系统。

cpio 命令是一个归档程序，可以将文件列表复制到单个大型输出文件中。此命令会在各个文件之间插入头，以便于进行恢复。可以使用 cpio 命令将完整的文件系统复制到其他片、其他系统或介质设备（如磁盘或软盘）。

由于 `cpio` 命令可以识别介质结尾并会提示您插入另一个卷，因此它是除了 `ufsdump` 以外最有效的命令，可用于创建需要多盒磁带或多张软盘的归档文件。

使用 `cpio` 命令时，经常将 `ls` 和 `find` 命令结合使用以列出和选择要复制的文件，然后将输出传输给 `cpio` 命令。

▼ 如何在文件系统之间复制目录 (cpio)

- 1 成为超级用户或承担等效角色。

- 2 转至相应的目录。

```
# cd filesystem1
```

- 3 通过将 `find` 命令与 `cpio` 命令结合使用，将目录树从 *filesystem1* 复制到 *filesystem2*。

```
# find . -print -depth | cpio -pdm filesystem2
```

. 从当前工作目录开始。

-print 列显文件名。

-depth 向下派生目录分层结构并从下到上列显文件名。

-p 创建文件列表。

-d 根据需要创建目录。

-m 在目录中设置正确的修改时间。

有关更多信息，请参见 `cpio(1)`。

系统将复制指定目录名中的文件。符号链接会被保留。

可能还要指定 `-u` 选项。此选项将强制执行无条件复制。否则，旧文件不会替代新文件。在需要目录的精确副本且复制的某些文件可能已存在于目标目录中时，此选项可能非常有用。

- 4 通过显示目标目录的内容，确认复制成功。

```
# cd filesystem2
```

```
# ls
```

- 5 删除源目录（如果可行）。

```
# rm -rf filesystem1
```

示例 29-3 在文件系统之间复制目录 (cpio)

```
# cd /data1

# find . -print -depth | cpio -pdm /data2

19013 blocks

# cd /data2

# ls

# rm -rf /data1
```

将文件和文件系统复制到磁带

可以使用 `tar`、`pax` 和 `cpio` 命令将文件和文件系统复制到磁带。所选的命令取决于复制所需的灵活性和精确度。由于所有这三个命令都使用原始设备，因此在使用磁带之前不需要在其中格式化或创建文件系统。

使用的磁带机和设备名称取决于每个系统的硬件配置。有关磁带设备名称的更多信息，请参见第 551 页中的“选择要使用的介质”。

将文件复制到磁带 (tar 命令)

以下是在使用 `tar` 命令将文件复制到磁带之前应了解的信息：

- 如果使用带有 `-c` 选项的 `tar` 命令将文件复制到磁带，则会破坏磁带中位于当前磁带位置以及该位置后的任何已有文件。
- 可以使用文件名替换通配符（`?` 和 `*`）作为复制文件时指定的文件名的一部分。例如，要复制带有 `.doc` 后缀的所有文档，请键入 `*.doc` 作为文件名参数。
- 从 `tar` 归档文件中提取文件时，不能使用文件名替换通配符。

▼ 如何将文件复制到磁带 (tar)

- 1 转到包含要复制的文件的目录。
- 2 将可写磁带插入磁带机中。

3 将文件复制到磁带。

```
$ tar cvf /dev/rmt/n filenames
```

c 指明要创建归档文件。

v 归档时显示每个文件的名称。

f /dev/rmt/n 指明应该将归档文件写入指定设备或文件。

filenames 指明要复制的文件和目录。使用空格分隔多个文件。

会将指定的文件名复制到磁带，并且会覆写该磁带上的任何现有文件。

4 从磁带机中取出磁带。在磁带标签中写入文件名称。

5 确认复制的文件位于该磁带中。

```
$ tar tvf /dev/rmt/n
```

有关列出 tar 磁带中的文件的更多信息，请参见第 537 页中的“[如何列出磁带中的文件 \(tar\)](#)”。

示例 29-4 将文件复制到磁带 (tar)

以下示例说明如何将三个文件复制到磁带机 0 中的磁带。

```
$ cd /export/home/kryten
```

```
$ ls reports
```

```
reportA reportB reportC
```

```
$ tar cvf /dev/rmt/0 reports
```

```
a reports/ 0 tape blocks
```

```
a reports/reportA 59 tape blocks
```

```
a reports/reportB 61 tape blocks
```

```
a reports/reportC 63 tape blocks
```

```
$ tar tvf /dev/rmt/0
```


▼ 如何列出磁带中的文件(tar)

1 将磁带插入磁带机。

2 显示磁带内容。

```
$ tar tvf /dev/rmt/n
```

t 列出磁带中的文件目录。

v 与 t 选项一同使用，提供有关磁带中的文件的详细信息。

f /dev/rmt/n 指明磁带设备。

示例 29-5 列出磁带中的文件(tar)

以下示例显示磁带机 0 中的磁带上的文件列表。

```
$ tar tvf /dev/rmt/0
```

```
drwxr-xr-x  0/1          0 Jul 28 15:00 2004 reports/
```

```
-r--r--r--  0/1    206663 Jul 28 15:00 2004 reports/reportA
```

```
-r--r--r--  0/1    206663 Jul 28 15:00 2004 reports/reportB
```

```
-r--r--r--  0/1    206663 Jul 28 15:00 2004 reports/reportC
```

▼ 如何从磁带中恢复文件(tar)

1 转到要放置文件的目录。

2 将磁带插入磁带机。

3 从磁带中恢复文件。

```
$ tar xvf /dev/rmt/n [filenames]
```

x 指明应从指定的归档文件中提取文件。指定磁带机中的磁带上的所有文件都会被复制到当前目录。

v 恢复时显示每个文件的名称。

f /dev/rmt/n 指明包含归档文件的磁带设备。

filenames 指定要恢复的文件。使用空格分隔多个文件。

有关更多信息，请参见 tar(1) 手册页。

4 确认文件已被复制。

```
$ ls -l
```

示例 29-6 检索磁带中的文件 (tar)

以下示例说明如何从磁带机 0 中的磁带恢复所有文件。

```
$ cd /var/tmp

$ tar xvf /dev/rmt/0

x reports/, 0 bytes, 0 tape blocks

x reports/reportA, 0 bytes, 0 tape blocks

x reports/reportB, 0 bytes, 0 tape blocks

x reports/reportC, 0 bytes, 0 tape blocks

x reports/reportD, 0 bytes, 0 tape blocks

$ ls -l
```

故障排除 从磁带中提取的文件名称必须与归档文件中存储的文件名称完全匹配。如果对文件的名称或路径有任何质疑，请首先列出磁带中的文件。有关列出磁带中的文件的说明，请参见第 537 页中的“如何列出磁带中的文件 (tar)”。

使用 pax 命令将文件复制到磁带

▼ 如何将文件复制到磁带 (pax)

- 1 转到包含要复制的文件的目录。
- 2 将可写磁带插入磁带机中。
- 3 将文件复制到磁带。

```
$ pax -w -f /dev/rmt/n filenames

-w          启用写模式。

-f /dev/rmt/n 确定磁带机。
```

filenames 指明要复制的文件和目录。使用空格分隔多个文件。
有关更多信息，请参见 `pax(1)` 手册页。

4 确认文件已被复制到磁带。

```
$ pax -f /dev/rmt/n
```

5 从磁带机中取出磁带。在磁带标签中写入文件名称。

示例 29-7 将文件复制到磁带 (pax)

以下示例说明如何使用 `pax` 命令来复制当前目录中的所有文件。

```
$ pax -w -f /dev/rmt/0 .
```

```
$ pax -f /dev/rmt/0
```

```
filea fileb filec
```

使用 cpio 命令将文件复制到磁带

▼ 如何将目录中的所有文件复制到磁带 (cpio)

1 转到包含要复制的文件的目录。

2 将可写磁带插入磁带机中。

3 将文件复制到磁带。

```
$ ls | cpio -oc > /dev/rmt/n
```

`ls` 为 `cpio` 命令提供文件名列表。

`cpio -oc` 指定 `cpio` 命令应该在复制出模式 (-o) 下操作，并以 ASCII 字符格式 (-c) 写入头信息。这些选项确保可移植到其他供应商的系统。

`> /dev/rmt/n` 指定输出文件。

会将目录中的所有文件复制到指定磁带机中的磁带，同时会覆写该磁带中的任何现有文件。屏幕将显示复制的总块数。

4 确认文件已被复制到磁带。

```
$ cpio -civt < /dev/rmt/n
```

- c 指定 cpio 命令应以 ASCII 字符格式读取文件。
- i 指定 cpio 命令应在复制入模式下操作，即使该命令仅列出了此时的文件。
- v 显示与 ls -l 命令的输出格式类似的输出。
- t 列出指定磁带机中的磁带上的文件目录。
- < /dev/rmt/*n* 指定现有 cpio 归档文件的输入文件。

5 从磁带机中取出磁带。在磁带标签中写入文件名称。

示例 29-8 将目录中的所有文件复制到磁带 (cpio)

以下示例说明如何将 /export/home/kryten 目录中的所有文件复制到磁带机 0 中的磁带。

```
$ cd /export/home/kryten

$ ls | cpio -oc > /dev/rmt/0

16 blocks

$ cpio -civt < /dev/rmt/0

-rw-r--r--    1 root    other      0 Jul 28 14:59 2004, filea
-rw-r--r--    1 root    other      0 Jul 28 14:59 2004, fileb
-rw-r--r--    1 root    other      0 Jul 28 14:59 2004, filec
drwxr-xr-x    2 root    other      0 Jul 28 14:59 2004, letters
drwxr-xr-x    2 root    other      0 Jul 28 15:00 2004, reports

16 blocks

$
```

▼ 如何列出磁带中的文件 (cpio)

注 – 列出磁带中的目录需要很长时间，因为 cpio 命令必须处理整个归档文件。

1 将归档文件磁带插入磁带机。

2 列出磁带中的文件。

```
$ cpio -civt < /dev/rmt/n
```

示例 29-9 列出磁带中的文件 (cpio)

以下示例说明如何列出磁带机 0 中的磁带上的文件。

```
$ cpio -civt < /dev/rmt/0

-rw-r--r--    1 root    other        0 Jul 28 14:59 2004, filea
-rw-r--r--    1 root    other        0 Jul 28 14:59 2004, fileb
-rw-r--r--    1 root    other        0 Jul 28 14:59 2004, filec
drwxr-xr-x    2 root    other        0 Jul 28 14:59 2004, letters
drwxr-xr-x    2 root    other        0 Jul 28 15:00 2004, reports

16 blocks

$
```

▼ 如何从磁带中恢复所有文件 (cpio)

如果已使用相对路径名创建了归档文件，则恢复文件时会在当前目录中以目录形式生成输入文件。但是，如果已使用绝对路径名创建了归档文件，则将使用相同的绝对路径在系统中重新创建该文件。



注意 – 使用绝对路径名可能会有危险，因为可能会覆写系统中的现有文件。

1 转到要放置文件的目录。

2 将磁带插入磁带机。

3 从磁带中提取所有文件。

```
$ cpio -icvd < /dev/rmt/n
```

- i 从标准输入中提取文件。
- c 指定 cpio 命令应以 ASCII 字符格式读取文件。
- v 在以与 ls 命令输出类似的格式恢复文件时，显示这些文件。

-d 根据需要创建目录。
< /dev/rmt/*n* 指定输出文件。

4 确认文件已被复制。

```
$ ls -l
```

示例 29-10 从磁带中恢复所有文件 (cpio)

以下示例说明如何从磁带机 0 中的磁带恢复所有文件。

```
$ cd /var/tmp

cpio -icvd < /dev/rmt/0

answers

sc.directives

tests

8 blocks

$ ls -l
```

▼ 如何从磁带中恢复特定文件 (cpio)

- 1 转到要放置文件的目录。
- 2 将磁带插入磁带机。
- 3 从磁带中恢复文件子集。

```
$ cpio -icv "*file" < /dev/rmt/n
```

-i 从标准输入中提取文件。

-c 指定 cpio 命令应以 ASCII 字符格式读取头。

-v 在以与 ls 命令输出类似的格式恢复文件时，显示这些文件。

*"*file"* 指定与该模式匹配的所有文件都将复制到当前目录。可以指定多种模式，但是每种模式都必须用双引号引起来。

< /dev/rmt/*n* 指定输入文件。

有关更多信息，请参见 cpio(1) 手册页。

4 确认文件已被复制。

```
$ ls -l
```

示例 29-11 从磁带中检索特定文件 (cpio)

以下示例说明如何从磁带机 0 中的磁带恢复带有 chapter 后缀的所有文件。

```
$ cd /home/smith/Book

$ cpio -icv "*chapter" < /dev/rmt/0

Boot.chapter

Directory.chapter

Install.chapter

Intro.chapter

31 blocks

$ ls -l
```

将文件复制到远程磁带设备

▼ 如何将文件复制到远程磁带设备 (tar 和 dd)

1 必须满足以下先决条件，才能使用远程磁带机：

- a. 本地主机名和 (可选) 执行复制的用户的用户名必须出现在远程系统的 /etc/hosts.equiv 文件中。或者，执行复制的用户在远程计算机上必须具有可访问的对应起始目录，而且在 \$HOME/.rhosts 中必须具有本地计算机名。
有关更多信息，请参见 hosts.equiv(4) 手册页。
- b. 远程系统的项必须位于本地系统的 /etc/inet/hosts 文件中，或位于名称服务 hosts 文件中。

2 要测试您是否具有执行远程命令的适当权限，请键入以下内容：

```
$ rsh remotehost echo test
```

如果系统向您回显了 `test`，则您有权执行远程命令。如果系统向您回显了 `Permission denied`，请按照步骤 1 所述检查您的设置。

3 转到要放置文件的目录。

4 将磁带插入磁带机。

5 将文件复制到远程磁带机。

```
$ tar cvf - filenames | rsh remote-host dd of=/dev/rmt/n obs=block-size
```

`tar cf` 创建磁带归档文件，归档时列出文件并指定磁带设备。

`v` 提供有关 `tar` 文件项的其他信息。

`-`（连字符） 表示磁带设备的占位符。

`filenames` 标识要复制的文件。使用空格分隔多个文件。

`rsh|remote-host` 将 `tar` 命令的输出传输到远程 shell。

`dd of= /dev/rmt/n` 表示输出设备。

`obs=block-size` 表示分块因子。

6 从磁带机中取出磁带。在磁带标签中写入文件名称。

示例 29-12 将文件复制到远程磁带机（`tar` 和 `dd`）

```
# tar cvf - * | rsh mercury dd of=/dev/rmt/0 obs=126b
```

```
a answers/ 0 tape blocks
```

```
a answers/test129 1 tape blocks
```

```
a sc.directives/ 0 tape blocks
```

```
a sc.directives/sc.190089 1 tape blocks
```

```
a tests/ 0 tape blocks
```

```
a tests/test131 1 tape blocks
```

```
6+9 records in
```

```
0+1 records out
```


▼ 如何从远程磁带设备中提取文件

1 将磁带插入磁带机。

2 转至临时目录。

```
$ cd /var/tmp
```

3 从远程磁带设备中提取文件。

```
$ rsh remote-host dd if=/dev/rmt/n | tar xvBpf -
```

`rsh remote-host` 指明使用 `dd` 命令从磁带设备中提取文件时启动的远程 shell。

`dd if=/dev/rmt/n` 指明输入设备。

`| tar xvBpf -` 将 `dd` 命令的输出传输到用于恢复文件的 `tar` 命令。

4 确认文件已被提取。

```
$ ls -l
```

示例 29-13 从远程磁带机中提取文件

```
$ cd /var/tmp
```

```
$ rsh mercury dd if=/dev/rmt/0 | tar xvBpf -
```

```
x answers/, 0 bytes, 0 tape blocks
```

```
x answers/test129, 48 bytes, 1 tape blocks
```

```
20+0 records in
```

```
20+0 records out
```

```
x sc.directives/, 0 bytes, 0 tape blocks
```

```
x sc.directives/sc.190089, 77 bytes, 1 tape blocks
```

```
x tests/, 0 bytes, 0 tape blocks
```

```
x tests/test131, 84 bytes, 1 tape blocks
```

```
$ ls -l
```

将文件和文件系统复制到软盘

将文件或文件系统复制到软盘之前，必须对软盘进行格式化。有关如何格式化软盘的信息，请参见第2章。

使用 `tar` 命令可将 UFS 文件复制到一张经过格式化的软盘。

如果需要将 UFS 文件复制到多张经过格式化的软盘，请使用 `cpio` 命令。`cpio` 命令可以识别介质结尾并会提示您插入下一张软盘。

将文件复制到软盘时应了解的信息

- 如果使用 `tar -c` 命令将文件复制到经过格式化的软盘，则会损坏该软盘上已有的任何文件。
- 不能挂载包含 `tar` 映像的软盘。
- 如果需要使用多卷交换实用程序，则请使用 `cpio` 命令。`tar` 命令仅仅是一个单卷实用程序。

有关更多信息，请参见 `tar(1)`。

▼ 如何将文件复制到单张经过格式化的软盘 (`tar`)

- 1 转到包含要复制的文件的目录。
- 2 将经过格式化但没有被写保护的软盘插入驱动器。
- 3 使软盘可用。

```
$ volcheck
```

- 4 重新格式化软盘（如果必要）。

```
$ rmformat -U /dev/rdiskette
```

```
Formatting will erase all the data on disk.
```

```
Do you want to continue? (y/n)y
```

- 5 将文件复制到软盘。

```
$ tar cvf /vol/dev/aliases/floppy0 filenames
```

会将指定的文件名复制到该软盘，同时会覆写该软盘上的任何现有文件。

- 6 确认文件已被复制。

```
$ tar tvf /vol/dev/aliases/floppy0
```

有关列出文件的更多信息，请参见第 547 页中的“如何列出软盘中的文件 (tar)”。

- 7 从驱动器中取出软盘。

- 8 在软盘标签中写入文件名称。

示例 29-14 将文件复制到单张经过格式化的软盘 (tar)

以下示例说明如何将名为 `evaluation*` 的文件复制到软盘。

```
$ cd /home/smith

$ volcheck

$ ls evaluation*

evaluation.doc  evaluation.doc.backup

$ tar cvf /vol/dev/aliases/floppy0 evaluation*

a evaluation.doc 86 blocks

a evaluation.doc.backup 84 blocks

$ tar tvf /vol/dev/aliases/floppy0
```

▼ 如何列出软盘中的文件 (tar)

- 1 将软盘插入驱动器。

- 2 使软盘可用。

```
$ volcheck
```

- 3 列出软盘中的文件。

```
$ tar tvf /vol/dev/aliases/floppy0
```

示例 29-15 列出软盘中的文件 (tar)

以下示例说明如何列出软盘中的文件。

```
$ volcheck

$ tar tvf /vol/dev/aliases/floppy0

rw-rw-rw-6693/10  44032 Jun  9 15:45 evaluation.doc

rw-rw-rw-6693/10  43008 Jun  9 15:55 evaluation.doc.backup

$
```

▼ 如何从软盘中恢复文件(tar)

1 转到要放置文件的目录。

2 将软盘插入驱动器。

3 使软盘可用。

```
$ volcheck
```

4 从软盘中恢复文件。

```
$ tar xvf /vol/dev/aliases/floppy0
```

软盘中的所有文件都被复制到当前目录。

5 确认文件已被恢复。

```
$ ls -l
```

6 从驱动器中取出软盘。

示例 29-16 从软盘中检索文件(tar)

以下示例说明如何从软盘中恢复所有文件。

```
$ cd /home/smith/Evaluations

$ volcheck

$ tar xvf /vol/dev/aliases/floppy0

x evaluation.doc, 44032 bytes, 86 tape blocks

x evaluation.doc.backup, 43008 bytes, 84 tape blocks
```

```
$ ls -l
```

以下示例说明如何从软盘中恢复单个文件。将从软盘中提取该文件，并会将其放置在当前工作目录中。

```
$ volcheck
```

```
$ tar xvf /vol/dev/aliases/floppy0 evaluation.doc
```

```
x evaluation.doc, 44032 bytes, 86 tape blocks
```

```
$ ls -l
```

将文件归档到多张软盘

如果您正在将大型文件复制到软盘上，则需要系统提示您用另一张经过格式化的软盘来替换已满的软盘。`cpio` 命令将提供此功能。`cpio` 命令的用法与将文件复制到磁带时的用法大致相同，不同的是将 `/vol/dev/aliases/floppy0` 指定为设备，而不是磁带设备名称。

有关如何使用 `cpio` 命令的信息，请参见第 539 页中的“[如何将目录中的所有文件复制到磁带 \(cpio\)](#)”。

管理磁带机（任务）

本章介绍如何在 Solaris™ 操作系统 (Solaris OS) 中管理磁带机。

以下是本章中逐步说明的列表。

- 第 554 页中的 “如何显示磁带机状态”
- 第 555 页中的 “重新张紧盒式磁带”
- 第 556 页中的 “反绕盒式磁带”

以下是本章中概述信息的列表。

- 第 551 页中的 “选择要使用的介质”
- 第 552 页中的 “备份设备名称”
- 第 554 页中的 “显示磁带机状态”
- 第 556 页中的 “磁带机维护和介质处理指南”

选择要使用的介质

通常使用以下磁带介质来备份 Solaris 系统：

- 1/2 英寸盘式磁带
- 1/4 英寸流盒式磁带
- 8 毫米盒式磁带
- 4 毫米盒式磁带 (DAT)

可以使用软盘来执行备份，但这样做非常耗时并且很麻烦。

选择哪一种介质取决于支持该介质的设备以及用于存储文件的介质（通常为磁带）是否可用。尽管必须从本地系统进行备份，但可以将文件写入远程设备。

下表显示了用于备份文件系统的典型磁带设备。每个设备的存储容量取决于驱动器的类型和写入磁带中的数据。

表 30-1 介质存储容量

备份介质	存储容量
1/2 英寸盘式磁带	140 MB (6250 bpi)
2.5 GB 1/4 英寸盒式 (QIC) 磁带	2.5 GB
DDS3 4 毫米盒式磁带 (DAT)	12-24 GB
14 GB 8 毫米盒式磁带	14 GB
DLT 7000 1/2 英寸盒式磁带	35-70 GB

备份设备名称

通过提供逻辑设备名称，可以指定用于备份的磁带或软盘。此名称指向包含“原始”设备文件的子目录，并且名称中包括磁带机的逻辑单元号。磁带机命名约定使用逻辑设备名称而不是物理设备名称。下表显示了此命名约定。

表 30-2 备份设备的基本设备名称

设备类型	名称
磁带	<code>/dev/rmt/n</code>
软盘	<code>/vol/dev/rdiskette0/unlabeled</code>

通常，可以按照下图所示指定磁带设备。

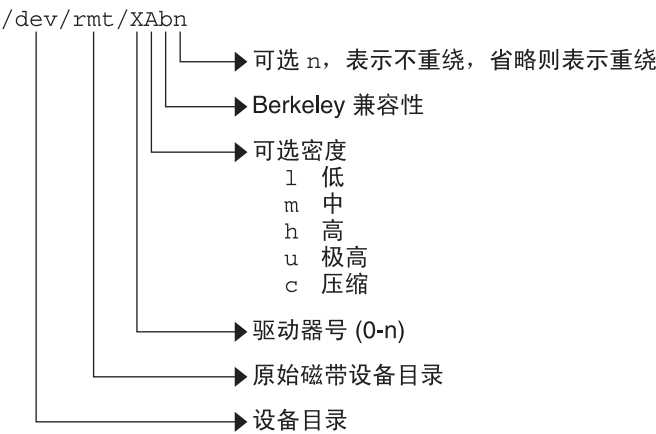


图 30-1 磁带机设备名称

如果不指定密度，则磁带机通常会按照其“首选”密度进行写入。首选密度通常表示

磁带机支持的最高密度。大多数 SCSI 磁带机可以自动检测磁带的密度或格式，并相应地进行读取。要确定磁带机支持的不同密度，请查看 `/dev/rmt` 子目录。该子目录中包含支持每种磁带的不同输出密度的磁带设备文件集。

另外，一个 SCSI 控制器最多可有七个 SCSI 磁带机。

指定磁带机的反绕选项

通常，可以按照逻辑单元号（范围从 0 到 n ）指定磁带机。下表介绍了如何使用反绕选项或不反绕选项来指定磁带设备名称。

表 30-3 指定磁带机反绕或不反绕

磁带机和反绕值	使用此选项
第一个磁带机，反绕	<code>/dev/rmt/0</code>
第一个磁带机，不反绕	<code>/dev/rmt/0n</code>
第二个磁带机，反绕	<code>/dev/rmt/1</code>
第二个磁带机，不反绕	<code>/dev/rmt/1n</code>

指定磁带机的不同密度

缺省情况下，磁带机按照“首选”密度写入，该密度通常为磁带机支持的最高密度。如果不指定磁带设备，则命令会按照设备支持的缺省密度写入磁带机编号 0。

要将磁带传输到其磁带机仅支持特定密度的系统，请指定按照所需密度写入的设备名称。下表介绍了如何指定磁带机的不同密度。

表 30-4 指定磁带机的不同密度

磁带机、密度和反绕值	使用此选项
第一个磁带机，低密度，反绕	<code>/dev/rmt/0l</code>
第一个磁带机，低密度，不反绕	<code>/dev/rmt/0ln</code>
第二个磁带机，中密度，反绕	<code>/dev/rmt/1m</code>
第二个磁带机，中密度，不反绕	<code>/dev/rmt/1mn</code>

其他密度值显示在[第 552 页中的“备份设备名称”](#)中。

显示磁带机状态

可以使用 `mt` 命令的 `status` 选项来获取有关磁带机的状态信息。`mt` 命令可报告有关 `/kernel/drv/st.conf` 文件中介绍的所有磁带机的信息。

▼ 如何显示磁带机状态

- 1 将磁带装入需要了解其相关信息的磁带机。
- 2 显示磁带机状态。
`# mt -f /dev/rmt/n status`
- 3 重复步骤 1–2，替换磁带机编号 0、1、2、3 等，以显示有关所有可用磁带机的信息。

示例 30-1 显示磁带机状态

以下示例显示了 QIC-150 磁带机 (`/dev/rmt/0`) 的状态：

```
$ mt -f /dev/rmt/0 status
```

Archive QIC-150 tape drive:

```
sense key(0x0)= No Additional Sense    residual= 0    retries= 0
```

```
file no= 0    block no= 0
```

以下示例显示了 Exabyte 磁带机 (`/dev/rmt/1`) 的状态：

```
$ mt -f /dev/rmt/1 status
```

Exabyte EXB-8200 8mm tape drive:

```
sense key(0x0)= NO Additional Sense residual= 0    retries= 0
```

```
file no= 0    block no= 0
```

以下示例说明了快速轮询系统并查找其所有磁带机的方法：

```
$ for drive in 0 1 2 3 4 5 6 7
```

```
> do
```

```
> mt -f /dev/rmt/$drive status
```

```
> done

Archive QIC-150 tape drive:

    sense key(0x0)= No Additional Sense    residual= 0    retries= 0

    file no= 0    block no= 0

/dev/rmt/1: No such file or directory

/dev/rmt/2: No such file or directory

/dev/rmt/3: No such file or directory

/dev/rmt/4: No such file or directory

/dev/rmt/5: No such file or directory

/dev/rmt/6: No such file or directory

/dev/rmt/7: No such file or directory

$
```

处理盒式磁带

如果读取磁带时出现错误，则可以重新张紧磁带、清洁磁带机，然后重试。

重新张紧盒式磁带

使用 `mt` 命令重新张紧盒式磁带。

例如：

```
$ mt -f /dev/rmt/1 retension

$
```

注 – 请勿重新张紧非 QIC 磁带机。

反绕盒式磁带

要反绕盒式磁带，请使用 `mt` 命令。

例如：

```
$ mt -f /dev/rmt/1 rewind  
  
$
```

磁带机维护和介质处理指南

不能读取的备份磁带是无用的。因此，请定期清洁和检查磁带机，以确保其正确操作。有关清洁磁带机的过程说明，请参见硬件手册。通过执行以下操作之一，可以检查磁带硬件：

- 将某些文件复制到磁带，读回这些文件，然后将原始文件与复制的文件进行比较。
- 使用 `ufsdump` 命令的 `-v` 选项借助源文件系统检验介质内容。必须取消挂载文件系统或使其完全处于空闲状态，`-v` 选项才能有效。

请注意，硬件可能会出现系统未报告的故障。

请始终在备份后对磁带进行标记。如果使用的备份策略与第 24 章中建议的策略类似，则应该在标签上标明“磁带 A”、“磁带 B”，依此类推。此标签绝不能更改。每次执行备份时，请创建另一个包含以下信息的磁带标签：

- 备份日期
- 备份的计算机和文件系统的名称
- 备份级别
- 磁带编号（1 至 n 中的任意一个数字，前提是备份跨多个卷）
- 特定于站点的任何信息

请将磁带存储在远离磁性设备的无尘安全位置。有些地方会将归档的磁带存储在偏僻位置的防火柜中。

应该创建一个日志并对其进行维护，该日志用于跟踪存储每个作业（备份）的介质（磁带卷）和每个备份文件的位置。

索引

数字和符号

/export/home 目录, 350
0 级别备份, 470
4.3 Tahoe 文件系统, 344
9660 CD 格式, 45

A

autofs, 358

B

BSD Fat Fast File 系统, 344

C

CacheFS 文件系统
 (概述), 391
 参数, 391
 查看 CacheFS 统计信息, 418
 查找 CacheFS 日志文件, 415
 创建(如何), 393
 创建压缩列表(如何), 407
 对 cachefspack 错误进行疑难解答, 410
 挂载(如何), 394
 删除(如何), 402
 设置 CacheFS 日志记录(如何), 415
 使用 cachefspack 命令进行压缩(概述), 404
 使用 cachefspack 命令进行压缩(如何), 405
 使用 fsck 命令进行检查(示例), 403

CacheFS 文件系统 (续)

 收集 CacheFS 统计信息(概述), 414
 停止 CacheFS 日志记录, 416
 显示已压缩的文件(如何), 405
 显示已压缩的文件(示例), 406
 显示有关信息(如何), 399

cachefspack 命令

 概述, 404
 如何使用, 405

CD

 ISO 9660 格式, 45
 UFS CD
 SPARC 与 x86 格式, 45
 名称, 44

cdwr 命令

 说明, 57
 限制访问(如何), 59
 写入数据 CD 和 DVD 及音频 CD(概述), 59

cfgadm

 PCI 热插拔(概述), 87
 SCSI 热插拔(概述), 87

cfsadmin 命令, 393, 402

clri 命令, 349

cpio 命令

 (概述), 533
 从磁带中提取所有文件(如何), 541
 列出磁带中的文件(如何), 541
 在文件系统之间复制目录(如何), 534

CTFS 文件系统, 347

D

datadm 命令, 191
dd 命令
 (概述), 530
 克隆磁盘(如何), 531
/dev/dsk 目录, 194
/dev/rdisk 目录, 194
devfsadm 命令, 193
df 命令, 195, 349
dfstab 文件, 为共享的本地可移除介质进行配置
 (如何), 50
DOS, 文件系统, 344
driver not attached 消息, 77
DVD-ROM, 345

E

EFI 标号
 (概述), 202
 安装系统, 204
 解决问题, 205
 限制, 203
 与 VTOC 标号比较, 202
eject 命令, 可移除介质(如何), 49
/etc/dfs/dfstab 文件, 为共享的可移除介质进行配
 置(如何), 50
/etc/dumpdates 文件, 522-523
/etc/rmmount.conf 文件, 共享可移除介质驱动器
 (如何), 50

F

FDFS 文件系统, 347
ff 命令, 349
FIFO inode, 435
FIFOFS 文件系统, 347
format.dat 文件
 创建项(概述), 238
 创建项(如何), 239
 关键字, 318, 321
 语法规则, 318
format 实用程序
 analyze 菜单, 314

format 实用程序 (续)

 defect 菜单, 316-317
 fdisk 菜单, 313
 partition 菜单, 312, 313
 (概述), 211
 标记磁盘
 示例, 231
 创建 Solaris fdisk 分区(如何), 267
 创建磁盘片和标记磁盘(如何)
 SPARC, 250
 x86, 276
 格式化磁盘(示例), 225
 功能和优势, 211
 何时使用, 211
 恢复损坏的磁盘标号(如何), 235
 确定磁盘是否已格式化(如何), 223
 确定系统中的磁盘(如何), 220
 确定系统中的磁盘(示例), 222
 使用帮助功能, 325
 使用指南, 212-213
 输入, 322, 324
 输入命令名称(如何), 323
 显示磁盘片信息(示例), 227
 指定块编号(如何), 323
 主菜单, 310
 自动配置 SCSI 磁盘驱动器(概述), 239
 自动配置 SCSI 磁盘驱动器(如何), 241
fsck 命令, 195, 349
 FSACTIVE 状态标志, 432
 FSBAD 状态标志, 432
 FSCLEAN 状态标志, 432
 FSSTABLE 状态标志, 432
 检查
 inode 列表大小, 434
 超级块, 434
 空闲 inode, 434
 空闲块, 434
 要修复的情况, 433
 以交互方式使用, 439
 语法和选项, 453
 整理, 444
 状态标志, 432
fsdb 命令, 349
fssnap 命令, 创建 UFS 快照(如何), 495

fstyp 命令, 349
 fstypes 文件, 358
 fuser 命令
 确定可移除介质是否在使用 (如何), 48
 中止访问可移除介质的进程 (如何), 48

G

grep 命令, 359
 GRUB
 Solaris 故障安全引导
 x86, 200
 使用 GRUB 管理磁盘
 x86, 200

H

High Sierra 文件系统, 344
 /home (已自动挂载), 358
 HSFS, 请参见 High Sierra 文件系统

I

I/O, 直接, 353
 InfiniBand 设备
 动态重新配置 (概述), 178
 概述, 175
 更新 IOC 配置 (如何), 189
 更新 IP p_key 表 (如何), 186
 配置 IB 端口、HCA_SVC 或 VPPA 设备 (如何), 182
 配置 IB 伪设备 (如何), 184
 配置 IOC 设备 (如何), 181
 取消配置 IB 端口、HCA_SVC 或 VPPA (如何), 182
 取消配置 IB 伪设备 (如何), 183
 取消配置 IOC 设备 (如何), 181
 取消配置与 HCA 连接的 IB 设备 (如何), 185
 删除现有的 IB 端口、HCA_SVC 或 VPPA 通信服务 (如何), 188
 添加 VPPA 通信服务 (如何), 187
 显示 HCA 的内核 IB 客户机 (如何), 184

InfiniBand 设备 (续)

 显示 IB 通信服务 (如何), 186
 显示 (如何), 178
 inode, 456-457
 directory, 435
 FIFO, 435
 常规, 435
 错误编号, 437
 大小, 436
 符号链接, 435
 检查格式和类型, 435
 块特殊, 435
 链接计数, 435
 字符特殊, 435
 字节数, 460
 inode 的格式, 435
 inode 的类型, 435
 inode 的链接计数, 435
 inode 列表大小, 434
 inode 状态, 435
 installboot 命令, 259
 installgrub 命令, 278
 iSCSI
 iSCSI 配置问题疑难解答 (如何), 301
 (概述), 281
 访问 iSCSI 磁盘 (如何), 290
 监视 iSCSI 配置 (如何), 291
 静态和动态目标搜索, 284
 配置 iSCSI 目标搜索 (如何), 288
 配置单向或双向 CHAP 验证 (如何), 286
 软件和硬件要求, 282
 删除搜索到的 iSCSI 目标 (如何), 289
 修改 iSCSI 启动器和目标参数 (如何), 295
 一般 iSCSI 错误消息, 304
 准备 Solaris iSCSI 配置 (如何), 285
 iscsiadm add 命令, 添加静态或动态目标 (示例), 288
 iscsiadm list, 显示 iSCSI 配置信息 (示例), 292
 iscsiadm modify 命令
 启用 CHAP (示例), 286
 启用或禁用静态或动态目标 (示例), 289
 iscsiadm remove 命令, 删除静态或动态目标 (示例), 289
 ISO 9660 文件系统, 344

ISO 标准, 9660 CD 格式, 45

K

/kernel/drv 目录, 76

L

labelit 命令, 349

lost+found 目录, 432

M

media was found 消息, 35

mkfile 命令, 426, 427

mkfs 命令, 349, 362

mkisofs 命令, 创建数据 CD 或 DVD 文件系统（如何）, 62

MNTFS 文件系统, 351

mnttab 文件, 355

mount 命令, 195

mountall 命令, 349

mt 命令, 555

N

NAMEFS 文件系统, 347

ncheck 命令, 349

newfs 命令, 195, 362

NFS

vfstab 项, 379

服务器说明, 358

说明, 357

nfsd 守护进程

启动, 50

验证是否正在运行, 50

no media was found 消息, 35

O

OBJFS 文件系统, 348

/opt 目录, 350

P

passwd 文件, 从磁带恢复（示例）, 510

PCFS 文件系统, 344

PCI 设备

对 PCI 配置问题进行疑难解答, 111

添加 PCI 适配卡（如何）, 108

显示 PCI 插槽配置信息（如何）, 104

移除 PCI 适配卡（如何）, 107

PCMCIA 内存卡

远程挂载（示例）, 56

在其他系统上访问（示例）, 56

/proc 目录, 347, 351

PROCFS 文件系统,（概述）, 347

prtvtoc 命令, 195

（示例）, 233

R

RCM 脚本

概述, 112

命令, 112

rmmount.conf 文件, 共享可移除介质驱动器（如何）, 50

Rock Ridge 扩展（HSFS 文件系统）, 344

S

SCSI 磁带机, 553

SCSI 磁盘驱动器, 239

SCSI 设备

对 SCSI 配置问题进行疑难解答, 101

更换 SCSI 控制器上的相同 SCSI 设备（如何）, 98

将 SCSI 设备添加到 SCSI 总线（如何）, 97

解析失败的 SCSI 取消配置操作（如何）, 103

连接 SCSI 控制器（如何）, 96

配置 SCSI 控制器（如何）, 93

SCSI 设备 (续)

- 配置 SCSI 设备 (如何), 94
- 取消配置 SCSI 控制器 (如何), 92
- 使用 `cfgadm` 命令断开连接 (如何), 95
- 显示有关信息 (如何), 92
- 移除 SCSI 设备 (如何), 100
- `share` 命令, 358
 - 使可移除介质可供其他系统使用 (如何), 50
- `shareall` 命令, 358
- Solaris `fdisk` 分区, 指南, 266-267
- Solaris 故障安全引导
 - x86, 200
 - (如何)
 - x86, 263
- SPECFS 文件系统, 348
- SunOS 缺省文件系统, 350
- `swap` 命令, 426
- `swapadd` 命令, 424
- SWAPFS 文件系统, 348
- `sysdef` 命令, 77

T**tar 命令**

- (概述), 535
- 从磁带中恢复文件 (如何), 537
- 从软盘中检索文件 (如何), 548
- 列出磁带中的文件 (如何), 537
- 列出软盘中的文件 (如何), 547
- 使用 `dd` 命令从远程磁带中检索文件 (如何), 545
- 使用 `dd` 命令将文件复制到远程磁带 (如何), 544
- `/tmp` 目录, 346, 350
- TMPFS 文件系统, 概述, 346

U**uDAPL**

- (概述), 189
- 更新 DAT 静态注册表 (如何), 191
- 启用 (如何), 190
- 在 DAT 静态注册表中注册 (如何), 191

uDAPL (续)

- 在 DAT 静态注册表中注销 (如何), 191
- UDF 文件系统, 345
- UFS CD, SPARC 格式与 x86 格式的对比, 45
- UFS 快照
 - 创建 (如何), 495
 - 创建完整备份 (如何), 499
 - 说明, 492
- UFS 日志记录, 概述, 352
- UFS 文件系统, 344, 351
 - 大文件系统, 351
 - 多 TB 文件系统, 351
 - 挂载, 379
 - 扩展的基本类型, 351
 - 日志, 351
 - 使用 `/etc/vfstab` 挂载, 380
 - 在不包含大文件的情况下挂载 (如何), 383
 - 状态标志, 351
- `ufsdump` 命令
 - 工作原理, 521-525
 - 介质结尾检测, 521-522
 - 如何复制数据, 522
 - 完整备份 (示例), 483, 487
 - 限制, 525
 - 选项和参数, 525
 - 增量备份 (示例), 485
- `ufsdump` 命令 (概述), 482
- `ufsrestore` 命令, 526
 - 确定要使用的磁带 (如何), 505
 - 准备使用 (概述), 504
- `umount` 命令, 349
- `umountall` 命令, 349
- UNIX 文件系统, 344
- USB 设备
 - 2.0 设备概述, 132
 - Solaris USB 体系结构 (Solaris USB Architecture, USB), 131
 - (概述), 127
 - 电缆, 135
 - 电源管理, 135
 - 复组合设备, 128
 - 滚轮鼠标支持, 134
 - 键盘和鼠标设备, 134
 - 可移除海量存储设备 (概述), 139

USB 设备（续）

- 连接 USB 设备（如何），172
- 名称，128
- 配置 USB 设备（如何），171
- 驱动程序，129
- 取消配置设备（如何），170
- 确定主音频设备（如何），165
- 热插拔（概述），142
- 软盘设备（概述），140
- 设备节点，131
- 设备类，129
- 使用不兼容的海量存储设备（概述），141
- 首字母缩略词，127
- 添加 USB 相机（如何），144
- 添加音频设备（如何），165
- 物理设备分层结构，127
- 显示 USB 设备信息（如何），147
- 显示总线信息（如何），169
- 以逻辑方式断开 USB 设备连接（如何），172
- 以逻辑方式断开 USB 设备子树连接（如何），172
- 音频
 - （概述），163
 - 更改主设备（如何），167
 - 设备拥有权，167
- 音频设备问题疑难解答，167
- 有关海量存储设备的疑难解答提示，161
- 在不运行 `vold` 的情况下格式化海量存储设备（如何），149
- 在不运行 `vold` 的情况下挂载海量存储设备（如何），160
- 在不运行 `vold` 的情况下添加海量存储设备（如何），143
- 在不运行 `vold` 的情况下卸载海量存储设备（如何），160
- 在不运行 `vold` 的情况下移除海量存储设备（如何），145
- 在不运行 `vold` 的情况下准备使用海量存储设备（如何），146
- 重置 USB 设备（如何），173
- 主机控制器和根集线器，134
- 总线供电设备，133
- 总线说明，127
- 组合设备，128

/usr 文件系统，350

V

- /var 目录，350
- `vfstab` 文件，358, 424
 - LOFS 的项，372
 - 挂载所有文件，379
 - 将交换添加到，424
 - 缺省，356
 - 添加项（如何），378
- `volcopy` 命令，349
- `volmgt start` 命令，46

安

- 安装引导块（如何），SPARC，259

备

备份

- 和恢复文件系统
 - 定义，464
 - 命令，464
- 类型，466
- 设备名称，552-553
- 完整和增量，定义，466
- 选择文件系统，465
- 原因，464
- 增量记录，522
- 准备（概述），480-481

备份计划

- 对于服务器，474-477
- 每日累积，每周累积备份，471
- 每日累积，每周增量备份，472
- 每日增量、每周累积备份，473
- 使用转储级别，470
- 示例，471, 477
- 指南，468, 469

不

不受支持的设备, 76

参

参数 (文件系统), 458-461

查

查看, CacheFS 统计信息, 418

查找

CacheFS 日志文件, 415

磁带设备名称, 504

文件系统的类型, 359

文件系统名称, 504

常

常规 inode, 435

超

超级块, 434, 446, 456

创

创建

UFS 快照

(示例), 495

UFS 快照 (如何), 495

UFS 快照信息的完整备份 (如何), 499

回送文件系统 (概述), 371

交换文件, 426

数据 CD 或 DVD 文件系统 (如何), 62

文件系统 (概述), 362

压缩列表 (如何), 407

磁

磁带

存储容量, 467, 551

大小, 467, 551

容量, 525

使用 tar 命令检索文件 (如何), 537

特征, 525

磁带机

反绕, 553

确定恢复类型, 504

维护, 556

最多的 SCSI, 553

磁带设备 (命名), 197

磁盘

创建磁盘片和标记磁盘 (如何)

SPARC, 250

创建磁盘片和标记磁盘 (示例)

SPARC, 252

格式化 (概述), 213

何时格式化 (概述), 222

恢复损坏的磁盘标号 (概述), 235

恢复损坏的磁盘标号 (如何), 235

连接辅助磁盘 (示例)

SPARC, 254

确定是否已格式化 (如何), 223

添加到 (概述)

SPARC, 248

x86, 262-279

修复有缺陷的扇区, 241, 244

自动配置 SCSI 驱动器, 239

磁盘标号

创建 (概述), 229

说明, 214

磁盘控制器, 196

磁盘片

定义, 207

对系统配置的要求, 210

确定要使用的片, 210

显示有关信息 (概述), 227-229

存

存储 (虚拟内存), 定义, 421

存储容量 (介质), 467, 551

错

错误的 . 和 .. 项, 437
错误的 inode 编号, 437
错误消息, iSCSI, 304

大

大文件选项, 375
大小
 inode, 436
 段, 459
 检查文件系统, 434

弹

弹出, 可移除介质 (如何), 49

动

动态重新配置, InfiniBand 设备, 178
动态重新配置 (概述), 87

断

断开连接
 SCSI 控制器 (如何), 95
 USB 设备, 以逻辑方式 (如何), 172
 USB 设备子树, 以逻辑方式 (如何), 172

段

段大小, 459

对

对文件系统的损坏, 432

访

访问
 iSCSI 磁盘 (如何), 290
 磁带设备, 197
 磁盘设备, 194
 可移除介质 (如何), 46

分

分区 (交换), 定义, 421

浮

浮动片 (free hog slice), 请参见提供片

符

符号链接, 435

辅

辅助磁盘
 连接到系统 (如何)
 SPARC, 250
 x86, 266
 说明, 210

复

复制
 使用 cpio 命令复制多组文件 (概述), 533
 使用 cpio 命令复制各个文件 (概述), 533
 使用 cpio 命令在文件系统之间复制目录 (概述), 533
 完整的文件系统 (dd), 530
 文件到软盘 (概述), 546

格

格式化,在不运行 `vold` 的情况下格式化 USB 海量存储设备 (如何), 149
 格式化磁盘, (概述), 213

根

根 (/) 文件系统, 350

更

更改,主 USB 音频设备 (如何), 167
 更换,SCSI 控制器上的相同 SCSI 设备 (如何), 98
 更新
 DAT 静态注册表 (如何), 191
 IOC 配置 (如何), 189
 IP `p_key` 表 (如何), 186

共

共享,文件, 357-358

挂**挂载**

NFS 文件系统, 379
 UFS 文件系统, 379
 UFS 文件系统 (如何)
 不包含大文件, 383
`vfstab` 文件中的所有文件, 379
 回送文件系统 (loopback file system, LOFS), 379
 可移除介质
 自动挂载与其比较, 30
 手动挂载远程可移除介质 (示例), 55
 文件系统, 使用 `/etc/vfstab`, 380
 在不运行 `vold` 的情况下挂载 USB 海量存储设备 (如何), 160
 在其他系统上挂载 PCMCIA 内存卡 (示例), 56
 自动挂载文件系统, 358
 挂载表, 355
 挂载点,定义, 353

管**管理**

磁盘 (使用 GRUB)
 x86, 200

归

归档,使用 `cpio` 命令将文件归档到多张软盘 (如何), 549

坏

坏的超级块, 446
 坏块编号, 436

恢

恢复坏的超级块, 446
 恢复文件
 非交互式恢复 (示例), 510
 交互式恢复 (示例), 508
 恢复文件系统
 磁带机类型, 504
 根目录和 `/usr` (SPARC) (示例), 518
 根目录或 `/usr` (x86) (示例), 519
 确定要使用的磁带 (如何), 505
 完整 (如何), 512
 完整 (示例), 514
 准备 (概述), 504

回

回送文件系统 (loopback file system, LOFS)
 创建 (概述), 371
 挂载, 379

基

基于 SPARC 的系统, UFS 格式, 45
 基于 x86 的系统, UFS 格式, 45

基于磁盘的文件系统, 344
基于网络的文件系统, 345

计

计划备份, 468

记

记录
 增量备份, 522
 转储, 522-523

监

监视, iSCSI 配置 (如何), 291

间

间接块, 436

检

检测介质结尾
 cpio 命令, 534
 ufsdump 命令, 521-522, 522
检查
 CacheFS 文件系统 (示例), 403
 inode 的格式和类型, 435
 inode 列表的一致性, 434
 和修复文件系统, 439
 空闲 inode, 434
 空闲块, 434
 文件系统大小, 434
检索, 使用 tar 命令从磁带中检索文件 (如何)
 , 537

交

交换分区, 定义, 421
交换文件
 创建, 426
 删除不用的, 428
 添加到 vfstab, 424
 显示, 425-426

解

解析, 失败的 SCSI 取消配置操作 (如何), 103

介

介质结尾检测
 cpio 命令, 534
 ufsdump 命令, 521-522, 522

进

进程文件系统 (Process File System, PROCFS), 347

卷

卷管理
 手动挂载与自动挂载的比较, 30
 停止 (如何), 46
 优点, 29
 重新启动 (如何), 46
 装入软盘 (如何), 35

可

可移除介质
 弹出 (如何), 49
 访问 (如何), 46
 访问 (示例), 46
挂载
 手动与自动的比较, 30
 挂载远程介质 (示例), 55

可移除介质（续）

- 名称, 44
- 确定介质是否在使用（如何）, 48
- 在其他系统上访问介质（示例）, 55
- 中止访问的进程（如何）, 48

空

- 空间优化类型, 460
- 空闲 inode, 434
- 空闲空间（最小）, 459-460
- 空闲块, 434, 457-458

块**块**

- 常规数据, 438
- 坏的, 436
- 间接, 436
- 空闲, 457-458
- 逻辑大小, 458-459
- 目录数据, 437
- 数据, 457
- 特殊 inode, 435
- 引导, 455
- 重复, 436
- 块磁盘设备接口
 - 定义, 195
 - 何时使用, 195

扩

- 扩展的基本类型（UFS 文件系统）, 351

连**连接**

- SCSI 控制器（如何）, 96
- USB 设备, 以逻辑方式（如何）, 172

临

- 临时文件系统 (temporary file system, TMPFS), 概述, 346

逻

- 逻辑块大小, 458-459
- 逻辑设备名称
 - 磁带, 197
 - 磁盘, 194
 - 定义, 194
 - 可移除介质, 197

每

- 每日独立备份, 471

面

- 面向总线的磁盘控制器, 196

目**目录**

- inode, 435
- /proc, 347
- /tmp, 346
- 使用 cpio 命令在文件系统之间复制（概述）, 533
- 未分配的块, 437

内

- 内存存储（虚拟）, 定义, 421

配**配置**

- IB 端口、HCA_SVC 或 VPPA 设备（如何）, 182

配置（续）

- IB 伪设备（如何），184
- IOC 设备（如何），181
- iSCSI 目标搜索（如何），288
- SCSI 控制器（如何），93
- SCSI 设备（如何），94
- USB 设备（如何），171
- 对 iSCSI 的单向或双向 CHAP 验证（如何），286

片

- 片（定义），207

启

- 启动
 - nfsd 守护进程，50
 - 卷管理（如何），46
- 启用,uDAPL，190

取

- 取消配置
 - IB 端口、HCA_SVC 或 VPPA 设备（如何），182
 - IB 伪设备（如何），183
 - IOC 设备（如何），181
 - SCSI 控制器（如何），92
 - USB 设备（如何），170
 - 与 HCA 连接的 IB 设备（如何），185

缺

- 缺省
 - SunOS 文件系统，350
 - /tmp 的文件系统 (TMPFS)，346

确

- 确定
 - 磁带机类型，504

确定（续）

- 磁带设备名称，504
- 挂载的文件系统，377
- 设备，77
- 文件系统类型，359
- 系统中的磁盘（如何），220
- 主 USB 音频设备（如何），165

热

热插拔

- PCI 设备（概述），104
 - （概述），87
- 更换 SCSI 控制器上的相同 SCSI 设备（如何），98
- 将 SCSI 设备添加到 SCSI 总线（如何），97
- 连接 SCSI 控制器（如何），96
- 配置 SCSI 控制器（如何），93
- 配置 SCSI 设备（如何），94
- 配置 USB 设备（如何），171
- 取消配置 SCSI 设备（如何），92
- 取消配置 USB 设备（如何），170
- 使用 cfgadm 命令断开 SCSI 控制器连接（如何），95
- 添加 PCI 适配卡（如何），108
- 移除 PCI 适配卡（如何），107
- 移除 SCSI 设备（如何），100
- 以逻辑方式断开 USB 设备连接（如何），172
- 以逻辑方式断开 USB 设备子树连接（如何），172
- 以逻辑方式连接 USB 设备（如何），172

日

- 日志（转储记录），522-523

软

软盘

- 使用 cpio 命令将文件归档到多张软盘（如何），549
- 通过卷管理装入（如何），35

删

删除

- CacheFS 文件系统（如何），402
- UFS 快照信息
（示例），498
- 不用的交换文件，428
- 搜索到的 iSCSI 目标（如何），289
- 现有的 IB 端口、HCA_SVC 或 VPPA 通信服务
（如何），188

设

设备, 访问, 193

设备名称

- 备份, 552-553
- 查找磁带, 504
- 查找文件系统名称, 504

设备驱动程序

- 定义, 75
- 添加, 86

设置, CacheFS 日志记录, 415

时

时间（优化类型），460

收

收集, CacheFS 统计信息（概述），414

手

手册页, 用于文件系统, 350

数

- 数据块, 438, 457
- 数据目录块, 437

添

添加

- PCI 适配卡（如何），108
- USB 相机（如何），144
- USB 音频设备（如何），165
- VPPA 通信服务（如何），187
- 磁盘（概述）
 - SPARC, 248
 - x86, 262-279
- 将 SCSI 设备添加到 SCSI 总线（如何），97
- 交换至 `vfstab`, 424
- 向 `/etc/vfstab` 文件添加项（如何），378
- 在不运行 `vold` 的情况下添加 USB 海量存储设备
（如何），143

停

停止

- CacheFS 日志记录, 416
- 卷管理（如何），46
- 文件系统的所有进程（如何），386
- 中止访问可移除介质的进程（如何），48

完

完整备份

- （示例），483, 487
- 定义, 466

维

维护磁带机, 556

伪

伪文件系统,（概述），346

未

未分配的 inode, 435

未分配的目录块, 437

文

文件

/etc/default/fs, 358

/etc/dfs/fstypes, 358

共享, 357-358

使用 cpio 命令归档到多张软盘 (如何), 549

使用 tar 命令从磁带中检索 (如何), 537

用于复制到介质的命令 (概述), 527

在 /proc 目录中, 347

文件系统

/, 350

4.3 Tahoe, 344

BSD Fat Fast, 344

CTFS, 347

DOS, 344

/export/home, 350

FDFS, 347

FIFOFS, 347

High Sierra, 344

ISO 9660, 344

MNTFS, 351

NAMEFS, 347

OBJFS, 348

/opt, 350

PCFS, 344

/proc, 351

PROCFS, (概述), 347

SPECFS, 348

SWAPFS, 348

TMPFS, 346

UFS, 344

UNIX, 344

/usr, 350

/var, 350

备份的原因, 464

查找类型, 359

创建 (概述)

 回送 (LOFS), 371

大, 375

导致不一致性问题的原因, 433

复制完整的 (dd), 530

文件系统 (续)

共享, 357-358

挂载表, 355

管理命令的说明, 349

基于磁盘, 344

基于网络, 345

检查大小, 434

检查和修复, 439

进程, (概述), 347

类型, 344

缺省 SunOS, 350

使可用 (概述), 373-377

手册页, 350

损坏, 432

停止进行访问的所有进程 (如何), 386

伪, (概述), 346

修复, 445

要备份的文件系统, 465

已高速缓存 (概述), 391

整理, 444, 445

柱面组结构, 455-458

自定义参数, 458-461

文件系统表, virtual, 356

文件系统的非一致性问题, 433

文件系统的类型, 344

文件系统的自定义参数, 458-461

文件系统名称, 504

文件系统损坏的原因, 432

物

物理设备名称

定义, 194

系

系统磁盘

连接 (如何)

 x86, 262

说明, 210

显**显示**

HCA 的内核 IB 客户机（如何），184
 IB 通信服务（如何），186
 InfiniBand 设备信息（如何），178
 PCI 插槽配置信息（如何），104
 USB 设备信息（如何），147
 USB 总线信息（如何），169
 磁盘片信息（概述），227
 交换空间，425-426
 可移除介质的用户（如何），48
 系统配置信息，77, 82
 已压缩的文件（如何），405
 已压缩的文件（示例），406
 有关 SCSI 设备的信息，92

限

限制, 访问可移除介质（如何），59

写

写入, 数据 CD 和 DVD 及音频 CD（概述），59

卸

卸载, 在不运行 `vol`d 的情况下卸载 USB 海量存储设备（如何），160

修

修复不一致的文件系统，445
 修改, iSCSI 启动器和目标参数（如何），295

虚

虚拟内存存储, 定义，421
 虚拟文件系统表，356

选

选项, 适用于 `ufsdump` 命令，525

验

验证, `nfsd` 守护进程正在运行，50

疑**疑难解答**

`cachefspack` 错误，410
 EFI 磁盘标号，205
 iSCSI 配置问题（如何），301
 PCI 配置问题，111
 SCSI 配置问题，101
 USB 海量存储设备，161
 USB 音频设备问题，167
 失败的 SCSI 取消配置操作，103

移**移除**

PCI 适配卡（如何），107
 SCSI 设备（如何），100
 在不运行 `vol`d 的情况下移除 USB 海量存储设备（如何），145

已

已分配的 `inode`，435

引

引导块，455

语

语法, `fsck` 命令，453

原

原始磁盘设备接口, 195

增

增量备份, 466, 522
 (示例), 485

整

整理文件系统, 444, 445

直

直接 I/O, 353

指

指定磁盘片, 196

中

中止
 访问可移除介质的进程 (如何), 48
 访问文件系统的所有进程 (如何), 386

重

重复块, 436
重新配置引导, 240
 SPARC 示例, 249
 x86 示例, 265
重置, USB 设备 (如何), 173

柱

柱面组, 455-458
柱面组的结构, 455-458

注

注册, DAT 静态注册表中的 (如何), 191
注销, DAT 静态注册表中的 (如何), 191

转

转储级别
 定义, 470
 每日增量备份, 471

装

装入, 具有卷管理的软盘 (如何), 35

状

状态标志
 fsck, 432
 UFS 文件系统, 351

准

准备
 Solaris iSCSI 配置 (如何), 285
 恢复文件 (概述), 504
 在不运行 `vold` 的情况下使用 USB 海量存储设备
 (如何), 146
 准备备份 (概述), 480-481

字

字符特殊 inode, 435
字节 (每 inode 的数目), 460

自

自动挂载, 与 `/home`, 358
自动配置过程, 76

最

最小空闲空间, 459-460

