

**HP-UX Reference**  
**Release 11i**  
**System Administration Commands**  
**Section 1M**  
**Part 1 of 2 (A-M)**  
**Volume 3 of 9**  
**Edition 1**

Customer Order Number: B2355-90688



**Manufacturing Part Number: B2355-90691**  
**E1200**

Printed in: United States  
© Copyright 1983-2000 Hewlett-Packard Company. All rights reserved.

---

## Legal Notices

The information in this document is subject to change without notice.

*Hewlett-Packard makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.* Hewlett-Packard shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

### Warranty

A copy of the specific warranty terms applicable to your Hewlett-Packard product and replacement parts can be obtained from your local Sales and Service Office.

### Restricted Rights Legend

Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 for DOD agencies, and subparagraphs (c) (1) and (c) (2) of the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 for other agencies.

HEWLETT-PACKARD COMPANY  
3000 Hanover Street  
Palo Alto, California 94304 U.S.A.

Use of this document and any supporting software media (CD-ROMs, flexible disks, and tape cartridges) supplied for this pack is restricted to this product only. Additional copies of the programs may be made for security and backup purposes only. Resale of the programs in their present form or with alterations is expressly prohibited.

### Copyright Notices

Copyright © 1983-2000 Hewlett-Packard Company. All rights reserved.

Reproduction, adaptation, or translation of this document without prior written permission is prohibited, except as allowed under the copyright laws.

Copyright © 1979, 1980, 1983, 1985-93 Regents of the University of California. This software is based in part on the Fourth Berkeley Software Distribution under license from the Regents of the University of California.

Copyright © 1988 Carnegie Mellon University.

Copyright © 1990-1995 Cornell University.

Copyright © 1986 Digital Equipment Corporation.

Copyright © 1997 Isogon Corporation.

Copyright © 1985, 1986, 1988 Massachusetts Institute of Technology.

Copyright © 1991-1997 Mentat, Inc.

Copyright © 1996 Morning Star Technologies, Inc.

Copyright © 1990 Motorola, Inc.

Copyright © 1980, 1984, 1986 Novell, Inc.

Copyright © 1989-1993 The Open Software Foundation, Inc.

Copyright © 1996 Progressive Systems, Inc.

Copyright © 1989-1991 The University of Maryland

Copyright © 1986-1992 Sun Microsystems, Inc.

#### Trademark Notices

Apple® and Macintosh® are trademarks of Apple Computer, Inc., registered in the United States and other countries.

AppleShare® is a registered trademark of Apple Computer, Inc.

CHAMELEON™ is a trademark of NetManage, Inc.

DIGITAL™ and PATHWORKS™ are trademarks of Digital Equipment Corporation.

DiskAccess® is a registered trademark of Intergraph.

EXCURSION™ is a trademark of Digital Equipment Corporation.

Exeed® is a registered trademark of Hummingbird Communications Ltd.

eXodus™ is a trademark of White Pine Software, Inc.

MS-DOS® and Microsoft® are U.S. registered trademarks of Microsoft Corporation.

NTRIGUE™ is a trademark of Insignia Solutions, Inc.

OSF/Motif™ is a trademark of the Open Software Foundation, Inc. in the U.S. and other countries.

PC\_Xware™ is a trademark, and WinCenter® is a registered trademark of Network Computing Devices, Inc.

REFLECTION® and WRQ® are registered trademarks of WRQ, Inc.

UNIX® is a registered trademark in the United States and other countries, licensed exclusively through The Open Group.

VERITAS® is a registered trademark of VERITAS Software Corporation.

VERITAS File System™ is a trademark of VERITAS Software Corporation.

WinDD™ is a trademark of Tektronix, Inc.

X Window System™ is a trademark of the Massachusetts Institute of Technology.

---

## **Publication History**

The manual publication date and part number indicate its current edition. The printing date will change when a new edition is printed. Minor changes may be made at reprint without changing the printing date. The manual part number will change when extensive changes are made.

Manual updates may be issued between editions to correct errors or document product changes. To ensure that you receive the updated or new editions, you should subscribe to the appropriate product support service. See your HP sales representative for details.

First Edition: December 2000 (HP-UX Release 11i)



# **Volume Three Table of Contents**

## **Section 1M**

# **Volume Three**

## **Table of Contents**

### **Section 1M**



## Table of Contents Volumes Three and Four

### Section 1M: System Administration Commands

Entry Name(Section):	name	Description
<b>intro(1M):</b>	intro	introduction to system maintenance commands and application programs
<b>accept(1M):</b>	accept, reject	allow or prevent LP printer queuing requests
<b>acct(1M):</b>	acctdisk, acctdusg, accton, acctwtmpt	overview of accounting and miscellaneous accounting commands
<b>acctcms(1M):</b>	acctcms	command summary from per-process accounting records
<b>acctcom(1M):</b>	acctcom	search and print process accounting files
<b>acctcon(1M):</b>	acctcon1, acctcon2	connect-time accounting
<b>acctcon1:</b>	connect-time accounting	see <b>acctcon(1M)</b>
<b>acctcon2:</b>	connect-time accounting	see <b>acctcon(1M)</b>
<b>acctdisk:</b>	miscellaneous accounting command	see <b>acct(1M)</b>
<b>acctdusg:</b>	miscellaneous accounting command	see <b>acct(1M)</b>
<b>acctmerg(1M):</b>	acctmerg	merge or add total accounting files
<b>accton:</b>	miscellaneous accounting command	see <b>acct(1M)</b>
<b>acctprc(1M):</b>	acctprc1, acctprc2	process accounting
<b>acctprc1:</b>	convert process accounting	see <b>acctprc(1M)</b>
<b>acctprc2:</b>	summarize process accounting	see <b>acctprc(1M)</b>
<b>acctsh(1M):</b>	chargefee, ckpacct, dodisk, lastlogin, monacct, nulladm, prctmp, prdaily, prtacct, shutacct, startup, turnacct	shell procedures for accounting
<b>acctwtmpt:</b>	miscellaneous accounting command	see <b>acct(1M)</b>
<b>AM60Srvr(1M):</b>	AM60Srvr	disk array server daemon
<b>amcfg(1M):</b>	amcfg	manage LUN configuration on disk array
<b>amdload(1M):</b>	amdload	download new firmware to disks and controllers
<b>amdsp(1M):</b>	amdsp	display status and configuration of disk array
<b>amfmt(1M):</b>	amfmt	disk format
<b>amlog(1M):</b>	amlog	display controller entries for disk array
<b>ammgr(1M):</b>	ammgr	manage operating characteristics of disk array
<b>amutil(1M):</b>	amutil	disk array management functions
<b>ARMServer(1M):</b>	ARMServer	disk array server daemon
<b>arp(1M):</b>	arp	address resolution display and control
<b>arraycfg(1M):</b>	arraycfg	configure LUNs and physical disks
<b>arraydsp(1M):</b>	arraydsp	display the status and operating configuration of the disk array
<b>arrayfmt(1M):</b>	arrayfmt	formats a LUN or an entire disk array
<b>arrayinfo(1M):</b>	arrayinfo	describe disk array characteristics
<b>arraylog(1M):</b>	arraylog	accesses and clears disk array logs
<b>arraymgr(1M):</b>	arraymgr	manages the operating environment of the disk array
<b>arrayrbl(1M):</b>	arrayrbl	controls rebuilding of disk array
<b>arrayrecover(1M):</b>	arrayrecover	reconstructs data mapping and array configuration
<b>arrayscan(1M):</b>	arrayscan	search system for disk arrays
<b>asecure(1M):</b>	asecure	control access to HP-UX Audio
<b>aserver(1M):</b>	aserver	Audio
<b>audevent(1M):</b>	audevent	change or display event or system call audit status
<b>audisp(1M):</b>	audisp	display audit information as requested by parameters
<b>audomon(1M):</b>	audomon	audit overflow monitor daemon
<b>audsys(1M):</b>	audsys	start or halt the auditing system and set or display audit file information
<b>audusr(1M):</b>	audusr	select users to audit
<b>authck(1M):</b>	authck	check internal consistency of Authentication database
<b>automount(1M):</b>	automount	automatically mount NFS file systems
<b>automountd(1M):</b>	automountd	autofs mount/unmount daemon
<b>autopush(1M):</b>	autopush	manage system database of automatically pushed STREAMS modules
<b>auto_parms(1M):</b>	auto_parms	Initial system configuration/DHCP support script
<b>backup(1M):</b>	backup	backup or archive file system
<b>bdf(1M):</b>	bdf	report number of free disk blocks (Berkeley version)
<b>biod:</b>	NFS block I/O daemons	see <b>nfstd(1M)</b>
<b>boot(1M):</b>	boot	bootstrap process
<b>bootpd(1M):</b>	bootpd	Internet Boot Protocol server

# Table of Contents

## Volumes Three and Four

Entry Name(Section): name	Description
<b>bootpquery(1M):</b> bootpquery .....	send BOOTREQUEST to BOOTP server
<b>cacheefsstat(1M):</b> cacheefsstat .....	cache file system statistics
<b>captoinfo(1M):</b> captoinfo .....	convert a termcap description into a terminfo description
<b>catman(1M):</b> catman .....	create the cat files for the manual
<b>cfl(1M):</b> cfl .....	configure a SCSI disk array LUN
<b>cfsadmin(1M):</b> cfsadmin .....	administer disk space used for caching file systems with CacheFS statistics
<b>chargefee:</b> shell procedures for accounting, charge fee to user .....	see <b>acctsh(1M)</b>
<b>chroot(1M):</b> chroot .....	change root directory for a command
<b>ch_rc(1M):</b> ch_rc .....	change system configuration file
<b>ckpacct:</b> shell procedures for accounting, check size of accounting file .....	see <b>acctsh(1M)</b>
<b>clri(1M):</b> clri .....	clear inode
<b>clrsvc(1M):</b> clrsvc .....	clear x25 switched virtual circuit
<b>config(1M):</b> config .....	configure and build an HP-UX system
<b>convertfs(1M):</b> convertfs .....	convert a file system to allow long file names
<b>convert_awk(1M):</b> convert_awk .....	converts old <b>sendmail.cf</b> files to new format
<b>cpset(1M):</b> cpset .....	install object files in binary directories
<b>crashconf(1M):</b> crashconf .....	configure system crash dumps
<b>crashutil(1M):</b> crashutil .....	manipulate crash dump data
<b>create_sysfile(1M):</b> create_sysfile .....	create a kernel system file
<b>cron(1M):</b> cron .....	timed-job execution daemon
<b>cuegetty(1M):</b> cuegetty .....	set terminal characteristics for cue
<b>dcc(1M):</b> dcc .....	controlling caching on HP SCSI disk arrays
<b>dcopy(1M):</b> dcopy .....	copy HFS file system with compaction
<b>devnm(1M):</b> devnm .....	device name
<b>df(1M):</b> df .....	report number of free file system disk blocks
<b>df_hfs(1M):</b> df .....	report number of free CDFS, HFS, or NFS file system disk blocks
<b>df_vxfs(1M):</b> df .....	report number of free disk blocks on VxFS file system
<b>dhcpcclient(1M):</b> dhcpcclient .....	Client for Dynamic Host Configuration Protocol Server
<b>dhcpdb2conf(1M):</b> dhcpdb2conf .....	DHCP client database converter
<b>dhcptools(1M):</b> dhcptools .....	command line tools for DHCP elements of bootpd
<b>diskinfo(1M):</b> diskinfo .....	describe characteristics of a disk device
<b>disksecn(1M):</b> disksecn .....	calculate default disk section sizes
<b>diskusg(1M):</b> diskusg .....	generate disk accounting data by user ID
<b>dif(1M):</b> dif .....	download firmware to an HP SCSI disk array
<b>dmesg(1M):</b> dmesg .....	collect system diagnostic messages to form error log
<b>dodisk:</b> shell procedures for accounting, perform disk accounting .....	see <b>acctsh(1M)</b>
<b>download(1M):</b> download .....	download new controller or disk firmware
<b>dpp(1M):</b> dpp .....	dedicated ports parser used by DDFA software
<b>drivetest(1M):</b> drivetest .....	test a disk in the array
<b>dsp(1M):</b> dsp .....	display status of an HP SCSI disk array
<b>dteststat(1M):</b> dteststat .....	display the results of disk testing, or cancel testing
<b>dump(1M):</b> dump, rdump .....	incremental file system dump
<b>dumpfs(1M):</b> dumpfs .....	dump file system information
<b>edquota(1M):</b> edquota .....	edit user quotas
<b>eisa_config(1M):</b> eisa_config .....	EISA configuration tool
<b>envd(1M):</b> envd .....	system physical environment daemon
<b>exportfs(1M):</b> exportfs .....	export and unexport directories to NFS clients
<b>extendfs(1M):</b> extendfs .....	extend file system size
<b>extendfs_hfs(1M):</b> extendfs .....	extend HFS file system size
<b>extendfs_vxfs(1M):</b> extendfs .....	extend VxFS file system size
<b>fbackup(1M):</b> fbackup .....	selectively back up files
<b>fcmsutil(1M):</b> fcmsutil .....	fibre channel diagnostic utility
<b>fdetach(1M):</b> fdetach .....	detach a STREAMS-based file descriptor from a filename
<b>ff(1M):</b> ff .....	list file names and statistics for file system
<b>ff_hfs(1M):</b> ff .....	list file names and statistics for HFS file system
<b>ff_vxfs(1M):</b> ff .....	fast find: list file names and statistics for VxFS file system
<b>fingerd(1M):</b> fingerd .....	remote user information server
<b>fixman(1M):</b> fixman .....	fix manual pages for faster viewing with <b>man(1)</b>
<b>format(1M):</b> format .....	format an HP SCSI disk array LUN
<b>frecover(1M):</b> frecover .....	selectively recover files
<b>freedisk(1M):</b> freedisk .....	recover disk space

## Table of Contents Volumes Three and Four

Entry Name(Section): name	Description
<b>frupower(1M):</b> frupower .....	turn on/off power or display status for cells and I/O chassis
<b>fsadm(1M):</b> fsadm .....	file system administration command
<b>fsadm_hfs(1M):</b> fsadm_hfs .....	HFS file system administration command
<b>fsadm_vxfs(1M):</b> fsadm .....	resize or reorganize a VxFS file system
<b>fscat_vxfs(1M):</b> fscat .....	cat a VxFS file system
<b>fsck(1M):</b> fsck .....	file system consistency check and interactive repair
<b>fsck_cacheufs(1M):</b> fsck_cacheufs .....	check integrity of data cached with CacheFS
<b>fsck_vxfs(1M):</b> fsck .....	check and repair VxFS file systems
<b>fsclean(1M):</b> fsclean .....	determine shutdown status of specified file system
<b>fsdb(1M):</b> fsdb .....	file system debugger (generic)
<b>fsdb_hfs(1M):</b> fsdb .....	HFS file system debugger
<b>fsdb_vxfs(1M):</b> fsdb .....	VxFS file system debugger
<b>fsirand(1M):</b> fsirand .....	install random inode generation numbers
<b>fstyp(1M):</b> fstyp .....	determine file system type
<b>ftpd(1M):</b> ftpd .....	file transfer protocol server
<b>fuser(1M):</b> fuser .....	list processes using a file or file structure
<b>fwtmp(1M):</b> fwtmp, wtmpfix .....	manipulate connect accounting records
<b>gated(1M):</b> gated .....	gateway routing daemon
<b>gdc(1M):</b> gdc .....	operations user interface for gated
<b>geocustoms(1M):</b> geocustoms .....	configure system language on multi-language systems
<b>gettext(1M):</b> gettext .....	get extent attributes (VxFS)
<b>getmemwindow(1M):</b> getmemwindow .....	extract window ids from processes
<b>getprpw(1M):</b> getprpw .....	display protected password database
<b>getty(1M):</b> getty .....	set terminal type, modes, speed, and line discipline
<b>getx25(1M):</b> getx25 .....	get x25 line
<b>groupadd(1M):</b> groupadd .....	add a new group to the system
<b>groupdel(1M):</b> groupdel .....	delete a group from the system
<b>groupmod(1M):</b> groupmod .....	modify a group on the system
<b>grpck:</b> group file checker .....	see <b>pwck(1M)</b>
<b>hosts_to_named(1M):</b> hosts_to_named .....	translate host table to name server file format
<b>hpux(1M):</b> hpux .....	HP-UX bootstrap and installation utility
<b>i4admin(1M):</b> i4admin .....	administer LicensePower/iFOR licensing
<b>i4lmd(1M):</b> i4lmd .....	start license server
<b>i4start(1M):</b> i4start .....	LicensePower/iFOR server start tool
<b>i4stop(1M):</b> i4stop .....	LicensePower/iFOR server stop tool
<b>i4target(1M):</b> i4target .....	print information about local LicensePower/iFOR target id
<b>i4tv(1M):</b> i4tv .....	verify Network License Servers are working
<b>identd(1M):</b> identd .....	TCP/IP IDENT protocol server
<b>ifconfig(1M):</b> ifconfig .....	configure network interface parameters
<b>inetd(1M):</b> inetd .....	Internet services daemon
<b>inetsvcs_sec(1M):</b> inetsvcs_sec .....	enable or disable secure internet services
<b>infocmp(1M):</b> infocmp .....	compare or print out terminfo descriptions
<b>init(1M):</b> init .....	process control initialization
<b>insf(1M):</b> insf .....	install special (device) files
<b>install(1M):</b> install .....	install commands
<b>install-sd(1M):</b> install-sd .....	install Software Distributor (SD)
<b>ioinit(1M):</b> ioinit .....	initialize I/O system
<b>ioscan(1M):</b> ioscan .....	scan the I/O system
<b>isl(1M):</b> isl .....	initial system loader
<b>itemap(1M):</b> itemap .....	load a keymap into the Internal Terminal Emulator (ITE)
<b>keyenvoy(1M):</b> keyenvoy .....	talk to the keyserv process
<b>keyserv(1):</b> keyserv .....	server for storing private encryption keys
<b>killall(1M):</b> killall .....	kill all active processes
<b>killsm(1M):</b> killsm .....	kills the sendmail daemon
<b>kl(1M):</b> kl .....	control kernel logging
<b>kmadmin(1M):</b> kmadmin .....	kernel module administration
<b>kminstall(1M):</b> kminstall .....	add, delete, update a kernel module
<b>kmmodreg(1M):</b> kmmodreg .....	register or unregister loadable kernel modules with the running kernel
<b>kmsystem(1M):</b> kmsystem .....	set, query configuration and loadable flags for a module
<b>kmtune(1M):</b> kmtune .....	query, set, or reset system parameter
<b>kmupdate(1M):</b> kmupdate .....	update default kernel files or specified kernel modules

# Table of Contents

## Volumes Three and Four

Entry Name(Section): name	Description
<b>krstd(1M)</b> .....	kernel registry services daemon
<b>krs_flush(1M)</b> .....	flushes kernel registry services data to disk
<b>labelit:</b> label for VxFS file system .....	see <b>volcopy_vxfs(1M)</b>
<b>labelit - copy</b> file systems with label checking .....	see <b>volcopy(1M)</b>
<b>labelit - copy</b> file systems with label checking .....	see <b>volcopy_hfs(1M)</b>
<b>lanadmin(1M):</b> lanadmin .....	local area network administration
<b>lanscan(1M):</b> lanscan .....	display LAN device configuration and status
<b>lastlogin:</b> shell procedures for accounting, show last login date .....	see <b>acctsh(1M)</b>
<b>libcadmin(1M):</b> libcadmin .....	libc administration command
<b>link(1M):</b> link, unlink .....	execute <b>link()</b> and <b>unlink()</b> system calls without error checking
<b>linkloop(1M):</b> linkloop .....	verify LAN connectivity with link-level loopback
<b>localedef(1M):</b> localedef .....	generate a locale environment file
<b>lockd(1M):</b> lockd .....	network lock daemon
<b>logins(1M):</b> logins .....	display system and user login data
<b>logprint(1M):</b> logprint .....	formats and prints contents of ARMServer log files
<b>lpadmin(1M):</b> lpadmin .....	configure the LP spooling system
<b>lpana(1M):</b> lpana .....	print LP spooler performance analysis information
<b>lpfence:</b> set LP scheduler priority fence .....	see <b>lpsched(1M)</b>
<b>lpmove:</b> move LP scheduler requests .....	see <b>lpsched(1M)</b>
<b>lpsched(1M):</b> lpshut, lpfence, lpmove, lpsched .....	start/stop the LP request scheduler and move requests
<b>lpshut:</b> stop LP scheduler requests .....	see <b>lpsched(1M)</b>
<b>lsdev(1M):</b> lsdev .....	list device drivers in the system
<b>lssf(1M):</b> lssf .....	list a special file
<b>lvchange(1M):</b> lvchange .....	change LVM logical volume characteristics
<b>lvcreate(1M):</b> lvcreate .....	create logical volume in LVM volume group
<b>lvdisplay(1M):</b> lvdisplay .....	display information about LVM logical volumes
<b>lvextend(1M):</b> lvextend .....	stripe, increase space, increase mirrors for LVM logical volume
<b>lvlnboot(1M):</b> lvlnboot .....	prepare LVM logical volume to be root, swap, or dump volume
<b>lvmerge(1M):</b> lvmerge .....	merge two LVM logical volumes into one logical volume
<b>lvmmigrate(1M):</b> lvmmigrate .....	prepare root file system for migration from partitions to LVM logical volumes
<b>lvreduce(1M):</b> lvreduce .....	decrease physical extents allocated to LVM logical volume
<b>lvremove(1M):</b> lvremove .....	remove logical volumes from LVM volume group
<b>lvrmboot(1M):</b> lvrmboot .....	remove LVM logical volume link to root, primary swap, or dump volume
<b>lvsplit(1M):</b> lvsplit .....	split mirrored LVM logical volume into two logical volumes
<b>lvsync(1M):</b> lvsync .....	synchronize stale mirrors in LVM logical volumes
<b>makedbm(1M):</b> makedbm .....	make a Network Information System database
<b>makemap(1M):</b> makemap .....	creates database maps for sendmail
<b>map-mbone(1M):</b> map-mbone, .....	multicast router connection mapper
<b>mc(1M):</b> mc .....	media changer manipulation utility
<b>mkboot(1M):</b> mkboot, rmboot .....	install, update, or remove boot programs from a disk device
<b>mkfs(1M):</b> mkfs .....	construct a file system (generic)
<b>mkfs_hfs(1M):</b> mkfs .....	construct an HFS file system
<b>mkfs_vxfs(1M):</b> mkfs .....	construct VxFS file system
<b>mklost+found(1M):</b> mklost+found .....	make a lost+found directory for <b>fsck(1M)</b>
<b>mknod(1M):</b> mknod .....	create special and FIFO files
<b>mkpdf(1M):</b> mkpdf .....	create Product Description File from an input
<b>mkssf(1M):</b> mkssf .....	make a special (device) file
<b>mk_kernel(1M):</b> mk_kernel .....	build a bootable HP-UX kernel or kernel module
<b>modprpw(1M):</b> modprpw .....	modify protected password database
<b>monacct:</b> shell procedures for accounting, create accounting summary .....	see <b>acctsh(1M)</b>
<b>mount(1M):</b> mount, umount .....	mount and unmount a file system
<b>mountall(1M):</b> mountall, umountall .....	mount and unmount multiple file systems
<b>mountd(1M):</b> mountd .....	NFS mount request server
<b>mount_cacheofs(1M):</b> mount_cacheofs .....	mount CacheFS file systems
<b>mount_cdfs(1M):</b> mount, umount .....	mount and unmount CDFS file systems
<b>mount_hfs(1M):</b> mount, umount .....	mount and unmount HFS file systems
<b>mount_lofs(1M):</b> mount .....	mount an LOFS file system
<b>mount_nfs(1M):</b> mount, umount .....	mount and unmount NFS file systems
<b>mount_vxfs(1M):</b> mount, umount .....	mount and unmount VxFS file system
<b>mrinfo(1M):</b> mrinfo, .....	multicast routing configuration information tool
<b>mrouted(1M):</b> mrouted .....	IP multicast routing daemon

## Table of Contents

### Volumes Three and Four

Entry Name(Section): name	Description
<b>mtail(1M):</b> mtail .....	displays the last part of the mail log
<b>mvdir(1M):</b> mvdir .....	move a directory
<b>naaagt(1M):</b> naaagt .....	Native Agent Adapter for SNMP
<b>named(1M):</b> named .....	Internet domain name server
<b>named-xfer(1M):</b> named-xfer .....	ancillary agent for inbound zone transfers
<b>ncheck(1M):</b> ncheck .....	generate path names from inode numbers
<b>ncheck_vxfs(1M):</b> ncheck .....	generate pathnames from inode numbers for VxFS file system
<b>ndd(1M):</b> ndd .....	network tuning
<b>netfmt(1M):</b> netfmt .....	format tracing and logging binary files
<b>nettl(1M):</b> nettl .....	control network tracing and logging
<b>nettlconf(1M):</b> nettlconf .....	configure tracing and logging commands
<b>newaliases(1M):</b> newaliases .....	rebuilds the database for the mail aliases file
<b>newarray(1M):</b> newarray .....	make a special file
<b>newfs(1M):</b> newfs .....	construct a new file system
<b>newfs_hfs(1M):</b> newfs .....	construct a new HFS file system
<b>newfs_vxfs(1M):</b> newfs_vxfs .....	construct new VxFS file system
<b>newkey(1M):</b> newkey .....	create a new key in publickey database file
<b>nfsd(1M):</b> biod, nfsd .....	NFS daemons
<b>nfsstat(1M):</b> nfsstat .....	Network File System statistics
<b>nisaddcred(1M):</b> nisaddcred .....	create NIS+ credentials
<b>nisaddent(1M):</b> nisaddent .....	create NIS+ tables from corresponding /etc files or NIS maps
<b>nisclient(1M):</b> nisclient .....	initialize NIS+ credentials for NIS+ principals
<b>nisd:</b> NIS+ service daemon .....	see <b>rpc.nisd(1M)</b>
<b>nisd_resolv:</b> NIS+ service daemon .....	see <b>rpc.nisd(1M)</b>
<b>nisinit(1M):</b> nisinit .....	NIS+ client and server initialization utility
<b>nislog(1M):</b> nislog .....	display the contents of the NIS+ transaction log
<b>nispasswdd():</b> NIS+ password update daemon .....	see <b>rpc.nispasswdd(1M)</b>
<b>nisping(1M):</b> nisping .....	send ping to NIS+ servers
<b>nispopulate(1M):</b> nispopulate .....	populate the NIS+ tables in a NIS+ domain
<b>nisserver(1M):</b> nisserver .....	set up NIS+ servers
<b>nissetup(1M):</b> nissetup .....	initialize a NIS+ domain
<b>nisshowcache(1M):</b> nisshowcache .....	NIS+ utility to print out the contents of the shared cache file
<b>nisstat(1M):</b> nisstat .....	report NIS+ server statistics
<b>nisupdkeys(1M):</b> nisupdkeys .....	update the public keys in a NIS+ directory object
<b>nis_cachemgr(1M):</b> nis_cachemgr .....	maintains a cache containing location information about NIS+ servers
<b>ntpdate(1M):</b> ntpdate .....	set time and date via NTP
<b>ntpq(1M):</b> ntpq .....	Network Time Protocol query program
<b>nulladm:</b> shell procedures for accounting, create null file .....	see <b>acctsh(1M)</b>
<b>ocd(1M):</b> ocd .....	outbound connection daemon used by DDFA software
<b>ocdebug(1M):</b> ocdebug .....	outbound connection daemon debug utility used by DDFA software
<b>opx25(1M):</b> opx25 .....	execute HALGOL programs
<b>ospf_monitor(1M):</b> ospf_monitor .....	monitor OSPF gateways
<b>owners(1M):</b> owners .....	lists owners of outgoing network connections
<b>parcreate(1M):</b> parcreate .....	create a new partition
<b>parmgr(1M):</b> parmgr .....	partition manager
<b>parmodify(1M):</b> parmodify .....	modify an existing partition
<b>parremove(1M):</b> parremove .....	remove an existing partition
<b>parunlock(1M):</b> parunlock .....	unlock Stable Complex or Partition Configuration Data
<b>pcnfsd(1M):</b> rpc.pcnfsd .....	PC-NFS authentication and print request server
<b>pcserver(1M):</b> pcserver .....	Basic Serial and HP AdvanceLink server
<b>pd(1M):</b> pdc .....	processor-dependent code (firmware)
<b>pddcesetup(1M)</b> .....	configure DCE for the HP Distributed Print Service
<b>pdfck(1M):</b> pdfck .....	compare Product Description File and file system
<b>pdfdiff(1M):</b> pdfdiff .....	compare two Product Description Files
<b>pdgwcfg(1):</b> pdgwcfg .....	displays the text and description of a HPDPS message at the command line
<b>pdstartclient(1M)</b> .....	start the HPDPS client daemon
<b>pdstartspl(1M)</b> .....	create or restart an HPDPS spooler
<b>pdstartsuv(1M)</b> .....	create or restart an HPDPS supervisor
<b>pdstopd(1M)</b> .....	stop the HPDPS client daemon
<b>pfsd(1M):</b> pfs .....	PFS daemon
<b>pfsd.rpc:</b> PFS daemon .....	see <b>pfsd(1M)</b>

# Table of Contents

## Volumes Three and Four

Entry Name(Section): name	Description
<b>pfs_exportfs(1M):</b> pfs_exportfs .....	export and unexport directories to PFS clients
<b>pfs_mount(1M):</b> pfs_mount .....	mount and unmount CD-ROM file systems
<b>pfs_mountd(1M):</b> pfs_mountd .....	PFS mount request server
pfs_mountd.rpc: PFS mount request server .....	see <b>pfs_mountd(1M)</b>
pfs_umount: unmount CD-ROM file systems .....	see <b>pfs_mount(1M)</b>
<b>ping(1M):</b> ping .....	send echo request packets to a network host; test host availability
<b>power_onoff(1M):</b> power_onoff .....	timed, system power on/off
prctmp: shell procedures for accounting, print session record file .....	see <b>acctsh(1M)</b>
prdaily: shell procedures for accounting, print daily report .....	see <b>acctsh(1M)</b>
prtacct: shell procedures for accounting, print accounting file .....	see <b>acctsh(1M)</b>
<b>pscan(1M):</b> pscan .....	scan HP SCSI disk array LUNs for parity consistency
<b>pvchange(1M):</b> pvchange ....	change characteristics and access path of physical volume in LVM volume group
<b>pvck(1M):</b> pvck .....	check or repair a physical volume in LVM volume group
<b>pvcreate(1M):</b> pvcreate .....	create physical volume for use in LVM volume group
<b>pvddisplay(1M):</b> pvddisplay .....	display information about physical volumes in LVM volume group
<b>pvmove(1M):</b> pvmove .....	move physical extents from one LVM physical volume to other physical volumes
<b>pvremove(1M):</b> pvremove .....	remove an LVM physical volume
<b>pwck(1M):</b> pwck, grpck .....	password/group file checkers
<b>pwconv(1M):</b> pwconv .....	update secure password facility
<b>pwgrd(1M):</b> pwgrd .....	password and group hashing and caching daemon
<b>pwgr_stat(1M):</b> pwgr_stat .....	password and group hashing and caching statistics
<b>quot(1M):</b> quot .....	summarize file system ownership
<b>quotacheck(1M):</b> quotacheck .....	file system quota consistency checker
<b>quotacheck_hfs(1M):</b> quotacheck_hfs .....	hfs file system quota consistency checker
<b>quotacheck_vxfs(1M):</b> quotacheck_vxfs .....	VxFS file system quota consistency checker
quotaoff: turn file system quotas off .....	see <b>quotaon(1M)</b>
<b>quotaon(1M):</b> quotaoff, quotaon .....	turn file system quotas on and off
<b>quot_hfs(1M):</b> quot .....	summarize HFS file system ownership
<b>quot_vxfs(1M):</b> quot .....	summarize file system ownership
<b>rad(1M):</b> rad .....	perform OLA/R functions without any comprehensive checks
<b>rarp(1M):</b> rarp .....	Reverse Address Resolution Protocol client
<b>rarpd(1M):</b> rarpd .....	Reverse Address Resolution Protocol daemon
<b>rbootd(1M):</b> rbootd .....	remote boot server
<b>rc(1M):</b> rc .....	general purpose sequencer invoked upon entering new run level
<b>rcancel(1M):</b> rcancel .....	remove requests from a remote line printer spooling queue
<b>rdpd(1M):</b> rdpd .....	router discovery protocol daemon
rdump: incremental file system dump across network .....	see <b>dump(1M)</b>
<b>reboot(1M):</b> reboot .....	reboot the system
reject: prevent LP printer queuing requests .....	see <b>accept(1M)</b>
<b>remshd(1M):</b> remshd .....	remote shell server
<b>renice(1):</b> renice .....	alter priority of running processes
<b>repquota(1M):</b> repquota .....	summarize file system quotas
<b>restore(1M):</b> restore, rrestore .....	restore file system incrementally, local or over a network
<b>revck(1M):</b> revck .....	check internal revision numbers of HP-UX files
<b>rex(1M):</b> rex .....	RPC-based remote execution server
<b>rexecd(1M):</b> rexecd .....	remote execution server
<b>ripquery(1M):</b> ripquery .....	query RIP gateways
<b>rlogind(1M):</b> rlogind .....	remote login server
<b>rlp(1M):</b> rlp .....	send LP line printer request to a remote system
<b>rlpdaemon(1M):</b> rlpdaemon .....	line printer daemon for LP requests from remote systems
<b>rlpstat(1M):</b> rlpstat .....	print status of LP spooler requests on a remote system
rmboot - install, update, or remove boot programs from a disk device .....	see <b>mkboot(1M)</b>
<b>rmsf(1M):</b> rmsf .....	remove a special (device) file
<b>rmt(1M):</b> rmt .....	remote magnetic-tape protocol module
<b>route(1M):</b> route .....	manually manipulate routing tables
<b>rpc.nisd(1M):</b> rpc.nisd_resolv, nisd, nisd_resolv .....	NIS+ service daemon
rpc.nisd_resolv: NIS+ service daemon .....	see <b>rpc.nisd(1M)</b>
<b>rpc.nispasswd(1M):</b> rpc.nispasswd(), nispasswd() .....	NIS+ password update daemon
rpc.pcnfsd: PC-NFS authentication and print request server .....	see <b>pcnfsd(1M)</b>
rpc.yppupdated: hex encryption and utility routines .....	see <b>yppupdated(1M)</b>
<b>rpcbind(1M):</b> rpcbind .....	universal addresses to RPC program number mapper

## Table of Contents

### Volumes Three and Four

Entry Name(Section): name	Description
<b>rpcinfo(1M):</b> rpcinfo .....	report RPC information
<b>rpr(1M):</b> rpr .....	repair parity information on an HP SCSI disk array LUN
<b>rquotad(1M):</b> rquotad .....	remote quota server
<b>rrestore:</b> restore file system incrementally over a network .....	see <b>restore(1M)</b>
<b>rstatd(1M):</b> rstatd .....	kernel statistics server
<b>runacct(1M):</b> runacct .....	run daily accounting
<b>rusersd(1M):</b> rusersd .....	network username server
<b>rvxdump:</b> incremental file system dump across network .....	see <b>vxdump(1M)</b>
<b>rvxrestore:</b> restore file system incrementally across network .....	see <b>vxrestore(1M)</b>
<b>rwall(1M):</b> rwall .....	write to all users over a network
<b>rwalld(1M):</b> rwalld .....	network rwall server
<b>rwhod(1M):</b> rwhod .....	system status server
<b>sa1(1M):</b> sa1, sa2, sadc .....	system activity report package
<b>sa2:</b> system activity report package .....	see <b>sa1(1M)</b>
<b>sadc:</b> system activity report package .....	see <b>sa1(1M)</b>
<b>sam(1M):</b> sam .....	system administration manager
<b>sar(1M):</b> sar .....	system activity reporter
<b>savecrash(1M):</b> savecrash .....	save a crash dump of the operating system
<b>scn(1M):</b> scn .....	scan HP SCSI disk array LUNs for parity consistency
<b>scsictl(1M):</b> scsictl .....	control a SCSI device
<b>sd:</b> create and monitor jobs .....	see <b>swjob(1M)</b>
<b>see(1M):</b> see .....	access EEPROM bytes in an HP SCSI disk array controller
<b>sendmail(1M):</b> sendmail .....	send mail over the Internet
<b>service.switch(1M):</b> service.switch .....	indicate lookup sources and fallback mechanism
<b>setboot(1M):</b> setboot .....	display and modify boot variables in stable storage
<b>setext(1M):</b> setext .....	set extent attributes (VxFS)
<b>setmemwindow(1M):</b> setmemwindow .....	set window id of a running process or start a program in a particular memory window
<b>setmnt(1M):</b> setmnt .....	establish file-system mount table, /etc/mnttab
<b>setprivgrp(1M):</b> setprivgrp .....	set special privileges for group
<b>setuname(1M):</b> setuname .....	change machine information
<b>set_parms(1M):</b> set_parms .....	Initial system set up program
<b>showmount(1M):</b> showmount .....	show all remote mounts
<b>shutacct:</b> shell procedures for accounting, turn off accounting .....	see <b>acctsh(1M)</b>
<b>shutdown(1M):</b> shutdown .....	terminate all processing
<b>sig_named(1M):</b> sig_named .....	send signals to the domain name server
<b>smrsh(1M):</b> smrsh .....	restricted shell for sendmail
<b>snmpd(1M):</b> snmpd .....	daemon that responds to SNMP requests
<b>softpower(1M):</b> softpower .....	determine if softpower hardware is installed
<b>spd(1M):</b> spd .....	set physical drive parameters for an HP SCSI disk array
<b>spray(1M):</b> spray .....	spray packets
<b>sprayd(1M):</b> sprayd .....	spray server
<b>sss(1M):</b> sss .....	set spindle synchronization state of drives in an HP SCSI disk array
<b>st(1M):</b> st .....	shared tape administration
<b>startup:</b> shell procedures for accounting, start up accounting .....	see <b>acctsh(1M)</b>
<b>statd(1M):</b> statd .....	network status monitor
<b>strace(1M):</b> strace .....	write STREAMS event trace messages to standard output
<b>strchg(1M):</b> strchg, strconf .....	change or query stream configuration
<b>strclean(1M):</b> strclean .....	remove outdated STREAMS error log files
<b>strconf:</b> query stream configuration .....	see <b>strchg(1M)</b>
<b>strddb(1M):</b> strddb .....	STREAMS debugging tool
<b>strerr(1M):</b> strerr .....	receive error messages from the STREAMS log driver
<b>strvf(1M):</b> strvf .....	STREAMS verification tool
<b>swacl(1M):</b> swacl .....	view or modify Access Control Lists
<b>swagent:</b> perform software management tasks as the agent of an SD command .....	see <b>swagentd(1M)</b>
<b>swagentd(1M):</b> swagentd, swagent .....	serve local or remote SD-UX software management tasks
<b>swapinfo(1M):</b> swapinfo .....	system paging space information
<b>swapon(1M):</b> swapon .....	enable device or file system for paging
<b>swask(1M):</b> swask .....	ask for user response for SD-UX
<b>swconfig(1M):</b> swconfig .....	configure, unconfigure, or reconfigure installed software
<b>swcopy:</b> copy software products for subsequent installation or distribution .....	see <b>swinstall(1M)</b>

# Table of Contents

## Volumes Three and Four

Entry Name(Section): name	Description
<b>swinstall(1M):</b> swcopy, swinstall .....	install and configure software products, copy software products
<b>swjob(1M):</b> swjob, sd .....	display job information, remove jobs, create and monitor jobs
<b>swlist(1M):</b> swlist .....	display information about software products
<b>swmodify(1M):</b> swmodify .....	modify software products in a target root or depot
<b>swpackage(1M):</b> swpackage .....	package software products into a target depot or tape
<b>swreg(1M):</b> swreg .....	register or unregister depots and roots
<b>swremove(1M):</b> swremove .....	unconfigure and remove software products
<b>swverify(1M):</b> swverify .....	verify software products
<b>sync(1M):</b> sync .....	synchronize file systems
<b>syncer(1M):</b> syncer .....	periodically sync for file system integrity
<b>sysdef(1M):</b> sysdef .....	display system definition
<b>syslogd(1M):</b> syslogd .....	log systems messages
<b>talkd(1M):</b> talkd .....	remote user communication server
<b>telnetd(1M):</b> telnetd .....	TELNET protocol server
<b>tftpd(1M):</b> tftpd .....	trivial file transfer protocol server
<b>tic(1M):</b> tic .....	terminfo compiler
<b>tsm.lpadmin(1M):</b> tsm.lpadmin .....	add or remove a printer for use with tsm
<b>ttsyncd(1M):</b> ttsyncd .....	Daemon to maintain the nis+ password table in sync with the nis+ trusted table
<b>tunefs(1M):</b> tunefs .....	tune up an existing HFS file system
<b>turnacct:</b> shell procedures for accounting, turn on or off process accounting .....	see <b>acctsh(1M)</b>
<b>udpublickey(1M):</b> udpublickey .....	updates the <b>publickey</b> database file and the NIS map
<b>umount:</b> mount and unmount CDFS file systems .....	see <b>mount_cdfs(1M)</b>
<b>umount:</b> mount and unmount HFS file systems .....	see <b>mount_hfs(1M)</b>
<b>umount:</b> mount and unmount NFS file systems .....	see <b>mount_nfs(1M)</b>
<b>umount:</b> unmount a file system .....	see <b>mount(1M)</b>
<b>unlink:</b> execute unlink() system call without error checking .....	see <b>link(1M)</b>
<b>unmount:</b> unmount VxFS file system .....	see <b>mount_vxfs(1M)</b>
<b>untic(1M):</b> untic .....	terminfo de-compiler
<b>update-ux(1M):</b> update-ux .....	updates the HP-UX operating system
<b>updaters(1M):</b> updaters .....	configuration file for NIS updating
<b>ups_mond(1M):</b> ups_mond .....	Uninterruptible Power System monitor daemon
<b>useradd(1M):</b> useradd .....	add a user login on the system
<b>userdel(1M):</b> userdel .....	delete a user login from the system
<b>usermod(1M):</b> usermod .....	modify a user login on the system
<b>uuccheck(1M):</b> uuccheck .....	check the uucp directories and permissions file
<b>uucico(1M):</b> uucico .....	transfer files for the uucp system
<b>uuclean(1M):</b> uuclean .....	uucp spool directory clean-up
<b>uucleanup(1M):</b> /0/0uucleanup .....	uucp spool directory clean-up
<b>uucpd():</b> server for supporting UUCP over TCP/IP networks .....	see <b>uucpd(1M)</b>
<b>uucpd(1M):</b> uucpd() .....	server for supporting UUCP over TCP/IP networks
<b>uugetty(1M):</b> uugetty .....	set terminal type, modes, speed and line discipline
<b>uuls(1M):</b> uuls .....	list spooled uucp transactions grouped by transaction
<b>uusched(1M):</b> uusched .....	schedule uucp transport files
<b>uusnap(1M):</b> uusnap .....	show snapshot of the UUCP system
<b>uusnaps(1M):</b> uusnaps .....	sort and embellish uusnap output
<b>uusub(1M):</b> uusub .....	monitor uucp network
<b>uuxqt(1M):</b> uuxqt .....	execute remote uucp or uux command requests
<b>vgcfgbackup(1M):</b> vgcfgbackup .....	create or update LVM volume group configuration backup file
<b>vgcfgrestore(1M):</b> vgcfgrestore .....	restore volume group configuration
<b>vgchange(1M):</b> vgchange .....	set LVM volume group availability
<b>vgchgid(1M):</b> vgchgid .....	modify the Volume Group ID (VGID) on a given set of physical devices
<b>vgcreate(1M):</b> vgcreate .....	create LVM volume group
<b>vgdisplay(1M):</b> vgdisplay .....	display information about LVM volume groups
<b>vgexport(1M):</b> vgexport .....	export an LVM volume group and its associated logical volumes
<b>vgextend(1M):</b> vgextend .....	extend an LVM volume group by adding physical volumes
<b>vgimport(1M):</b> vgimport .....	import an LVM volume group onto the system
<b>vgreduce(1M):</b> vgreduce .....	remove physical volumes from an LVM volume group
<b>vgremove(1M):</b> vgremove .....	remove LVM volume group definition from the system
<b>vgscan(1M):</b> vgscan .....	scan physical volumes for LVM volume groups
<b>vgsync(1M):</b> vgsync .....	synchronize stale logical volume mirrors in LVM volume groups
<b>vipw(1M):</b> vipw .....	edit the password file



## Table of Contents

### Volumes Three and Four

Entry Name(Section): name	Description
<b>volcopy(1M):</b> volcopy, labelit .....	copy file systems with label checking
<b>volcopy_hfs(1M):</b> volcopy, labelit .....	copy file systems with label checking
<b>volcopy_vxfs(1M):</b> volcopy, labelit .....	copy VxFS file system with label checking
<b>vtdaemon(1M):</b> vtdaemon .....	respond to vt requests
<b>vxdiskusg(1M):</b> vxdiskusg .....	generate disk accounting data of VxFS file system by user ID
<b>vxdump(1M):</b> rvxdump, vxdump .....	incremental file system dump, local or across network
<b>vxenablef(1M):</b> vxenablef .....	enable VxFS DMAPI or OnLineJFS functionality in the kernel
<b>vxfsconvert(1M):</b> vxfsconvert .....	convert file system to vxfs file system
<b>vxlicense(1M):</b> vxlicense .....	VxFS licensing key utility
<b>vxrestore(1M):</b> vxrestore, rvxrestore .....	restore file system incrementally, local or across network
<b>vxtunefs(1M):</b> vxtunefs .....	tune a VxFS file system
<b>vxupgrade(1M):</b> vxupgrade .....	upgrade the disk layout of a VxFS file system
<b>wall(1M):</b> wall .....	write message to all users
<b>whodo(1M):</b> whodo .....	which users are doing what
<b>wtmpfix:</b> manipulate connect accounting records .....	see <b>fwtmp(1M)</b>
<b>xntpd(1M):</b> xntpd .....	Network Time Protocol daemon
<b>xntpd(1M):</b> xntpd .....	special NTP query program
<b>ypbind:</b> Network Information Service (NIS) binder processes .....	see <b>ypserv(1M)</b>
<b>ypinit(1M):</b> ypinit .....	build and install Network Information Service databases
<b>ypmake(1M):</b> ypmake .....	create or rebuild Network Information Service databases
<b>yppasswd(1M):</b> yppasswdd .....	daemon for modifying Network Information Service passwd database
<b>yppoll(1M):</b> yppoll .....	query NIS server for information about an NIS map
<b>yppush(1M):</b> yppush .....	force propagation of a Network Information Service database
<b>ypserv(1M):</b> ypserv, ypbind, ypxfrd .....	Network Information Service (NIS) server and binder processes
<b>ypset(1M):</b> ypset .....	bind to particular Network Information Service server
<b>ypupdated(1M):</b> ypupdated, rpc.ypupdated .....	server for changing NIS information
<b>ypxfr(1M):</b> ypxfr, ypxfr_1perday, ypxfr_1perhour, ypxfr_2perday .....	transfer NIS database from NIS server to local node
<b>ypxfrd:</b> Network Information Service (NIS) transfer processes .....	see <b>ypserv(1M)</b>

## Notes

**Section 1M**

**Part 1**

**System Administration Commands  
(A-M)**

# **Section 1M**

## **Part 1**

### **System Administration Commands (A-M)**

**NAME**

intro - introduction to system maintenance commands and application programs

**DESCRIPTION**

This section describes commands that are used chiefly for system maintenance and administration purposes. The commands in this section should be used in conjunction with other sections of this manual, as well as the HP-UX System Administration manuals for your system.

**Command Syntax**

Unless otherwise noted, commands described in this section accept options and other arguments according to the following syntax:

*name* [ *option* ( *s* ) ] [ *cmd\_arg* ( *s* ) ]

where the elements are defined as follows:

*name*      Name of an executable file.

*option*      One or more *options* can appear on a command line. Each takes one of the following forms:

    -*no\_arg\_letter*

        A single letter representing an option without an argument.

    -*no\_arg\_letters*

        Two or more single-letter options combined into a single command-line argument.

    -*arg\_letter*<>*opt\_arg*

        A single-letter option followed by a required argument where:

*arg\_letter*

            is the single letter representing an option that requires an argument,

*opt\_arg*

            is an argument (character string) satisfying the preceding *arg\_letter*,

        <>      represents optional white space.

*cmd\_arg*      Path name (or other command argument) *not* beginning with -, or - by itself indicating the standard input. If two or more *cmd\_args* appear, they must be separated by white space.

**RETURN STATUS**

Upon termination, each command returns two bytes of status, one supplied by the system giving the cause for termination, and (in the case of "normal" termination) one supplied by the program (for descriptions, see *wait(2)* and *exit(2)*). The system-supplied byte is 0 for normal termination. The byte provided by the program is customarily 0 for successful execution and non-zero to indicate errors or failure such as incorrect parameters in the command line, or bad or inaccessible data. Values returned are usually called variously "exit code", "exit status", or "return code", and are described only where special conventions are involved.

**WARNINGS**

Some commands produce unexpected results when processing files containing null characters. These commands often treat text input lines as strings and therefore become confused upon encountering a null character (the string terminator) within a line.

**SEE ALSO**

getopt(1), exit(2), wait(2), getopt(3C), hier(5), introduction(9).

Web access to HP-UX documentation at <http://docs.hp.com>.

**NAME**

accept, reject - allow/prevent LP printer queuing requests

**SYNOPSIS**

```
/usr/sbin/accept destination ...
```

```
/usr/sbin/reject [-r[reason]] destination ... [-r[reason] destination ...] ...
```

**DESCRIPTION**

The **accept** command permits the **lp** command (see *lp(1)*) to accept printing requests for each named LP printer or printer class *destination* queue.

The **reject** command causes the **lp** command to reject subsequent printing requests for each named *destination* queue. Requests already queued will continue to be processed for printing by the **lpsched** scheduler (see *lpsched(1M)*).

Use the **lpstat** command (see *lpstat(1)*) to find the status of destination queues.

For an overview of LP command interactions, see *lp(1)*.

**Options**

The **reject** command can have the following option.

**-r[reason]** Specifies a string that is used to explain why the **lp** command is not accepting requests for a destination. *reason* applies to all queues mentioned up to the next **-r** option. If *reason* or **-r[reason]** is omitted, the default is **"reason unknown"**. The maximum length of *reason* is 80 bytes.

*reason* is reported by the **lpstat** command and by the **lp** command when users direct requests to a rejected destination.

**EXTERNAL INFLUENCES****Environment Variables**

The **LANG** variable determines the language in which messages are displayed. If **LANG** is not specified or is set to the empty string, it defaults to "C" (see *lang(5)*).

If any internationalization variable contains an invalid setting, all internationalization variables default to "C" (see *environ(5)*).

**International Code Set Support**

Single- and multibyte character code sets are supported.

**EXAMPLES**

These examples assume you have a system with two printers named **laser1** and **jet2**, and one class named **lj** that includes both printers.

**Example 1**

To allow all destinations to accept print requests:

```
accept laser1 jet2 lj
```

**Example 2**

To reject requests to the **lj** class destination, requiring users to choose a printer:

```
reject lj
```

**Example 3**

To reject requests to the individual printer destinations, requiring all requests to go through the class destination:

```
accept lj
reject -r"use the lj destination" laser1 jet2
```

**WARNINGS**

**accept** and **reject** operate on the local system only.

**FILES**

<code>/etc/lp</code>	Directory of spooler configuration data
<code>/var/adm/lp</code>	Directory of spooler log files
<code>/var/spool/lp</code>	Directory of LP spooling files and directories

**SEE ALSO**

enable(1), lp(1), lpstat(1), lpadmin(1M), lpsched(1M), rcancel(1M), rlp(1M), rlpdaemon(1M), rlpstat(1M).

**a**

## NAME

acctdisk, acctdusg, accton, acctwtmp, closewtmp, utmp2wtmp - overview of accounting and miscellaneous accounting commands

## SYNOPSIS

```
/usr/sbin/acct/acctdisk
/usr/sbin/acct/acctdusg [-u file] [-p file]
/usr/sbin/acct/accton [file]
/usr/sbin/acct/acctwtmp reason
/usr/sbin/acct/closewtmp
/usr/sbin/acct/utmp2wtmp
```

## DESCRIPTION

Accounting software is structured as a set of tools (consisting of both C programs and shell procedures) that can be used to build accounting systems. The shell procedures, described in *acctsh*(1M), are built on top of the C programs.

Connect time accounting is handled by various programs that write records into */etc/utmp*, as described in *utmp*(4). The programs described in *acctcon*(1M) convert this file into session and charging records which are then summarized by *acctmerg* (see *acctmerg*(1M)).

Process accounting is performed by the HP-UX system kernel. Upon termination of a process, one record per process is written to a file (normally */var/adm/pacct*). The programs in *acctprc*(1M) summarize this data for charging purposes; *acctcms* is used to summarize command usage (see *acctcms*(1M)). Current process data can be examined using *acctcom* (see *acctcom*(1M)).

Process accounting and connect time accounting (or any accounting records in the format described in *acct*(4)) can be merged and summarized into total accounting records by *acctmerg* (see *tacct* format in *acct*(4)). *prtacct* is used to format any or all accounting records (see *acctsh*(1M)).

*acctdisk* reads lines that contain user ID, login name, and number of disk blocks, and converts them to total accounting records that can be merged with other accounting records.

*acctdusg* reads its standard input (usually from *find -print*) and computes disk resource consumption (including indirect blocks) by login. Only files found under login directories (as determined from the password file) are accounted for. All files under a login directory are assumed to belong to that user regardless of actual owner. If *-u* is given, records consisting of those file names for which *acctdusg* charges no one are placed in *file* (a potential source for finding users trying to avoid disk charges). If *-p* is given, *file* is the name of the password file. This option is not needed if the password file is */etc/passwd*. (See *diskusg*(1M) for more details.)

*accton* turns process accounting off if the optional *file* argument is omitted. If *file* is given, it must be the name of an existing file, to which the kernel appends process accounting records (see *acct*(2) and *acct*(4)).

*acctwtmp* writes a *utmp*(4) record to its standard output. The record contains the current time and a string of characters that describe the *reason* for writing the record. A record type of ACCOUNTING is assigned (see *utmp*(4)). The string argument *reason* must be 11 or fewer characters, numbers, \$, or spaces. For example, the following are suggestions for use in reboot and shutdown procedures, respectively:

```
acctwtmp `uname` >> /var/adm/wtmp
acctwtmp "file save" >> /var/adm/wtmp
```

*closewtmp* writes a DEAD\_PROCESS record, for each user currently logged in, to the file */var/adm/wtmp*. This program is invoked by *runacct* to close the existing *wtmp* file before creating a new one.

*utmp2wtmp* writes a USER\_PROCESS record, for each user currently logged in, to the file */var/adm/wtmp*. This program is invoked by *runacct* to initialize the newly created *wtmp* file.

## FILES

<i>/usr/sbin/acct</i>	Holds all accounting commands listed in section (1M) of this manual.
<i>/var/adm/pacct</i>	Current process accounting file.



<code>/etc/passwd</code>	Used for converting login name to user ID
<code>/var/adm/wtmp</code>	Login/logoff history file.

**SEE ALSO**

acctcms(1M), acctcom(1M), acctcon(1M), acctmerg(1M), acctprc(1M), acctsh(1M), diskusg(1M), fwtmp(1M), runacct(1M), acct(2), acct(4), utmp(4).

**STANDARDS CONFORMANCE**

**acctdisk:** SVID2, SVID3

**accton:** SVID2, SVID3

**acctwtmp:** SVID2, SVID3

  
**a**

**NAME**

acctcms - command summary from per-process accounting records

**SYNOPSIS**

`/usr/sbin/acct/acctcms` [*options*] *files*

**DESCRIPTION**

**acctcms** reads one or more *files*, normally in the form described in *acct*(4). It adds all records for processes that executed identically-named commands, sorts them, and writes them to the standard output, normally using an internal summary format.

**Options**

**acctcms** recognizes the following options:

- a Print output in ASCII rather than in the internal summary format. The output includes command name, number of times executed, total kcore-minutes, total CPU minutes, total real minutes, mean size (in K), mean CPU minutes per invocation, "hog factor", characters transferred, and blocks read and written, as in *acctcom*(1M). Output is normally sorted by total kcore-minutes.
- c Sort by total CPU time, rather than total kcore-minutes.
- j Combine all commands invoked only once under **\*\*\*other**.
- n Sort by number of command invocations.
- s Any file names encountered hereafter are already in internal summary format.
- t Process all records as total accounting records. The default internal summary format splits each field into prime- and non-prime-time parts. This option combines the prime and non-prime time parts into a single field that is the total of both, and provides upward compatibility with old (i.e., UNIX System V) style **acctcms** internal summary format records.

The following options can be used only with the **-a** option.

- p Output a prime-time-only command summary.
- o Output a non-prime- (offshift) time only command summary.

When **-p** and **-o** are used together, a combination prime and non-prime time report is produced. All the output summaries are total usage except number of times executed, CPU minutes, and real minutes which are split into prime and non-prime.

**EXAMPLES**

A typical sequence for performing daily command accounting and for maintaining a running total is:

```
acctcms file ... >today
cp total previous total
acctcms -s today previous total >total
acctcms -a -s today
```

**WARNINGS**

Unpredictable output results if **-t** is used on new-style internal-summary-format files, or if it is not used with old style internal summary format files.

**SEE ALSO**

*acct*(1M), *acctcom*(1M), *acctcon*(1M), *acctmerg*(1M), *acctprc*(1M), *acctsh*(1M), *fwtmp*(1M), *runacct*(1M), *acct*(2), *acct*(4), *utmp*(4).

**STANDARDS CONFORMANCE**

**acctcms**: SVID2, SVID3

**NAME**

acctcom - search and print process accounting files

**SYNOPSIS**

`/usr/sbin/acct/acctcom` *[[option]... [file]] ...*

**DESCRIPTION**

The **acctcom** command reads *file*, standard input, or `/var/adm/pacct`, in the form described in *acct(4)* and writes selected records to standard output. Each record represents the execution of one process. The output has the following column titles:

```
COMMAND NAME
USER
TTYNAME
START TIME
END TIME
REAL (SECS)
CPU (SECS)
MEAN SIZE(K)
```

Optionally, the following can be displayed:

```
F          fork() / exec() flag: 1 for fork() without exec()
STAT       System exit status
HOG FACTOR
KCORE MIN
CPU FACTOR
CHARS TRNSFD
BLOCKS READ      Total blocks read and written
PRMID           PRM process resource group ID
```

The command name is preceded by a # if a privileged user is *required* to execute the command.

For example, if a user is logged in as **root**, and executes the **date** command to check the time, this does not require a privileged user, and will be shown by **acctcom** without the # character on the line. If the user executes the command **date 0731180092** to set the time, this requires a privileged user, and so will be marked with a # by **acctcom**.

If a process is not associated with a known terminal, a ? is printed in the **TTYNAME** field.

The system exit status **STAT** is 0 if the process terminated by calling **exit**. If it is not 0, it is the signal number that caused the process to terminate. If a core file image was produced as a result of the signal (see *signal(5)*), the value is the signal number plus 0200.

If no *files* are specified, and if standard input is associated with a terminal or `/dev/null` (as is the case when using **&** in a shell), **acctcom** reads `/var/adm/pacct`. Otherwise, it reads standard input.

If any *file* arguments are given, they are read in their respective order. Each file is normally read forward, that is, in chronological order by process-completion time. The file `/var/adm/pacct` is usually the current file to be examined. A busy system may need several such files of which all but the current file are found in `/var/adm/pacct[1-9]`.

**Options**

**acctcom** recognizes the following values for the *option* argument. Listing options together has the effect of a logical AND.

- a** Show some average statistics about the processes selected. Statistics are printed after the output records.
- b** Read backwards, showing latest commands first. This option has no effect when standard input is read.
- f** Print in octal the **F** flag and system exit status columns in the output.
- h** Instead of mean memory size, **MEAN SIZE(K)**, show the fraction of total available CPU time consumed by the process during its execution. This **HOG FACTOR** is computed as:

*total-CPU-time / elapsed-time*

- i** Print columns containing the I/O counts in the output.
- k** Instead of memory size, show total kcore-minutes.
- m** Show mean core size (the default).
- P** Show the PRM process resource group ID (**PRMID**) of each process. See DEPENDENCIES.
- r** Show CPU factor:  

$$\text{user-time} / (\text{system-time} + \text{user-time})$$
- t** Show separate system and user CPU times.
- v** Exclude column headings from the output.
- l line** Show only processes belonging to terminal */dev/line*.
- u user** Show only processes belonging to *user*, specified as: a user ID, a login name that is then converted to a user ID, a # which designates only those processes executed by a privileged user, or ? which designates only those processes associated with unknown user IDs. The # and ? characters should be preceded by a backslash (\) and typed as \# and \? to prevent the shell from interpreting the # as the start of a comment, or the ? as a pattern.
- g group** Show only processes belonging to *group*, specified as either the group ID or group name.
- s time** Select processes existing at or after *time*, given in the format:  

$$\text{hour}[:\text{minute}[:\text{second}]]$$
- e time** Select processes existing at or before *time*; see **-s**.  
 Using the same *time* for both **-s** and **-e** shows the processes that existed at *time*; see **-s**.
- S time** Select processes starting at or after *time*; see **-s**.
- E time** Select processes ending at or before *time*; see **-s**.
- n pattern** Show only commands matching *pattern*, where *pattern* is a regular expression as in *ed(1)* except that + means one or more occurrences.
- q** Do not print any output records. Just print the average statistics as with the **-a** option.
- o ofile** Copy selected process records in the input data format to *ofile*. Suppress standard output printing.
- H factor** Show only processes that exceed *factor*, where *factor* is the "hog factor" as explained in option **-h**.
- O time** Show only those processes with operating system CPU time exceeding *time*; see **-s**.
- C sec** Show only processes with total CPU time, system plus user, exceeding *sec* seconds.
- I chars** Show only processes transferring more characters than the cut-off number given by *chars*.
- R prmggroup** Show only processes belonging to process resource group *prmggroup*, specified as either process resource group name or ID number. See DEPENDENCIES.

## WARNINGS

**acctcom** only reports on processes that have terminated. For active processes, use the **ps** command (see *ps(1)*).

If *time* exceeds the current system clock time, *time* is interpreted as occurring on the previous day.

The accounting flag is not cleared when one processes exec's another, but only when one process forks another. One side-effect of this is that some processes will be marked with #, when users do not expect them to be.

For example, the **login** command requires a privileged user to assume the identity of the user who is logging-in, setting the ASU bit in the accounting flag (which ultimately causes the # symbol in the **acctcom** output). After assuming the user's identity, **login** exec's the user's shell. Since the exec does not clear the ASU flag, the shell will inherit it, and be marked with a # in the **acctcom** output.

**DEPENDENCIES****HP Process Resource Manager**

The **-P** and **-R** options require the optional HP Process Resource Manager (PRM) software to be installed and configured. See *prmconfig(1)* for a description of how to configure HP PRM, and *prmconf(4)* for the definition of process resource group.

**FILES**

**/etc/group**  
**/etc/passwd**  
**/var/adm/pacct**

**SEE ALSO**

**ps(1)**, **su(1)**, **acct(1M)**, **acctcms(1M)**, **acctcon(1M)**, **acctmerg(1M)**, **acctprc(1M)**, **acctsh(1M)**, **fwtmp(1M)**, **runacct(1M)**, **acct(2)**, **wait(2)**, **acct(4)**, **utmp(4)**, **signal(5)**.

HP Process Resource Manager: **prmconfig(1)**, **prmconf(4)** in *HP Process Resource Manager User's Guide*.

**STANDARDS CONFORMANCE**

**acctcom**: SVID2, SVID3

**a**

**NAME**

acctcon, acctcon1, acctcon2 - connect-time accounting

**SYNOPSIS**

```
/usr/sbin/acct/acctcon [options]
/usr/sbin/acct/acctcon1 [options]
/usr/sbin/acct/acctcon2
```

**DESCRIPTION**

The **acctcon1** command converts a sequence of login/logoff records read from its standard input to a sequence of records, one per login session. Its input should normally be redirected from **/var/adm/wtmp**. Its output is ASCII, giving device, user ID, login name, prime connect time (seconds), non-prime connect time (seconds), session starting time (numeric), and starting date and time. Prime connect time is defined as the connect time within a specific prime period on a non-holiday weekday (Monday through Friday). The starting and ending time of the prime period and the year's holidays are defined in file **/etc/acct/holidays**.

**acctcon2** expects as input a sequence of login session records, produced by **acctcon1**, and converts them into total accounting records (see **tacct** format in **acct(4)**).

**acctcon** combines the functionality of **acctcon1** and **acctcon2** into one program. It takes the same input format as **acctcon1** and writes the same output as **acctcon2**.

**acctcon1** recognizes the following *options*:

- p** Print input only, showing line name, login name, and time (in both numeric and date/time formats).
- t** **acctcon1** maintains a list of lines on which users are logged in. When it reaches the end of its input, it emits a session record for each line that still appears to be active. It normally assumes that its input is a current file, so that it uses the current time as the ending time for each session still in progress. The **-t** flag causes it to use, instead, the last time found in its input, thus ensuring reasonable and repeatable numbers for non-current files.

**acctcon1** and **acctcon** recognize the following *options*:

- l file** *file* is created to contain a summary of line usage showing line name, number of minutes used, percentage of total elapsed time used, number of sessions charged, number of logins, and number of logoffs. This file helps track line usage, identify bad lines, and find software and hardware oddities. Hang-up, termination of **login** (see **login(1)**), and termination of the login shell each generate logoff records, so that the number of logoffs is often three to four times the number of sessions. See **init(1M)** and **utmp(4)**.
- o file** *file* is filled with an overall record for the accounting period, giving starting time, ending time, number of reboots, and number of date changes.

**EXAMPLES**

These commands are typically used as shown below. The file **ctmp** is created only for the use of commands described by the **acctprc(1M)** manual entry:

```
acctcon1 -t -l lineuse -o reboots < wtmp | sort +1n +2 > ctmp
acctcon2 < ctmp | acctmerg > ctacct
```

or

```
acctcon -t -l lineuse -o reboots < wtmp | acctmerg > ctacct
```

**FILES**

```
/var/adm/wtmp
/etc/acct/holidays
```

**WARNINGS**

The line usage report is confused by date changes. Use **wtmpfix** (see **fwtmp(1M)**) to correct this situation.

**SEE ALSO**

acct(1M), acctcms(1M), acctcom(1M), acctmerg(1M), acctprc(1M), acctsh(1M), fwtmp(1M), init(1M), login(1), runacct(1M), acct(2), acct(4), utmp(4).

**STANDARDS CONFORMANCE**

**acctcon1:** SVID2, SVID3

**acctcon2:** SVID2, SVID3

**a**

**NAME**

acctmerg - merge or add total accounting files

**SYNOPSIS**

`/usr/sbin/acct/acctmerg` [*options*] [*file*] ...

**DESCRIPTION**

**acctmerg** reads its standard input and up to nine additional files, all in the **tacct** format (see *acct(4)*) or an ASCII version thereof. It merges these inputs by adding records whose keys (normally user ID and name) are identical, and expects the inputs to be sorted on those keys.

**Options**

**acctmerg** recognizes the following options:

- a** Produce output in ASCII version of **tacct**.
- i** Input files are in ASCII version of **tacct**.
- p** Print input with no processing.
- t** Produce a single record that totals all input.
- u** Summarize by user ID, rather than user ID and name.
- v** Produce output in verbose ASCII format, with more precise notation for floating point numbers.

**EXAMPLES**

The following sequence is useful for making "repairs" to any file kept in this format:

```
acctmerg -v < file1 > file2  
    edit file2 as desired ...  
acctmerg -i < file2 > file1
```

**SEE ALSO**

acct(1M), acctcms(1M), acctcom(1M), acctcon(1M), acctprc(1M), acctsh(1M), fwtmp(1M), runacct(1M), acct(2), acct(4), utmp(4).

**STANDARDS CONFORMANCE**

**acctmerg**: SVID2, SVID3



**NAME**

acctprc, acctprc1, acctprc2 - process accounting

**SYNOPSIS**

```
/usr/sbin/acct/acctprc
/usr/sbin/acct/acctprc1 [ctmp]
/usr/sbin/acct/acctprc2
```

**DESCRIPTION**

**acctprc1** reads input in the form described by *acct(4)*, adds login names corresponding to user IDs, then writes for each process an ASCII line giving user ID, login name, prime CPU time (tics), non-prime CPU time (tics), and mean memory size (in memory segment units). If **ctmp** is given, it is expected to contain a list of login sessions in the form described in *acctcon(1M)*, sorted by user ID and login name. If this file is not supplied, it obtains login names from the password file. The information in **ctmp** helps it distinguish among different login names that share the same user ID.

**acctprc2** reads records in the form written by **acctprc1**, summarizes them by user ID and name, then writes the sorted summaries to the standard output as total accounting records.

**acctprc** combines the functionality of **acctprc1** and **acctprc2** into one program. It takes the same input format as **acctprc1** (but does not accept the **ctmp** argument) and writes the same output as **acctprc2**.

These commands are typically used as shown below:

```
acctprc1 ctmp < /var/adm/pacct | acctprc2 > ptacct
or
acctprc < /var/adm/pacct > ptacct
```

**EXTERNAL INFLUENCES****Environment Variables**

For the output of **acctprc2**, if the user IDs are identical, **LC\_COLLATE** determines the order in which the user names are sorted.

If **LC\_COLLATE** is not specified in the environment or is set to the empty string, the value of **LANG** is used as a default. If **LANG** is not specified or is set to the empty string, a default of "C" (see *lang(5)*) is used instead of **LANG**. If any internationalization variable contains an invalid setting, **acctprc2** behaves as if all internationalization variables are set to "C" (see *environ(5)*).

**WARNINGS**

Although it is possible to distinguish among login names that share user IDs for commands run normally, it is difficult to do this for those commands run from **cron** for example (see *cron(1M)*). More precise conversion can be done by faking login sessions on the console via the **acctwtmpt** program in *acct(1M)*.

A memory segment of the mean memory size is a unit of measure for the number of bytes in a logical memory segment on a particular processor.

**FILES**

/etc/passwd

**SEE ALSO**

acct(1M), acctcms(1M), acctcom(1M), acctcon(1M), acctmerg(1M), acctsh(1M), cron(1M), fwtmpt(1M), runacct(1M), acct(2), acct(4), utmp(4).

**STANDARDS CONFORMANCE**

**acctprc1**: SVID2, SVID3

**acctprc2**: SVID2, SVID3

**NAME**

chargefee, ckpacct, dodisk, lastlogin, monacct, nulladm, prctmp, prdaily, prtacct, shutacct, startup, turnacct - shell procedures for accounting

**SYNOPSIS**

```
/usr/sbin/acct/chargefee login-name number
/usr/sbin/acct/ckpacct [blocks]
/usr/sbin/acct/dodisk [-o] [files ...]
/usr/sbin/acct/lastlogin
/usr/sbin/acct/monacct number
/usr/sbin/acct/nulladm file
/usr/sbin/acct/prctmp
/usr/sbin/acct/prdaily [-l] [-c] [mmd]
/usr/sbin/acct/prtacct file [heading]
/usr/sbin/acct/shutacct [reason]
/usr/sbin/acct/startup
/usr/sbin/acct/turnacct on | off | switch
```

**DESCRIPTION**

- chargefee** Can be invoked to charge a *number* of units to *login-name*. A record is written to */var/adm/fee*, to be merged with other accounting records during the night.
- ckpacct** Should be initiated via *cron*(1M). It periodically checks the size of */var/adm/pacct*. If the size exceeds *blocks*, 1000 by default, *turnacct* is invoked with argument *switch*. If the number of free disk blocks in the */var* file system falls below 500, *ckpacct* automatically turns off the collection of process accounting records via the *off* argument to *turnacct*. When at least this number of blocks is restored, the accounting will be activated again. This feature is sensitive to the frequency at which *ckpacct* is executed, usually by *cron*.
- dodisk** Should be invoked by *cron* to perform the disk accounting functions. By default, it will do disk accounting on the special files in */etc/fstab*. If the *-o* flag is used, it does a slower version of disk accounting by login directory. *files* specifies the one or more filesystem names where disk accounting is to be done. If *files* is used, disk accounting will be done on these filesystems only. If the *-o* flag is used, *files* should be mount points of mounted filesystem. If omitted, they should be the special file names of mountable filesystems.
- lastlogin** Invoked by *runacct* to update */var/adm/acct/sum/loginlog* which shows the last date on which each user logged in (see *runacct*(1M)).
- monacct** Should be invoked once each month or each accounting period. *number* indicates which month or period it is. If *number* is not given, it defaults to the current month (01 through 12). This default is useful if *monacct* is to be executed via *cron* on the first day of each month. *monacct* creates summary files in */var/adm/acct/fiscal* and restarts summary files in */var/adm/acct/sum*.
- nulladm** Creates *file* with mode 664 and ensures that owner and group are *adm*. It is called by various accounting shell procedures.
- prctmp** Can be used to print the session record file normally */var/adm/acct/nite/ctmp* created by *acctcon1* (see *acctcon*(1M)).
- prdaily** Invoked by *runacct* (see *runacct*(1M)) to format a report of the previous day's accounting data. The report resides in */var/adm/acct/sum/rprtmmdd* where *mmdd* is the month and day of the report. The current daily accounting reports may be printed by typing *prdaily*. Previous days' accounting reports can be printed by using the *mmdd* option and specifying the exact report date desired. The *-l* flag prints a report of exceptional usage by login id for the specified date. Previous daily reports are cleaned up and therefore inaccessible after each invocation of *monacct*. The *-c* flag prints a report of exceptional

	resource usage by command, and can be used on current day's accounting data only.
<b>prtacct</b>	Can be used to format and print any total accounting ( <b>tacct</b> ) file.
<b>shutacct</b>	Should be invoked during a system shutdown to turn process accounting off and append a "reason" record to <b>/var/adm/wtmp</b> .
<b>startup</b>	Should be called by system startup scripts to turn the accounting on whenever the system is brought up.
<b>turnacct</b>	An interface to <b>accton</b> (see <b>acct(1M)</b> ) to turn process accounting <b>on</b> or <b>off</b> . The <b>switch</b> argument turns accounting off, moves the current <b>/var/adm/pacct</b> to the next free name in <b>/var/adm/pacctincr</b> then turns accounting back on again. ( <i>incr</i> is a number starting with 1 and incrementing by one for each additional <b>pacct</b> file.) <b>tur-nacct</b> is called by <b>ckpacct</b> , and thus can be run under <b>cron</b> and used to keep <b>pacct</b> to a reasonable size.

## FILES

<b>/usr/sbin/acct</b>	holds all accounting commands listed in section (1M) of this manual
<b>/var/adm/fee</b>	accumulator for fees
<b>/var/adm/acct/nite</b>	working directory
<b>/var/adm/pacct</b>	current file for per-process accounting
<b>/var/adm/pacct*</b>	used if <b>pacct</b> gets large, and during execution of daily accounting procedure
<b>/usr/sbin/acct/ptecms.awk</b>	contains the limits for exceptional usage by command name
<b>/usr/sbin/acct/ptelus.awk</b>	contains the limits for exceptional usage by login id
<b>/var/adm/acct/sum</b>	summary directory, should be saved
<b>/var/adm/wtmp</b>	login/logoff summary

## SEE ALSO

**acct(1M)**, **acctcms(1M)**, **acctcom(1M)**, **acctcon(1M)**, **acctmerg(1M)**, **acctprc(1M)**, **cron(1M)**, **diskusg(1M)**, **fwtmp(1M)**, **runacct(1M)**, **acct(2)**, **acct(4)**, **utmp(4)**.

## STANDARDS CONFORMANCE

**chargefee**: SVID2, SVID3  
**ckpacct**: SVID2, SVID3  
**dodisk**: SVID2, SVID3  
**lastlogin**: SVID2, SVID3  
**monacct**: SVID2, SVID3  
**prctmp**: SVID2, SVID3  
**prdaily**: SVID2, SVID3  
**prtacct**: SVID2, SVID3  
**shutacct**: SVID2, SVID3  
**startup**: SVID2, SVID3  
**turnacct**: SVID2, SVID3

**NAME**

AM60Srvr - disk array server daemon

**SYNOPSIS**

/opt/hparray/bin

**DESCRIPTION**

**AM60Srvr** is the server portion of the Array Manager 60 management software. It monitors the operation and performance of the disk array, and services external requests from clients executing disk array commands. **AM60Srvr** monitors disk array performance and status, maintains disk array logs, and allows clients to examine and change disk array configuration.

**AM60Srvr** must be running to allow management of the disk array using the command line utilities. Host I/Os are not dependent on **AM60Srvr** and are serviced regardless of whether it is running or not. Because of its importance in managing the disk arrays, **AM60Srvr** is launched automatically when the system is booted.

**SECURITY CONFIGURATION**

This command is modified for all security configurations.

**Security Behavior/Restrictions**

Use of this command is restricted to authorized users only.

**Command Authorizations**

This command requires the **sysadmin** authorization to successfully execute.

**Privileges**

The command has been modified to support least privilege. The potential privileges possessed by the command and their uses include:

<b>allowdacread</b>	This privilege is raised to provide discretionary read access to the devices.
<b>allowdacwrite</b>	This privilege is raised to provide discretionary write access to the devices.
<b>allowmacread</b>	This privilege is raised to provide mandatory read access to the devices.
<b>allowmacwrite</b>	This privilege is raised to provide mandatory write access to the devices.
<b>filesysops</b>	This privilege is raised to allow the <i>mknod(2)</i> system call to succeed.
<b>writeaudit</b>	The command generates its own audit records and submits these directly to the system audit trail. This privilege is raised whenever the command needs to write an audit record.

**DIAGNOSTICS**

All significant changes in disk array status detected by **AM60Srvr** are entered in */var/adm/syslog/syslog.log*. These entries form a history of disk array operation and can be used to track operation.

**AUTHOR**

**AM60Srvr** was developed by HP.

**FILES**

The following files support the operation of **AM60Srvr**. These files are typically located in */opt/hparray/lib/nls/msg/C/*.

<b>AM60Srvr.cat</b>	Message catalog file
<b>am60cl.cat</b>	Message catalog file for all command line clients
<b>fwerrcod.cat</b>	Message catalog file for command line clients
<b>oemmsg01.cat</b>	Message catalog for OEM-specific messages

**SEE ALSO**

amdsp(1M), amcfg(1M), amutil(1M), amlog(1M), ammgr(1M).

**NAME**

amcfg - manages LUN (logical drive) configuration on the disk array

**SYNOPSIS**

```
amcfg -L cntrlrID:LUN -d channel:ID,channel:ID,...
      {-r RAIDlevel [-c capacity] [-s SegmentSize] [-force]} [-V] [-?] ArrayID

amcfg -R cntrlrID:LUN -d channel:ID,channel:ID,...
      {-r RAIDlevel [-c capacity] [-s SegmentSize] [-force]} [-V] [-?] ArrayID

amcfg -C -d channel:ID,channel:ID,... {-r RAIDlevel [-s SegmentSize]} [-V] [-?] ArrayID

amcfg -D LUN [-V] [-?] ArrayID

amcfg -M LUN -c cntrlrID [-V] [-?] ArrayID
```

**DESCRIPTION**

**amcfg** manages the LUN configuration on the disk array identified by *ArrayID*. This involves creating, deleting, replacing, and assigning ownership of all LUNS on the disk array.

**WARNING:** Deleting a LUN will destroy all the data on the LUN. Backup all vital data before deleting a LUN.

The *ArrayID* used to address the disk array can be the disk array serial number or name, if one has been assigned to the disk array.

**Identifying Disk Modules**

Disk modules are identified within Array Manager 60 using a numbered pair of the form *n:n*. The first number identifies the SCSI channel (or bus) connecting the array controller to the enclosure containing the disk module. The channel number is indicated on the back of the array controller enclosure. The second number is the disk module SCSI ID. The SCSI ID is determined by the slot in which the disk module is installed, but is not the same as the physical slot number (0-9).

For example, the numbered pair **2:1** identifies the disk module on channel 2 with a SCSI ID of 1. Refer to the *Disk Array FC/60 User's Guide* for more information on disk module addressing.

**Options**

**amcfg** supports the following options:

**-c capacity** Indicates the capacity of the LUN being created. The capacity can be specified in megabytes (M), or gigabytes (G) by appending the appropriate letter to the value. If not specified, the default is gigabytes.

By default, the LUN will use the entire capacity available from all the disks. It is possible to specify a lower value for LUN capacity, but this will result in unused disk capacity. Any capacity not included in the LUN will be inaccessible and is essentially wasted capacity.

**-C** Calculate the capacity of a LUN using the specified disks, RAID level, and stripe segment size. This command does not actually create the LUN. It simply returns the capacity available if a LUN is created using the specified values.

**-d channel:ID,channel:ID**

Identifies the disks used for the LUN. Multiple disks can be specified. Each disk is identified by channel number (1-6) and SCSI ID (0-4, 8-12). Note that multiple disks must be specified with no spaces between each disk.

To ensure high-availability, each disk should be in a different disk enclosure. This protects the LUN against an enclosure failure.

**NOTE:** When selecting disks for a RAID 0/1 LUN, the order in which you specify disks is important. The first half of the disks you specify will be the primary data disks, and the second half of the disks will be the disk mirrors. To maintain data availability, the disk mirrors must be in a different enclosure than the data disks.

For example, assume you are creating a 4-disk RAID 0/1 LUN using one disk enclosure on channel 1, and a second disk enclosure on channel 2. Specifying the disks in the order **1:2, 1:3, 2:2, 2:3** would result in mirrored pairs of **1:2/2:2** and **1:3/2:3**. This would ensure availability because the data disks are on channel 1,

a

and the mirror disks are on channel 2. However, specifying disks in the order 1:2, 2:2, 1:3, and 2:3 would result in mirrored pairs of 1:2/1:3 and 2:2/2:3. This would put the data disk and mirror disk of each pair in the same enclosure, making the LUN vulnerable to an enclosure failure.

- D LUN** Delete the LUN identified by *LUN*.
- force** Allows a LUN to be created using two or more disks in the same enclosure. This option allows you to override the high-availability protection designed into the LUN binding process. Using this option you can specify more than one disk per enclosure. You can also use this option to create a RAID 5 LUN that includes more than six disks.
- L cntrlrID:LUN** Create a LUN with the number specified by *LUN*. The LUN will be owned by the controller identified by *cntrlrID*.
- M LUN -c cntrlrID** Change the ownership of the LUN specified by *LUN* to the controller specified by *cntrlrID*.
- r RAIDlevel** Identifies the RAID level used for the LUN. RAID levels that can be specified are 1 and 5. A RAID 0/1 LUN is created by selecting RAID 1 with more than two disks.
- R cntrlrID:LUN** Replace (unbind and rebind) the LUN with the number specified by *LUN* on the controller identified by *cntrlrID*.
- s SegmentSize** Identifies the stripe segment size used for the LUN. The stripe segment size is specified in Kbytes and must be a multiple of the current cache page size setting. For example, if the cache page size is set to 4 Kbytes, valid settings would be 4, 8, 16, 64, etc.
- v** Verbose mode displays additional command execution, state, and/or status messages.
- ?** Display expanded usage message. This option overrides all others.

## SECURITY CONFIGURATION

This command is modified for all security configurations.

### Security Behavior/Restrictions

Use of this command is restricted to authorized users only.

### Command Authorizations

This command requires the **sysadmin** authorization to successfully execute.

### Privileges

The command has been modified to support least privilege. The potential privileges possessed by the command and their uses include:

- allowdacread** This privilege is raised to provide discretionary read access to the devices.
- allowdacwrite** This privilege is raised to provide discretionary write access to the devices.
- allowmacread** This privilege is raised to provide mandatory read access to the devices.
- allowmacwrite** This privilege is raised to provide mandatory write access to the devices.
- filesysops** This privilege is raised to allow the *mknod(2)* system call to succeed.
- writeaudit** The command generates its own audit records and submits these directly to the system audit trail. This privilege is raised whenever the command needs to write an audit record.

## EXTERNAL INFLUENCES

### Environment Variables

**LC\_MESSAGES** determines the language in which messages are displayed. The current language settings can be checked with *locale(1)*.

**RETURN VALUE**

**amcfg** returns the following values:

- 0 Successful completion.
- 1 An error in execution (I/O, subsystem, security, etc.) occurred.
- 2 An error in command syntax occurred: for example, an unknown command-line option was passed.
- 3 Timeout in communication to server. May indicate **AM60Srvr** is not running.
- 4 The state of the disk array has changed since **AM60Srvr** was started. A **RESCAN** should be performed using **amdsp(1M)** to update **AM60Srvr** status.

**DIAGNOSTICS**

The following messages can be generated by **amcfg**:

**Usage:** **amcfg** { **-L** | **-R** } <CntrlrID>:<LUN>  
**-d** <Channel:Id>[,<Channel:Id>...] **-r** <RAIDLevel>  
**[-c** <Capacity>] **[-s** <SegmentSize>] **[-force]**  
**[-V]** **[-S]** **[-?]** <ArrayID>

**amcfg -C -d** <Channel:Id>[,<Channel:Id>...]  
**-r** <RAIDLevel> **[-s** <SegmentSize>]  
**[-V]** **[-S]** **[-?]** <ArrayID>

**amcfg -D** <LUN> **[-V]** **[-S]** **[-?]** <ArrayID>

**amcfg -M** <LUN> **-c** <CntrlrID>  
**[-V]** **[-S]** **[-?]** <ArrayID>

An error in command syntax has occurred. Re-enter the command with all necessary arguments.

**amcfg: The <ArrayID> entered does not identify a known, supported array**  
 The specified *ArrayID* does not exist or does not identify a device that is communicating with the system. Verify the array exists and is operational.

**amcfg: The specified controller is not physically installed.**  
 A controller-specific command was issued to a non-existent or inaccessible controller.

**amcfg: The specified LUN does not exist in this array.**  
 The specified LUN does not exist.

**amcfg: The capacity specified exceeds total available for disks.**  
**LUN not created.**  
 The capacity requested exceeds available capacity for the disks requested.

**amcfg: Unknown capacity subscript: x**  
 A capacity subscript of **M** or **G** was expected, or none at all. Found *x* instead. Re-enter command with correct subscript.

**amcfg: Capacity must be >= 10 MB**  
 An attempt was made to create a LUN with a capacity less than 10 MB. A minimum size of 10 MB is required. Re-issue the command with a larger capacity specification.

**amcfg: LUN already owned by specified controller**  
 An attempt was made to change the ownership of a LUN to the same controller that already owns it. No operation is performed.

**amcfg: LUN 0 may not be deleted. Use -R option to replace LUN 0 instead.**  
 An attempt was made to delete LUN 0, which is not allowed. LUN 0 may, however, be reconfigured using **amcfg -R**. No operation is performed.

**amcfg: LUN's owning controller not responding.**  
**Check controller status or bind LUN to the other controller.**  
 An attempt was made to bind a LUN to a controller which is not responding. No operation is performed.

**amcfg: LUN's new owning controller not responding.**  
**Check controller status or bind LUN to the other controller.**  
 An attempt was made to replace (unbind and rebind) a LUN, but the new owning controller does not respond. No operation is performed.

a

amcfg: The specified operation cannot be performed on a passive controller.  
An attempt was made to bind or replace a LUN on a passive controller.

amcfg: Error in command execution, <Additional Error Info>:  
<Error Info Decode>  
The command failed due to a device error, an internal error, or a system error. The *Additional Error Info* and *Error Info Decode* fields will hold specifics about the failure and its cause.

amcfg: A RAID 5 LUN with more than one disk on a channel is not a recommended high availability configuration. Use -force to override.

amcfg: A RAID 5 LUN with more than one disk in an enclosure is not a recommended high availability configuration. Use -force to override.

amcfg: A RAID 5 LUN with more than six disks in this array is not a recommended high availability configuration. Use -force to override.

amcfg: A RAID 0/1 LUN with both mirrors residing on the same channel is not a recommended high availability configuration. Use -force to override.

amcfg: A RAID 0/1 LUN with both mirrors residing in the same enclosure is not a recommended high availability configuration. Use -force to override.

amcfg: A RAID 0/1 LUN may contain no more than 30 disks.  
The absolute maximum number of disks that the array will support in a RAID 0/1 LUN has been exceeded.

amcfg: A RAID 5 LUN may contain no more than 20 disks.  
The absolute maximum number of disks that the array will support in a RAID 5 LUN has been exceeded.

## EXAMPLES

Bind a 5-disk, RAID 5 LUN on disk array RACK\_51. The disk array includes five disk enclosures, each on its own channel. The capacity of the LUN will default to the total capacity available from the five disks. The LUN is owned by controller A, is assigned number 2, and uses a stripe segment size of 16 Kbytes. Note that each disk is in a different enclosure for high availability, and that there are no spaces between the individual disk parameters.

```
amcfg -L A:2 -d 1:1,2:2,3:1,4:4,5:3 -r 5 -s 16 RACK_51
```

Bind a RAID 0/1 LUN on disk array RACK\_51. The LUN is owned by controller B, is assigned number 4, and uses a stripe segment size of 4 Kbytes. Note that the disks selected create mirrored pairs that are in separate enclosures (1:3, 2:3 and 1:4, 2:4). This maintains high availability. Although RAID 1 is specified, the inclusion of more than two disks causes the disk array to create a RAID 0/1 LUN.

```
amcfg -L B:4 -d 1:3,1:4,2:3,2:4 -r 1 -s 4 RACK_51
```

Calculate the available capacity from a five-disk RAID 5 LUN on disk array RACK\_51.

```
amcfg -C -d 1:2,2:2,3:2,4:6 -r 5 -s 32 RACK_51
```

Delete LUN 3 on the disk array RACK\_51:

```
amcfg -D 3 RACK_51
```

Change the ownership of LUN 0 on disk array RACK\_51 to controller B:

```
amcfg -M 0 -c B RACK_51
```

## DEPENDENCIES

AM60Srvr must be running to execute this command. See *AM60Srvr(1M)*.

## AUTHOR

amcfg was developed by HP.



**SEE ALSO**

amdsp(1M), ammgr(1M), amutil(1M), amlog(1M), AM60Srvr(1M).

**a**

**NAME**

amdload - downloads new firmware to the disk array controllers, the disk system BCC controllers, and disks

**SYNOPSIS**

```
amdload -D { all | channel:ID,channel:ID,... } codefile ArrayID
amdload -C { all | cntrlrID } codefile ArrayID
amdload -S { all | channel,channel, ... } [-force] codefile ArrayID
amdload -i ArrayID
amdload -?
```

**Remarks**

Downloading firmware should only be performed by service-trained personnel. If firmware is not downloaded properly, it may cause the disk array or some its components to become inoperative.

**DESCRIPTION**

**amdload** copies new firmware code to the disk array controllers, the disk system BCC controllers, or the disks in the disk array identified by *ArrayID*. The new code is stored in the file identified by *codefile*.

The *ArrayID* used to address the disk array can be the disk array serial number or name, if one has been assigned to the disk array.

**Identifying Disk Modules**

Disk modules are identified within Array Manager 60 using a numbered pair of the form *n:n*. The first number identifies the SCSI channel (or bus) connecting the array controller to the enclosure containing the disk module. The channel number is indicated on the back of the array controller enclosure. The second number is the disk module SCSI ID. The SCSI ID is determined by the slot in which the disk module is installed, but is not the same as the physical slot number (0-9).

For example, the numbered pair **2:1** identifies the disk module on channel 2 with a SCSI ID of 1. Refer to the *Disk Array FC/60 User's Guide* for more information on disk module addressing.

**Options**

**amdload** supports the following options:

- C Download new firmware to the disk array controllers. The new code is stored in the file identified by *codefile*.  
If the **all** option is included, download firmware to both disk array controllers.  
If the *cntrlrID* option is included, download firmware to the specified controller (A or B).  
NOTE! There are three controller firmware files and they must be downloaded in the proper sequence. When upgrading firmware, download the bootware code file first, followed by the firmware code file, and finally the NVSRAM file.
- D Download new firmware to the disks. The new code is stored in the file identified by *codefile*.  
If the **all** option is included, download firmware to all disks on the disk array.  
If the *channel:ID* option is included, download firmware only to the specified disks. Disk are identified by channel (1-6) and SCSI ID (0-4, 8-12).
- i Display the firmware revisions for all hardware components of the specified disk array.
- S Download new firmware to the disk system BCC controllers. The new code is stored in the file identified by *codefile*.  
If the **all** option is included, download firmware to all disk system BCC controllers on the disk array.  
If the *channel* option is included, download firmware only to the specified disk systems. Disk systems are identified by channel (1-6). Firmware will be downloaded to both BCC controllers on the disk system.

- If the **-force** option is included, non-optimal disk states will be ignored.
- ?** Display expanded usage message. This option overrides all others.

## SECURITY CONFIGURATION

This command is modified for all security configurations.

### Security Behavior/Restrictions

Use of this command is restricted to authorized users only.

### Command Authorizations

This command requires the **sysadmin** authorization to successfully execute.

### Privileges

The command has been modified to support least privilege. The potential privileges possessed by the command and their uses include:

- allowdacread** This privilege is raised to provide discretionary read access to the devices.
- allowdacwrite** This privilege is raised to provide discretionary write access to the devices.
- allowmacread** This privilege is raised to provide mandatory read access to the devices.
- allowmacwrite** This privilege is raised to provide mandatory write access to the devices.
- filesysops** This privilege is raised to allow the *mknod(2)* system call to succeed.
- writeaudit** The command generates its own audit records and submits these directly to the system audit trail. This privilege is raised whenever the command needs to write an audit record.

## EXTERNAL INFLUENCES

### Environment Variables

**LC\_MESSAGES** determines the language in which messages are displayed. The current language settings can be checked with *locale(1)*.

## RETURN VALUE

**amdload** returns the following values:

- 0 Successful completion.
- 1 An error in execution (I/O, subsystem, security, etc.) occurred.
- 2 An error in command syntax occurred: for example, an unknown command-line option was passed.
- 3 Timeout in communication to server. May indicate **AM60Srvr** is not running.
- 4 The state of the disk array has changed since **AM60Srvr** was started. A **RESCAN** should be performed using *amdsp(1M)*. to update **AM60Srvr** status.

## DIAGNOSTICS

The following messages can be generated by **amdload**:

```
amdload -D {all | channel:ID,channel:ID...} codefile <ArrayID>
amdload -C { all | <cntrlrID> } codefile <ArrayID>
amdload -S { all | channel,channel,...} [-force] codefile <ArrayID>
amdload -i <ArrayID>
amdload -?
```

An error in command syntax has occurred. Re-enter the command with all necessary arguments.

### amdload: Arg out of range

One of the arguments has exceeded its maximum or minimum size, or is incorrect in form. Check the size and form of each argument.

### amdload: The <ArrayID> entered does not identify a known, supported array

The specified *ArrayID* does not exist or does not identify a device that is communicating with the system. Verify the array exists and is operational.

a

**amdload:** Firmware download has been aborted.  
 Disk <disk> does not exist in the subsystem.  
 The specified disk does not exist in the subsystem. Re-enter the list of disks, making sure all disks are physically present. A similar message will appear for channels that do not exist.

**amdload:** Firmware download has been aborted.  
 There is more than one type of disk mechanism among the disks selected for update.  
 Any particular firmware file is only valid for one type of disk mechanism. This message will appear if the disks specified for update are of more than one mechanism type. Re-enter the list of disks, only including disks of one mechanism type.

**amdload:** The disk <disk> was repeated in your list.  
 Re-enter the command, listing each item once.  
 This message will appear if a disk or channel is repeated in the input list. Be sure to list each item once.

**amdload:** Firmware download has been aborted.  
 LUN <lun> is not in an Optimal state. Wait until all LUNs are in an Optimal state before attempting further firmware downloads.  
 LUNs must be in an Optimal state before attempting disk or array controller downloads. This message will appear if any LUNs are not Optimal prior to a download, or if a LUN enters a non-Optimal state between disk downloads.

**amdload:** Error in command execution, <Additional Error Info>:  
 <Error Info Decode>  
 The command failed due to a device error, an internal error, or a system error. The *Additional Error Info* and *Error Info Decode* fields will hold specifics about the failure and its cause.

## EXAMPLES

Download new firmware from file **coderev2.1** to both disk array controllers on disk array **RACK\_51**:

```
amdload -C all coderev2.1 RACK_51
```

Download new firmware from file **diskrev3.2** to the specified disks (channel 6, SCSI ID 2 and channel 6, SCSI ID 4) in disk array **RACK\_51**:

```
amdload -D 6:2, 6:4 diskrev3.2 RACK_51
```

Download new firmware from file **bccrev14** to the BCC controllers on all the disk systems on disk array **RACK\_51**:

```
amdload -S all bccrev14 RACK_51
```

## DEPENDENCIES

**AM60Srvr** must be running to execute this command. See *AM60Srvr(1M)*.

## AUTHOR

**amdload** was developed by HP.

## SEE ALSO

**amdsp(1M)**, **amcfg(1M)**, **amutil(1M)**, **amlog(1M)**, **ammgr(1M)**, **AM60Srvr(1M)**.

**NAME**

amdsp - display the status and operating configuration of the disk array

**SYNOPSIS**

```
amdsp [ -l [LUN] | -g | -d | -c | -s | -h | -a | -A | -r ] [-V] [-S] [-?] ArrayID
amdsp -p [-V] [-S] [-?] DeviceFile
amdsp -i [-V] [-S] [-?]
amdsp -R [-V] [-?]
```

**DESCRIPTION**

**amdsp** displays status and configuration information for the disk array identified by *ArrayID*. Logical configuration, physical configuration, and current status can all be displayed using **amdsp**. A list of all the disk arrays recognized by the host can also be displayed.

The *ArrayID* used to address the disk array can be the disk array serial number or name, if one has been assigned to the disk array.

**Identifying Disk Modules**

Disk modules are identified within Array Manager 60 using a numbered pair of the form *n:n*. The first number identifies the SCSI channel (or bus) connecting the array controller to the enclosure containing the disk module. The channel number is indicated on the back of the array controller enclosure. The second number is the disk module SCSI ID. The SCSI ID is determined by the slot in which the disk module is installed, but is not the same as the physical slot number (0-9).

For example, the numbered pair **2:1** identifies the disk module on channel 2 with a SCSI ID of 1. Refer to the *Disk Array FC/60 User's Guide* for more information on disk module addressing.

**Options**

**amdsp** supports the following options:

- |                             |   |
|-----------------------------|---|
| <b>none</b>                 | Display general information about the disk array. This includes product and vendor information, array state, and capacity usage.  |
| <b>-a</b>                   | Display all information presented by the <b>-c</b> , <b>-d</b> , <b>-g</b> , <b>-h</b> , <b>-s</b> , <b>-l</b> , and <b>-r</b> options. This is a quick way of displaying all configuration and status information about the disk array. This option will display information for all LUNs. |
| <b>-c</b>                   | Display information for both disk array controllers and the BCC disk system controllers.  |
| <b>-d</b>                   | Display information for all disks installed in the array. Each disk is identified by a numbered pair of the form <i>channel:ID</i> . The enclosure and slot in which the disk is installed are also displayed.  |
| <b>-g</b>                   | Display disk group information. Information will be displayed for each disk group on the disk array. There is a disk group for each LUN.  |
| <b>-h</b>                   | Display information about the disk array hardware. This includes the hardware components in the disk array subsystem and each disk system.  |
| <b>-i</b>                   | Display the ID of all disk arrays currently connected to and recognized by the server. If the disk array has been assigned a name, it will also be displayed. Because this option is device-independent, it does not use <i>ArrayID</i> .   |
| <b>-l [LUN]</b>             | Display information for the LUN identified by <i>LUN</i> . If <i>LUN</i> is not specified, display information for all LUNs on the disk array.  |
| <b>-p <i>DeviceFile</i></b> | Display hardware path information for the controller corresponding to the specified device file.  |
| <b>-r</b>                   | Display the progress of all rebuilds currently in progress on the disk array.   |
| <b>-R</b>                   | Rescan for disk arrays. The host will scan for all supported disk arrays and update the current list. This may be useful if a new disk array has been added but does not yet show up in the disk array list returned by the <b>-i</b> option.   |
| <b>-s</b>                   | Display disk array state information. This includes cache settings for the disk array.  |

- S Raw output display. Data is output as a colon-delimited ASCII text string. Raw output format is near the end of this page.
- V Verbose mode displays additional command execution, state, and/or status messages.
- ? Display extended usage message. This option overrides all others.

## SECURITY CONFIGURATION

This command is modified for all security configurations.

### Security Behavior/Restrictions

Use of this command is restricted to authorized users only.

### Command Authorizations

This command requires the **sysadmin** authorization to successfully execute.

### Privileges

The command has been modified to support least privilege. The potential privileges possessed by the command and their uses include:

- allowdacread** This privilege is raised to provide discretionary read access to the devices.
- allowdacwrite** This privilege is raised to provide discretionary write access to the devices.
- allowmacread** This privilege is raised to provide mandatory read access to the devices.
- allowmacwrite** This privilege is raised to provide mandatory write access to the devices.
- filesysops** This privilege is raised to allow the *mknod(2)* system call to succeed.
- writeaudit** The command generates its own audit records and submits these directly to the system audit trail. This privilege is raised whenever the command needs to write an audit record.

## EXTERNAL INFLUENCES

### Environment Variables

**LC\_MESSAGES** determines the language in which messages are displayed. The current language settings can be checked with *locale(1)*.

## RETURN VALUE

**amdsp** returns the following values:

- 0 Successful completion.
- 1 An error in execution (I/O, subsystem, security, etc.) occurred.
- 2 An error in command syntax occurred: for example, an unknown command-line option was passed.
- 3 Timeout in communication to server. May indicate **AM60Srvr** is not running.
- 4 The state of the disk array has changed since **AM60Srvr** was started. A **RESCAN** should be performed using the **-R** option to update **AM60Srvr** status.

## DIAGNOSTICS

The following messages can be generated by **amdsp**:

```
amdsp [-l [LUN] | -g | -d | -c |
      -s | -h | -a | -A | -r
      [-V] [-S] [-?] <ArrayID>
```

```
amdsp {-i | -R} [-V] [-S] [-?]
```

```
amdsp -p [-V] [-S] [-?] <DeviceFile>
```

An error in command syntax has occurred. Reenter the command with all necessary arguments.

**amdsp: Arg out of range**

One of the arguments has exceeded its maximum or minimum size, or is incorrect in form. Check the size and form of each argument.

**amdsp: The <ArrayID> entered does not identify a known, supported array**

The specified *ArrayID* does not exist or does not identify a device that is communicating with the system. Verify the array exists and is operational.

**amdsp: Array just powered-on or reset.**

The array was just powered-on or was reset. Reissue the command.

**amdsp: The specified controller is not physically installed.**

A controller-specific command was issued to a non-existent or inaccessible controller.

**amdsp: The specified LUN does not exist in this array.**

The specified LUN does not exist.

**amdsp: No array controllers installed**

The software was unable to identify an installed controller in the array. This may be due to controller failure, or cabling problems. Correct the problem and re-issue the command.

**amdsp: No controller hardware path was found for the given device file**

A controller-specific command was issued to a non-existent or inaccessible controller.

**amdsp: Error in command execution, <Additional Error Info>:  
<Error Info Decode>**

The command failed due to a device error, an internal error, or a system error. The *Additional Error Info* and *Error Info Decode* fields will hold specifics about the failure and its cause.

## EXAMPLES

Display general information about disk array RACK\_51:

```
amdsp RACK_51
```

Display information for LUN 2 on disk array named RACK\_51:

```
amdsp -l 2 RACK_51
```

Display information for all LUNs on disk array RACK\_51:

```
amdsp -l RACK_51
```

Display information for all disks installed in disk array named RACK\_51:

```
amdsp -d RACK_51
```

List the serial numbers of all of the disk arrays recognized by the host:

```
amdsp -i
```

## RAW OUTPUT FORMAT

The contents of the raw output text string for each display option are listed here. Note that the raw output for the **-a** option comprises the individual strings displayed by the **-l**, **-d**, **-c**, **-s**, **-g**, **-A**, and **-h** options. For clarification when evaluating Boolean expressions, **TRUE=1** and **FALSE=0**.

### OPTION RAW OUTPUT DESCRIPTION

**none**     **D:VendorID:ProductID:ArrayID:ServerName:ArrayType:  
MfgProdCode:PhysCapacity:LUNCapacity:  
HotSpareCapacity:UnallocatedCapacity:<ArrayState>:  
<WarnInd>:<ArrayAlias>**

The leading **D** is literal to identify this as a DEFAULT record.

All capacities are in GB where **1GB=1024<sup>3</sup>**, except LUN capacity, which is always in MB in raw output mode, where **1MB=1024<sup>2</sup>**.

**-c**     The following is output for each disk array controller:

**C:Subsystem:CntrlrID:CompState:CntrlrMode:Quiesced:VendorID:  
ModelNum:SerNumber:FWRevision:BootRevision:HPRev:LoopID:ALPA:  
PreferredALPA:CntrlrDate:CntrlrTime:CacheBattAge:ArrayID:NVRAMChecksum**

The following is output for each disk enclosure controller:

**C:Subsystem:CntrlrID:CompState:CntrlrMode:Quiesced:VendorID:  
ModelNum:SerNumber:FWRevision:EncSerNum:EncID:SubsysChannel:  
SubsysScsiID:ThumbwheelSetting:CntrlrDate:CntrlrTime:CacheBattAge:ArrayID**

The leading **C** is literal to identify this as a CONTROLLER record.

The above will be displayed for each controller installed. The first type of record shown above will be displayed for each Subsystem whose value is 0. The second record type will be displayed when the Subsystem is greater than 0.

*Quiesced* is a boolean field indicating whether the controller is quiescent.

*CntrlrMode*, *Quiesced*, *CntrlrDate*, *CntrlrTime*, and *CacheBattAge* are only defined when the *Subsystem* is 0 (array controller enclosure). Otherwise they are undefined.

*EncSerNum* is the serial number of the disk subsystem enclosure.

*CntrlrTime* is a colon delimited field having a format of *HH:MM:SS*.

If *CompState* is not **GOOD** (1), or *CntrlrMode* is **FAILED** (4), information in the remaining fields may be undefined or absent.

*ALPA* and *PreferredALPA* are displayed in hexadecimal.

*CacheBattAge* is the time in days (rounded to the nearest 90 days) since the cache battery age was last reset.

-d The following output is provided for each installed disk:

```
PD:Channel:ID:EnclosureID:SlotID:DiskState:DiskGroupID:DGType:
Capacity:Manufacturer:Model:FWRevision:SerialNum:ArrayID
```

OR

```
PD:Channel:ID:EnclosureID:SlotID:DiskState:DiskGroupID:DGType:
Capacity:Manufacturer:Model:FWRevision:SerialNum:ArrayID:
SparedChannel:SparedID
```

The leading **PD** is literal to identify this as a PHYSICAL DISK record.

The above will be displayed for each disk slot. For disks which have been bound to a LUN and then removed, or assigned as a hot spare and then removed, the *Manufacturer*, *Model*, *FWRevision* and *SerialNum* fields may be undefined.

Unassigned disks which are removed will have no PD record.

Capacity is displayed in GB, rounded up to the nearest GB.

If *DGType* is 1 (HOT SPARE), and if this disk is currently sparing another disk, then two extra fields will be printed, designating the channel and SCSI ID of the spared disk.

-g One or more LUN records will be displayed in the following format:

```
G:VendorID:ProductID:<DiskGroupID>:<DGType>:
NumLuns:<LUN>...:RemCapacity:RAIDLevel:SegmentSize:nDisks:
Channel-ID,Channel-ID...:<ArrayID>
```

The leading **G** is literal to identify this as a Disk Group record.

The above fields will be displayed for each disk group.

The following fields are undefined when *DGType* is 1 or 2: *NumLUNs*, *LUN*, *RemCapacity*, *RAIDLevel*, and *SegmentSize*.

Note that capacity will always be expressed in MB for raw output.

-h One or more "component state records" will be displayed in the following format:

```
H:VendorID:ProductID:<SubSystem>:CntrlrAStatus:CntrlrBStatus:
PS1Status:PS2Status:Fan1Status:Fan2Status:TsensorStatus:
BattStatus:EncSerNum:ArrayID
```

OR

```
H:VendorID:ProductID:SubSystem:CntrlrAStatus:CntrlrBStatus:
PwrSupplyAStatus:PwrSupplyBStatus:FanAStatus:FanBStatus:
TsensorStatus:BattStatus:EncSerNum:ArrayID
```

The leading **H** is literal to identify this as a Hardware Status record.

When the *SubSystem* is greater than 0 (disk enclosure), the *BattStatus* is undefined.



When the *SubSystem* is greater than 0 (disk enclosure), the power supplies and fans are designated as A and B, rather than 1 and 2. The left to right orientation of the disk enclosure power supplies may not match that of the array controller power supplies.

*EncSerNum* is the disk enclosure serial number, and is only defined when the *SubSystem* is greater than 0; otherwise, it is undefined.

- i The following output includes a separate field (*ArrayID:Alias*) for each disk array identified:

**I:ArrayID:Alias:ArrayID:Alias:...**

The leading **I** is literal to identify this as an Identify record.

- l **L:VendorID:ProductID:LunNumber:Present:LunState:  
CacheState:Capacity:Owner:RaidLevel:SegmentSize:  
DiskGroup:nDisks:Channel-ID,Channel-ID,...:ArrayID:  
CME:CWOB:RCD:WCA:RCA**

The leading **L** is literal to identify this as a LUN record.

*Present* is a Boolean, TRUE if the LUN is configured, FALSE otherwise.

If no option is specified, the above will be displayed for all LUNs, both present and not present.

*CacheState* is a two-bit field in which the high order bit is the boolean *CacheMirroringActive* (CMA) state of the LUN's owning controller, and the low order bit is the boolean *WriteCacheEnabled* (WCE) state for the controller.

*CME* (Cache Mirroring Enabled), *CWOB* (Cache Without Batteries), *RCD* (Read Cache Disabled), *WCA* (Write Cache Active), and *RCA* (Read Cache Active) indicate the state of the caching parameters.

Capacities displayed in the raw output are always shown in MB, where **1MB=1024<sup>2</sup>**.

- p **P:CntrlrID:SerNumber:<ArrayID>  
P:AltCntrlrID:AltSerNumber:<ArrayID>**

The leading **P** is literal to identify this as a HARDWARE PATH record.

The first hardware path record is for the controller that corresponds to the given device file. The second record is for the alternate controller. If the alternate controller is not present, its serial number will be 0.

- r **R:VendorID:ProductID:CntrlrID:LUN:RebuildProgress:Freq:  
Amt:ArrayID**

The leading **R** is literal, and identifies this as a LUN rebuild record.

One rebuild record is returned for each LUN that is currently rebuilding. If no LUN is rebuilding, no record will be returned.

*RebuildProgress* above is percent complete

*Freq* is the rebuild frequency in tenths of a second

*Amt* is the rebuild amount in blocks per rebuild command

- s **S:VendorID:ProductID:ValidData-A:ValidData-B:  
CacheBlkSize-A:CacheBlkSize-B:CacheFlushThresh-A:  
CacheFlushThresh-B:CacheFlushLim-A:CacheFlushLim-B:  
CacheSize-A:CacheSize-B:ArrayState:<ArrayID>**

The leading **S** is literal to identify this as an ARRAY STATE record.

## DEPENDENCIES

**AM60Srvr** must be running to execute this command. See *AM60Srvr(1M)*.

## AUTHOR

**amdsp** was developed by HP.

## SEE ALSO

*ammgr(1M)*, *amcfg(1M)*, *amutil(1M)*, *amlog(1M)*, *AM60Srvr(1M)*.

**NAME**

amfmt - disk format, not supported

**DESCRIPTION**

The **amfmt** command is a data destructive command and is not supported. If an attempt is made to execute the **amfmt** command, data may be destroyed.

**AUTHOR**

**amfmt** was developed by HP.

**SEE ALSO**

ammgr(1M), amutil(1M), amcfg(1M), amdsp(1M), amlog(1M), amdload(1M), AM60Srvr(1M).

**a**

**NAME**

amlog - displays host-based controller log entries for a disk array

**SYNOPSIS**

```
amlog [-s StartTime] [-e EndTime] [-d LogDir] [-S] [-a ArrayID]
amlog -?
```

**DESCRIPTION**

**amlog** displays the contents of the disk array controller logs maintained by the host. These logs contain information useful for diagnosing and troubleshooting the disk array. The host maintains multiple log files containing entries for each disk array.

The *ArrayID* used to address the disk array must be the disk array ID. An alias name cannot be used because alias names are not recorded in the log.

**Identifying Disk Modules**

Disk modules are identified within Array Manager 60 using a numbered pair of the form *n:n*. The first number identifies the SCSI channel (or bus) connecting the array controller to the enclosure containing the disk module. The channel number is indicated on the back of the array controller enclosure. The second number is the disk module SCSI ID. The SCSI ID is determined by the slot in which the disk module is installed, but is not the same as the physical slot number (0-9).

For example, the numbered pair **2:1** identifies the disk module on channel 2 with a SCSI ID of 1. Refer to the *Disk Array FC/60 User's Guide* for more information on disk module addressing.

**Options**

**amlog** supports the following options:

- a *ArrayID* Identify a specific disk array for which the logs will be displayed. The default is to display the logs for all disk arrays.
- d *LogDir* Specify the path name of the log directory. The default is `/var/opt/hparray/log`.
- e *EndTime* Specify the ending date and time. Log records with a later date and time will not be printed. The default is the time of the last log record. Uses the same format as *StartTime*.
- s *StartTime* Specify the starting date and time. Log records with an earlier date and time will not be printed. The default is the time of the oldest log record. The format for entering the date and time is *MMddhhmm[yy]*, where
  - MM* = Month (01-12)
  - dd* = Day (01-31)
  - hh* = Hour (00-23)
  - mm* = Minute (00-59)
  - yy* = Year (00-99) [optional]. Years earlier than 90 are interpreted as 2000 + *YY*.
- S Raw output display. Data is output as a colon-delimited ASCII text string. Raw output format is described below.
- ? Display expanded usage message. This option overrides all others.

**SECURITY CONFIGURATION**

This command is modified for all security configurations.

**Security Behavior/Restrictions**

Use of this command is restricted to authorized users only.

**Command Authorizations**

This command requires the **sysadmin** authorization to successfully execute.

**Privileges**

The command has been modified to support least privilege. The potential privileges possessed by the command and their uses include:

a

<b>allowdacread</b>	This privilege is raised to provide discretionary read access to the devices.
<b>allowdacwrite</b>	This privilege is raised to provide discretionary write access to the devices.
<b>allowmacread</b>	This privilege is raised to provide mandatory read access to the devices.
<b>allowmacwrite</b>	This privilege is raised to provide mandatory write access to the devices.
<b>filesysops</b>	This privilege is raised to allow the <i>mknod(2)</i> system call to succeed.
<b>writeaudit</b>	The command generates its own audit records and submits these directly to the system audit trail. This privilege is raised whenever the command needs to write an audit record.

**EXTERNAL INFLUENCES****Environment Variables**

**LC\_MESSAGES** determines the language in which messages are displayed. The current language settings can be checked with *locale(1)*.

**RETURN VALUE**

**amlog** returns the following values:

- 0 Successful completion.
- 1 An error in execution (I/O, subsystem, security, etc.) occurred.
- 2 An error in command syntax occurred: for example, an unknown command-line option was passed.

**DIAGNOSTICS**

The following message can be generated by **amlog**:

**Usage:** **amlog** [-s <StartTime>] [-e <EndTime>]  
 [-d <LogDir>] [-S] [-a <ArrayID>]

**amlog -?**

An error in command syntax has occurred. Re-enter the command with all necessary arguments.

**EXAMPLES**

Display the controller log for disk array 000200A0B805E798. Display the log entries entered since 6/1/99.

```
amlog -s 0601000099 -a 000200A0B805E798
```

Display the controller log for all disk arrays on the host. Use the default settings to display all log entries.

```
amlog
```

**RAW OUTPUT FORMAT**

The contents of the raw output text string for log entries is as follows:

```
AL:LogFilename:AddSenAvailable:SenseDate:SenseTime:FRUCode:  
FRUCodeQual:SK:ASC:ASCQ:ArrayID:LogDate:LogTime:LUN
```

One record will be printed for each log entry between the start and end times.

*LogFilename* is the name of the server log file where the record is found.

The leading **AL** is literal, and identifies this as an AM60 LOG record.

*AddSenAvailable* is a boolean that indicates whether the *SenseDate*, *SenseTime* and *FRUCodeQual* are available from the SCSI sense data that generates these fields. If FALSE (0), these fields will be undefined.

*SenseDate* and *SenseTime* correspond to the time of the reported event, while *LogDate* and *LogTime* correspond to the time when the event was written to the log file. The format of the *SenseDate* is *MMDDYY*, while that of *LogDate* is *MMDDYYYY*. The format for the *SenseTime* and *LogTime* is *HHMMSS*.

The SCSI sense data fields, *FRUCode*, *SK*, *ASC*, *ASCQ*, are one byte hexadecimal; the *FRUCodeQual* field is two bytes hexadecimal.

If the LUN information in the sense data is valid, it will be included.

**DEPENDENCIES**

There are no dependencies for this command.

**AUTHOR**

**amlog** was developed by HP.

**SEE ALSO**

amdsp(1M), amcfg(1M), amutil(1M), ammgr(1M), AM60Srvr(1M).

  
**a**

**NAME**

ammgr - manages the operating characteristics of the disk array

**SYNOPSIS**

**ammgr** **-D** *ArrayAlias* **[-V]** **[-?]** *ArrayID*

**ammgr** [*Options*] **[-V]** **[-?]** *ArrayID*

**DESCRIPTION**

**ammgr** manages the operating characteristics of the disk array by providing access to the settings used to control disk array operation. In most cases, these settings control the operation of the entire disk array; consequently, every LUN on the disk array will be affected by any changes made using this command. The exception is a parity scan, which impacts only the specified LUN.

The *ArrayID* used to address the disk array can be the disk array serial number or name, if one has been assigned to the disk array.

**Identifying Disk Modules**

Disk modules are identified within Array Manager 60 using a numbered pair of the form *n:n*. The first number identifies the SCSI channel (or bus) connecting the array controller to the enclosure containing the disk module. The channel number is indicated on the back of the array controller enclosure. The second number is the disk module SCSI ID. The SCSI ID is determined by the slot in which the disk module is installed, but is not the same as the physical slot number (0-9).

For example, the numbered pair **2:1** identifies the disk module on channel 2 with a SCSI ID of 1. Refer to the *Disk Array FC/60 User's Guide* for more information on disk module addressing.

**Options**

**ammgr** supports the following options:

- b** Reset battery age to zero. This should be done when the battery is replaced.
- d channel:ID** Remove the role of hot spare for the disk at address *channel:ID*. The disk is identified by channel number (1-6) and SCSI ID (0-4, 8-12).
- D ArrayAlias** Assigns the text string specified by *ArrayAlias* to the disk array. This name is another mechanism for identifying the disk array when executing a command. The name can be up to 16 characters in length and can include letters, numbers, number sign (#), underscore (\_), and period (.).

Disk names can be used in a variety of ways to help identify disk arrays in large systems. For example, by assigning numbers to racks and to the shelf positions within the racks, each disk array can be uniquely identified using an appropriate alias. If a rack is assigned number 12, the disk array installed on shelf 3 of the rack could be identified using an alias of 12\_03. This technique simplifies locating the disk array should it need service.

- h channel:ID** Assign the role of hot spare to the disk at address *channel:ID*. The disk is identified by channel number (1-6) and SCSI ID (0-4, 8-12).
- H LUN** Halt a parity scan currently in progress on the specified LUN. If a parity scan is not in progress on the specified LUN, clear the status of the last completed parity scan for the LUN.

- L cntrlrID:percent** Sets the cache flush limit for the controller identified by *cntrlrID* to the value specified by *percent*. When flushing write cache pages to the disk media, the flush operation will stop when the indicated percentage is reached. The value is expressed as a percentage of the current cache flush threshold.

- p {4|16}** Set the cache page size to 4 KB or 16 KB. The cache page size is set for both disk array controllers. NOTE: The cache page size may not be set to 16 KB unless the segment sizes of all existing LUNs are multiples of 16 KB.

- P LUN** Perform a parity scan on the specified LUN.

NOTE: If errors are detected during a parity scan, it is recommended that you contact your Hewlett-Packard service representative immediately. The occurrence of parity errors may indicate a potential problem with the disk array hardware.

- s *LUN* Show the status of the parity scan in progress on the specified LUN. The final status continues to be displayed until cleared using the -H option.
- t Set the date and time stamp for both disk array controllers to match that of the host.
- T *cntrlrID:percent* Sets the cache flush threshold for the controller identified by *cntrlrID* to the value specified by *percent*. When the contents of the write cache reaches the indicated percentage, pages are flushed (written) to the disk media.
- V Verbose mode displays additional command execution, state, and/or status messages.
- ? Display extended usage message. This option overrides all others.

## SECURITY CONFIGURATION

This command is modified for all security configurations.

### Security Behavior/Restrictions

Use of this command is restricted to authorized users only.

### Command Authorizations

This command requires the **sysadmin** authorization to successfully execute.

### Privileges

The command has been modified to support least privilege. The potential privileges possessed by the command and their uses include:

- allowdacread** This privilege is raised to provide discretionary read access to the devices.
- allowdacwrite** This privilege is raised to provide discretionary write access to the devices.
- allowmacread** This privilege is raised to provide mandatory read access to the devices.
- allowmacwrite** This privilege is raised to provide mandatory write access to the devices.
- filesysops** This privilege is raised to allow the *mknod(2)* system call to succeed.
- writeaudit** The command generates its own audit records and submits these directly to the system audit trail. This privilege is raised whenever the command needs to write an audit record.

## EXTERNAL INFLUENCES

### Environment Variables

**LC\_MESSAGES** determines the language in which messages are displayed. The current language settings can be checked with *locale(1)*.

## RETURN VALUE

**ammgr** returns the following values:

- 0 Successful completion.
- 1 An error in execution (I/O, subsystem, security, etc.) occurred.
- 2 An error in command syntax occurred: for example, an unknown command-line option was passed.
- 3 Timeout in communication to server. May indicate **AM60Srvr** is not running.
- 4 The state of the disk array has changed since **AM60Srvr** was started. A **RESCAN** should be performed using *amdsp(1M)* to update **AM60Srvr** status.

## DIAGNOSTICS

The following messages can be generated by **ammgr**:

```
Usage: ammgr -D <ArrayAlias> | -b
      -h <Channel:ID> | -d <Channel:ID> |
      -T <CntrlrID>:<percent> |
      -L <CntrlrID>:<percent> |
      -p <pagesize> | -P <LUN> |
      -H <LUN> | -s <LUN> | -t
      [-S] [-V] [-?] <ArrayID>
```

An error in command syntax has occurred. Re-enter the command with all necessary arguments.

**ammgr: Arg out of range**

One of the arguments has exceeded its maximum or minimum size, or is incorrect in form. Check the size and form of each argument.

**ammgr: The <ArrayID> entered does not identify a known, supported array**

The specified *ArrayID* does not exist or does not identify a device that is communicating with the system. Verify the array exists and is operational.

**ammgr: The specified controller is not physically installed.**

A controller-specific command was issued to a non-existent or inaccessible controller.

**ammgr: The specified LUN does not exist in this array.**

The specified LUN does not exist.

**ammgr: Invalid Array alias. Alias name characters must be**

alphanumeric, pound (#), underscore (\_), or dot (.)

Illegal syntax for <ArrayID>.

**ammgr: Invalid array alias. Another array already has the specified alias.**

Another array on the host already has the specified alias. Duplicate aliases are not allowed.

**ammgr: Invalid array alias. Alias name cannot be longer than 16 characters.**

An attempt was made to set an array alias with more than the maximum allowed number of characters.

**ammgr: A parity scan is already in progress on the specified LUN.**

An attempt was made to initiate a parity scan on a LUN which already has a parity scan in progress.

**ammgr: No parity scan in progress on the specified LUN.**

An attempt was made to halt a parity scan on a LUN which has no parity scan in progress.

**ammgr: Cache page size may not be set to 16 KB unless the segment sizes of all existing LUNs are multiples of 16 KB.**

Illegal cache page size requested. Segment sizes of existing LUNs must be multiples of the requested cache page size.

**ammgr: Error in command execution, <Additional Error Info>:  
<Error Info Decode>**

The command failed due to a device error, an internal error, or a system error. The *Additional Error Info* and *Error Info Decode* fields will hold specifics about the failure and its cause.

## EXAMPLES

Assign a name to the disk array identified with ID 0000005EBD20. Use an alias that identifies the rack the disk array is installed in (51 for this example), and the shelf position the disk array is on (03 for this example):

```
ammgr -D RACK_51 0000005EBD20
```

Add a hot spare to disk array RACK\_51 using the disk on channel 3 with SCSI ID 4:

```
ammgr -h 3:4 RACK_51
```

Initiate a parity scan of LUN 1 on disk array RACK\_51:

```
ammgr -P 1 RACK_51
```

Set the cache flush threshold to 80% for controller A on disk array RACK\_51:

```
ammgr -T A:80 RACK_51
```

## DEPENDENCIES

AM60Srvr must be running to execute this command. See *AM60Srvr(1M)*.

## AUTHOR

ammgr was developed by HP.



**SEE ALSO**

amdsp(1M), amcfg(1M), amutil(1M), amlog(1M), AM60Srvr(1M).

**a**

**NAME**

amutil - controls various disk array management functions

**SYNOPSIS**

```
amutil { [ -f channel:ID,channel:ID... ] | [ -F ] | [ -s ] | [ -R LUN -f Freq -a Amt ] |
        [ -l ] } [ -v ] ArrayID

amutil -?
```

**DESCRIPTION**

**amutil** controls a variety of management operations on the disk array identified by *ArrayID*. Locating hardware components and managing the rebuild process are done using **amutil**.

The *ArrayID* used to address the disk array can be the disk array serial number or name, if one has been assigned to the disk array.

**Identifying Disk Modules**

Disk modules are identified within Array Manager 60 using a numbered pair of the form *n:n*. The first number identifies the SCSI channel (or bus) connecting the array controller to the enclosure containing the disk module. The channel number is indicated on the back of the array controller enclosure. The second number is the disk module SCSI ID. The SCSI ID is determined by the slot in which the disk module is installed, but is not the same as the physical slot number (0-9).

For example, the numbered pair **2:1** identifies the disk module on channel 2 with a SCSI ID of 1. Refer to the *Disk Array FC/60 User's Guide* for more information on disk module addressing.

**Options**

**amutil** supports the following options:

**-f** *channel:ID,channel:ID...*

Flash the amber LED on the specified disks. Each disk is identified by *channel:ID*. *Channel:ID* is determined by the disk enclosure channel number (1-6) and disk SCSI ID (0-4, 8-12).

**-F**

Flash the amber LEDs on all the disks in the disk array.

**-l**

Flush server log file. This will retrieve the current log entries from the disk array controller.

**-R** *LUN -f req -a Amt*

Set the rebuild priority settings of the specified LUN. The *freq* value identifies the rate at which the disk array attempts to execute rebuild commands. Specified in tenths of a second, this value can be 1 to 50, or 0.1 seconds to 5.0 seconds. A low setting increases the frequency at which rebuild commands are issued, giving higher priority to the rebuild but reducing I/O performance. A high value reduces the rebuild command frequency, giving higher priority to host I/Os. The default value for this setting is 1, or 0.1 seconds.

The *Amt* value identifies the number of blocks to rebuild at a time. This value can be from 1 to 64K and specifies the number of 512-byte blocks processed during each rebuild command. The higher the setting the more blocks which will be processed, thus reducing I/O performance. A lower setting gives priority to host I/Os, delaying the completion of the rebuild. The default value for this setting is 64 blocks, or 32 Kbytes of data.

**-s**

Stop flashing disk activity lights. This option is used to stop the flashing on all disks.

**-v**

Verbose mode displays additional command execution, state, and/or status messages.

**-?**

Display extended usage message. This option overrides all others.

**SECURITY CONFIGURATION**

This command is modified for all security configurations.

**Security Behavior/Restrictions**

Use of this command is restricted to authorized users only.

**Command Authorizations**

This command requires the **sysadmin** authorization to successfully execute.

**Privileges**

The command has been modified to support least privilege. The potential privileges possessed by the command and their uses include:

<b>allowdacread</b>	This privilege is raised to provide discretionary read access to the devices.
<b>allowdacwrite</b>	This privilege is raised to provide discretionary write access to the devices.
<b>allowmacread</b>	This privilege is raised to provide mandatory read access to the devices.
<b>allowmacwrite</b>	This privilege is raised to provide mandatory write access to the devices.
<b>filesysops</b>	This privilege is raised to allow the <i>mknod(2)</i> system call to succeed.
<b>writeaudit</b>	The command generates its own audit records and submits these directly to the system audit trail. This privilege is raised whenever the command needs to write an audit record.

**EXTERNAL INFLUENCES****Environment Variables**

**LC\_MESSAGES** determines the language in which messages are displayed. The current language settings can be checked with *locale(1)*.

**RETURN VALUE**

**amutil** returns the following values:

- 0 Successful completion.
- 1 An error in execution (I/O, subsystem, security, etc.) occurred.
- 2 An error in command syntax occurred: for example, an unknown command line option was passed.
- 3 Timeout in communication to server. May indicate the **AM60Srvr** is not running.
- 4 The state of the disk array has changed since **AM60Srvr** was started. A **RESCAN** should be performed using *amdsp(1M)* to update **AM60Srvr** status.

**DIAGNOSTICS**

The following messages can be generated by **amutil**:

```
amutil {[-f Channel:Id] | [-F] | [-s] |
-R <LUN> -f <Freq> -a <Amt> | -l ]
[-V] <ArrayID>}
```

**amutil -?**

An error in command syntax has occurred. Reenter the command with all necessary arguments.

**amutil: Arg out of range**

One of the arguments has exceeded its maximum or minimum size, or is incorrect in form. Check the size and form of each argument.

**amutil: The <ArrayID> entered does not identify a known, supported array**

The specified *ArrayID* does not exist or does not identify a device that is communicating with the system. Verify the array exists and is operational.

**amutil: The specified controller is not physically installed.**

A controller-specific command was issued to a non-existent or inaccessible controller.

**amutil: The specified LUN does not exist in this array.**

The specified LUN does not exist.

**amutil: Error in command execution, <Additional Error Info>:  
<Error Info Decode>**

The command failed due to a device error, an internal error, or a system error. The *Additional Error Info* and *Error Info Decode* fields will hold specifics about the failure and its cause.

**EXAMPLES**

Flash the amber LEDs on all the disks in disk array RACK\_51:

```
amutil -F RACK_51
```

Stop flashing the amber LEDs on disk array RACK\_51:

```
amutil -s RACK_51
```

The following example alters the rebuild priority settings. It assigns a value of 5 seconds to the rebuild command rate, and sets the data block amount to 16 blocks on LUN 4 on disk array RACK\_51. This gives host I/Os higher priority than the default settings.

```
amutil -R 4 -f 50 -a 16 RACK_51
```

**DEPENDENCIES**

**AM60Srvr** must be running to execute this command. See *AM60Srvr(1M)*.

**AUTHOR**

**amutil** was developed by HP.

**SEE ALSO**

*ammgr(1M)*, *amcfg(1M)*, *amdsp(1M)*, *amlog(1M)*, *AM60Srvr(1M)*.

**NAME**

ARMServer - disk array server daemon

**SYNOPSIS**

/opt/hparray/bin

**DESCRIPTION**

**ARMServer** is the server portion of the HP Advanced Disk Array management software. It monitors the operation and performance of the disk array, and services external requests from clients executing disk array commands. **ARMServer** monitors disk array performance and status, maintains disk array logs, initiates diagnostics, and allows clients to examine and change disk array configuration.

**ARMServer** must be running to allow management of the disk array using the command line utilities. Host I/Os are not dependent on **ARMServer** and are serviced regardless of whether it is running or not. Because of its importance in managing the disk arrays, **ARMserver** is launched automatically when the system is booted.

One of the most important functions provided by **ARMServer** is retrieving and storing performance and status information from the disk array logs. If **ARMServer** is not running, the disk array will continue to log data, but when the array logs are full older entries will be lost to make room for new entries. A "Log Full" warning will be displayed if this condition occurs, alerting you to the fact that **ARMServer** is not running.

**DIAGNOSTICS**

All significant changes in disk array status detected by **ARMServer** are entered in /var/adm/syslog/syslog.log. These entries form a history of disk array operation and can be used to track operation.

**DEPENDENCIES**

The **arraymond** disk array monitor requires **ARMServer** to be running to check the status of the disk array. **arraymond** uses the **arraydsp** utility to retrieve status information about the disk array at regular intervals. When performing maintenance on the disk array (for example, downloading new firmware code), **arraymond** can be turned off to avoid error messages generated because the disk array is not accessible. To turn off **arraymond** and **ARMServer** use the following command line:

```
/sbin/init.d/hparray stop
```

To restart **arraymond** and **ARMServer** use the following command line:

```
/sbin/init.d/hparray start
```

**VIRTUALVAULT SECURITY INFORMATION****Security Configuration**

This command is modified for all security configurations.

**Security Behavior/Restrictions**

Use of this command is restricted to authorized users only.

**Command Authorizations**

This command requires the **sysadmin** authorization to successfully execute.

**Privileges**

The command has been modified to support least privilege. The potential privileges possessed by the command and their uses include:

**allowdacread** This privilege is raised to provide discretionary read access to the devices.

**allowdacwrite** This privilege is raised to provide discretionary write access to the devices.

**allowmacread** This privilege is raised to provide mandatory read access to the devices.

**allowmacwrite** This privilege is raised to provide mandatory write access to the devices.

**filesysops** This privilege is raised to allow the **mknod(2)** system call to succeed.

**writeaudit** The command generates its own audit records and submits these directly to the system audit trail. This privilege is raised whenever the command needs to write an audit record.

**AUTHOR**

**ARMServer** was developed by HP.

**FILES**

The following files support the operation of **ARMServer**. These files are typically located in `/opt/hparray/lib/nls/msg/C/`:

**ARMServer.cat** Message catalog file  
**hpraidcl.cat** Message catalog file for all command line clients  
**fwerrcod.cat** Message catalog file for command line clients  
**oemmsg01.cat** Message catalog for OEM-specific messages

**SEE ALSO**

`arraycfg(1M)`, `arraydsp(1M)`, `arrayfmt(1M)`, `arraylog(1M)`, `arraymgr(1M)`, `arrayrbld(1M)`, `arrayrecover(1M)`, `download(1M)`, `drivetest(1M)`, `dteststat(1M)`, `logprint(1M)`.

**NAME**

arp - address resolution display and control

**SYNOPSIS**

```
arp hostname
arp -a[n] [system] [core]
arp [-d | -D] hostname
arp -f filename
arp -s hostname hw_address [temp] [pub] [rif rif_address]
arp -sfc hostname nport_id
```

**DESCRIPTION**

The **arp** command displays and modifies the Internet-to-Ethernet and Internet-to-Fibre Channel address translation tables used by the Address Resolution Protocol (ARP).

**Options**

**arp** has the following keyletter options:

- hostname* (first form above) Display the current ARP entry for *hostname*, which must appear in the *hostname* database (see *hosts*(4)), or for the DARPA Internet address expressed in Internet standard "dot" notation.
- a[*n*]** Display all current ARP entries by reading the table from file *core* (default */dev/kmem*) based on the kernel file *system* (default */stand/vmunix*). The **-an** option provides the same information as the **-a** option except that network addresses are not displayed symbolically.
- d** If an ARP entry exists for the host called *hostname*, delete it. This option cannot be used to delete a permanent ARP entry whose IP address is an interface on the local system.
- D** (Not recommended). Delete a permanent ARP entry whose IP address is an interface on the local system. The removal of such an ARP entry may result in loss or limitation of network connectivity with remote machines. The local system will no longer respond to ARP requests for this IP address. Consequently, communication with remote systems is possible only when that communication is initiated by the local system. This option should be used with extreme caution.
- f** Read file *filename* and set multiple entries in the ARP tables. Fibre Channel entries in the file should be of the form:  

```
-sfc hostname nport_id
```

Other entries in the file should be of the form:

```
hostname hw_address
[temp]
[pub]
[rif
rif_address]
```

The argument meanings are the same as for the **-s** option.

- s** Create an ARP entry for the host called *hostname* with the hardware station address *hw\_address*. The hardware station address is given as six hexadecimal bytes separated by colons. If an ARP entry already exists for *hostname*, the existing entry is updated with the new information.

The entry is permanent unless the word **temp** is given in the command.

If the word **pub** is specified, the entry is published, which means that this system will act as an ARP server responding to requests for *hostname* even though the host address is not its own.

The word **rif** specifies source routing information used for token ring networks. This information allows you to specify the particular bridge route which the token ring packet should be delivered. *rif\_address* is given as an even number of hexadecimal bytes

separated by colons, up to a maximum of 16 bytes.

**-sfc** Create a permanent ARP entry for the Fibre Channel host called *hostname* with the N\_Port address *nport\_id*. The N\_Port address is given as three hexadecimal bytes separated by colons. If an ARP entry already exists for *hostname*, the existing entry is updated with the new information.

You need superuser privilege to use the **-d**, **-D**, **-f**, **-s** and **-sfc** options.

#### AUTHOR

**arp** was developed by HP and the University of California, Berkeley.

#### SEE ALSO

ifconfig(1M), inet(3N), hosts(4), arp(7P).



**NAME**

arraycfg - configures LUNs (logical drives) and physical disks

**SYNOPSIS**

```
arraycfg -L LUN { -a capacity | -d | -r newLUN } [-v] [-?] array-id
```

```
arraycfg -D slot { -a | -d [-v] [-R|-Z] } [-V] [-?] array-id
```

**DESCRIPTION**

**arraycfg** is used to manage both the logical and physical configuration of the disk array identified by *array-id*. Logical configuration involves creating, deleting, and renumbering LUNS (logical drives) on the disk array. Physical configuration involves including or downing disks from the array configuration.

**WARNING:** Deleting a LUN will destroy all the data on the LUN. Backup all vital data before deleting a LUN.

The *array-id* used to address the disk array can be the disk array serial number, the character device file name of any LUN on the array (LUN 0 if no LUNs are created), or the alias text string assigned to the disk array.

**Options**

**arraycfg** supports the following options:

- a *capacity* If **-L** is specified, create the indicated LUN. The *capacity* parameter indicates the size of the LUN. By default, the size is specified in Mbytes. The size can also be specified in Gbytes by appending a "G" to the value, or in Kbytes by appending a "K" to the value.  
If **-D** is specified, include the indicated disk in the array configuration. When performing this operation, *capacity* is not included.
- d [ -v ] [ -R | -Z ]  
If **-L** is specified, delete the indicated LUN. The remaining options are not used.  
If **-D** is specified, bring down the indicated disk. When downing a disk, the following supplemental options are used
  - v Specifies whether the down operation is destructive or nondestructive. This determines whether or not the disk array will assume there is any valid data on the disk when it is returned to the array configuration. If the **-v** option is included, the down is nondestructive and the array assumes any data on the disk that was not updated in the disk's absence is valid. By default, the array performs a destructive down and assumes no valid data remains on the disk when it is returned to the array configuration.
  - R Allows the disk to be downed even if a loss of redundancy would result, but *not* data unavailability. When using this option, the disk array will continue to offer full data availability but the disk array is not protected against another disk failure. If **-R** is not included, the command to down a disk will fail if it would result in a loss of redundancy.
  - Z Allows the disk to be downed even if data unavailability would occur. If **-Z** is not included, the command to down a disk will fail if it would result in data unavailability.

**WARNING:** Specifying the **-Z** option could cause the disk array to become unavailable.
- D *slot* Perform a physical configuration operation on the disk installed in the cabinet slot identified by *slot*. Slot numbers must be of the form *An* or *Bn*, where *A* or *B* correspond to a cabinet column, and *n* corresponds to a shelf position (1-6).
- L *LUN* Perform a logical configuration operation on the LUN identified by *LUN*.
- r *newLUN* Renumber the LUN specified by *LUN* to the value specified by *newLUN*. This option is only used with the **-L** option.
- v Verbose mode displays additional command execution, state, and/or status messages.
- ? Display expanded usage message. This option overrides all others.

**EXTERNAL INFLUENCES****Environment Variables**

**LC\_MESSAGES** determines the language in which messages are displayed. The current language settings can be checked with *locale(1)*.

a

**RETURN VALUE**

**arraycfg** returns the following values:

- 0 Successful completion.
- 1 An error in execution (I/O, subsystem, security, etc.) occurred.
- 2 An error in command syntax occurred: for example, an unknown command-line option was passed.
- 3 Timeout in communication to server. May indicate **ARMserver** is not running.

**DIAGNOSTICS**

The following messages can be generated by **arraycfg**:

**Usage: arraycfg -L LUN {-a capacity | -d | -r newLUN}**  
**[-V] [-?] <array-id>**

An error in command syntax has occurred. Re-enter the command with all necessary arguments.

**Usage: arraycfg -D slot {-a | -d [-v] [-R | -Z]}**  
**[-V] [-?] <array-id>**

An error in command syntax has occurred. Re-enter the command with all necessary arguments.

**arraycfg: Arg out of range**

One of the arguments has exceeded its maximum or minimum size, or is incorrect in form. Check the size and form of each argument.

**arraycfg: Unknown argument**

An invalid argument was specified. Check command usage.

**arraycfg: No such file or directory**

The specified *array-id* does not exist or does not identify a disk array LUN that is communicating with the system. Verify the correct *array-id* with *ioscan(1M)* or *arraydsp(1M)*.

**arraycfg: Error in command execution, <Additional Error Info>:**  
**<Error Info Decode>**

The command failed due to a device error, an internal error, or a system error. The *Additional Error Info* and *Error Info Decode* fields will hold specifics about the failure and its cause.

**arraycfg: The disk is already included.**

An attempt was made to include a disk that is already included in the array configuration.

**arraycfg: Previously downed disk has been downed.**

A down command was issued to a disk that was already downed. The type of down (destructive or nondestructive) specified by the latest command will be implemented.

**arraycfg: LUN nn does not exist in the array.**

An attempt was made to perform an operation on a nonexistent LUN.

**arraycfg: Not enough space to create LUN.**

An attempt was made to create a LUN that was larger than the available unallocated capacity. Reduce the size of the LUN, or make more capacity available by adding a disk or deleting an existing LUN.

**arraycfg: The specified LUN exists, cannot create.**

An attempt was made to create a LUN that already exists. Use a different number to create the LUN.

**arraycfg: The specified LUN does not exist or is invalid.**

An attempt was made to perform an operation on a LUN that does not exist.

**arraycfg: The specified LUN is invalid.**

An attempt was made to perform an operation on an invalid LUN.

**arraycfg: The specified newLUN already exists, cannot renumber.**

An attempt was made to renumber a LUN using the number of an existing LUN. Select an unused number for the LUN being renumbered.

**arraycfg: Disk was not downed, redundancy or data would be lost.**

The specified operation cannot be performed because it would result in loss of data availability or data redundancy. The **-R** and **-Z** options can be used to override this error.

**arraycfg: Disk was not downed, data would be lost.**

The specified operation cannot be performed because it would result in loss of data. The **-Z** option can be used to override this error.

**arraycfg: A diagnostic is currently running on this disk, cannot include disk until the diagnostic completes.**

An attempt was made to include a disk that is currently undergoing diagnostic testing. Wait until the diagnostic completes, or cancel the diagnostic using the **dteststat** utility.

**arraycfg: The specified disk is not installed.**

An attempt was made to perform an operation on a disk that is not installed in the array.

**arraycfg: The array is in a state that cannot process this command.**

The disk array is unable to execute the command. This typically occurs because the disk array is shut-down. Check disk array status.

**arraycfg: Array was just powered-on or reset.**

The command failed because the disk array has been powered-on or reset. Reissue the command and it should succeed.

## EXAMPLES

Create LUN 6 with a size of 500 Mbytes on the disk array identified by device file **/dev/rdisk/c2t0d0**:

```
arraycfg -L 6 -a 500 /dev/rdisk/c2t0d0
```

Renumber LUN 1 to LUN 2 on disk array serial number **00786b5c0000**:

```
arraycfg -L 1 -r 2 00786b5c0000
```

Delete LUN 3 on the disk array identified by device file **/dev/rdisk/c2t0d0**:

```
arraycfg -L 3 -d /dev/rdisk/c2t0d0
```

Include the disk in slot B4 in the configuration of disk array serial number **00786b5c0000**:

```
arraycfg -D B4 -a 00786b5c0000
```

Down the disk in slot A3 on disk array serial number **00786b5c0000**. Allow the command to execute even if redundancy will be lost:

```
arraycfg -D A3 -d -R 00786b5c0000
```

## DEPENDENCIES

**ARMServer** must be running to execute this command. See *ARMServer(1M)*.

## SECURITY CONFIGURATION

This command is modified for all security configurations.

### Security Behavior/Restrictions

Use of this command is restricted to authorized users only.

### Command Authorizations

This command requires the **sysadmin** authorization to successfully execute.

### Privileges

The command has been modified to support least privilege. The potential privileges possessed by the command and their uses include:

**allowdacread** This privilege is raised to provide discretionary read access to the devices.

**allowdacwrite** This privilege is raised to provide discretionary write access to the devices.

**allowmacread** This privilege is raised to provide mandatory read access to the devices.

**allowmacwrite** This privilege is raised to provide mandatory write access to the devices.

**filesysops** This privilege is raised to allow the *mknod(2)* system call to succeed.

**writeaudit** The command generates its own audit records and submits these directly to the system audit trail. This privilege is raised whenever the command needs to write an

audit record.

**AUTHOR**

**arraycfg** was developed by HP.

**SEE ALSO**

ARMServer(1M), arraydsp(1M), arrayfmt(1M), arraylog(1M), arraymgr(1M), arrayrbld(1M), arrayrecover(1M), download(1M), drivetest(1M), dteststat(1M), logprint(1M).

**a**

**NAME**

arraydsp - display the status and operating configuration of the disk array

**SYNOPSIS**

```
arraydsp [-l [LUN] | -d | -c | -s | -v | -h | -a] [-V] [-S] [-?] array-id
arraydsp { -r stime etime | -m stime etime [ int] } [-V] [-S] [-?] array-id
arraydsp -i [-V] [-S] [-?]
arraydsp -R [-V] [-?]
```

**DESCRIPTION**

**arraydsp** displays status and configuration information for the disk array identified by *array-id*. Logical configuration, physical configuration, current status, and performance data can all be displayed using **arraydsp**. A list of the serial numbers of all the disk arrays recognized by the host can also be displayed.

The *array-id* used to address the disk array can be the disk array serial number, the character device file name of any LUN on the array (LUN 0 if no LUNs are created), or the alias text string assigned to the disk array.

**Options**

**arraydsp** supports the following options:

- |                              |   |
|------------------------------|---|
| <b>none</b>                  | Display general information about the disk array. This includes product and vendor information, array state, and capacity usage.  |
| <b>-a</b>                    | Display the information presented by the <b>-l</b> , <b>-d</b> , <b>-c</b> , <b>-s</b> , <b>-v</b> and <b>-h</b> options. This is a quick way of displaying all configuration and status information about the disk array. This option will display information for all LUNs.   |
| <b>-c</b>                    | Display information for each controller installed in the disk array.  |
| <b>-d</b>                    | Display information for all disks installed in the array.   |
| <b>-h</b>                    | Display status information about the disk array hardware. This includes controller, power supply, and fan status.   |
| <b>-i</b>                    | Display the serial number of all disk arrays currently connected to and recognized by the server. Because this option is device-independent, it does not use <i>array-id</i> .  |
| <b>-l [LUN]</b>              | Display information for the LUN identified by <i>LUN</i> . If <i>LUN</i> is not specified, display information for all LUNs on the disk array.  |
| <b>-m stime etime [ int]</b> | <p>Display detailed performance metrics. These are the metrics on which the performance recommendations are based. Performance is analyzed over the period of time beginning at <i>stime</i> and ending at <i>etime</i>. A detailed description of the performance metrics is included later in this manual page.</p> <p>The <i>int</i> (interval) value controls the resolution of the data display. Each interval increment equals 15 minutes, with a default of 4 (1 hour).</p> <p>The format for entering time is <i>mmddhhmm[yy]</i>:</p> <p><i>mm</i> Month (01-12)<br/> <i>dd</i> Day (01-31)<br/> <i>hh</i> Hour (00-23)<br/> <i>mm</i> Minute (00-59)<br/> <i>yy</i> Year (optional)</p> |
| <b>-r stime etime</b>        | Display recommendations for improving disk array performance. The recommendations are based on various performance metrics maintained by the disk array. Performance is analyzed over the period of time beginning at <i>stime</i> and ending at <i>etime</i> . See above for the format for entering time.   |
| <b>-R</b>                    | Rescan for disk arrays. The host will scan for all supported disk arrays and update the current list. This may be useful if a new disk array has been added but does not yet show up in the array list. It is also useful if the state of the disk array has changed (for example, following a firmware update).  |

a

- s Display general configuration information for the disk array. This includes the configuration settings that control the operation of the entire disk array.
- S Raw output display. Data is output as a colon-delimited ASCII text string. Raw output format is near the end of this page.
- v Display virtual information about the disk array. This includes the amount of capacity currently being used for RAID 0/1, and the amount of unallocated capacity.
- V Verbose mode displays additional command execution, state, and/or status messages.
- ? Display extended usage message. This option overrides all others.

## EXTERNAL INFLUENCES

### Environment Variables

**LC\_MESSAGES** determines the language in which messages are displayed. The current language settings can be checked with *locale(1)*.

## RETURN VALUE

**arraydsp** returns the following values:

- 0 Successful completion.
- 1 An error in execution (I/O, subsystem, security, etc.) occurred.
- 2 An error in command syntax occurred: for example, an unknown command-line option was passed.
- 3 Timeout in communication to server. May indicate **ARMServer** is not running.

## DIAGNOSTICS

The following messages can be generated by **arraydsp**:

**Usage:** **arraydsp** [-l [LUN] | -d | -c | -s | -v | -h | -a |  
-r stime etime | -m stime etime [int]]  
[-v] [-?] <array-id>

An error in command syntax has occurred. Reenter the command with all necessary arguments.

**Usage:** **arraydsp** {-i | -R} [-V] [-?]

An error in command syntax has occurred. Reenter the command with all necessary arguments.

**Usage:** **arraydsp** -r stime etime | -m stime etime [int] array-id

The values entered for the performance analysis start and stop times are invalid. The format for entering the time is *mmddhhmm[yy]*.

**arraydsp: Arg out of range**

One of the arguments has exceeded its maximum or minimum size, or is incorrect in form. Check the size and form of each argument.

**arraydsp: Unknown argument**

An invalid argument was specified. Check command usage.

**arraydsp: No such file or directory**

The specified *array-id* does not exist or does not identify a disk array LUN that is communicating with the system. Verify the correct *array-id* with *ioscan(1M)* or *arraydsp(1M)*.

**arraydsp: Error in command execution, <Additional Error Info>:**

**<Error Info Decode>**

The command failed due to a device error, an internal error, or a system error. The *Additional Error Info* and *Error Info Decode* fields will hold specifics about the failure and its cause.

**arraydsp: Device was just powered-on or reset.**

The command failed because the disk array has been powered-on or reset. Reissue the command and it should succeed.

## EXAMPLES

Display general information about the disk array identified by device file */dev/rdisk/c2t0d0*:

```
arraydsp /dev/rdisk/c2t0d0
```

Display information for LUN 2 on disk array serial number 00786b5c0000:

```
arraydsp -l 2 00786b5c0000
```

Display information for all LUNs on the disk array identified by alias **AUTORAID4**:

```
arraydsp -l AUTORAID4
```

Display information for all disks installed in disk array serial number **00786b5c0000**:

```
arraydsp -d 00786b5c0000
```

List the serial numbers of all of the disk arrays recognized by the host:

```
arraydsp -i
```

Display performance recommendations for disk array serial number **00786b5c0000**. Analyze performance for the time period from 8:00 AM to 5:00 PM (1700) on March 15.

```
arraydsp -r 03150800 03151700 00786b5c0000
```

Display performance metrics for disk array serial number **00786b5c0000**. Analyze performance for the time period from 1200 PM to 6:00 PM (1800) on April 6. Use a display interval of 30 minutes.

```
arraydsp -m 04061200 04061800 2 00786b5c0000
```

PERFORMANCE METRICS

The following information describes the various performance metrics returned by the **-m** option.

Metric	Description
Drive Diff	Values (Typical/Limit): 0-1/>1  Indicates if the disks are properly distributed across the internal SCSI channels. If the disks are not installed properly, one of the SCSI channels may be handling more than its share of activity, reducing performance. Redistributing the disks will solve this problem.
Relocate Blocks	Values (Typical/Limit): 0.008/>0.07  Indicates how much data is being moved between RAID 0/1 and RAID 5. If too much data is being transferred between RAID 0/1 and RAID 5, performance declines.
Working Set	Values (Typical/Limit): 1/>1.1  Indicates the ratio of the write working set to the amount of RAID 0/1 space available. To maintain performance, the amount of RAID 0/1 space should equal or exceed the write working set (a value of 1 or less). A value greater than 1 indicates that the working set is larger than the available RAID 0/1 space. In this case, the disk array must service writes from RAID 5 space, which degrades performance.
Cache Utilization	Values (Typical/Limit): 0.1/0.2  Indicates how efficiently the write cache on the disk array controller is being utilized. If the write cache is consistently full, too many commands begin stacking up in the cache queue, resulting in decreased performance.
Disk Queue	Values (Typical/Limit): 2-4/>=10 (per disk)  Indicates the average number of transactions on all disk drives waiting in the internal disk SCSI queues. Too many commands in the queues decrease performance.
Concurrency	Values (Typical/Limit): >= drive count/< drive count  Indicates the number of internal simultaneous disk accesses the disk array is making. The disk array achieves optimum performance when it is accessing multiple disks at the same time, so some concurrency is desirable. Too low a value indicates that the host is not requesting enough concurrent I/Os to keep the disk array busy.
Cache Queue	Values (Typical/Limit): 0.05/>=0.32  Indicates the number of commands in the cache queue. An excessive number may reflect address conflicts in host I/Os, caused by repeated access to the same block(s) of data.
Host Queue	Values (Typical/Limit): 0/>32  Indicates the number of host commands waiting in the host queue. A problem here typically reflects another bottleneck further along in the controller data path. For example, if the cache queue is full, commands will stack up in the host queue waiting for the



a

	cache queue to empty. The counter does not show until 32 commands are in the queue.
Throughput	Values (Typical/Limit): Not Used  Indicates the average number of bytes per second being transferred over the SCSI channel. This metric reflects channel activity, and does not indicate any problems with the disk array.
I/O Rate	Values (Typical/Limit): Not Used  Indicates the average number of I/Os per second being serviced by the disk array. This metric reflects channel activity; it does not indicate problems with the disk array

RAW OUTPUT FORMAT

The contents of the raw output text string for each display option are listed here. Note that the raw output for the **-a** option comprises the individual strings displayed by the **-l**, **-d**, **-c**, **-s**, **-v** and **-h** options. For clarification when evaluating Boolean expressions, **TRUE=1** and **FALSE=0**.

Option    Raw Output Description

none	D:HP:ProductID:ArraySN:ServerName:ArrayType: MfgProdCode:PhysCapacity:LUNCapacity:ActiveSpareCapacity: DiskNonIncludedCapacity:RedundantCapacity: UnallocatedCapacity:<ArrayState>:<WarnInd>:<Alias>  The leading "D" is literal to identify this as a DEFAULT record. All capacities are in MB.
-c	C:<CntrlrID>:<CompState>:HP:ProductID:Primary: BkplaneSlot:Address:SerNumber:PairedSerNum: ControllerType:ProdRev:MfgProdCode:FirmwareRev:<arraySN>  The leading "C" is literal to identify this as a CONTROLLER record. Output will be displayed for each controller installed. If <CompState> is not "Good", information in the remaining fields will be undefined or absent.
-d	The following output is provided for each installed disk: PD:<SlotID>:<DiskState>:Vendor:ProductID:Capacity: MaxBlkAddr:BlkLen:SerialNum:FwRev:SeqNum:VolSetSerNum:<arraySN>: RecoveryMapsPresent  The leading "PD" is literal to identify this as a PHYSICAL DISK record. The above will be displayed for each disk slot. For disks with a <DiskState> of NOT_PRESENT, the only field that will follow <DiskState> will be <arraySN>. "Capacity" is displayed in MB, rounded up to the nearest MB.  The following output is provided for each missing disk. A disk is missing if it is no longer accessible to the array (due to removal or failure) but is still required to maintain full redundancy. After the data on the missing disk is rebuilt, the disk is no longer required and is not considered missing.  MD:Vendor:ProductID:Capacity:MaxBlkAddr:BlkLen: SerialNum:DiskIDNum:VolSetSerNum:<arraySN>  The leading "MD" is literal to identify this as a MISSING DISK record. The above will be displayed for each missing disk. "Capacity" is displayed in MB, rounded up to the nearest MB.
-h	One or more "component state records" will be displayed in the following format: H:HP:ProductID:<CompID>:<nn>:<CntrlrID>:<CompState>:<arraySN> H:HP:ProductID:<CompID>:<nn>:<CntrlrID>:<BattState>:<arraySN> H:HP:ProductID:<CompID>:<nn>:<CntrlrID>:<SimmState>:<arraySN>  The leading "H" is literal to identify this as a HARDWARE record. "<CompID>" is the unique ID of the component. The CompIDs are interpreted as follows:  0 = Fans 1 = Power supplies 2 = Controllers 3 = RESERVED



4 = Controller X DRAM  
 5 = Controller X NVRAM  
 6 = Controller Y DRAM  
 7 = Controller Y DRAM  
 8 = Controller X batteries  
 9 = Controller Y batteries

Because there may be multiple components of type "<CompID>", the "<nn>" field further identifies the component. "<nn>" will always be 0 for a "<CompID>" = CONTROLLER.

The "<CntrlrID>" identifies the associated controller. If no association exists for this component with a controller, this field will be empty. "<CompState>" and "<BattState>" indicate the state of the component or battery, respectively.

- i The following single-line output includes a separate field (xxxxxxxxxx) for each disk array identified:

I:xxxxxxxxxx:xxxxxxxxxx

The leading "I" is literal to identify this as an Identify record.

- l L:HP:ProductID:LunNumber:Present:MaxBlockAddr:  
BlockLength:<arraySN>:<Alias>

The leading "L" is literal to identify this as a LUN record. "Present" is a Boolean, TRUE if the LUN is configured, FALSE otherwise. If a LUN is not specified, the above will be displayed for all LUNs.

- m PM:HP:ProductID:<requestedStime>:  
<requestedEtime>:<actualStime>:<actualEtime>:<interval>:<arraySN>  
:<PerfMetricId>:<nn>:<nn>: ...  
:<PerfMetricId>:<nn>:<nn>: ...  
:<PerfMetricId>:<nn>:<nn>: ...

The leading "PM" is literal to identify this as a PERFORMANCE METRIC record.

- r PR:HP:ProductID:<stime>:<etime>:  
<Recommendation1>:<Recommendation2>:...

The leading "PR" is literal to identify this as a PERFORMANCE RECOMMENDATIONS record.

- s S:HP:ProductID:VolSetPartition:FmtPatternFill:  
AutoRebuild:AutoInclude:Parity:SDTR:WDTR:TermPwr:UnitAtn:  
ActiveSpare:LogFullWarn:WrtCache:RdCache:RebuildPri:  
CapDepThresh:WrtWrkSetInt:SubsysTypeID:  
Language:CntXaddr:CntYaddr:Rebuilding:RebuildPercentage:  
Balancing:Optimizing:<ArrayState>:<WarnInd>:DRR:<ArraySN>:  
LUNCreateLimit:MaxLUNCreateLimit:SecContOffline:VeryEarlyBusy  
QueueFullThreshold:MaxQueueFullThreshold:  
DisableSingleControllerWarning:LockWriteCacheOn:  
DisableNVRAMonWCEfalse:DisableNVRAMonSingleController:  
DisableNVRAMonUPSabsent:<ForceUnitAccessResponse>:  
DisableReadHits:RecoveryInProgress:<RecoveryProgress>:  
<ResiliencyThreshold>

The leading "S" is literal to identify this as an ARRAY STATE record. "<WarningState>" will be a hexadecimal number representing the "Warning Indications" field returned by the "Check Array State" command, which is described in the "HP XLR1200 Advanced Disk Array SCSI Interface Specification" (5010-3128).

- v V:HP:ProductID:Raid1Blocks:Raid1BlockSize:<arraySN>

The leading "V" is literal to identify this as a VIRTUAL record.

## DEPENDENCIES

**ARMServer** must be running to execute this command. See *ARMServer*(1M).

## SECURITY CONFIGURATION

This command is modified for all security configurations.

**Security Behavior/Restrictions**

Use of this command is restricted to authorized users only.

**Command Authorizations**

This command requires the **sysadmin** authorization to successfully execute.

**Privileges**

The command has been modified to support least privilege. The potential privileges possessed by the command and their uses include:

<b>allowdacread</b>	This privilege is raised to provide discretionary read access to the devices.
<b>allowmacread</b>	This privilege is raised to provide mandatory read access to the devices.
<b>filesysops</b>	This privilege is raised to allow the <i>mknod(2)</i> system call to succeed.

**AUTHOR**

**arraydsp** was developed by HP.

**SEE ALSO**

ARMServer(1M), arraycfg(1M), arrayfmt(1M), arraylog(1M), arraymgr(1M), arrayrbld(1M), arrayrecover(1M), download(1M), drivetest(1M), dteststat(1M), logprint(1M).

**NAME**

arrayfmt - formats a LUN or an entire disk array

**SYNOPSIS**

**arrayfmt** **-L** *LUN* [-V] [-?] *array-id*

**arrayfmt** **-F** [-h] [-V] [-?] *array-id*

**DESCRIPTION**

**arrayfmt** formats the LUN (logical devices) identified by *LUN* or the entire disk array identified by *array-id*.

**WARNING:** This is a destructive command. Data on the selected LUN or disk array will be destroyed following successful completion of this command. Backup all vital data before performing a format.

The *array-id* used to address the disk array can be the disk array serial number, the character device file name of any LUN on the array (LUN 0 if no LUNs are created), or the alias text string assigned to the disk array.

**Options**

**arrayfmt** supports the following options:

- F** [ *-h* ]      Format the entire disk array identified by *array-id*. By default, only disks currently included in the array configuration will be formatted. If the *-h* option is specified, all disks installed in the cabinet will be formatted, even those that are not included in the array configuration.
- L** *LUN*          Format the LUN specified by *LUN*.
- V**                Verbose mode displays additional command execution, state, and/or status messages.
- ?**                Display expanded usage message. This option overrides all others.

**EXTERNAL INFLUENCES****Environment Variables**

**LC\_MESSAGES** determines the language in which messages are displayed. The current language settings can be checked with *locale(1)*.

**RETURN VALUE**

**arrayfmt** returns the following values:

- 0 Successful completion.
- 1 An error in execution (I/O, subsystem, security, etc.) occurred.
- 2 An error in command syntax occurred: for example, an unknown command-line option was passed.
- 3 Timeout in communication to server. May indicate **ARMServer** is not running.

**DIAGNOSTICS**

The following messages can be generated by **arrayfmt**:

**Usage: arrayfmt {-L LUN | -F [-h]} [-V] [-?] <array-id>**

An error in command syntax has occurred. Re-enter the command with all necessary arguments.

**arrayfmt: Arg out of range**

One of the arguments has exceeded its maximum or minimum size, or is incorrect in form. Check the size and form of each argument.

**arrayfmt: Unknown argument**

An invalid argument was specified. Check command usage.

**arrayfmt: No such file or directory**

The specified *array-id* does not exist or does not identify a disk array that is communicating with the system. Verify the correct *array-id* with *ioscan(1M)*.

**arrayfmt: Error in command execution, <Additional Error Info>:**

**<Error Info Decode>**

The command failed due to a device error, an internal error, or a system error. The *Additional Error Info* and *Error Info Decode* fields will hold specifics about the failure and its cause.

**arrayfmt: Cannot format array while a diagnostic is running.**

An attempt was made to format the array while a diagnostic was in progress. Allow the diagnostic to complete or cancel the test using **dteststat**.

**arrayfmt: Device was just powered-on or reset.**

A command failed because the disk array has been powered-on or reset. Reissue the command and it should succeed.

## EXAMPLES

Format LUN 1 on disk array serial number 00786b5c0000:

```
arrayfmt -L 1 00786b5c0000
```

Format the entire disk array identified by device file `/dev/rdisk/c2t0d0`:

```
arrayfmt -F /dev/rdisk/c2t0d0
```

## DEPENDENCIES

**ARMServer** must be running to execute this command. See *ARMServer(1M)*.

## SECURITY CONFIGURATION

This command is modified for all security configurations.

### Security Behavior/Restrictions

Use of this command is restricted to authorized users only.

### Command Authorizations

This command requires the **sysadmin** authorization to successfully execute.

### Privileges

The command has been modified to support least privilege. The potential privileges possessed by the command and their uses include:

**allowdacread** This privilege is raised to provide discretionary read access to the devices.

**allowdacwrite** This privilege is raised to provide discretionary write access to the devices.

**allowmacread** This privilege is raised to provide mandatory read access to the devices.

**allowmacwrite** This privilege is raised to provide mandatory write access to the devices.

**filesysops** This privilege is raised to allow the *mknod(2)* system call to succeed.

**writeaudit** The command generates its own audit records and submits these directly to the system audit trail. This privilege is raised whenever the command needs to write an audit record.

## AUTHOR

**arrayfmt** was developed by HP.

## SEE ALSO

*ARMServer(1M)*, *arraycfg(1M)*, *arraydsp(1M)*, *arraylog(1M)*, *arraymgr(1M)*, *arraybld(1M)*, *arrayrecover(1M)*, *download(1M)*, *drivetest(1M)*, *dteststat(1M)*, *logprint(1M)*.

**NAME**

arrayinfo - describe general characteristics of a disk array

**SYNOPSIS**

**arrayinfo** [-j | -m | -s | -ar | -dr] *device\_file*

**DESCRIPTION**

**arrayinfo** displays summarized information for the SCSI disk array associated with the character device file *device\_file*.

By default **arrayinfo** returns the following information:

- array vendor ID
- array product ID
- number of attached disk mechanisms
- vendor/product type of attached disk mechanisms. (Assumes all are the same type)

NOTE: The array vendor ID, and product ID information are constant, regardless of the type and quantity of disks attached.

**Options:**

**arrayinfo** recognizes the following options:

- j Displays the current setting of certain jumper switches on each disk mechanism, including:
 

Automatic Spin Up	( 0 Disable / 1 Enable )
Parity Error Detect	( 0 Disable / 1 Enable )
Unit Attention	( 0 Enable / 1 Disable )
Initiate Synchronous Data Transfer	( 0 Disable / 1 Enable )

 SCSI target address of the mechanism
- m Displays array mapping information, including:
  - The disk vendor, and model type of each disk in the array
  - The current status of each disk in the array, as determined by the array controller.
  - The array sub-channel, and sub-channel addresses for each disk in the array.
- s Displays serial numbers. This option displays serial number information for the disk array controller, and all attached disk mechanisms.
- ar Displays array revision information. This option displays revision information for the hardware, firmware, and software of the array controller.
- dr Display disk revisions. This option displays revision information for the hardware, and firmware of each disk in the array.

**RETURN VALUE**

**arrayinfo** returns the following values:

- 0 Successful completion
- 1 Command failed (an error occurred).

**DEPENDENCIES**

This utility is only compatible with HP C2430 disk arrays.

**Series 700**

**arrayinfo** must be used with a device file mapped to a unit address that is not in use by the array controller (unconfigured). By convention unit addresses 6 and 7 should not be configured. Array information should be accessible by addressing either of these unit addresses.

**Series 800**

Any device file (LU) that is mapped to the disk array can be used to access the array information.

**AUTHOR**

**arrayinfo** was developed by Hewlett-Packard.

**arrayinfo(1M)**

**arrayinfo(1M)**

**SEE ALSO**  
dsp(1M).

**a**

**NAME**

arraylog - accesses and clears disk array logs

**SYNOPSIS**

```
arraylog [-u] [-e] [ -d slot [-p pagecode] ] [ -C { -c | -d slot } ] [-?] array-id
```

**DESCRIPTION**

**arraylog** provides access to the controller and disk logs maintained by the disk array identified by *array-id*. These logs contain information useful for diagnosing and troubleshooting the disk array. The logs can also be cleared using **arraylog**.

The *array-id* used to address the disk array can be the disk array serial number, the character device file name of any LUN on the array (LUN 0 if no LUNs are created), or the alias text string assigned to the disk array.

**Options**

**arraylog** supports the following options:

**-C { -c | -d slot }**

Clear the specified logs. If **-c** is specified, clear the disk array controller usage and event logs. Both logs will be cleared when using this option. It is not possible to clear just one of the logs.

If **-d** is specified, clear the log for the disk installed in the cabinet slot identified by *slot*.

**-d slot [-p pagecode]**

Display the contents of the log for the disk installed in the cabinet slot identified by *slot*. Slot numbers must be of the form *An* or *Bn*, where *A* or *B* correspond to a cabinet column, and *n* corresponds to a shelf position (1-6).

The **-p** option returns the log information identified by *pagecode*. The use of the **-p** option is intended primarily for accessing logs on unsupported disks.

**-e** Display the contents of the disk array controller event log.

**-u** Display the contents of the disk array controller usage log.

**-?** Display expanded usage message. This option overrides all others.

**EXTERNAL INFLUENCES****Environment Variables**

**LC\_MESSAGES** determines the language in which messages are displayed. The current language settings can be checked with *locale(1)*.

**RETURN VALUE**

**arraylog** returns the following values:

- 0 Successful completion.
- 1 An error in execution (I/O, subsystem, security, etc.) occurred.
- 2 An error in command syntax occurred: for example, an unknown command-line option was passed.
- 3 Timeout in communication to server. May indicate **ARMServer** is not running.

**DIAGNOSTICS**

The following messages can be generated by **arraylog**:

**Usage: arraylog { [-u] [-e] [-d n [-p n]] [-C -c] [-C -d n] } [-?]**  
**<array-id>**

An error in command syntax has occurred. Re-enter the command with all necessary arguments.

**arraylog: Arg out of range**

One of the arguments has exceeded its maximum or minimum size, or is incorrect in form. Check the size and form of each argument.

**arraylog: No such file or directory**

The specified *array-id* does not exist or does not identify a disk array that is communicating with the system. Verify the correct *array-id* with *ioscan(1M)* or *arraydsp(1M)*.

**arraylog: Unknown argument**

An invalid argument was specified. Check command usage.

**arraylog: Error in command execution, <Additional Error Info>:  
<Error Info Decode>**

The command failed due to a device error, an internal error, or a system error. The *Additional Error Info* and *Error Info Decode* fields will hold specifics about the failure and its cause.

**arraylog: Cannot clear logs on an unsupported disk.**

An attempt was made to clear the logs on an unsupported disk. The logs on an unsupported disk cannot be cleared using this command.

**arraylog: No disk in slot n.**

There is no disk in the indicated slot.

**Use the -p [pagecode] option to attempt to recover any log information.**

**Any recoverable log data will be displayed as a hex dump.**

The logs on an unsupported disk cannot be interpreted by the **arraylog** utility. Using the **-p [pagecode]** option will cause the utility to retrieve any log information it can from the disk.

**arraylog: Array was just powered-on or reset.**

The command failed because the disk array has been powered-on or reset. Reissue the command and it should succeed.

## EXAMPLES

Display the controller usage log for the disk array identified by device file **/dev/rdisk/c2t0d0**:

```
arraylog -u /dev/rdisk/c2t0d0
```

Display the controller event log for disk array serial number **00786b5c0000**:

```
arraylog -e 00786b5c0000
```

Display the log information for the disk installed in slot A2 of the disk array identified by device file **/dev/rdisk/c2t0d0**:

```
arraylog -d A2 /dev/rdisk/c2t0d0
```

Clear the controller event and usage logs for disk array serial number **00786b5c0000**:

```
arraylog -C -c 00786b5c0000
```

Clear the log for the disk installed in slot B1 of disk array serial number **00786b5c0000**:

```
arraylog -C -d B1 00786b5c0000
```

## DEPENDENCIES

**ARMServer** must be running to execute this command. See *ARMServer(1M)*.

## SECURITY CONFIGURATION

This command is modified for all security configurations.

### Security Behavior/Restrictions

Use of this command is restricted to authorized users only.

### Command Authorizations

This command requires the **sysadmin** authorization to successfully execute.

### Privileges

The command has been modified to support least privilege. The potential privileges possessed by the command and their uses include:

**allowdacread** This privilege is raised to provide discretionary read access to the devices.

**allowdacwrite** This privilege is raised to provide discretionary write access to the devices.

**allowmacread** This privilege is raised to provide mandatory read access to the devices.

**allowmacwrite** This privilege is raised to provide mandatory write access to the devices.

**filesysops** This privilege is raised to allow the *mknod(2)* system call to succeed.



**writeaudit**      The command generates its own audit records and submits these directly to the system audit trail. This privilege is raised whenever the command needs to write an audit record.

**AUTHOR**

**arraylog** was developed by HP.

**SEE ALSO**

ARMServer(1M), arraycfg(1M), arraydsp(1M), arrayfmt(1M), arraymgr(1M), arrayrbld(1M), arrayrecover(1M), download(1M), drivetest(1M), dteststat(1M), logprint(1M).

**a**

**NAME**

arraymgr - manages the operating environment of the disk array.

**SYNOPSIS**

```
arraymgr -s { shut | start } [-V] [-?] array-id
arraymgr -R [-V] [-?] array-id
arraymgr -D alias_name [-V] [-?] array-id
arraymgr -J { SingleController | Secure | Normal | HighPerformance }
           [-V] [-?] array-id
arraymgr [Options] [-V] [-?] array-id
```

**DESCRIPTION**

**arraymgr** manages the operating environment of the disk array by providing access to the settings used to control disk array operation. These settings control the operation of the entire disk array; consequently, every LUN on the disk array will be affected by any changes made using this command. **arraymgr** also allows you shutdown, restart, and reset the disk array.

The *array-id* used to address the disk array can be the disk array serial number, the character device file name of any LUN on the array (LUN 0 if no LUNs are created), or the alias text string assigned to the disk array.

Changing some of the disk array settings may require that the disk array be reset using the **-R** option. You will be prompted to initiate a reset if the setting you changed requires this action.

NOTE: Several of the **arraymgr** options are used to set the disk array SCSI operating parameters. These parameters control the transfer of data over the SCSI channel connecting the host and the disk array and typically do not need to be changed. Before changing a SCSI setting, make sure you understand what affect it will have on disk array operation. Selecting an incorrect setting may make it impossible for the host to access the disk array.

**Options**

**arraymgr** supports the following options:

- a { on | off }** Set Auto Rebuild on or off. On allows the disk array to automatically begin a rebuild in the event of a disk failure. Off requires a rebuild to be started manually. In either case, the disk array must have enough available capacity to perform the rebuild.
- c { X | Y }** Set controller X or controller Y as the active controller for the disk array. This option is only available if there are two controllers installed in the disk array.
- C { X | Y } addr** Set the SCSI address of the specified controller (X or Y) to the value of *addr*. The address must be a value from 0 to 15 and must not conflict with any other SCSI devices connected to the same SCSI channel.
- D alias\_name** Assigns the text string specified by *alias\_name* as the alias for the disk array. The alias is another mechanism for identifying the disk array when executing a command. The alias name can be up to 12 characters in length and can include upper case letters, numbers, spaces, number sign (#), underscore (\_), and period (.).  
  
Aliasing can be used in a variety of ways to help identify disk arrays in large systems. For example, by assigning numbers to racks and to the shelf positions within the racks, each disk array can be uniquely identified using an appropriate alias. If a rack is assigned number 12, the disk array installed on shelf 3 of the rack could be identified using an alias of 12\_03. This technique simplifies locating the disk array should it need service.
- e { on | off }** Change the apparent state of read cache on or off. This switch only changes the read cache setting presented to the operating system and does not affect the operation of the disk array, which always has read cache enabled. This switch is provided for operating system interoperability.
- f { on | off }** Set Format Pattern Fill on or off. This feature impacts the performance of the disk array. When disabled the disk array does not write a format pattern to the unused portion of each 64K data block. This improves the write performance of the disk array, but leaves invalid data in the unused portion of the block. Enabling the format

pattern fill causes the disk array to fill the unused portion of each block with a known data pattern. This increases the amount of time required to perform each write, but it ensures that any unused portions of the disk contain a recognizable data pattern. The need to use this feature depends on how well your OS can detect unwritten portions of the disk. If your OS requires a known format pattern to distinguish areas of the disk that do not contain valid data, then enable write pattern fill.

- F** Force server lock. Used in multi-host configurations, this option allows a host to take immediate ownership of the disk array lock. This lock gives the owner exclusive ability to perform destructive operations, such as reading and clearing the disk array logs.
- g { start | stop }** Starts and stops background disk integrity testing. When starting, the testing will begin within 60 seconds of issuing the command. Disk integrity testing checks the condition of the disk media, looking for potential problems and data errors.
- G value** Controls the interval at which disk integrity testing is performed. A value of 1 to 365 specifies the number of days between testing. If a value of 0 is specified, testing is disabled and will not be performed. The default value is 9 days. If the disk array is reset or power cycled, the default value of 9 days will be restored.
- h { on | off }** Set Active Spare on or off. On enables the disk array to allocate enough capacity to rebuild a failed disk. Off disables Active Spare.
- i { on | off }** Set Auto Include on or off. On allows the disk array to automatically include any disk when it is installed in the disk array cabinet. Off requires that a disk be manually included after it is installed.
- J** This option sets the level of protection offered by data resiliency. It determines how often the content of the controller maps are copied to the disk. Keeping the map information on the disks protect against controller map loss.  
  
**SingleController** is used if the disk array is operating with only one controller. This suppresses the single controller warning messages that are normally generated when the disk array is operating with one controller.  
  
**Secure** continually updates the disks with any changes in the controller maps. This is the highest level of data protection, but it may result in decreased I/O performance.  
  
**Normal** updates the maps on the disks at regular intervals (typically 4 seconds). This option offers both data protection and good performance.  
  
**HighPerformance** updates the disk maps only during shutdown of the disk array. This is the lowest level of data protection, but it offers the highest level of performance.
- l** Flush current server logs to disk. The contents of the server log are copied to a disk file.
- m { on | off }** Set Terminator Power on or off.
- o { on | off }** Set Secondary Controller Offline. When set on, this option causes the secondary controller to remove itself from the SCSI bus. The secondary controller no longer responds to host commands and appears to be disconnected from the SCSI bus. This allows the secondary controller to be set to the same SCSI ID as the primary controller. If the primary controller fails, the secondary controller takes over and the host driver does not have to contend with a change in SCSI ID. This is useful for systems that do not use a driver that can gracefully handle a change in SCSI ID on the disk array.
- p { high | low }** Set Rebuild Priority high or low. High sets the rebuild to the same priority as host I/Os. This allows the rebuild to complete as quickly as possible, but slows down the servicing of I/Os. Low sets the rebuild priority lower than host I/Os. This ensures that host I/Os are serviced more quickly, but it delays the completion of the rebuild.
- P { on | off }** Set SCSI Parity Checking on or off.
- r { on | off }** Set Disable Remote Reset on or off. On disables Remote Reset, and off enables it. Remote Reset is useful when the disk array controllers are connected to separate

a

SCSI buses. In some situations, a reset on one SCSI bus can cause the clearing of all commands on the other bus. When this occurs, the host would typically have to wait for a timeout before realizing that commands had been lost. If Remote Reset is enabled (Disable Remote Reset off), the other controller will automatically reset its own SCSI bus, thus alerting the host that commands have been lost. This allows the host to respond quickly and reissue any lost commands.

- R** Reset the disk array. This operation performs a reset of the disk array.
- s { shut | start }** Shutdown (**shut**) or restart (**start**) the disk array. A shutdown takes the disk array offline, making all data on the array unavailable to the host. A restart brings the disk array back online.
- t value** Set the capacity threshold warning to the percentage specified by *value*. The capacity threshold warning generates an alert when the percentage of disk array capacity specified by *value* is in use.
- T { on | off }** Set SDTR on or off.
- V** Verbose mode displays additional command execution, state, and/or status messages.
- w { on | off }** Change the apparent state of write cache on or off. This switch only changes the write cache setting presented to the operating system and does not affect the operation of the disk array, which always has write cache enabled. The disk array write cache is stored in NVRAM which eliminates the need to disable write cache for protection against power loss. This switch is provided for interoperability with those operating systems that require write cache be turned off.
- W { on | off }** Set WDTR on or off.
- ?** Display extended usage message. This option overrides all others.

## EXTERNAL INFLUENCES

### Environment Variables

**LC\_MESSAGES** determines the language in which messages are displayed. The current language settings can be checked with *locale(1)*.

## RETURN VALUE

**arraymgr** returns the following values:

- 0 Successful completion.
- 1 An error in execution (I/O, subsystem, security, etc.) occurred.
- 2 An error in command syntax occurred: for example, an unknown command-line option was passed.
- 3 Timeout in communication to server. May indicate **ARMServer** is not running.

## DIAGNOSTICS

The following messages can be generated by **arraymgr**:

**Usage: arraymgr {[-s shut|start] | [-D alias\_name] | -R | -F}**  
**[-V] [-?] <array-id>**

An error in command syntax has occurred. Re-enter the command with all necessary arguments.

**Usage: arraymgr {[-c X|Y] [-C X|Y addr] [-i on|off]**  
**[-t value] [-e on|off] [-w on|off] [-h on|off]**  
**[-a on|off] [-p high|low] [-P on|off] [-T on|off]**  
**[-W on|off] [-m on|off] [-f on|off] [-r on|off]**  
**[-q value] [-n value] [-o on|off] [-b on|off]**  
**[-g start|stop] [-G value] [-l] [-L]}**  
**[-S] [-V] [-?] [-E num] <array-id>**

An error in command syntax has occurred. Re-enter the command with all necessary arguments.

**Usage: arraymgr -J SingleController|Secure|Normal|HighPerformance**  
**[-V] [-?] <array-id>**

An error in command syntax has occurred. Re-enter the command with all necessary arguments.

**arraymgr: Arg out of range**

One of the arguments has exceeded its maximum or minimum size, or is incorrect in form. Check the size and form of each argument.

**arraymgr: Unknown argument**

An invalid argument was specified. Check command usage.

**arraymgr: No such file or directory**

The specified *array-id* does not exist or does not identify a disk array LUN that is communicating with the system. Verify the correct *array-id* with *ioscan(1M)*.

**arraymgr: Error in command execution, <Additional Error Info>:  
<Error Info Decode>**

The command failed due to a device error, an internal error, or a system error. The *More Error Info* and *Error Info Decode* fields will hold specifics about the failure and its cause.

**arraymgr: The array did not start up following the reset.**

The disk array has been reset, but it did not start up again. Check the control panel for possible hardware failure.

**arraymgr: The array is not shutdown, no restart performed.**

An attempt was made to restart a disk array that was not shutdown.

**arraymgr: The array did not restart in the expected time.**

Following a restart command, the disk array took too long to come back on line.

**arraymgr: The array is already shutdown.**

An attempt was made to shutdown a disk array that was already shutdown.

**arraymgr: The specified controller is not physically installed.**

An attempt was made to perform an operation on a controller that is not installed in the disk array.

**arraymgr: The specified controller is already active.**

An attempt was made to make the currently active controller active again.

**arraymgr: Active Spare Desired flag has been set, however the array is reporting the Active Spare is not available.**

The Active Spare feature has been enabled, but the disk array does not have enough unallocated capacity to create the Active Spare. Add another disk or delete an existing LUN to increase the amount of unallocated capacity so the Active Spare can be created.

**arraymgr: Array was just powered-on or reset.**

The command failed because the disk array has been powered-on or reset. Reissue the command and it should succeed.

## EXAMPLES

Shutdown the disk array identified by device file `/dev/rdisk/c2t0d0`:

```
arraymgr -s shut /dev/rdisk/c2t0d0
```

Restart disk array serial number 00786b5c0000:

```
arraymgr -s start 00786b5c0000
```

Set **Y** as the active controller on disk array serial number 00786b5c0000:

```
arraymgr -C Y 00786b5c0000
```

Assign an alias to the disk array identified by device file `/dev/rdisk/c2t0d0`. Use an alias that identifies the rack the disk array is installed in (51 for this example), and the shelf position the disk array is on (03 for this example):

```
arraymgr -D RACK_51_03 /dev/rdisk/c2t0d0
```

Set the data resiliency protection level to the highest level of protection on the disk array identified by alias **AUTORAID2**.

```
arraymgr -J Secure AUTORAID2
```

## DEPENDENCIES

**ARMServer** must be running to execute this command. See *ARMServer(1M)*.

## SECURITY CONFIGURATION

This command is modified for all security configurations.

**Security Behavior/Restrictions**

Use of this command is restricted to authorized users only.

**Command Authorizations**

This command requires the **sysadmin** authorization to successfully execute.

**Privileges**

The command has been modified to support least privilege. The potential privileges possessed by the command and their uses include:

<b>allowdacread</b>	This privilege is raised to provide discretionary read access to the devices.
<b>allowdacwrite</b>	This privilege is raised to provide discretionary write access to the devices.
<b>allowmacread</b>	This privilege is raised to provide mandatory read access to the devices.
<b>allowmacwrite</b>	This privilege is raised to provide mandatory write access to the devices.
<b>filesysops</b>	This privilege is raised to allow the <i>mknod(2)</i> system call to succeed.
<b>writeaudit</b>	The command generates its own audit records and submits these directly to the system audit trail. This privilege is raised whenever the command needs to write an audit record.

**AUTHOR**

**arraymgr** was developed by HP.

**SEE ALSO**

ARMServer(1M), arraycfg(1M), arraydsp(1M), arrayfmt(1M), arraylog(1M), arrayrbld(1M), arrayrecover(1M), download(1M), drivetest(1M), dteststat(1M), logprint(1M).

**NAME**

arrayrbl - controls rebuilding of the disk array

**SYNOPSIS**

```
arrayrbl -r | -c | -p | -a { on | off } | -P { high | low } [-V] [-?] array-id
```

**DESCRIPTION**

**arrayrbl** manages the rebuild process on the disk array identified by *array-id*. A rebuild can be started, canceled, prioritized, and monitored using **arrayrbl**.

The *array-id* used to address the disk array can be the disk array serial number, the character device file name of any LUN on the array (LUN 0 if no LUNs are created), or the alias text string assigned to the disk array.

**Options**

**arrayrbl** supports the following options:

- a { on | off } Set Auto Rebuild. On allows the disk array to automatically begin a rebuild in the event of a disk failure. Off requires a rebuild to be started manually.
- c Cancel a rebuild currently in progress. A rebuild started by Auto Rebuild cannot be canceled.
- p Display rebuild status and control settings. The following information is displayed:
  - Vendor and product ID.
  - Rebuild progress in percent complete, or a message indicating that no rebuild is currently in progress.
  - Current setting of Auto Rebuild (on or off).
  - Current setting of Rebuild Priority (high or low).
  - Current rebuild status indicating if a rebuild is in progress, or if a rebuild failed.
- P { high | low } Set Rebuild Priority. High sets the rebuild to the same priority as host I/Os. This allows the rebuild to complete as quickly as possible, but slows down the servicing of I/Os. Low sets the rebuild priority lower than host I/Os. This ensures that host I/Os are serviced first, but it delays the completion of the rebuild.
- r Start a rebuild on the disk array.
- V Verbose mode displays additional command execution, state, and/or status messages.
- ? Display extended usage message. This option overrides all others.

**EXTERNAL INFLUENCES****Environment Variables**

**LC\_MESSAGES** determines the language in which messages are displayed. The current language settings can be checked with *locale(1)*.

**RETURN VALUE**

**arrayrbl** returns the following values:

- 0 Successful completion.
- 1 An error in execution (I/O, subsystem, security, etc.) occurred.
- 2 An error in command syntax occurred: for example, an unknown command-line option was passed.
- 3 Timeout in communication to server. May indicate **ARMserver** is not running.

**DIAGNOSTICS**

The following messages can be generated by **arrayrbl**:

```
Usage: arrayrbl [-r | -c] | [-p] | [-a on|off] [-P high|low]
[-V] [-?] <array-id>
```

An error in command syntax has occurred. Re-enter the command with all necessary arguments.

**arrayrbl: Arg out of range**

One of the arguments has exceeded its maximum or minimum size, or is incorrect in form. Check the size and form of each argument.

a

**arraybld: Unknown argument**

An invalid argument was specified. Check command usage.

**arraybld: No such file or directory**

The specified *array-id* does not exist or does not identify a disk array LUN that is communicating with the system. Verify the correct *array-id* with *ioscan(1M)* or *arraydsp(1M)*.

**arraybld: Error in command execution, <Additional Error Info>:  
<Error Info Decode>**

The command failed due to a device error, an internal error, or a system error. The *Additional Error Info* and *Error Info Decode* fields will hold specifics about the failure and its cause.

**arraybld: The array does not need to be rebuilt, rebuild not started.**

An attempt was made to start a rebuild when it was not required.

**arraybld: The array is already rebuilding.**

An attempt was made to start a rebuild with one already in progress.

**arraybld: Cannot cancel because the array is not rebuilding.**

An attempt was made to cancel a rebuild, but none was in progress.

**arraybld: Cannot cancel a rebuild when array has Auto Rebuild enabled.**

An attempt was made to cancel a rebuild that was started automatically by Auto Rebuild. A rebuild started in this way cannot be canceled unless Auto Rebuild is first disabled.

**arraybld: Device was just powered-on or reset.**

The command failed because the disk array has been powered-on or reset. Reissue the command and it should succeed.

**EXAMPLES**

Rebuild disk array serial number 00786b5c0000:

```
arraybld -r 00786b5c0000
```

Cancel a rebuild currently in progress on the disk array identified by device file */dev/rdisk/c2t0d0*:

```
arraybld -c /dev/rdisk/c2t0d0
```

Display the rebuild status on disk array identified by alias **AUTORAIID3**:

```
arraybld -p AUTORAIID3
```

Set Auto Rebuild on and Rebuild Priority low on the disk array identified by device file */dev/rdisk/c2t0d0*:

```
arraybld -a on -P low /dev/rdisk/c2t0d0
```

**DEPENDENCIES**

**ARMServer** must be running to execute this command. See *ARMServer(1M)*.

**SECURITY CONFIGURATION**

This command is modified for all security configurations.

**Security Behavior/Restrictions**

Use of this command is restricted to authorized users only.

**Command Authorizations**

This command requires the **sysadmin** authorization to successfully execute.

**Privileges**

The command has been modified to support least privilege. The potential privileges possessed by the command and their uses include:

**allowdacread** This privilege is raised to provide discretionary read access to the devices.

**allowdacwrite** This privilege is raised to provide discretionary write access to the devices.

**allowmacread** This privilege is raised to provide mandatory read access to the devices.

**allowmacwrite** This privilege is raised to provide mandatory write access to the devices.



**fileysops**

This privilege is raised to allow the *mknod(2)* system call to succeed.

**writeaudit**

The command generates its own audit records and submits these directly to the system audit trail. This privilege is raised whenever the command needs to write an audit record.

**AUTHOR**

**arrayrbld** was developed by HP.

**SEE ALSO**

ARMServer(1M), arraycfg(1M), arraydsp(1M), arrayfmt(1M), arraylog(1M), arraymgr(1M), arrayrecover(1M), download(1M), drivetest(1M), dteststat(1M), logprint(1M).

**a**

**NAME**

arrayrecover - reconstructs data mapping and array configuration in the event of the loss of NVRAM contents

**SYNOPSIS**

**arrayrecover** [ **[-s]** | **[-l[c]]** | **[-p]** | **[-v *volume-set-id*]** ] **[-c]** **[-override]** **[-?]** *array-id*

**DESCRIPTION**

**arrayrecover** reconstructs data mapping and array configuration in the event of the loss of NVRAM contents. The data maps are reconstructed using the latest copy of mapping information stored on the disk drives. Part of the recovery operation includes performing a parity scan on the contents of the entire disk array to validate the accuracy of the maps and to correct any drive parity inconsistencies. This process can take up to several hours depending on the amount of data on the disk array.

The *array-id* used to address the disk array can be either the disk array serial number, the character device file name of any LUN on the array (LUN 0 if no LUNs are created), or the alias text string assigned to the disk array.

**Options**

**arrayrecover** supports the following options:

- |                                |   |
|--------------------------------|---|
| <b>none</b>                    | Begin the recovery in interactive mode. The utility will display all recoverable volume sets, then prompt for the number of the volume set to recover.  |
| <b>-c</b>                      | Used with other options, this will cause continuous polling during the recovery. The recovery is monitored and the current progress is displayed at regular intervals.  |
| <b>-l</b>                      | Displays the array controller logs on standard output.  |
| <b>-lc</b>                     | Displays the array controller logs on standard output, and then clears the logs.  |
| <b>-p</b>                      | Begin the recovery in non-interactive mode. If there are multiple volume sets on the disk array, a list of the volume set numbers will be displayed. If there is only one volume set on the disk array, the recovery will be done on it. This option provides a mechanism to pass volume set information to a script designed to perform a recovery.        |
| <b>-s</b>                      | Returns the recoverability status of the disk array, as well as the status of a recovery in progress. The status indicates if recovery is needed, and what percentage of the entire recovery has been completed. When used with the <b>-c</b> option, status will be returned at regular intervals allowing continuous monitoring of the recovery progress. |
| <b>-v <i>volume-set-id</i></b> | Starts a recovery on the volume set indicated by <i>volume-set-id</i> . This option is intended for use only in environments where there are multiple volume sets on the disk array.  |
| <b>-override</b>               | When specified with other options this will bypass checks on warning states and allow a recovery command to be issued. This option should be used with caution.   |
| <b>-?</b>                      | Display expanded usage message. This option overrides all others.   |

**EXTERNAL INFLUENCES****Environment Variables**

**LC\_MESSAGES** determines the language in which messages are displayed. The current language settings can be checked with *locale(1)*.

**RETURN VALUE**

**arrayrecover** returns the following values:

- 0 Successful completion.
- 1 An error in execution (I/O, subsystem, security, etc.) occurred.
- 2 An error in command syntax occurred: For example, an unknown command-line option was passed.
- 3 Timeout in communication to server. May indicate the server is not running.

**DIAGNOSTICS**

The following messages can be generated by **arrayrecover**:

Usage: arrayrecover [[-s] | [-l[c]] | [-p] | [-v volume-set-id]]  
 [-c] [override] [-?]<array-id>

An error in command syntax has occurred. Re-enter the command with all necessary arguments.

**arrayrecover: Arg out of range**

One of the arguments has exceeded its maximum or minimum size, or is incorrect in form. Check the size and form of each argument.

**arrayrecover: Unknown argument**

An invalid argument was specified. Check command usage.

**arrayrecover: No such file or directory**

The specified *array-id* does not exist or does not identify a disk array that is communicating with the system. Verify the correct *array-id* with *ioscan*(1M).

**arrayrecover: Error in command execution, <Additional Error Info>:**

**<Error Info Decode>**

The command failed due to a device error, an internal error, or a system error. The *Additional Error Info* and *Error Info Decode* fields will hold specifics about the failure and its cause.

## EXAMPLES

Perform a recovery on the default volume set on disk array serial number 00786b5c0000:

```
arrayrecover -p 00786b5c0000
```

Retrieve status for the recovery in progress on the disk array serial number 00786b5c0000. Also have the status continually updated until completion.

```
arrayrecover -s -c 00786b5c0000
```

## DEPENDENCIES

**arrayscan** must be installed and **ARMserver** must be running to execute this command. See *ARMserver*(1M).

## SECURITY CONFIGURATION

This command is modified for all security configurations.

### Security Behavior/Restrictions

Use of this command is restricted to authorized users only.

### Command Authorizations

This command requires the **sysadmin** authorization to successfully execute.

### Privileges

The command has been modified to support least privilege. The potential privileges possessed by the command and their uses include:

**allowdacread** This privilege is raised to provide discretionary read access to the devices.

**allowdacwrite** This privilege is raised to provide discretionary write access to the devices.

**allowmacread** This privilege is raised to provide mandatory read access to the devices.

**allowmacwrite** This privilege is raised to provide mandatory write access to the devices.

**filesysops** This privilege is raised to allow the *mknod*(2) system call to succeed.

**writeaudit** The command generates its own audit records and submits these directly to the system audit trail. This privilege is raised whenever the command needs to write an audit record.

## AUTHOR

**arrayrecover** was developed by HP.

## SEE ALSO

*ARMserver*(1M), *arraycfg*(1M), *arraydsp*(1M), *arrayfmt*(1M), *arraylog*(1M), *arraymgr*(1M), *arrayrbld*(1M), *download*(1M), *drivetest*(1M), *dteststat*(1M), *logprint*(1M).

**NAME**

arrayscan - search system for disk arrays

**SYNOPSIS**

arrayscan

**DESCRIPTION**

**arrayscan** searches the system I/O buses to locate the address(es) of attached HP disk array devices. The utility can also be used to determine which logical units are configured on a disk array.

**arrayscan** performs several functions, including:

- Ensuring device special files exist.  
**arrayscan** verifies that block and character device special files exist for all LUNs configured. On Series 700 systems, device files are created for all possible LUNs.
- Ensuring disk array software was downloaded.  
**arrayscan** verifies that the disk array software has been downloaded for each disk array it encounters. If **arrayscan** encounters a disk array that does not have disk array software loaded, it automatically downloads the array software.
- Updating **monitor**, and **pscan** device lists.  
Two files, `/etc/hpC2400/hparray.devs`, and `/etc/hpC2400/hparray.luns` are updated by **arrayscan**. `/etc/hpC2400/hparray.devs` is used by the monitor daemon (`/usr/sbin/hpC2400/arraymond`) to determine which devices to monitor. `/etc/hpC2400/hparray.luns` is used by the parity scan utilities (**pscan**, **scn**, and **rpr**) to determine which LUNs to monitor.

**RETURN VALUE**

**arrayscan** returns the following values:

- 0 Successful completion
- 1 Command failed (an error occurred).

**DIAGNOSTICS AND ERRORS**

Errors can originate from problems with:

- **arrayscan**
- SCSI (device level) communications
- system calls

**Error messages generated by arrayscan:**

**arrayscan:** Cannot access lock file. Create an empty file <FILE>

Two semaphore files are used by **arrayscan**, `/etc/hpC2400/pscan.lock`, and `/etc/hpC2400/monitor.lock`. If these files do not exist when **arrayscan** begins, it assumes that the monitor daemon is executing. If the period of time required for the monitor daemon to execute expires, and the files still do not exist, it is assumed that they need to be created. You can create these files, if necessary, using the **touch** command (see *touch(1)*);

**arrayscan:** Unable to open Array Parity Scan list <FILE>

**arrayscan** updates `/etc/hpC2400/hparray.luns`, and `/etc/hpC2400/hparray.devs`. **arrayscan** was unable to write to this file.

**arrayscan:** Error from process **insf**.

An error occurred while executing **insf** (see *insf(1M)*). **insf** is used by **arrayscan** on Series 800 systems to create device files for newly configured disk array devices.

**arrayscan:** Error from process **ioscan**.

An error occurred while executing **ioscan**. **ioscan** is used by **arrayscan** to scan for all devices. Disk array devices are filtered from the **ioscan** output.

**arrayscan:** No SCSI devices identified. Check SCSI connections.

No SCSI devices were identified. Check SCSI cables and power connections and retry the command.

```
arrayscan: Unable to create char device special file for path <FILE>
arrayscan will create character, and block device files for all disk array devices it encounters.
arrayscan was unable to create the device file.

arrayscan: Insufficient dynamic memory
An attempt to allocate dynamic memory failed.
```

**DEPENDENCIES**

This utility is supported only on HP C2425D, HP C2430D, HP C3595A and HP C3596A disk array devices.

**AUTHOR**

**arrayscan** was developed by HP.

**FILES**

```
/etc/hpC2400/hparray.luns
/etc/hpC2400/hparray.devs
/etc/hpC2400/pscan.lock
/etc/hpC2400/monitor.lock
```

**NAME**

asecure - control access to Audio on a workstation

**SYNOPSIS**

```
/opt/audio/bin/asecure [-Cdelp] [+h host] [-h host] [+p user] [-p user]
[+u user] [-u user] [+b host,user] [-b host,user]
```

**DESCRIPTION**

On Series 700 workstations, audio is secured so that only the user on the local workstation can access audio. You use the **asecure** command to modify audio security. This command does not apply to X stations; on an X station, access to audio is unrestricted.

To modify audio security, become root on the local workstation where you want make a change. Then, use **asecure** as follows:

```
/opt/audio/bin/asecure -C
```

When prompted, enter any meaningful password. Issuing **asecure -C** creates the Audio Security File (ASF). The ASF contains information that determines which hosts and users can access the Aserver, and which users (other than the superuser) can modify the ASF.

If needed, you can allow unrestricted access to audio on this workstation. To remove audio security, issue this command:

```
/opt/audio/bin/asecure -d
```

If instead, you wish to modify security, you use **asecure** to make changes to the information in the ASF. (Because the ASF is a binary file, we do not recommend using an editor on this file.) You can use **asecure** to make these types of changes:

- Allow all clients from a remote host to access the server.
- Allow specific users from all other hosts to access the server.
- Allow a specific user from a specific host to access the server.
- Disable access control, allowing complete unrestricted access to the server, but leaving the ASF intact.

Every operation that creates, reinitializes, or changes the contents of the ASF is logged in the `/var/adm/audio/asecure_log` file, so that you can track any changes to the ASF.

**OPTIONS**

**asecure** supports the following options:

**+b** | **-b** *host,user*

Add/delete *hostname,username* pair. You must be either superuser or a **privileged user** to do this. You can supply more than one *hostname,username* pair separated by blanks.

To use either the **+b** or **-b** options, you **MUST** supply at least one *hostname,username* pair. This option will not work without a pair.

**-C**

Create a new ASF file, called the **audio.sec** file. Access control default is enabled with no entries in the access list. Aserver can now be accessed only by local users on the host machine. If an **audio.sec** file already exists, it is re-initialized.

You must be superuser to execute this option. This option is mutually-exclusive of all other options.

This option requires a password. This is an extra layer of protection for the contents of the ASF. It is designed to prevent surreptitious manipulation of the ASF. If you are creating a new ASF, you are prompted for a password and an encrypted copy of that password is stored in the new ASF.

If the ASF already exists, you are prompted for the password. If your password matches the password stored in the ASF, the ASF is then re-initialized.

**-d**

Disable access control to the Aserver. This allows unrestricted access by all clients.

**-e**

Enable access control to the Aserver. This restricts access to clients listed in the ASF. Enabled is the default state.

- +h** | **-h** *host* Add/delete *hostnames* for ALL users. You must be either superuser or a **privileged user** to do this. You can supply more than one *hostname* separated by blanks.
- l** List the contents of the ASF. This option shows a list of the *hostnames* and/or *usernames* that have access to the Aserver.
- P** Change password for **audio.sec** file. You must be superuser to do this. You are prompted once for the old password, then prompted twice for the new password.
- +p** | **-p** *user* Add/delete **privileged users**. You must be superuser to do this and must enter the password given when the ASF was created (see **-C** option). To see a list of privileged users, you must be superuser and use the **-l** option.
- +u** | **-u** *user* Add/delete *usernames* for ALL hosts. You must be either superuser or a **privileged user** to do this. You can supply more than one *username* separated by blanks.

## EXAMPLES

List entries in access list.

```
/opt/audio/bin/asecure -l
```

Disable access control. This means anyone can connect to Aserver without restriction.

```
/opt/audio/bin/asecure -d
```

Add **moonbeam** host for all users to access list. Remove **pluto** host for all users from access list.

```
/opt/audio/bin/asecure +h moonbeam -h pluto
```

Add user **comet** for hosts **saturn** and **mercury** to access list.

```
/opt/audio/bin/asecure +b saturn,comet mercury,comet
```

Add user **comet** to access list for all hosts. Remove users **venus** and **neptune** from access list for all hosts.

```
/opt/audio/bin/asecure +u comet -u venus neptune
```

Create new access list.

```
/opt/audio/bin/asecure -C
```

## AUTHOR

**asecure** was developed by HP.

## FILES

```
/var/opt/audio/asecure_log  asecure log pathname
/etc/opt/audio/audio.sec    ASF pathname
```

## SEE ALSO

audio(5), asecure(1M), aserver(1M), attributes(1), convert(1), send\_sound(1).

*Using the Audio Developer's Kit*

**NAME**

Aserver - start the audio server

**SYNOPSIS**

```
/opt/audio/bin/Aserver -f
```

**DESCRIPTION**

The **Aserver** command starts the HP-UX Audio server, which can run on a system with audio hardware. See *Audio(5)* for information about which systems have audio hardware. The **-f** option forces the starting of the Audio server; this option is only needed if the Aserver has problems starting.

**The Audio Server**

Before using any audio tools such as the **Audio Editor**, the system or X station must be running two audio server processes, called **Aserver**. On a Series 700, the Remote Procedure Call daemon (**rpcd**) must also be running.

Normally, the Aserver processes and **rpcd** start automatically when the system is booted. If problems occur on an ENTRIA or ENVIZEX X station, see the X station owner's manual. On a Series 700 Audio hardware, first check if **rpcd** is running. Type the following:

```
ps -e | grep rpcd
```

If it is running, you see a line similar to the following.

```
604 ? 0:36 rpcd
```

If it is not running, see HP 9000/DCE documentation for information on restarting it. If **rpcd** is running, verify that the Aserver is running. Type:

```
ps -e | grep Aserver
```

If the Aserver is running you will see lines similar to the following, which indicate the presence of the two Aserver processes:

```
1 ? 0:00 Aserver
224 ? 0:00 Aserver
```

If it is not running, become root and restart it as follows:

```
/opt/audio/bin/Aserver
```

If it fails to start, reissue the command with the **-f** option:

```
/opt/audio/bin/Aserver -f
```

**Using Audio over the Network**

From a workstation, you can also use the Audio Editor and Control Panel over the network. However, the remote system is where the actual playback and recording occur.

The local workstation (or audio client) can be any Series 700 system. The remote system (or audio server) can be a Series 700 or an X station with audio hardware and must have the Aserver processes running. If the server is a workstation, it must also allow access from remote clients (see *asecure(1M)*) and must have **rpcd** running.

To make the system an audio client, set the **AUDIO** variable by modifying the **\$HOME/.vueprofile** file as follows:

```
Korn, Bourne, and POSIX Shells:  AUDIO=system_name; export AUDIO
C Shell:                          setenv AUDIO system_name
```

For *system\_name*, identify the workstation or X Station running the Aserver.

If the **AUDIO** variable is not set, the Audio Library attempts to use the Aserver on the system defined by the **DISPLAY** variable. If neither **DISPLAY** nor **AUDIO** is set, the Aserver on the local machine is used.

**DEPENDENCIES**

The Audio Server must run on a system that has audio hardware. Note that HP-UX for the 8MB 705 System does not include audio software.



**AUTHOR**

The Audio Server was developed by HP.

**SEE ALSO**

audio(5), asecure(1M), attributes(1), convert(1), send\_sound(1).

*Using the Audio Developer's Kit*

  
**a**

**NAME**

audevent - change or display event or system call audit status

**SYNOPSIS**

**audevent** [-P|-p] [-F|-f] [-E] [[-e *event*] ...] [-S] [[-s *syscall*] ...]

**DESCRIPTION**

**audevent** changes the auditing status of the given events or system calls. The *event* is used to specify names associated with certain self-auditing commands; *syscall* is used to select related system calls.

If neither **-P**, **-p**, **-F**, nor **-f** is specified, the current status of the selected events or system calls is displayed. If no events or system calls are specified, all events and system calls are selected.

If the **-E** option is supplied, it is redundant to specify events with the **-e** option; this applies similarly to the **-S** and **-s** options.

**audevent** takes effect immediately. However, the events and system calls specified are audited only when called by a user currently being audited (see *audusr*(1M)). A list of valid events and associated syscalls is provided in *audit*(5).

Only the super-user can change or display audit status.

**Options**

**audevent** recognizes the following options and command-line arguments:

- P**            Audit successful events or system calls.
- p**            Do not audit successful events or system calls.
- F**            Audit failed events or system calls.
- f**            Do not audit failed events or system calls.
- E**            Select all events for change or display.
- e *event***    Select *event* for change or display.
- S**            Select all system calls for change or display.
- s *syscall***   Select *syscall* for change or display.

The following is a list of the valid *events* and the associated *syscalls* (if any):

- create**        Object creation (*creat()*, *mknod()*, *msgget()*, *pipe()*, *semget()*, *shmat()*, *shmget()*)
- delete**        Object deletion (*ksem\_unlink()*, *mq\_unlink()*, *msgctl()*, *rmdir()*, *semctl()*, *shm\_unlink()*)
- readdac**       Discretionary access control (DAC) information reading (*access()*, *fstat()*, *fstat64()*, *getaccess()*, *lstat()*, *lstat64()*, *stat()*, *stat64()*)
- moddac**        Discretionary access control (DAC) modification (*acl()*, *chmod()*, *chown()*, *fchmod()*, *fchown()*, *fsetacl()*, *lchmod()*, *lchown()*, *putpmsg()*, *semop()*, *setacl()*, *umask()*)
- modaccess**    Non-DAC modification (*chdir()*, *chroot()*, *link()*, *lockf()*, *lockf64()*, *rename()*, *setgid()*, *setgroups()*, *setpgid()*, *setpgrp()*, *setregid()*, *setresgid()*, *setresuid()*, *setsid()*, *setuid()*, *shmctl()*, *shmdt()*, *symlink()*, *unlink()*)
- open**          Object opening (*execv()*, *execve()*, *ftruncate()*, *ftruncate64()*, *kload()*, *ksem\_open()*, *mmap()*, *mmap64()*, *mq\_open()*, *open()*, *ptrace()*, *shm\_open()*, *truncate()*, *truncate64()*)
- close**          Object closing (*close()*, *ksem\_close()*, *mq\_close()*, *munmap()*)
- process**       Process operations (*exit()*, *fork()*, *kill()*, *mlock()*, *mlockall()*, *munlock()*, *munlockall()*, *nsp\_init()*, *plock()*, *rtprio()*, *setcontext()*, *setrlimit64()*, *sigqueue()*, *ulimit64()*, *vfork()*)
- removable**    Removable media events (*exportfs()*, *mount()*, *umount()*, *vfsmount()*)

<b>login</b>	Logins and logouts
<b>admin</b>	administrative and superuser events ( <b>acct()</b> , <b>adjtime()</b> , <b>audctl()</b> , <b>audswitch()</b> , <b>clock_settime()</b> , <b>mpctl()</b> , <b>reboot()</b> , <b>sched_setparam()</b> , <b>sched_setscheduler()</b> , <b>serialize()</b> , <b>setauditid()</b> , <b>setaudproc()</b> , <b>setdomainname()</b> , <b>setevent()</b> , <b>sethostid()</b> , <b>setpriority()</b> , <b>setprivgrp()</b> , <b>settimeofday()</b> , <b>stime()</b> , <b>swapon()</b> , <b>toolbox()</b> , <b>utssys()</b> )
<b>ipccreat</b>	Interprocess Communication (IPC) object creation ( <b>bind()</b> , <b>ipccreate()</b> , <b>ipcd-est()</b> , <b>socket()</b> , <b>socket2()</b> , <b>socketpair()</b> )
<b>ipcopen</b>	IPC object opening ( <b>accept()</b> , <b>connect()</b> , <b>fattach()</b> , <b>ipccconnect()</b> , <b>ipclookup()</b> , <b>ipcrecvcn()</b> )
<b>ipcclose</b>	IPC object deletion ( <b>fdetach()</b> , <b>ipcsshutdown()</b> , <b>shutdown()</b> )
<b>ipcdgram</b>	IPC datagram ( <b>sendto()</b> and <b>recvfrom()</b> )
<b>uevent1</b>	User-defined event 1
<b>uevent2</b>	User-defined event 2
<b>uevent3</b>	User-defined event 3

**WARNINGS**

All modifications made to the audit system are lost upon reboot. To make the changes permanent, set **AUDEVENT\_ARGS1**, **AUDEVENT\_ARGS2**, or **AUDEVENT\_ARGS3** in **/etc/rc.config.d/auditing**.

**AUTHOR**

**audevent** was developed by HP.

**SEE ALSO**

**audisp(1M)**, **audomon(1M)**, **audsys(1M)**, **audusr(1M)**, **getevent(2)**, **setevent(2)**, **audit(4)**, **audit(5)**.



## NAME

**audisp** - display the audit information as requested by the parameters

## SYNOPSIS

```
audisp [-u username] [-e eventname] [-c syscall] [-p] [-f] [-l ttyid] [-t start_time]
        [-s stop_time] [-y2|-y4] audit_filename ...
```

## DESCRIPTION

**audisp** analyzes and displays the audit information contained in the specified *audit\_filename* audit files. The audit files are merged into a single audit trail in time order. Although the entire audit trail is analyzed, **audisp** allows you to limit the information displayed, by specifying options. This command is restricted to privileged users.

Any unspecified option is interpreted as an unrestricted specification. For example, a missing **-u** *username* option causes all users' audit information in the audit trail to be displayed as long as it satisfies all other specified options. By the same principle, citing **-t** *start\_time* without **-s** *stop\_time* displays all audit information beginning from *start\_time* to the end of the file.

**audisp** without any options displays all recorded information from the start of the audit file to the end.

Specifying an option without its required parameter results in error. For example, specifying **-e** without any *eventname* returns with an error message.

## Options

- u** *username* Specify the login name (*username*) about whom to display information. If no (*username*) is specified, **audisp** displays audit information about all users in the audit file.
- e** *eventname* Display audit information of the specified event types. The defined event types are **admin**, **close**, **create**, **delete**, **ipcclose**, **ipccreat**, **ipcdgram**, **ipcopen**, **login**, **modaccess**, **moddac**, **open**, **process**, **readdac**, **removable**, **uevent1**, **uevent2**, and **uevent3** (see *audevent(1M)*).
- c** *syscall* Display audit information about the specified system calls.
- p** Display only successful operations that were recorded in the audit trail. No user event that results in a failure is displayed, even if *username* and *eventname* are specified.  
  
The **-p** and the **-f** options are mutually exclusive; do not specify both on the same command line. To display both successful and failed operations, omit both **-p** and **-f** options.
- f** Display only failed operations that are recorded in the audit trail.
- l** *ttyid* Display all operations that occurred on the specified terminal (*ttyid*) and were recorded in the audit trail. By default, operations on all terminals are displayed.
- t** *start\_time* Display all audited operations occurring since *start\_time*, specified as *mmddhhmm[yy]* (month, day, hour, minute, year). If the year is specified and is greater than 70, it is interpreted as in the twentieth century. Otherwise, it is interpreted as in the twenty-first century. If no year is given, the current year is used. No operation in the audit trail occurring before the specified time is displayed.
- s** *stop\_time* Display all audited operations occurring before *stop\_time*, specified as *mmddhhmm[yy]* (month, day, hour, minute, year). If the year is specified and is greater than 70, it is interpreted as in the twentieth century. Otherwise, it is interpreted as in the twenty-first century. If no year is given, the current year is used. No operation in the audit trail occurring after the specified time is displayed.
- y2|-y4** The year is displayed as a two digit number (with **-y2**), or as a four digit number (with **-y4**). The default is **-y2**. Note that *start\_time* and *stop\_time* must still be specified as two digit numbers.

## AUTHOR

**audisp** was developed by HP.

## SEE ALSO

*audevent(1M)*, *audit(4)*, *audit(5)*.

**NAME**

audomon - audit overflow monitor daemon

**SYNOPSIS**

```
/usr/sbin/audomon [-p fss] [-t sp_freq] [-w warning] [-v] [-o output_tty]
```

**DESCRIPTION**

**audomon** monitors the capacity of the current audit file and the file system on which the audit file is located, and prints out warning messages when either is approaching full. It also checks the audit file and the file system against 2 switch points: *FileSpaceSwitch* (FSS) and *AuditFileSwitch* (AFS) and if either is reached, audit recording automatically switches to the backup audit file if it is available.

The *FileSpaceSwitch* (FSS) is specified as a percentage of the total disk space available. When the file system reaches this percentage, **audomon** looks for a backup audit file. If it is available, recording is switched from the audit file to the backup file.

The *AuditFileSwitch* (AFS) is specified (using *audsys*(1M)) by the size of the audit file. When the audit file reaches the specified size, **audomon** looks for a backup audit file. If it is available, recording is switched from the audit file to the backup file (see *audsys*(1M) for further information on use of this parameter).

If either switch point is reached but no backup file is available, **audomon** issues a warning message.

**audomon** is typically spawned by */sbin/init.d/auditing* (as part of the *init*(1M) start-up process) when the system is booted up. Once invoked, **audomon** monitors, periodically sleeping and “waking up” at intervals. Note that **audomon** does not produce any messages when the audit system is disabled.

**audomon** is restricted to privileged users.

**Options**

- p *fss*** Specify the *FileSpaceSwitch* by a number ranging from 0 to 100. When the audit file's file system has less than *fss* percent free space remaining, **audomon** looks for a backup file. If available, the backup file is designated as the new audit file. If no backup file is available, **audomon** issues a warning message.  
  
The *fss* parameter should be a larger number than the *min\_free* parameter of the file system to ensure that the switch takes place before *min\_free* is reached. By default, *fss* is 20 percent.
- t *sp\_freq*** Specify the wake-up switch-point frequency in minutes. The wake-up frequency at any other time is calculated based on *sp\_freq* and the current capacity of the audit file and the file system. The calculated wake-up frequency at any time before the switch points is larger than *sp\_freq*. As the size of the audit file or the file system's free space approaches the switch points, the wake-up frequency approaches *sp\_freq*. *sp\_freq* can be any positive real number. Default *sp\_freq* is 1 (minute).
- w *warning*** Specify that warning messages be sent before the switch points. *warning* is an integer ranging from 0 through 100. The higher the *warning*, the closer to the switch points warning messages are issued. For example, *warning* = 50 causes warning messages to be sent half-way before the switch points are reached. *warning* = 100 causes warning messages to be sent only after the designated switch points are reached and a switch is not possible due to a missing backup file. By default, *warning* is 90.
- v** Make audomon more verbose. This option causes **audomon** to also print out the next wake-up time.
- o *output\_tty*** Specify the tty to which warning messages are directed. By default, warning messages are sent to the console. Note that this applies only to the diagnostic messages **audomon** generates concerning the status of the audit system. Error messages caused by wrong usage of **audomon** are sent to the standard output (where **audomon** is invoked).

**WARNINGS**

All modifications made to the audit system are lost upon reboot. To make the changes permanent, set **AUDOMON\_ARGS** in */etc/rc.config.d/auditing*.

**AUTHOR**

**audomon** was developed by HP.

**SEE ALSO**

audsys(1M), audit(5).

**a**

**NAME**

audsys - start or halt the auditing system and set or display audit file information

**SYNOPSIS**

**audsys** [-nf] [-c *file* -s *cafs*] [-x *file* -z *xafs*]

**DESCRIPTION**

*audsys* allows the user to start or halt the auditing system, to specify the auditing system "current" and "next" audit files (and their switch sizes), or to display auditing system status information. This command is restricted to super-users.

The "current" audit file is the file to which the auditing system writes audit records. When the "current" file grows to either its Audit File Switch (AFS) size or its File Space Switch (FSS) size (see *audomon*(1M)), the auditing system switches to write to the "next" audit file. The auditing system switches audit files by setting the "current" file designation to the "next" file and setting the new "next" file to NULL. The "current" and "next" files can reside on different file systems.

When invoked without arguments, *audsys* displays the status of the auditing system. This status includes information describing whether auditing is on or off, the names of the "current" and "next" audit files, and a table listing their switch sizes and the sizes of file systems on which they are located, as well as the space available expressed as a percentage of the switch sizes and file system sizes.

**Options**

*audsys* recognizes the following options:

- n Turn on the auditing system. The system uses existing "current" and "next" audit files unless others are specified with the -c and -x options. If no "current" audit file exists (such as when the auditing system is first installed), specify it by using the -c option.
- f Turn off the auditing system. The -f and -n options are mutually exclusive. Other options specified with -f are ignored.
- c *file* Specify a "current" file. Any existing "current" file is replaced with the *file* specified; the auditing system immediately switches to write to the new "current" file. The specified *file* must be empty or nonexistent, unless it is the "current" or "next" file already in use by the auditing system.
- s *cafs* Specify *cafs*, the "current" audit file switch size (in kbytes).
- x *file* Specify the "next" audit file. Any existing "next" file is replaced with the *file* specified. The specified *file* must be empty or nonexistent, unless it is the "current" or "next" file already in use by the auditing system.
- z *xafs* Specify *xafs*, the "next" audit file switch size (in kbytes).

If -c but not -x is specified, only the "current" audit file is changed; the existing "next" audit file remains. If -x but not -c is specified, only the "next" audit file is changed; the existing "current" audit file remains.

The -c option can be used to manually switch from the "current" to the "next" file by specifying the "next" file as the new "current" file. In this instance, the file specified becomes the new "current" file and the "next" file is set to NULL.

In instances where no next file is desired, the -x option can be used to set the "next" file to NULL by specifying the existing "current" file as the new "next" file.

The user should take care to select audit files that reside on file systems large enough to accommodate the Audit File Switch (AFS) desired. *audsys* returns a non-zero status and no action is performed, if any of the following situations would occur:

- The Audit File Switch size (AFS) specified for either audit file exceeds the space available on the file system where the file resides.

- The AFS size specified for either audit file is less than the file's current size.

- Either audit file resides on a file system with no remaining user space (exceeds minfree).

**WARNINGS**

All modifications made to the audit system are lost upon reboot. To make the changes permanent, set **AUDITING**, **PRI\_AUDFILE**, **PRI\_SWITCH**, **SEC\_AUDFILE**, and **SEC\_SWITCH** in */etc/rc.config.d/auditing*.

  
**a**

A user process will be blocked in the kernel if all of the following events occurs:

- the file system containing current audit file is full,
- there is no next audit file or the next audit file is removed, and
- the user process makes an auditable system call or generates an auditable event.

To recover from the resulting deadlock, the session leader of the console is killed so that the administrator can login. Hence sensitive applications should not be run as session leaders on the console.

**AUTHOR**

**audsys** was developed by HP.

**FILES**

**/ .secure/etc/audnames** File maintained by **audsys** containing the "current" and "next" audit file names and their switch sizes.

**SEE ALSO**

audit(5), audomon(1M), audctl(2), audwrite(2), audit(4), setsid(2).



**NAME**

**audusr** - select users to audit

**SYNOPSIS**

**audusr** [[-a *user*] ...] [[-d *user*] ...] [-A|-D]

**DESCRIPTION**

**audusr** is used to specify *users* to be audited or excluded from auditing. If no arguments are specified, **audusr** displays the audit setting of every user. **audusr** is restricted to super-users.

**Options**

**audusr** recognizes the following options:

- a *user*      Audit the specified *user*. The auditing system records audit records to the “current” audit file when the specified *user* executes audited events or system calls. Use **audevent** to specify events to be audited (see *audevent(1M)*).
- d *user*      Do not audit the specified *user*.
- A            Audit all users.
- D            Do not audit any users.

The -A and -D options are mutually exclusive: that is, if -A is specified, -d cannot be specified; if -D is specified, -a cannot be specified.

Users specified with **audusr** are audited (or excluded from auditing) beginning with their next login session, until excluded from auditing (or specified for auditing) with a subsequent **audusr** invocation. Users already logged into the system when **audusr** is invoked are unaffected during that login session; however, any user who logs in after **audusr** is invoked is audited or excluded from auditing accordingly.

**AUTHOR**

**audusr** was developed by HP.

**FILES**

/tcb/files/auth/\*/\*      File containing flags to indicate whether users are audited.

**SEE ALSO**

*audevent(1M)*, *setaudproc(2)*, *audswitch(2)*, *audwrite(2)*, *audit(5)*.

**NAME**

authck - check internal consistency of Authentication database

**SYNOPSIS**

**authck** [-p] [-t] [-a] [-v] [-d [ *domainname* ]]

**DESCRIPTION**

**authck** checks both the overall structure and internal field consistency of all components of the Authentication database. It reports all problems it finds. Only users who have the *superuser* capability can run this command. When **pwck** is used with the **-s** option, **authck** is run with the **-p** option automatically.

**Options**

**authck** recognizes the following options and tests:

- p** Check the Protected Password database. The Protected Password database and **/etc/passwd** are checked for completeness such that neither contains entries not in the other. The cross references between the Protected Password database and **/etc/passwd** are checked to make sure that they agree. However, if **Nis+** is configured in your system, the password table is also checked before reporting a discrepancy. This means that a discrepancy would not be reported for a user that does NOT exist in **/etc/passwd** but exists in the Protected Password database as well as the **Nis+ passwd** table. Fields in the Protected Password database are then checked for reasonable values. For example, all time stamps of past events are checked to make sure that they have times less than the times returned by *time(2)*.
- t** Fields in the Terminal Control database are checked for reasonable values. All time stamps of past events are checked to make sure they have times less than those returned by *time(2)*.
- a** Shorthand equivalent of using the **-p** and **-t** options together in a single command.
- v** Provide running diagnostics as the program proceeds. Produce warnings when unusual conditions are encountered that might not cause program errors in *login*, *password* and *su* programs.
- d** Removes Protected Password database entries that are not found in the **Nis+ passwd** table. **Nis+** users may have an entry in the Protected database and not in **/etc/passwd**. Thus, this option removes orphaned Protected database entries: orphaned entries can exist for deleted **Nis+** users. The optional *domainname* specifies the desired **Nis+** domain to use for the **passwd** table. If *domainname* is not specified, the local domain name is used.

**FILES**

<b>/etc/passwd</b>	System password file
<b>/tcb/files/auth/**</b>	Protected Password database
<b>/tcb/files/ttys</b>	Terminal Control database
<b>/tcb/files/auth/system/default</b>	System Defaults database
<b>/usr/sbin/authck</b>	

**AUTHOR**

**authck** was developed by HP.

**SEE ALSO**

getprpwent(3), getprtcent(3), getprdfent(3), authcap(4).

## NAME

**automount** - install automatic mount points (**autofs**) or automatically mount NFS file systems; see "Remarks" below in SYNOPSIS.

## SYNOPSIS

```
/usr/sbin/automount [-f master-file] [-t duration] [-v]
```

## Remarks

This manpage contains two versions of **automount**. The **autofs** version is presented first, followed by the previous automounter version.

The **/usr/sbin/automount** script checks the **AUTOFS** variable in **/etc/rc.config.d/nfsconf**. If the **AUTOFS** variable is set to one, then **/usr/lib/netsvc/fs/autofs/automount** is executed. The first half of this manpage represents the **autofs automount**.

If the **AUTOFS** variable is set to 0 (zero) or does not exist in **/etc/rc.config.d/nfsconf**, then the **automount** daemon is executed in **/usr/lib/netsvc/fs/automount/automount**. The second half of this manpage represents the older **automount**.

In future releases, only the **autofs** version of **automount** will be supported.

## DESCRIPTION

**automount** is a command that installs **autofs** mount points and associates an **automount** map with each mount point. The **autofs** filesystem monitors attempts to access directories within it and notifies the **automountd** daemon (See **automountd(1M)**). The daemon uses the map to locate a filesystem, which it then mounts at the point of reference within the **autofs** filesystem. You can assign a map to an **autofs** mount using an entry in the **/etc/auto\_master** map or a direct map.

If the file system is not accessed within an appropriate interval (five minutes by default), the **automountd** daemon unmounts the file system.

The file **/etc/auto\_master** determines the locations of all **autofs** mount points. By default, this file contains the following entry:

```
# Master map for automounter
#
/net          -hosts          -soft
```

The first field in the master file specifies a directory on which an **autofs** mount will be made, and the second field specifies the automounter map to be associated with it. Mount options may be supplied as an optional third field in the entry. These options are used for any entries in the map that do not specify mount options explicitly. The **automount** command is usually run without arguments. It compares the entries **/etc/auto\_master** with the current list of **autofs** mounts in **/etc/mnttab** and adds, removes or updates **autofs** mounts to bring the **/etc/mnttab** up to date with the **/etc/auto\_master**. At boot time it installs all **autofs** mounts from the master map. Subsequently, it may be run to install **autofs** mounts for new entries in the master map or a direct map, or to perform unmounts for entries that have been removed.

The automounter maps, including the **auto\_master** map, may be distributed by NIS or NIS+. The Name Service Switch configuration file, **/etc/nsswitch.conf**, determines where the **automount** command will look for the maps.

## Options

**-f** *master-file* Specify a local master file for initialization.

When the **-f** option is used and the master file specified is not found, then **automount** defaults to **/etc/auto\_master** and then to the NIS **auto\_master** map.

**-t** *duration* Specify a *duration*, in seconds, that a file system is to remain mounted when not in use. The default is 5 minutes.

**-v** Verbose mode. Notify of **autofs** mounts, unmounts or other non-essential information. Messages are written to **stderr**.

## Map Entry Format

A simple map entry (mapping) takes the form:

*key* [ *-mount-options* ] *location*...

where *key* is the full pathname of the directory to mount when used in a direct map, or the simple name of a subdirectory in an indirect map. *mount-options* is a comma-separated list of mount options, and *location* specifies a file system from which the directory may be mounted. In the case of a simple NFS mount, *location* takes the form:

*host:pathname*

*host* is the name of the host from which to mount the file system (it may be omitted if the pathname refers to a local device on which the filesystem resides) and *pathname* is the pathname of the directory to mount.

Default mount options can be assigned to an entire map when specified as an optional third field in the master map. These options apply only to map entries that have no mount options.

### Replicated Filesystems

Multiple *location* fields can be specified for replicated NFS filesystems, in which case **automount** chooses a server with preference given to a server on the local subnet or net.

If each *location* in the list shares the same *pathname* then a single *location* may be used with a comma-separated list of hostnames:

*hostname,hostname...:pathname*

The multiple location feature for NFS mounts allows the **automountd** daemon to choose the most appropriate server at mount time. While such a mount is in effect, the daemon does not monitor the status of the server. If the server crashes, **automountd** will not select an alternative server from the list.

Requests for a server may be weighted, with the weighting factor appended to the server name as an integer in parentheses. Servers without a weighting are assumed to have a value of zero (most likely to be selected). Progressively higher values decrease the chance of being selected. In the example,

```
man -ro alpha,bravo,charlie(1),delta(4):/usr/share/man
```

hosts **alpha** and **bravo** have the highest priority; host **delta**, has the lowest priority.

NOTE: Server proximity takes priority in the selection process. In the example above, if the server **delta** is on the same network segment as the client, but the others are on different network segments, then **delta** will be selected — the weighting value is ignored. The weighting has effect only when selecting between servers with the same network proximity.

In cases where each server has a different export point, you can still apply the weighting. For example:

```
man -ro alpha:/usr/man bravo,charlie(1):/usr/share/man
delta(3):/export/man
```

A mapping can be continued across input lines by escaping the NEWLINE with a `'\'` (backslash). Comments begin with a `'#'` (number sign) and end at the subsequent NEWLINE.

### Map Key Substitution

The `'&'` (ampersand) character is expanded to the value of the *key* field for the entry in which it occurs. In this case:

```
amy rowboatserver:/home/&
```

the `&` expands to **amy**.

### Wildcard Key

The `'*'` (asterisk) character, when supplied as the *key* field, is recognized as the catch-all entry. Such an entry will match any *key* not previously matched. For instance, if the following entry appeared in the indirect map for `/config`:

```
* &:/export/config/&
```

this would allow automatic mounts in `/config` of any remote file system whose location could be specified as:

```
hostname:/export/config/hostname
```

### Variable Substitution

Client specific variables can be used within an **automount** map. For instance, if `$HOST` appeared within a map, **automount** would expand it to its current value for the client's host name. Supported variables

## (autofs)

are:

**HOST** The output of **uname -n**. The host name. For example "rowboat"  
**OSNAME** The output of **uname -s**. The OS name. For example "HP-UX"  
**OSREL** The output of **uname -r**. The OS release name. For example "B.11.0"  
**OSVERS** The output of **uname -v**. The OS version. For example "C"

If a reference needs to be protected from affixed characters, you can surround the variable name with '{' (curly braces).

### Multiple Mounts

A multiple mount entry takes the form:

```
key [-mount-options] [[mountpoint] [-mount-options] location...]....
```

The initial `/[mountpoint]` is optional for the first mount and mandatory for all subsequent mounts. The optional *mountpoint* is taken as a pathname relative to the directory named by *key*. If *mountpoint* is omitted in the first occurrence, a *mountpoint* of `/` (root) is implied.

Given an entry in the indirect map for `/src`:

```
beta -ro
/      svr1,svr2:/export/src/beta
/1.0   svr1,svr2:/export/src/beta/1.0
/1.0/man svr1,svr2:/export/src/beta/1.0/man
```

**automount** would automatically mount `/src/beta`, `/src/beta/1.0`, and `/src/beta/1.0/man`, as needed, from either **svr1** or **svr2**, whichever host is nearest and responds first.

The autofs mount points must not be hierarchically related. **automount** does not allow an **autofs** mount point to be created within another **autofs** mount.

### Other Filesystem Types

The automounter assumes NFS mounts as a default filesystem type. Other filesystem types can be described using the **fstype** mount option. Other mount options specific to this filesystem type can be combined with the **fstype** option. The location field must contain information specific to the filesystem type. If the location field begins with a slash, a colon character must be prepended, for instance, to mount a CD filesystem:

```
cdrom-fstype=hsfs,ro :/dev/sr0
```

or to perform an **autofs** mount:

```
src -fstype=autofs auto_src
```

Mounts using CacheFS are most useful when applied to an entire map as map defaults (see *cfsadmin(1M)*). The following entry in the master map describes cached home directory mounts. It assumes the default location of the cache directory, `/cache`.

```
/home auto_home -fstype=cachefs,backfstype=nfs
```

### Indirect Maps

An indirect map allows you to specify mappings for the subdirectories you wish to mount under the directory indicated in the `/etc/auto_master` map or on the command line. In an indirect map, each *key* consists of a simple name that refers to the subdirectory where one or more filesystems that are to be mounted as needed.

Entries in both direct and indirect maps can be modified at any time. The new information is used when **automountd** next uses the map entry to do a mount.

### Direct Maps

Entries in a direct map are associated directly with **autofs** mount points. Each *key* is the full pathname of an **autofs** mount point. The direct map as a whole is not associated with any single directory.

Since each direct map entry results in a new **autofs** mount, such maps should be kept short.

If a directory contains direct map mount points, then an `ls -l` in the directory will force all the direct map mounts to occur.

Entries in both direct and indirect maps can be modified at any time. The new information is used when **automountd** next uses the map entry to do a mount.

New entries added to a master map or direct map will not be useful until the **automount** command is run to install them as new autofs mount points. New entries added to an indirect map may be used immediately.

An autofs directory associated with an indirect map shows only currently-mounted entries. This is a deliberate policy to avoid inadvertent mounting of every entry in a map via an **ls -l** of the directory.

### Included Maps

The contents of another map can be included within a map with an entry of the form:

**+mapname**

If *mapname* begins with a slash then it is assumed to be the pathname of a local file. Otherwise the location of the map is determined by the policy of the name service switch according to the entry for the automounter in **/etc/nsswitch.conf**, such as

**automount: nis files**

If the name service is **files** then the name is assumed to be that of a local file in **/etc**. If the *key* being searched for is not found in the included map, the search continues with the next entry.

### Special Maps

There are two special maps available: **-hosts** and **-null**. The **-hosts** map is used with the **/net** directory and assumes that the map key is the hostname of an NFS server. The **automountd** daemon dynamically constructs a map entry from the server's list of exported filesystems. For instance, a reference to **/net/hermes/usr** would initiate an automatic mount of all exported file systems from **hermes** that are mountable by the client. References to a directory under **/net/hermes** will refer to the corresponding directory relative to **hermes** root.

The **-hosts** map must mount all of the exported NFS filesystems from a server. If frequent access to just a single filesystem is required, it is more efficient to access the filesystem with a map entry that is tailored to mount just the filesystem of interest.

The **-null** map, when indicated on the command line, cancels a previous map for the directory indicated. This is most useful in the **/etc/auto\_master** map for cancelling entries that would otherwise be inherited from the **+auto\_master** include entry. To be effective, the **-null** entries must be inserted before the included map entry.

### Executable Maps

Local maps that have the execute bit set in their file permissions will be executed by the automounter and provided with a key to be looked up as an argument. The executable map is expected to return the content of an automounter map entry on its **stdout** or no output if the entry cannot be determined.

### Configuration and the auto\_master Map

When initiated without arguments, **automount** consults the master map for a list of **autofs** mount points and their maps. It mounts any **autofs** mounts that are not already mounted, and unmounts **autofs** mounts that have been removed from the master map or direct map.

The master map is assumed to be called **auto\_master** and its location is determined by the name service switch policy. Normally the master map is located initially as a local file, **/etc/auto\_master**.

### Network Information Service (NIS) and Yellow Pages (YP)

The Network Information Service (NIS) was formerly known as Sun Yellow Pages (YP). The functionality of the two remains the same.

### EXIT STATUS

**automount** returns:

- 0 successful
- 1 failure
- 3 map not found

**AUTHOR**

**automount** was developed by Sun Microsystems, Inc.

**FILES**

**/etc/auto\_master**        master automount map.  
**/etc/nsswitch.conf**     name service switch configuration file.  
**/usr/lib/netsvc/fs/autofs/automount**  
                         autofs **automount** command.  
**/usr/sbin/automount**    previous automounter **automount** command.

**SEE ALSO**

**automountd(1M)**, **cfsadmin(1M)**, **mount(1M)**.

  
**a**

**NAME**

automount - automatically mount NFS file systems. This manpage contains two versions of **automount**. See "Remarks" below in SYNOPSIS.

**SYNOPSIS**

```
automount [-nTv] [-D name = value] [-f master-file] [-M mount-directory] [-tl duration]
           [-tm interval] [-tw interval] [directory map [-mount-options] ] ...
```

**Remarks**

This manpage contains two versions of **automount**. The autofs version is presented first, followed by the previous automounter version.

The `/usr/sbin/automount` script checks the **AUTOFS** variable in `/etc/rc.config.d/nfsconf`. If the **AUTOFS** variable is set to one, then `/usr/lib/netsvc/fs/autofs/automount` is executed. The first half of this manpage represents the autofs **automount**.

If the **AUTOFS** variable is set to 0 (zero) or does not exist in `/etc/rc.config.d/nfsconf`, then the automount daemon is executed in `/usr/lib/netsvc/fs/automount/automount`. The second half of this manpage represents the older automount.

In future releases, only the autofs version of **automount** will be supported.

**DESCRIPTION**

**automount** is a daemon that automatically and transparently mounts NFS file systems as needed. It monitors attempts to access directories that are associated with an **automount** map, along with any directories or files that reside under them. When a file is to be accessed, the daemon mounts the appropriate NFS file system. Maps can be assigned to a directory by using an entry in a direct **automount** map, or by specifying an indirect map on the command line.

**automount** interacts with the kernel in a manner closely resembling an NFS server:

- **automount** uses the map to locate an appropriate NFS file server, exported file system, and mount options.
- It then mounts the file system in a temporary location, and replaces the file system entry for the directory or subdirectory with a symbolic link to the temporary location.
- If the file system is not accessed within an appropriate interval (five minutes by default), the daemon unmounts the file system and removes the symbolic link.
- If the specified directory has not already been created, the daemon creates it, and then removes it upon exiting.

Since name-to-location binding is dynamic, updates to an **automount** map are transparent to the user. This obviates the need to mount shared file systems prior to running applications that contain internally hard-coded references to files.

If the dummy directory (`/-`) is specified, **automount** treats the *map* argument that follows as the name of a direct map. In a direct map, each entry associates the full path name of a mount point with a remote file system to mount.

If the *directory* argument is a path name, the *map* argument points to an indirect map. An indirect map, contains a list of the subdirectories contained within the indicated *directory*. With an indirect map, it is these subdirectories that are mounted automatically.

A map can be a file or a NIS/NIS+ map; if a file, the *map* argument must be a full path name.

The *-mount-options* argument, when supplied, is a comma-separated list of options to the **mount** command (see **mount(1M)**) preceded by a `-`. However, any conflicting mount options specified in the indicated map take precedence.

**Options**

**automount** recognizes the following options:

- m           Option not supported.
- n           Disable dynamic mounts. With this option, references through the **automount** daemon succeed only when the target filesystem has been previously mounted. This can



## (autounter)

- be used to prevent NFS servers from cross-mounting each other.
- T Trace. Expand each NFS call and log it in `/var/adm/automount.log` file.
  - v Verbose. Log status messages to the system log file (see `syslogd(1M)`).
  - D *envar* = *value*  
Assign *value* to the indicated `automount` (environment) variable *envar*.
  - f *master-file* Read the local `master_file` before reading `auto_master` map.
  - M *mount-directory*  
Mount temporary file systems in the named directory instead of in `/tmp/mnt`.
  - tl *duration* Specify a *duration* (in seconds) that a file system is to remain mounted when not in use. The default is 5 minutes.
  - tm *interval* Specify an *interval* (in seconds) between attempts to mount a filesystem. The default is 30 seconds.
  - tw *interval* Specify an *interval* (in seconds) between attempts to unmount filesystems that have exceeded their cached times. The default is 1 minute.

**Map Entry Format**

A simple map entry (mapping) takes the form:

*directory* [ *-mount-options* ] *location* ...

where *directory* is the full path name of the directory to mount, when used in a direct map, or the basename of a subdirectory in an indirect map. *mount-options* is a comma-separated list of `mount` options, and *location* specifies a remote filesystem from which the directory may be mounted. In the simple case, *location* takes the form:

*host:pathname*

Multiple *location* fields can be specified, in which case `automount` pings all servers in the list and then selects the first host that responds to serve that mount point.

If *location* is specified in the form:

*host:path:subdir*

*host* is the name of the host from which to mount the file system, *path* is the path name of the directory to mount, and *subdir*, when supplied, is the name of a subdirectory to which the symbolic link is made. This can be used to prevent duplicate mounts when multiple directories in the same remote file system might be accessed. Assume a map for `/home` resembling:

```
mike      hpserver1:/home/hpserver1:mike
dianna    hpserver1:/home/hpserver1:dianna
```

Attempting to access a file in `/home/mike` causes `automount` to mount `hpserver1:/home/hpserver1` and creates a symbolic link called `/home/mike` to the `mike` subdirectory in the temporarily-mounted filesystem. A subsequent file access request in `/home/dianna` results in `automount` simply creating a symbolic link that points to the `dianna` subdirectory because `/home/hpserver1` is already mounted. Given the map:

```
mike      hpserver1:/home/hpserver1/mike
dianna    hpserver1:/home/hpserver1/dianna
```

`automount` would have to mount the filesystem twice.

A mapping can be continued across input lines by escaping the newline character with a backslash (`\`). Comments begin with a `#` and end at the subsequent newline character.

**Directory Pattern Matching**

The `&` character is expanded to the value of the *directory* field for the entry in which it occurs. Given an entry of the form:

```
mike      hpserver1:/home/hpserver1:&
```

the `&` expands to `mike`.

The `*` character, when supplied as the *directory* field, is recognized as the catch-all entry. Such an entry resolves to any entry not previously matched. For example, if the following entry appeared in the indirect

map for /home:

```
*      &: /home/&
```

this would allow automatic mounts in /home of any remote file system whose location could be specified as:

```
hostname : /home hostname
```

### Hierarchical Mappings

A hierarchical mapping takes the form:

```
directory [ / [subdirectory] [-mount-options] location ... ] ...
```

The initial / within the /[subdirectory] is required; the optional subdirectory is taken as a file name relative to the directory. If subdirectory is omitted in the first occurrence, the / refers to the directory itself.

Given the direct map entry:

```
/usr/local    \
/      -ro,intr  shasta:/usr/local      ranier:/usr/local    \
/bin  -ro,intr  ranier:/usr/local/bin  shasta:/usr/local/bin \
/man  -ro,intr  shasta:/usr/local/man  ranier:/usr/local/man
```

automount automatically mounts /usr/local, /usr/local/bin, and /usr/local/man, as needed, from either shasta or ranier, whichever host responded first.

### Direct Maps

A direct map contains mappings for any number of directories. Each directory listed in the map is automatically mounted as needed. The direct map as a whole is not associated with any single directory.

### Indirect Maps

An indirect map allows specifying mappings for the subdirectories to be mounted under the directory indicated on the command line. It also obscures local subdirectories for which no mapping is specified. In an indirect map, each directory field consists of the basename of a subdirectory to be mounted as needed.

### Included Maps

The contents of another map can be included within a map with an entry of the form:

```
+mapname
```

mapname can either be a file name, or the name of an NIS/NIS+ map, or one of the special maps described below. If mapname begins with a slash then it is assumed to be the pathname of a local file. Otherwise the location of the map is determined by the policy of the name service switch according to the entry for the automounter in /etc/nsswitch.conf, such as

```
automount: nis files
```

If the name service is files then the name is assumed to be that of a local file in /etc. If the key being searched for is not found in the included map, the search continues with the next entry.

### Special Maps

Three special maps, -hosts, -passwd, and -null, are currently available: The -hosts map uses the gethostbyname() map to locate a remote host when the hostname is specified (see gethostent(3N)). This map specifies mounts of all exported file systems from any host. For example, if the following automount command is already in effect:

```
automount /net -hosts
```

a reference to /net/hermes/usr initiates an automatic mount of all file systems from hermes that automount can mount, and any subsequent references to a directory under /net/hermes refer to the corresponding directory on hermes. The -passwd map uses the passwd(4) database to attempt to locate a user's home directory. For example, if the following automount command is already in effect:

```
automount /homes -passwd
```

if the home directory for a user has the form /dir/server/username, and server matches the host system on which that directory resides, automount mounts the user's home directory as: /homes /username.

For this map, the tilde character (~) is recognized as a synonym for username.

The **-null** map, when indicated on the command line, cancels a previous map for the directory indicated. It can be used to cancel a map given in **auto\_master**.

### Configuration and the auto\_master Map

**automount** normally consults the **auto\_master** configuration map for a list of initial **automount** maps, and sets up automatic mounts for them in addition to those given on the command line. If there are duplications, the command-line arguments take precedence. This configuration database contains arguments to the **automount** command rather than mappings.

Maps given on the command line, or those given in a local master file specified with **-f** override those in the **auto\_master** map. For example, given the command:

```
automount /homes /etc/auto.homes /- /etc/auto.direct
```

and the master map file **auto\_master** containing:

```
/homes -passwd
```

**automount** mounts home directories using the **/etc/auto.homes** map instead of the special **-passwd** map in addition to the various directories specified in the **/etc/auto.direct** map.

### EXTERNAL INFLUENCES

#### Environment Variables

Environment variables can be used within an **automount** map. For example, if **\$HOME** appears within a map, **automount** expands it to the current value of the **HOME** environment variable.

To protect a reference from affixed characters, surround the variable name with curly braces. Environment variables cannot appear as the key entry in maps.

### WARNINGS

Do not send the **SIGKILL** signal (**kill -9**, or **kill -KILL**) to the **automount** daemon. Doing so causes any processes accessing mount directories served by **automount** to hang. A system reboot may be required to recover from this state.

Do not start an **automount** daemon while another is still running. If restarting **automount**, make sure the first daemon and all of its children are not running.

When **automount** receives signal **SIGHUP**, it rereads the **/etc/mnttab** file to update its internal record of currently mounted file systems. If a file system mounted by **automount** is unmounted by a **umount** command, **automount** should be forced to reread the file by sending the **SIGHUP** signal (see **kill(1)**).

Shell file name expansion does not apply to objects not currently mounted.

Since **automount** is single-threaded, any request that is delayed by a slow or nonresponding NFS server delays all subsequent automatic mount requests until it completes.

Programs that read **/etc/mnttab** and then touch files that reside under automatic mount points introduce further entries to the file.

Automatically-mounted file systems are mounted with type **ignore**; they do not appear in the output of either **mount** or **bdf** (see **mount(1M)** and **bdf(1M)**).

### AUTHOR

**automount** was developed by Sun Microsystems, Inc.

### FILES

<b>/tmp_mnt</b>	directory under which filesystems are dynamically mounted
<b>/etc/mnttab</b>	mount table
<b>/etc/nsswitch.conf</b>	the name service switch configuration file.

### SEE ALSO

**mount(1M)**, **bdf(1M)**, **passwd(4)**.

**NAME**

automountd - autofs mount/unmount daemon

**SYNOPSIS**

automountd [-Tv] [-D *name=value*]

**DESCRIPTION**

**automountd** is an RPC server that answers file system mount and unmount requests from the **autofs** filesystem. It uses local files or name service maps to locate filesystems to be mounted. These maps are described with the **automount** command (see *automount(1M)*).

The **automountd** daemon is automatically invoked if the **AUTOFS** variable and the **AUTOMOUNT** variable are set to 1 in **/etc/rc.config.d/nfsconf**.

**Options**

**-T** Trace. Expand each RPC call and display it to **/var/adm/automount.log**.

**-v** Verbose. Log status messages to **/var/adm/syslog/syslog.log**.

**-D** *name=value*

Assign *value* to the indicated **automount** map substitution variable. These assignments cannot be used to substitute variables in the master map **auto\_master**.

**Tracing**

While **automountd** is running, the signal **SIGUSR2** can be sent to **automountd** to turn tracing on. It logs messages in **/var/adm/automount.log**. To turn tracing off, send the signal **SIGUSR2** again.

**The automount Daemon**

In some releases of HP-UX, there are two versions of **automount**, **autofs** and the old automounter. They are both documented in *automount(1M)*. To use **autofs** instead of the old automounter daemon, set the **AUTOFS** variable to 1 in **/etc/rc.config.d/nfsconf**.

**EXIT STATUS**

**automountd** returns:

0 successful

1 failure

**EXAMPLES**

To turn on tracing, enter:

```
/usr/lib/netsvc/fs/autofs/automountd -T
```

or

```
automountd -T
```

No messages are displayed by the trace. Check the results in **/var/adm/automount.log**.

**AUTHOR**

**automountd** was developed by Sun Microsystems, Inc.

**FILES**

<b>/etc/auto_master</b>	master map for automounter
<b>/var/adm/automount.log</b>	log file for automountd
<b>/usr/lib/netsvc/fs/autofs/automountd</b>	autofs daemon

**SEE ALSO**

*automount(1M)*.

**NAME**

autopush - manage system database of automatically pushed STREAMS modules

**SYNOPSIS**

```
autopush -f file
autopush -g -M major -m minor
autopush -r -M major -m minor
```

**DESCRIPTION**

**autopush** manages the system database that is used for automatic configuration of STREAMS devices. The command is used in three different ways as dictated by the **-f**, **-g**, and **-r** command-line options described below.

**Options**

**autopush** recognizes the following command-line options and arguments:

**-f *file*** Using the configuration information contained in *file*, load the system database with the names of the STREAMS devices and a list of modules to use for each device. When a device is subsequently opened, the HP-UX STREAMS subsystem pushes the modules onto the stream for the device.

*file* must contain one or more lines of at least four fields separated by a space as shown below:

```
major minor lastminor module1 module2 ... moduleN
```

The first field *major* can be either an integer or a device name. The device name is the name for the device used in the **master** file. The next two fields are integers. If *minor* is set to -1, then all minor devices for the specified *major* are configured and *lastminor* is ignored. If *lastminor* is 0, then only a single minor device is configured. To configure a range of minor devices for a major device, *minor* must be less than *lastminor*. The remaining field(s) list one or more module names. Each module is pushed in the order specified. A maximum of eight modules can be pushed. Any text after a # character in *file* is treated as a comment for that line only.

This option is also used to restore device configuration information previously removed by **autopush -r**. However, when used in such a manner, the entire database is restored, not just the information that was previously removed.

**-g -M *major* -m *minor*** Display current configuration information from the system database for the STREAMS device specified by the *major* device number (or device name for the device from the **master** file) and *minor* number.

If a range of minors has been previously configured then **autopush -g** returns the configuration information for the first minor in the range, in addition to other information.

**-r -M *major* -m *minor*** Remove configuration information from the system database for the STREAMS device specified by the *major* device number (or device name for the device from the **master** file) and *minor* number. Removal is performed on the database only, not on the original configuration file. Therefore, the original configuration can be restored by using the **-f *file*** option. To permanently exclude a STREAMS device from the database, its information must be removed from the configuration file.

If *minor* matches the first minor of a previously configured range then **autopush -r** removes the configuration information for the entire configured range.

**EXAMPLES**

If the file `/tmp/autopush.example` contains:

```
75 -1 0 modA modB
test 0 5 modC modA
```

Then **autopush -f /tmp/autopush.example** will cause **modA** and **modB** to be pushed whenever major device # 75 is opened, and **modC** and **modA** to be pushed for the first six opens of device

**test.**

This next example lists information about the stream for major device 75 and its minor device -2:

```
autopush -g -M 75 -m -2
```

**a**

**FILES**

`/usr/lib/nls/msg/C/autopush.cat`

NLS catalog for **autopush**.

**SEE ALSO**

`sad(7)`, `streamio(7)`.

**NAME**

auto\_parms - initial system configuration/DHCP support script

**SYNOPSIS**

auto\_parms

**DESCRIPTION**

**auto\_parms** is a system initialization script whose primary responsibility lies in handling first time boot configuration and ongoing management of the DHCP lease(s). **auto\_parms** is invoked at boot time by the **/sbin/rc** script. Initially, it will load a list of available ethernet interfaces and begin requesting a DHCP lease on each interface, stopping when a valid lease is secured or the list is exhausted.

As a part of checking for the availability of a lease on a particular interface, **auto\_parms** will also consult **/etc/rc.config.d/netconf** and examine the variable **DHCP\_ENABLE[index]**. If **DHCP\_ENABLE[index]** is set to '1', **auto\_parms** will attempt to request a lease on the interface designated by 'index'. If **DHCP\_ENABLE[index]** is set to '0' (the default case) or does not exist in **/etc/rc.config.d/netconf**, **auto\_parms** will not attempt the DHCP request over the interface.

Once a lease is secured, the information supplied with the lease will be used to initialize key networking parameters (see *dhcplib2conf(1M)*).

If **auto\_parms** detects that the system is going through a "first time boot" (keyed by the hostname for the system not being set), it will invoke **set\_parms** for the purpose of verifying the DHCP supplied parameters as well as collecting any parameters not supplied by DHCP.

For all subsequent boots, the data supplied by a DHCP lease is assumed to be definitive and will be recognized as such by **auto\_parms**. Note that in an environment (non-mobile) where DHCP is being used for IP address management, the lease information will not change from boot to boot under normal conditions. This is accomplished by **auto\_parms** ensuring that the **dhcplib** is placed in "lease maintenance mode" prior to exiting.

**FILES**

**/sbin/auto\_parms**  
**/sbin/set\_parms.util**

**EXAMPLES**

See **/sbin/rc** for invocation context

**SEE ALSO**

*dhcplib2conf(1M)*.

**NAME**

backup - backup or archive file system

**SYNOPSIS**

**/usr/sbin/backup** [-A] [-archive] [-fsck]

**DESCRIPTION**

The *backup* command uses *find*(1) and *cpio*(1) to save a *cpio* archive of all files that have been modified since the modification time of **/var/adm/archivedate** on the default tape drive (**/dev/update.src**). *backup* should be invoked periodically to ensure adequate file backup.

The **-A** option suppresses warning messages regarding optional access control list entries. *backup*(1M) does not backup optional access control list entries in a file's access control list (see *acl*(5)). Normally, a warning message is printed for each file having optional access control list entries.

The **-archive** option causes *backup* to save all files, regardless of their modification date, and then update **/var/adm/archivedate** using *touch*(1).

*backup* prompts you to mount a new tape and continue if there is no more room on the current tape. Note that this prompting does not occur if you are running *backup* from *cron*(1M).

The **-fsck** option causes *backup* to start a file system consistency check (without correction) after the backup is complete. For correct results, it is important that the system be effectively single-user while *fsck* is running, especially if **-fsck** is allowed to automatically fix whatever inconsistencies it finds. *backup* does not ensure that the system is single-user.

You can edit **/usr/sbin/backup** to customize it for your system. Several local values are used that can be customized:

BACKUPDIRS	specifies which directories to back up recursively (usually <b>/</b> , meaning all directories);
BACKUPLOG	file name where start and finish times, block counts, and error messages are logged;
ARCHIVE	file name whose date is the date of the last archive;
REMINDD	file name that is checked by <b>/etc/profile</b> to remind the next person who logs in to change the backup tape;
FCKLOG	file name where start and finish times and <i>fsck</i> output is logged.

You may want to make other changes, such as whether or not *fsck* does automatic correction (according to its arguments), where *cpio* output is directed, other information logging, etc.

In all cases, the output from *backup* is a normal *cpio* archive file (or volume) which can be read using *cpio* with the **c** option.

**File Recovery**

*backup* creates archive tapes with all files and directories specified relative to the root directory. When recovering files from an archive tape created by *backup*, you should be in the root directory and specify the directory path names for recovered files relative to the root directory (**/**). When specifying the directory path name for file recovery by *cpio*, do not precede the leading directory name with a slash. If you prefer, you can also use *cpio* with a **-t** option to determine how files and directories are named on the archive tape before attempting recovery.

**WARNINGS**

Refer to WARNINGS in *cpio*(1).

When *cpio* runs out of tape, it sends an error to standard error and demands a new special file name from **/dev/tty**.

To continue, rewind the tape, mount the new tape, type the name of the new special file at the system console, and press **Return**.

If *backup* is being run unattended from *cron*(1M) and the tape runs out, *backup* terminates, leaving the *find* process still waiting. Kill this process when you return.

**FILES**

**/var/adm/archivedate** parameterized file names



**SEE ALSO**

cpio(1), find(1), touch(1), cron(1M), fbackup(1M), frecover(1M), fsck(1M), acl(5).

  
**b**

**NAME**

bdf - report number of free disk blocks (Berkeley version)

**SYNOPSIS**

```
/usr/bin/bdf [-b] [-i] [-l] [-t type | [filesystem|file] ... ]
```

**DESCRIPTION**

The **bdf** command displays the amount of free disk space available either on the specified *filesystem* (*/dev/dsk/c0d0s0*, for example) or on the file system in which the specified *file* (such as **\$HOME**), is contained. If no file system is specified, the free space on all of the normally mounted file systems is printed. The reported numbers are in kilobytes.

**Options**

The **bdf** command recognizes the following options:

- b** Display information regarding file system swapping.
- i** Report the number of used and free inodes.
- l** Display information for local file systems only (for example, HFS and CDFS file systems).
- t *type*** Report on the file systems of a given *type* (for example, **nfs** or **hfs**).

**RETURN VALUE**

The **bdf** command returns 0 on success (able to get status on all file systems), or returns 1 on failure (unable to get status on one or more file systems).

**WARNINGS**

If file system names are too long, the output for a given entry is displayed on two lines.

The **bdf** command does not account for any disk space reserved for swap space, or used for the HFS boot block (8 KB, 1 per file system), HFS superblocks (8 KB each, 1 per disk cylinder), HFS cylinder group blocks (1 KB - 8 KB each, 1 per cylinder group), and inodes (currently 128 bytes reserved for each inode). Non-HFS file systems may have other items not accounted for by this command.

**AUTHOR**

**bdf** was developed by the University of California, Berkeley.

**FILES**

- /etc/fstab** Static information about the file systems.
- /etc/mnttab** Mounted file system table.
- /dev/dsk/\*** File system devices.

**SEE ALSO**

df(1M), fstab(4), mnttab(4).

**NAME**

boot - bootstrap process

**DESCRIPTION**

The Series 700 and 800 bootstrap process involves the execution of three software components:

- **pd**c (see *pd*(1M),
- **isl** (see *isl*(1M), and
- **hpux**.

After the processor is RESET, **pd**c, the **processor-dependent code** (firmware), performs a self-test and initializes the processor. It then loads and transfers control to **isl**, the operating-system-independent **initial system loader**. **isl**, in turn, loads and transfers control to the **hpux** utility, the HP-UX-specific bootstrap loader. **hpux** then downloads the HP-UX kernel object file from an HP-UX file system and transfers control to the loaded kernel image.

**SEE ALSO**

hpux(1M), isl(1M), pdc(1M).

**NAME**

bootpd - Internet Boot Protocol server

**SYNOPSIS**

```
/usr/sbin/bootpd [-d debuglevel] [-t timeout] [configfile [dumpfile]]
```

**DESCRIPTION**

The **bootpd** daemon implements three functions: a Dynamic Host Configuration Protocol (DHCP) server as defined in RFC1541, an Internet Boot Protocol (BOOTP) server as defined in RFC951 and RFC1395, and a DHCP/BOOTP relay agent as defined in RFC1542. It also contains some of the useful fields as defined in RFC2132.

**bootpd** is run through **inetd** (see *inetd*(1M)). It is run by **/etc/inetd** when the following line (or equivalent) is included in the file **/etc/inetd.conf**:

```
bootps dgram udp wait root /usr/sbin/bootpd bootpd
```

**bootpd** starts when a boot request arrives. If it has not received another boot request after 500 minutes, **bootpd** exits. The **-t** option can be used to specify a different timeout value in minutes (such as **-t20**). With a timeout value of zero (**-t0**), **bootpd** never exits.

The **-d** option sets the verbosity level (1-3) of the logging emitted by the daemon via **syslog** (see *syslog*(3C)). For improved performance, this option should not be used. If this option is not used, no logging is done by **syslog** except for fatal errors.

When **bootpd** receives a DHCP/BOOTP request, it first checks if the hardware address of the client is listed in the **/etc/dhcpydeny** database. If yes, this client is denied lease. If the client is not listed in the **dhcpydeny** database, it checks whether the client information is in the **/etc/bootptab** database. If the client information is available, **bootpd** sends back the reply. Otherwise, it checks whether there is any matched relay information for the client in the **/etc/bootptab** database. If so, **bootpd** goes through a series of checks to see if it should relay the request. If no matched relay information was found, **bootpd** checks whether the client information is matched by a pool or device group in the **/etc/dhcptab** database. If a match is found, **bootpd** sends back a reply. The request is dropped if no matched group information is found.

To reply to a DHCP or BOOTP request the server puts together a BOOTREPLY message and does a number of checks to ensure the message is sent to the correct destination.

**bootpd** first checks the **ciaddr** (client IP address) field of the DHCP/BOOTP packet. If this field is nonzero, the BOOTREPLY message is sent to the IP address identified in **ciaddr**.

If the **ciaddr** field is zero, **bootpd** checks the **giaddr** field. If this field is not zero, **bootpd** sends the BOOTREPLY message to the *relay agent* specified in **giaddr** field and the *relay agent* delivers the BOOTREPLY message to the client. If the **giaddr** field is zero, **bootpd** sends the BOOTREPLY message to the client. In both cases, the BOOTREPLY will either be sent to the IP address specified in the **yiaddr** (your IP address) field or as a broadcast message. On HP-UX, there are two ways to specify that the BOOTREPLY should be sent as a broadcast message.

1. The client sets the broadcast flag bit in the **flag** field (bit 0) of the DHCP/BOOTP request packet.
2. Define the **ba** tag in the **bootptab** file (see *Tags for client entries* below)

For the case where the **bootpd** has matched a relay entry in **/etc/bootptab**, it attempts to forward the request to the configured DHCP/BOOTP server.

**bootpd** first checks whether the relay function is enabled for the requesting client. The relay capability is configurable. If the relay function is disabled, then the request packet is dropped.

Before **bootpd** relays the request, it also examines the **giaddr** (gateway IP address) field. The client sets the **giaddr** field to zero when it sends out the request. If the relay agent finds this field is zero, it fills this field with the primary IP address of the interface on which the request was received; otherwise, the relay agent does not change this field. Then **bootpd** increments the value of the **hops** field, and relays the request to the DHCP/BOOTP servers that have been configured for this client.

If the relay function is enabled for this client, **bootpd** checks the **hops** field of the DHCP/BOOTP request packet. The client sets the **hops** field to 0 when it sends out the DHCP/BOOTP request. The **hops** value is increased every time the request packet is relayed by a relay agent. The maximum hop number can be configured. The maximum possible hop number allowed is 16. The default maximum is set to 4. The request packet is dropped if the hop value exceeds the configured maximum.

Then **bootpd** compares the value of the **secs** (seconds since the client began booting) field of the DHCP/BOOTP packet to the **threshold** value. The client sets the **secs** field to zero when it first sends out the request. The client repeats the request if it does not receive a reply. When the client repeats the request, it sets the **secs** value to the number of seconds since the first request was sent. **bootpd** does not relay the request if the value of the **secs** field is less than the **threshold** value. The **threshold** value can be configured. The default value is 0.

### Configuration

Upon startup, **bootpd** reads its configuration files to build its internal database, then listens for boot request packets. The default configuration files are, **/etc/dhcdeny**, **/etc/bootptab**, and **/etc/dhcptab**. The **bootptab** file can be specified in the command line. **bootpd** rereads its configuration file when it receives a hangup signal, **SIGHUP**, or when it receives a boot request packet and detects that the configuration file has been updated. If hosts are added, deleted, or modified, their entries in the **bootpd** internal database are updated accordingly when the configuration files are reread. The **/etc/dhcdeny** database contains the list of hardware addresses of the clients that will not be served by this server.

If **bootpd** receives a **SIGUSR1** signal, it dumps its memory-resident database to the file **/var/tmp/bootpd.dump** or the *dumpfile* specified in the command line.

The configuration file can contain two types of host entries:

1. The client entries, which contains the client information.
2. The relay entries, which contains the configuration to relay DHCP/BOOTP requests for one or more clients.

The configuration uses two-character, case-sensitive tag symbols to represent host parameters. These parameter declarations are separated by colons (:). The general format is:

*hostname* : *tg=value* : ... : *tg=value* : ... : *tg=value* : ...

where *hostname* is the actual name of a DHCP/BOOTP client in the client entries, and in the case of a relay entry, it can be the actual name of a client if it is an individual relay entry, or it can be a name for a group of clients if it is a group relay entry. *tg* is a two-character tag symbol. Most tags must be followed by an equals-sign, and a value as above. Some can appear in a boolean form with no value (that is, **:tg:**).

Blank lines and lines beginning with **#** are ignored in the configuration file. Host entries are separated from one another by newlines; a single host entry can be extended over multiple lines if the lines end with a backslash (\). It is also acceptable for lines to be longer than 80 characters. Tags can appear in any order with the following exceptions: The host name must be the very first field in an entry, and the hardware type tag, **ht**, must precede the hardware address tag, **ha**, and the hardware mask tag, **hm**.

IP addresses are specified in standard Internet dot notation, and can use decimal, octal, or hexadecimal numbers (octal numbers begin with 0, hexadecimal numbers begin with 0x or 0X). Certain tags accept a list of one or more IP addresses (*ip\_address\_list*). When more than one IP address is listed, the addresses must be separated by whitespace.

The types of tags can be grouped into three categories:

1. The tags that can be used for both the client and the relay entries.
2. The tags that can only be used in the relay entries.
3. The tags that can only be used in the client information entries.

Tag **ip** is used to differentiate a client entry from a relay entry. An entry with tag **ip** defined is treated as a client entry. A relay entry can contain the relay configuration for an individual client, also a hardware address mask mechanism is provided to configure the relay entry for a group of clients. The group client relay entries are kept in a linear sorted table by **bootpd**. When a client does not have an individual relay specification, the linear table is searched to see if there is a match for the client. If there are multiple matched entries in the sorted table, only the first one is used. Tag **hm** is used to differentiate an individual client relay entry from a group relay entry. The linear sorted table is sorted on the value of tag **hm**. The search and match mechanism is explained in the discussion of tag **hm**.

### Tags for both kinds of entries

**ha**=*hardware-address*

This tag specifies the hardware address of the client. The *hardware address* must be specified in hexadecimal; optional periods and/or a leading 0x can be included for readability. The **ha** tag

must be preceded by the **ht** tag (either explicitly or implicitly; see **tc** below).

**ht=***hardware-type*

This tag specifies the hardware type code. *hardware-type* can be an unsigned decimal, octal, or hexadecimal integer corresponding to one of the ARP Hardware Type codes specified in RFC1010. It can also be specified by the symbolic names **ethernet** or **ether** for 10-Mb Ethernet; **ethernet3** or **ether3** for 3-Mb experimental Ethernet; **ieee802**, **tr**, or **token-ring** for IEEE 802 networks; **pronet** for Proteon ProNET Token Ring; **chaos**, and **arcnet**, for Chaos and ARCNET, respectively.

**tc=***template-host*

This tag indicates a table continuation. Often, many host entries share common values for certain tags (such as domain servers, etc.). Rather than repeatedly specifying these tags, a full specification can be listed for one host entry and shared by others via the **tc** mechanism.

The *template-host* is a dummy host that does not actually exist and never sends boot requests. Information explicitly specified for a host always overrides information implied by a **tc** tag symbol. The value of *template-host* can be the host name or IP address of any host entry previously listed in the configuration file.

Sometimes it is necessary to delete a specific tag after it has been inferred via **tc**. This can be done using the construction *tag@* which removes the effect of *tag*. For example, to completely undo an RFC1034 domain name server specification, use **:ds@:** at an appropriate place in the configuration entry. After removal with @, a tag is eligible to be set again through the **tc** mechanism.

### Tags for relay entries

**bp=***bootp-servers*

This tag specifies the BOOTP servers that DHCP/BOOTP requests will be relayed to. The value of *bootp-servers* can be one or more individual IP addresses, and/or one or more network broadcast addresses. A relay entry with this tag configured indicates that the relay function is on for the clients specified in this entry. A relay entry missing this symbol means that the relay function is off for the clients specified in this entry.

**th=***threshold*

This tag specifies the *threshold* value in seconds for the entry. The default value is 0.

**hp=***hops*

This tag specifies the maximum *hops* value. If the *hops* value exceeds 16, it is set to 16. The default value is 4.

**hm=***hardware-address-mask*

This tag specifies the mask for the hardware address **ha**. *hardware-address-mask* must be specified in hexadecimal. An optional leading **0x** can be included for readability. The **hm** tag must be preceded by the **ht** tag (either explicitly or implicitly; see **tc** above). Each 0 bit in **hm** specifies that the corresponding bit in **ha** is a "don't-care" bit, each 1 bit in **hm** specifies that the corresponding bit in the **ha** value is ANDed with the **hm** value. If the result is the same and also the hardware type matches, then a match is found. For example,

```
if ((hm & ha)==(client_hw_addr & hm))
    && (ht == client_hw_type))
    then a match is found
    else continue the search
```

### Tags for client entries

**ba** This tag specifies that **bootpd** should broadcast the boot reply to the client. As a boolean tag, it causes **bootpd** to send the boot reply on the configured broadcast address of each network interface. You can also assign the tag an IP-address value, which specifies the specific IP or broadcast address for the boot reply.

**bf=***filename*

This tag specifies the *filename* of the bootfile that the client should download. The client's boot request, and the values of the **hd** (see below) and **bf** symbols, determine the contents of the bootfile field in the boot reply packet.

If the client specifies an absolute path name (in its boot request), and that file is accessible on the server machine (see below), **bootpd** returns that path name in the reply packet. If the file is

not accessible, the request is discarded; no reply is sent. If the client specifies a relative path name, **bootpd** constructs a full path name by appending the relative path name to the value of the **hd** tag, and tests to determine if the full path name is accessible. If the full path name is accessible, it is returned in the boot reply packet; if not, the request is discarded.

Clients that do not specify boot files in their boot requests always elicit a reply from the server. The exact reply depends on the values of the **hd** and **bf** tags. If the **bf** tag specifies an absolute path name, and the file is accessible, that path name is returned in the reply packet. Otherwise, if the **hd** and **bf** tags together specify an accessible file, that file name is returned in the reply. If a complete file name cannot be determined, or the file is not accessible publicly, the reply contains a zeroed-out bootfile field.

If the **tftp** pseudo-user exists, **bootpd** treats all path names (absolute or relative) as being relative to the home directory of **tftp** and checks there first. If the file is not accessible under the **tftp** home directory or the **tftp** pseudo-user does not exist, **bootpd** checks for the file relative to **/**.

For a file to be available, it must exist, and be publicly readable.

All file names are first tried as *filename.hostname* and then simply as *filename*. However, in the case when the **tftp** pseudo-user exists, but *filename.hostname* and *filename* are not accessible under the **tftp** home directly, only *filename* is checked relative to **/**.

Note that a file considered to be accessible relative to **/** might not actually be accessible via **tftp** if the command line arguments to **tftp** disallow that path.

**bs=***size*

This tag specifies the size of the bootfile. The parameter *size* can be either a decimal, octal, or hexadecimal integer specifying the size of the bootfile in 512-octet blocks, or the keyword **auto**, which causes the server to automatically calculate the bootfile size at each request. Specifying the **bs** symbol as a boolean has the same effect as specifying **auto** as its value.

**ci=***client\_ID*

This tag specifies the client identifier of the client. The parameter *client\_ID* can be either a hexadecimal integer, or a string contained in double quotes. The *client\_ID* is a unique identifier that the DHCP client may use to identify itself to the server. If present, the client identifier supersedes the hardware address, so a client and an entry will only match in one of two situations: one, they both have the same client identifier, or two they both have the same hardware address and neither has a client identifier. If a request has a client identifier, then that is used to match the client up with an entry in the **bootp** configuration file. One common client ID used is to concatenate the hardware type (e.g. 0x01 for ethernet) with the hardware address.

**cs=***ip\_address\_list*

This tag specifies the IP addresses of RFC865 Quote of the Day (cookie) servers.

**dn=***domain\_name*

This tag specifies the domain name of the client for Domain Name Server resolution (see RFC1034).

**ds=***ip\_address\_list*

This tag specifies the IP addresses of RFC1034 Domain Name servers.

**ef=***filename*

Specifies the name of an extensions file. The file, retrievable via TFTP, contains information which can be interpreted in the same way as the 64-octet vendor-extension field within the BOOTP response. The maximum length of the file is unconstrained. All references to an extensions filename within the file are ignored.

**gw=***ip\_address\_list*

This tag specifies the IP addresses of gateways for the client's subnet. If one of multiple gateways is preferred, it should be listed first.

**hd=***home-directory*

This tag specifies a directory name to which the bootfile is appended (see the **bf** tag above). The default value of the **hd** tag is **/**.

**hn** The presence of this tag indicates that the client's host name should be sent in the boot reply. The **hn** tag is a boolean tag. **bootpd** attempts to send the entire host name as it is specified in the configuration file or hosts database. The configuration file is checked first, if the host name

is not found, the hosts(4) database is then checked. If the hostname cannot fit into the reply packet, an attempt is made to shorten the name to just the host field (up to the first period, if present) and then tried. In no case is an arbitrarily truncated host name sent. If nothing reasonable can fit, nothing is sent.

**im**=*ip\_address\_list*

This tag specifies the IP addresses of Impress network image servers.

**ip**=*ip-address*

This tag specifies the IP address of the DHCP/BOOTP client.

**lg**=*ip\_address\_list*

This tag specifies the IP addresses of MIT-LCS UDP log servers.

**lp**=*ip\_address\_list*

This tag specifies the IP addresses of Berkeley 4BSD printer servers.

**md**=*merit\_dump\_file*

This tag specifies the name of a file to dump the core of a client.

**ms**=*ip\_address\_list*

This tag specifies the IP address(es) of SMTP servers available to the client (RFC2132).

**na**=*ip\_address\_list*

This tag specifies the IP address(es) of RFC 1001/1002 NetBIOS name server(s) in order of preference.

**nb**=*ip\_address\_list*

This tag specifies the IP address(es) of RFC 1001/1002 NetBIOS datagram distribution server(s) in order of preference.

**nc**=*NetBIOS\_node\_type*

Specifies the NetBIOS node type code. Allows NetBIOS over TCP/IP clients to be configured as described in RFC1001/1002. The *NetBIOS\_node\_type* can be an unsigned decimal, octal, or hexadecimal integer corresponding to one of the client types as follows:

0x1 or **B-node** for B-node;  
 0x2 or **P-node** for P-node;  
 0x4 or **M-node** for M-node;  
 0x8 or **H-node** for H-node.

**nd**=*string*

This tag specifies the NetBIOS over TCP/IP scope parameter for the client as specified in RFC 1001/1002.

**ns**=*ip\_address\_list*

This tag specifies the IP addresses of IEN-116 name servers.

**nt**=*ip\_address\_list*

This tag specifies the IP addresses of Network Time Protocol servers. Servers should be listed in order of preference.

**pd**=*NIS+-domain-name*

This tag specifies the name of clients NIS+ domain name (RFC2132).

**ps**=*ip\_address\_list*

This tag specifies the IP address(es) of NIS+ servers available to the client (RFC2132).

**rl**=*ip\_address\_list*

This tag specifies the IP addresses of RFC887 Resource Location Protocol servers.

**rp**=*root\_path*

This tag specifies a path name to be mounted as a root disk.

**sm**=*subnet-mask*

This tag specifies the client's subnet mask. *subnet-mask* is specified as a single IP address.

**sr**=*destination\_ip\_address gateway\_ip\_address ...*

This tag specifies a list of static routes that the client should put in its routing cache. Each route consists of a pair of IP addresses. The first address is the destination address, and the second is the router. Use the **gw**= option to specify the default route (0.0.0.0) as it is not a legal destination address.



**ss**=*ip\_address*

This tag specifies the IP address of a swap server.

**Tnnn**=*generic-data*

This is a generic tag where *nnn* is an RFC1533 option field tag number. Use this option to configure RFC1533 options not currently supported with **bootpd** tag names. This option allows one to immediately take advantage of future extensions to RFC1533. The *generic-data* data can be represented as either a stream of hexadecimal numbers or as a quoted string of ASCII characters. The length of the generic data is automatically determined and inserted into the proper fields of the RFC1541-style boot reply.

**to**=*offset*

This tag specifies the client's time zone offset in seconds from UTC. The time *offset* can be either a signed decimal integer or the keyword **auto** which uses the server's time zone offset. Specifying the **to** symbol as a boolean has the same effect as specifying **auto** as its value.

**ts**=*ip\_address\_list*

This tag specifies the IP addresses of RFC868 Time Protocol servers.

**yd**=*NIS-domain-name*

Specifies the name of the client's NIS domain.

**ys**=*ip\_address\_list*

Specifies the IP address(es) of NIS servers available to the client. Servers should be listed in order of preference.

**vm**=*magic-cookie*

This tag specifies the RFC1048 vendor information magic cookie. *magic-cookie* can be one of the following keywords: **auto** (indicating that vendor information is determined by the client's request), **rfc1048** (which always forces an RFC1048-style reply), or **cmu** (which always forces a CMU-style reply).

**Vnnn**=*generic-data*

This is a generic tag for vendor specific information where *nnn* is a vendor defined option field tag number. The *generic-data* data can be represented as either a stream of hexadecimal numbers or as a quoted string of ASCII characters. The length of the generic data is automatically determined and inserted into the vendor specific field of the RFC1541-style boot reply.

**xd**=*ip\_address\_list*

This tag specifies the IP addresses of systems that are running the X Window System Display Manager and are available to the client. Addresses should be listed in order of preference.

**xf**=*ip\_address\_list*

This tag specifies the IP addresses of X window System font servers available to the client. Servers should be listed in order of preference.

### Dhcpdeny Configuration

The configuration file **/etc/dhcpdeny** contains the list of hardware addresses, one address per line, for clients that will not be served by our server. If we know about some bad clients in the network and we don't want to serve them, add the hardware address of those clients in this file. This file, like other configuration files, takes **#** character as the starting of a comment.

### Dhcptab Configuration

The configuration file **/etc/dhcptab** defines groups of IP addresses that to be leased out to clients. It also specifies certain general behaviors of the server, such as whether or not to give addresses from these groups to **bootp** clients or only to DHCP clients.

The configuration file has a format similar to the **/etc/bootptab** configuration file, with a keyword followed by one or more tag symbols. These tag symbols are separated by colons (:). The general format is:

*keyword*: *tg=value*: ... : *tg=value*: ... : *tg=value*: ...

where *keyword* is one of four allowed (non-case-sensitive) symbols and *tg* is a two or more (case-sensitive) character tag symbol. Most tags must be followed by an equals-sign and a value as above. Some can also appear in a boolean form with no value (i.e. **:tg:**).

Blank lines and lines beginning with **#** are ignored in the configuration file. Keyword entries are separated from one another by newlines; a single host entry may be extended over multiple lines if each continued line ends with a backslash (\). Lines may be longer than 80 characters. Tags can appear in any

order.

IP addresses must be specified in standard Internet “dot” notation, and can use decimal, octal, or hexadecimal numbers (octal numbers begin with 0, hexadecimal numbers begin with 0x or 0X). Certain tags accept a list of one or more IP addresses (*ip\_address\_list*). When more than one IP address is listed, they must be separated by white space.

The currently recognized keywords are:

#### **dhcp\_pool\_group**

This keyword is followed by tags defining a group of IP addresses to give out to clients on the same subnet, and the characteristics of that group. In addition to the tags defined for DHCP groups, all of the two-letter tags for bootp entries may also be used (except for **ht**, the hardware type tag, **ha**, the hardware address tag, or **ci**, the client ID tag. Required tags are: **subnet-mask**, **addr-pool-start-address**, and **addr-pool-last-address**.

#### **dhcp\_device\_group**

This keyword is used to define a group of IP addresses on a subnet much like **dhcp\_pool\_group**, but with one exception: all clients in a device group must have the same client class (specified with tag **class-id**). This allows different types of clients to receive different parameters from the server. Required tags are: **class-id**, **subnet-mask**, **addr-pool-start-address**, and **addr-pool-last-address**.

#### **dhcp\_default\_client\_settings**

This keyword is followed by tags to be applied to all groups. These tag values can be overridden for a specific group if that tag is defined for that specific group. This keyword simply saves one from entering the same tag for every group. Thus most tags that may be used for **dhcp\_pool\_group**, and **dhcp\_device\_group**, may be used here. The tag descriptions specify if a tag may not be used here.

#### **dhcp\_server\_settings**

This keyword is followed by tags that specify a few general behaviors for the dhcp server as a whole.

The currently supported tags for **dhcp\_server\_settings**:

#### **dhcpdb-write-perf=**

This parameter takes a small integer (like 2 or 5) as input. If set, the write to the */etc/dhcpdb* file will be delayed by the server. This will increase performance for busy servers. If set to a value greater than 2, the server will spawn a new process to do the writing, which will be a considerable performance improvement.

#### **callback-style=old|new**

Callbacks are a powerful feature that allow the system administrator to customize the operation of the server. A user-supplied executable file (typically a shell script) is executed each time one of the main server actions is performed (example: granting a lease). An argument list is passed in with information about the individual client and the lease. The **callback-style=** tag specifies whether the *old* (and confusing) argument list should be used with the **call-on-xxx** feature described below. The *new* (and recommended) argument list is much simpler to use, and is identical for all of the **call-on-xxx** functions. The *new* style simply inserts a value of “00” for fields that are not sensible for a particular callback. The *new* argument list is:

*filename: client-id htype haddr ipaddr subnet-mask lease-expiration hostname gateway*

The *old* argument list is described for each of the individual callbacks below.

#### **call-on-unrequited=filename**

This tag specifies an executable file *filename* that will be called when the server receives a request to which it cannot send a response. Certain arguments will be passed in; the call executed will be:

*filename: client-id htype haddr [gateway]*

where *client-id* is the client ID in hex if present, or 00 if there is no client ID. *htype* is the hardware type as per the ARP section of the “Assigned Numbers” RFC. *haddr* is the hardware address in hex. *gateway* is the IP address of the bootp relay agent. If the packet was not relayed, then this field is absent.

#### **call-on-assignment=filename**

This tag specifies the fully qualified *filename* to be called when an IP address has been assigned

to a new client. Some arguments will be passed in, the call will be made as follows:

*filename: client-id htype haddr ipaddr subnet-mask lease-expiration [hostname]*

where *client-id* is the client ID in hex if present, or 00 if there is no client ID. *htype* is the hardware type as per the ARP section of the "Assigned Numbers" RFC. *haddr* is the hardware address in hex. *ipaddr* is the IP address that was assigned to the client. *subnet-mask* is the subnet mask of the client represented as an IP address. *lease-expiration* is the **bootpd** internal representation of when the lease will expire (based on a C call to `time()`), a value of **ffffffff** represents an infinite lease. If there is a *hostname* associated with this address, then it is the final argument.

#### **call-on-decline=filename**

This tag specifies the fully qualified *filename* to be called when an IP address has been declined by a new client. Some arguments will be passed in, the call will be made as follows:

*filename: client-id htype haddr ipaddr subnet-mask*

where *client-id* is the client ID in hex if present, or 00 if there is no client ID. *htype* is the hardware type as per the ARP section of the "Assigned Numbers" RFC. *haddr* is the hardware address in hex. *ipaddr* is the IP address that was declined by the client. *subnet-mask* is the subnet mask of the client represented as an IP address.

#### **call-on-discard=filename**

This tag specifies the fully qualified *filename* to be called when an IP address has been discarded due to a conflict. Some arguments will be passed in, the call will be made as follows:

*filename: client-id htype haddr ipaddr subnet-mask*

where *client-id* is the client ID in hex if present, or 00 if there is no client ID. *htype* is the hardware type as per the ARP section of the "Assigned Numbers" RFC. *haddr* is the hardware address in hex. *ipaddr* is the IP address that was declined by the client. *subnet-mask* is the subnet mask of the client represented as an IP address.

#### **call-on-release=filename**

This tag specifies the fully qualified *filename* to be called when an IP address has been released by a client. Some arguments will be passed in, the call will be made as follows:

*filename: client-id htype haddr ipaddr lease-expiration*

where *client-id* is the client ID in hex if present, or 00 if there is no client ID. *htype* is the hardware type as per the ARP section of the "Assigned Numbers" RFC. *haddr* is the hardware address in hex. *ipaddr* is the IP address that was released by the client. *lease-expiration* is the **bootpd** internal representation of when the lease would have expired, a value of **ffffffff** represents an infinite lease.

#### **call-on-lease-extend=filename**

This tag specifies the fully qualified *filename* to be called when an IP address lease for a client has been extended. Some arguments will be passed in, the call will be made as follows:

*filename: client-id htype haddr ipaddr subnet-mask lease-expiration*

where *client-id* is the client ID in hex if present, or 00 if there is no client ID. *htype* is the hardware type as per the ARP section of the "Assigned Numbers" RFC. *haddr* is the hardware address in hex. *ipaddr* is the IP address that was assigned to the client. *subnet-mask* is the subnet mask of the client represented as an IP address. *lease-expiration* is the **bootpd** internal representation of when the lease will expire (based on a C call to `time()`), a value of **ffffffff** represents an infinite lease.

#### **call-on-discover=filename**

This tag specifies the fully qualified *filename* to be called when the server receives a discover. It should be noted that this callback can only be used when *callback-style* is set to *new*. The format of the arguments passed to this callback is same as the format specified for *callback-style=new*. If a particular parameter is not known or not required, 00 can be used in its place.

#### **call-on-offer=filename**

This tag specifies the fully qualified *filename* to be called when the server sends an offer to a client. It should be noted that this callback can only be used when *callback-style* is set to *new*. The format of the arguments passed to this callback is same as the format specified for *callback-style=new*. If a particular parameter is not known or not required, 00 can be used in

it's place.

The currently supported tags for `dhcp_pool_group`, `dhcp_device_group`, and `dhcp_default_client_settings`:

**class-name=***classname*

This tag specifies a name to refer to a device group by. It is only applicable to `dhcp_device_group`. The only use that `bootpd` makes of this field is in logging errors found in the configuration of the group.

**pool-name=***poolname*

This tag specifies a name to refer to a pool group by. It is only applicable to `dhcp_pool_group`. The only use that `bootpd` makes of this field is in logging errors found in the configuration of the group.

**class-id=***client-class*

This tag specifies the *client-class* that clients must have to be assigned to this group. This tag is required for `dhcp_device_group` and is inappropriate for any other keyword. Some DHCP clients send out a *client-class* that identifies a class that a client belongs to. For an IP address to be assigned from a device group address pool, not only must the client be on the right subnet, it must send a request with a *client-class* that matches that defined for the `class-id`. This may be specified in either hex or in ASCII (an ASCII string must be enclosed in double quotes).

**subnet-mask=***mask*

This tag specifies the subnet mask for the addresses in the group being defined. It is specified as an IP address. This tag is required for both `dhcp_device_group` and `dhcp_pool_group`, and is inappropriate for `dhcp_default_client_settings`.

**addr-pool-start-address=***IP-address*

This tag specifies the lowest address in the pool group to be assigned. This tag is required for both `dhcp_device_group` and `dhcp_pool_group`, and is inappropriate for `dhcp_default_client_settings`.

**addr-pool-last-address=***ip-address*

This tag specifies the highest address in the pool group to be assigned. This address and the `addr-pool-start-address` define a range of addresses that can be assigned to clients. For the server, no two group address ranges may overlap.

**reserved-for-other=***ip-address-list*

This tag is followed by one address that falls in the range of the group. This address is reserved, and will not be assigned to any clients by the DHCP server. Alternatively, a range of addresses may be defined by giving 2 addresses, with the range being the addresses from the first address up to the second address, inclusively. This tag may be repeated to reserve more addresses in the same group. It is not appropriate for `dhcp_default_client_settings`.

**lease-time=***seconds*

This tag specifies the time in seconds that a lease should be given to each client. The word "infinite" may be used to specify leases that never expire. The default is "infinite." Note that if a client asks for a shorter lease than is configured for it, it will get that shorter lease time. A lease time shorter than 120 seconds will be silently upgraded to 120.

**lease-grace-period=***percent*

This tag specifies the time after a lease expires during which that lease will not be assigned to a new client. *percent* is the percentage of the configured lease time that this grace period lasts. The default is 5%.

**tr=***percent*

This tag specifies the DHCP IP lease renewal time (T1). This is the time interval from lease assignment until when the client attempts to renew the lease. RFC1541 states that T1 defaults to half the lease duration. The minimum value is 40 percent. T1 must always be smaller than T2.

**tv=***percent*

This tag specifies the DHCP IP lease rebind time (T2). This is the time interval from lease assignment until when the client attempts to obtain a new lease from any server. RFC1541 states that T2 defaults to 0.875 times the lease duration. The minimum value is 50 percent. T2 must always be greater than T1.

- lease-policy=***policy*  
This tag specifies whether or not the assigning of new leases can be done. If *policy* is set to **reject-new-clients** then no new clients can get a lease, and only clients with existing leases will get a response. *accept-new-clients* is the default.
- allow-bootp-clients=***boolean*  
This tag specifies whether or not bootp clients can be members of the group being defined. The default is **false**. If *boolean* is **TRUE**, then an IP address may be assigned to a client that doesn't have an entry in the **bootptab** file and that is on the same subnet as the group being defined. This address is treated as an infinite lease, and a boot reply is sent to the client. This tag is is not appropriate for **dhcp\_device\_group**, since bootp clients don't have a client class (and therefore a bootp client would be incapable of matching the client class of the device group). If this tag is used for **dhcp\_default\_client\_settings**, then it is only applicable to pool groups.
- ddns=***ip-address*  
This tag specifies the IP address of the Domain Name Server (DNS) to which dynamic update requests are sent.
- pcsn**  
This tag specifies that the name sent by client should be given preference. As a boolean tag, if set it causes bootpd to accept the name sent by the client (if any). If name is not sent by the client, bootpd tries to find one.
- sp** As a boolean tag, if set it causes bootpd to not use pre-requisite section in the update request when an update request is to be sent to DNS.

DHCP/BOOTP Packet

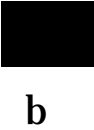
The DHCP/BOOTP packet has the following format:

```
struct dhcp {
    unsigned char    op;                /* packet opcode type */
    unsigned char    htype;             /* hardware addr type */
    unsigned char    hlen;             /* hardware addr length */
    unsigned char    hops;             /* gateway hops */
    unsigned long    xid;               /* 4 bytes transaction ID */
    unsigned short   secs;             /* seconds since boot began */
    unsigned short   flags;            /* if giaddr!=0, client flags */
    struct in_addr   ciaddr;           /* client IP address */
    struct in_addr   yiaddr;           /* 'your' IP address */
    struct in_addr   siaddr;           /* server IP address */
    struct in_addr   giaddr;           /* gateway IP address */
    unsigned char    chaddr[16];       /* client hardware address */
    unsigned char    sname[64];        /* server host name */
    unsigned char    file[128];        /* boot file name */
    unsigned char    options[312];     /* options area */
};
```

DHCP Option Numbers

The DHCP/BootP options discussed above correspond to the option numbers in RFC1533 as follows:

Number	Tag	Description
1	sm	Subnet Mask
2	to	Time Offset
3	gw	Gateways
4	ts	Time Servers
5	ns	IEN 116 Name Servers
6	ds	Domain Name Servers
7	lg	Log Servers
8	cs	Cookie Servers
9	lp	LPR Servers



b

10	im	Impress Servers
11	rl	Resource Location Servers
12	hn	Send Host Name in reply
13	bs	Boot File Size
14	md	Merit Dump File
15	dn	Domain Name
16	ss	Swap Server
17	rp	Root Path
18	ef	Extensions Path
28	ba	Broadcast Address
33	sr	Static Routes
40	yd	NIS Domain
41	ys	NIS Servers
42	nt	NTP Servers
43	v###	Vendor Specific Information
44	na	NetBIOS Name Servers
45	nb	NetBIOS Datagram Distribution Servers
46	nc	NetBIOS Node Type
47	nd	NetBIOS Scope
48	xf	X Font Servers
49	xd	X Display Manager
51	lease-time	IP Address Lease Time
58	tr	Lease Renewal Time (T1)
59	tv	Lease Rebinding Time (T2)
60	class-id	Class Identifier
61	ci	Client Identifier
64	pd	NIS+ Domain
65	ps	NIS+ Servers
69	ms	SMTP Servers

**EXAMPLES**

This is an example of a `/etc/bootptab` file:

```
# Common entry

global.defaults:\
    bf=C2300A:\
    hd=/usr/lib/X11/:\
    hn:\
    ht=ether:\
    vm=rfc1048

# Now the actual individual entries

xterm1:\
    tc=global.defaults:\
    ha=08000903212F:\
    ip=190.40.101.22

xterm2:\
    tc=global.defaults:\
    ha=0800090324AC:\
    ip=190.40.101.35

# Common relay entry.

relay-default:\
    ht=ethernet:\
    bp=15.4.3.136 15.13.6.192:\
    th=2:\
    hp=5:

# Relay entry for node2
```

```

node2:\
    tc=relay-default:\
    ha=08000902CA00:

# Group relay entry

group-machines:\
    tc=relay-default:\
    ha=080009000000:\
    hm=080009000000:

# Turn the relay off (block the relay) for the following machines.

blocked-machines:\
    ht=ethernet:\
    ha=07000A000000:\
    hm=07000A000000:

# Relay definition for all other machines.

all:\
    tc=relay-default:\
    ha=000000000000:\
    hm=000000000000:

```

This is an example of a `/etc/dhcptab` file:

```

# The first entry is for options which define the server's operation.

DHCP_SERVER_SETTINGS:\
    call-on-unrequited="/tmp/unrequited.script" :\
    dhcpdb-write-perf= 3 :\

# The next entry is for options that will be applied to all groups.
# Individual options may be overridden for a specific group if the group
# also configures the option.

DHCP_DEFAULT_CLIENT_SETTINGS:\
    hn:\
    lease-time=10080:\

# The next entry defines an address pool for devices with the class
# id "xterminal" on subnet 15.14.128. Address leases will be granted
# for up to 1 week. The server will use a broadcast message to
# respond to all client requests.

DHCP_DEVICE_GROUP:\
    ba:\
    class-name=SUBNET_128_XTERMINAL_GROUP:\
    class-id="xterminal:"\
    subnet-mask=255.255.255.0 :\
    addr-pool-start-address= 15.14.128.1 :\
    addr-pool-last-address= 15.14.128.254 :\
    lease-time=604800 :\
    lease-grace-period=5 :\

# The next entry grants IP leases to any device on subnet
# 15.13.128. The script /usr/local/bin/assignment.script will be
# run whenever a new lease is granted.

DHCP_POOL_GROUP:\
    pool-name=RED_SUBNET_POOL:\

```

```
call-on-assignment="/usr/local/bin/assignment.script" :\
subnet-mask=255.255.255.0 :\
addr-pool-start-address= 15.13.128.100 :\
addr-pool-last-address= 15.13.128.254 :\
gw=15.13.128.1 :\
```

**b**

This is an example of a `/etc/dhcpdeny` file:

```
# Sample /etc/dhcpdeny file
080009000001
0x080009000002
# Above two clients will not be served!
```

**WARNINGS**

Individual host entries must not exceed 1024 characters.

**AUTHOR**

`bootpd` was developed by Carnegie Mellon University, Stanford University, and HP.

**FILES**

```
/etc/bootptab
/etc/dhcptab
/etc/services
```

**SEE ALSO**

`bootpquery(1M)`, `dhcptools(1M)`, `inetd(1M)`, `tftpd(1M)`, `syslog(3C)`, `hosts(4)`.

DARPA Internet Requests For Comments: RFC865, RFC868, RFC887, RFC951, RFC1010, RFC1034, RFC1048, RFC1084, RFC1395, RFC1533, RFC1534, RFC1541, RFC1542.



**NAME**

bootpquery - send BOOTREQUEST to BOOTP server

**SYNOPSIS**

*/usr/sbin/bootpquery haddr [htype] [options]*

**DESCRIPTION**

**bootpquery** is a diagnostic function used to check the configuration of the Internet Bootstrap Protocol (BOOTP) server, *bootpd*(1M). This function can only be run by the superuser, since it uses reserved ports.

**bootpquery** constructs a boot request with the supplied parameters to send to the BOOTP server, and prints the contents of the BOOTP server reply (as shown in EXAMPLES, below). Note that **bootpquery** formats and prints RFC-1048 or CMU-style vendor information included in the BOOTREPLY.

The BOOTREQUEST packet is broadcast on the BOOTP server port, **bootps**. If a BOOTP server is configured to respond to the request, it returns a BOOTREPLY packet on the BOOTP client port, **bootpc**. **bootpquery** can only display BOOTREPLY packets when the BOOTP server broadcasts the reply on the client port or when the hardware address and IP address supplied in the BOOTREQUEST are those of the host on which **bootpquery** is run.

The following options provide the information for the BOOTREQUEST:

**haddr** Hardware address of the BOOTP client; used in the BOOTREQUEST. A BOOTP server responds if it has configuration information for a host with this link-level address.

**htype** Type of address specified as *haddr*; may be **ether** or **ieee802**. The default address type is **ether**.

**-i ipaddr**

Specify the internet address of the BOOTP client to be used in the BOOTREQUEST. If the BOOTP client does not know its IP address, the BOOTP server supplies it in the BOOTREPLY. Otherwise, the server returns the BOOTREPLY directly to ipaddr.

**-s server**

Specify the name of the BOOTP server to receive BOOTREQUEST. When the BOOTP server is known, the BOOTREQUEST is not broadcast.

**-v vendor**

Specify a vendor name to include vendor information in the BOOTREPLAY. *vendor* can be specified as **rfc1048** or **cmu**. For any other *vendor* specification, the first four characters of the parameter are used as the vendor magic cookie.

**-f**

Specify that **bootpd** should broadcast the reply back. This option is only valid for **bootpd** on the HP-UX 10.0 (or later) release(s).

**-b bootfile**

Specify a boot file needed by the BOOTP client. If a boot file is specified in the BOOTREQUEST, the BOOTP server responds only if the server host can make the file available.

**EXAMPLES**

```
/usr/sbin/bootpquery 02608cee018e ether -s hpserver
```

Received BOOTREPLY from hpserver.hp.com (15.9.18.119)

Hardware Address: 02:60:8c:ee:01:8e

Hardware Type: ethernet

IP Address: 15.9.18.113

Boot file: /export/tftpd/hp-gw2-config

RFC 1048 Vendor Information:

Subnet Mask: 255.255.248.0

Bootfile Size: 6 512 byte blocks

Domain Name Server: 15.9.18.119

Host Name: hp-gw2

**AUTHOR**

**bootpquery** was developed by HP.

**SEE ALSO**

bootpd(1M), tftp(1), tftpd(1M).

DARPA Internet Request For Comments RFC951, RFC1048, RFC1084, RFC1395, RFC1542 Assigned Numbers.

**b**

**NAME**

cachefsstat - Cache File System statistics

**SYNOPSIS****cachefsstat** [ **-z** ] [ *path...* ]**DESCRIPTION**

The **cachefsstat** command displays statistical information about the cache filesystem mounted on *path*. The statistical information includes cache hits and misses, consistency checking, and modification operations. If *path* is not specified, all mounted cache filesystems are used. **cachefsstat** can also be used to reinitialize this information (see **-z** option). The statistical information has the following format:

```
<cache hit rate>
<consistency checks>
<modifies>
```

where:

*cache hit rate*

The percentage of cache hits over the total number of attempts, followed by the actual numbers of hits and misses.

*consistency checks*

The number of consistency checks performed, followed by the number that passed, and the number that failed.

*modifies*

The number of modify operations, including writes, creates, etc.

**Options**

The following option is supported:

**-z** Zero (reinitialize) statistics. Execute **cachefsstat -z** before executing **cachefsstat** again to gather statistics on the cache performance. This option can only be used by the super-user. The statistics printed reflect those just before the statistics are re-initialized.

**EXAMPLES**

```
example% cachefsstat /home/sam
cache hit rate:  73% (1234 hits, 450 misses)
consistency checks:  700 (650 pass, 50 fail)
modifies:  321
```

**EXIT STATUS**

The following exit values are returned:

0 Success.  
non-zero An error has occurred.

**AUTHOR**

**cachefsstat** was developed by Sun Microsystems, Inc.

**SEE ALSO**

cfsadmin(1M).

## NAME

captoinfo - convert a termcap description into a terminfo description

## SYNOPSIS

captoinfo [-lv] [-wn] [*filenames*]

## DESCRIPTION

**captoinfo** looks in *filenames* for *termcap*(3X) descriptions. For each one found, an equivalent *terminfo*(4) description is written to standard output along with any comments found. The short two letter name at the beginning of the list of names in a **termcap** entry, a holdover from Version 6 UNIX, is removed. Any description that is expressed relative to another description (as specified in the **termcap** *tc=* field) is reduced to the minimum superset before output.

If no *filename* is given, the environment variable **TERMCAP** is used for the filename or entry. If **TERMCAP** is a full pathname to a file, only the terminal whose name is specified in the environment variable **TERM** is extracted from that file. If the environment variable **TERMCAP** is not set, the file **/usr/share/lib/termcap** is read.

## Options

*captoinfo* recognizes the following options:

- l Print one field per line. If this option is not selected, multiple fields are printed on each line up to a maximum width of 60 characters.
- v Print (verbose) tracing information as the program runs. Additional -v options print more information (for example -v -v -v or -vvv).
- wn Change the output width to *n* characters.

## DIAGNOSTICS

**tgetent failed with return code n (reason).**

The termcap entry is not valid. In particular, check for an invalid 'tc=' entry.

**unknown type given for the termcap code 'cc'.**

The termcap description had an entry for 'cc' whose type was not boolean, numeric or string.

**wrong type given for the boolean (numeric, string) termcap code 'cc'.**

The boolean termcap entry 'cc' was entered as a numeric or string capability.

**the boolean (numeric, string) termcap code 'cc' is not a valid name.**

An unknown termcap code was specified.

**tgetent failed on TERM=term.**

The terminal type specified could not be found in the termcap file.

**TERM=term: cap cc (info ii) is NULL: REMOVED**

The termcap code was specified as a null string. The correct way to cancel an entry is with an @, as in :bs@:. Giving a null string could cause incorrect assumptions to be made by any software that uses termcap or terminfo.

**a function key for 'cc' was specified, but it already has the value 'vv'.**

When parsing the 'ko' capability, the key 'cc' was specified as having the same value as the capability 'cc', but the key 'cc' already had a value assigned to it.

**the unknown termcap name 'cc' was specified in the 'ko' termcap capability.**

A key that could not be handled was specified in the 'ko' capability.

**the vi character 'v' (info 'ii') has the value 'xx', but 'ma' gives 'n'.**

The 'ma' capability specified a function key with a value different from that specified in another setting of the same key.

**the unknown vi key 'v' was specified in the 'ma' termcap capability.**

A vi key unknown to captoinfo was specified in the 'ma' capability.

**Warning: termcap sg (nn) and termcap ug (nn) had different values.**

terminfo assumes that the sg (now xmc) and ug values were the same.

**Warning: the string produced for 'ii' may be inefficient.**

The parameterized string being created should be rewritten by hand.

Null **termname** given.

The terminal type was null. This occurs when **\$TERM** is null or not set.

cannot open "file" for reading.

The specified file could not be opened.

**Warning:** cannot translate *capability* (unsupported in **terminfo**).

This termcap capability is no longer supported in **terminfo**, and therefore cannot be translated.

## **WARNINGS**

Certain **termcap** defaults are assumed to be true. For example, the bell character (**terminfo** *bel*) is assumed to be **^G**. The linefeed capability (**termcap** *nl*) is assumed to be the same for both **cursor\_down** and **scroll\_forward** (**terminfo** *cul1* and *ind*, respectively). Padding information is assumed to belong at the end of the string.

The algorithm used to expand parameterized information for **termcap** fields such as **cursor\_position** (**termcap** *cm*, **terminfo** *cup*) sometimes produces a string which, though technically correct, may not be optimal. In particular, the rarely used **termcap** operation **%n** produces strings that are especially long. Most occurrences of these less than optimal strings are flagged with a warning message, and may need to be recoded by hand.

HP supports only terminals listed on the current list of supported devices. However, the **terminfo** database contains both supported and nonsupported terminals. If you use nonsupported terminals, they may not work correctly.

## **AUTHOR**

**captoinfo** was developed by AT&T.

## **SEE ALSO**

**tic(1M)**, **untic(1M)**, **curses(3X)**, **termcap(3X)**, **terminfo(4)**, **infocmp(1M)**.

**NAME**

catman - create the cat files for the manual

**SYNOPSIS**

```
/usr/sbin/catman [-A alt-path] [-p] [-m] [-n] [-w] [-z] [sections]
```

**DESCRIPTION**

The **catman** command creates the formatted versions of the online manual from *nroff*(1)-compatible source files. Each manual entry in the **man\*.Z** and **man\*** directories is examined, and those whose formatted versions are missing or out-of-date are recreated. **catman** formats the most recent of the entries, compresses it, and puts it into the appropriate **cat\*.Z** directory.

If any changes are made, **catman** recreates the **/usr/share/lib/whatis** database. By default, the **/usr/share/lib/whatis** database is overwritten. If the **MANPATH** environment variable is set to a non-default set of paths, the old database file is saved in **/usr/share/lib/whatis.old** so that, if desired, the system administrator may merge them together.

By default, **catman** searches the **man\*.Z** and **man\*** subdirectories under the following man directories:

- **/usr/share/man**
- **/usr/contrib/man**
- **/usr/local/man**

If **MANPATH** is set in the environment, the directories given in **MANPATH** are checked instead of the default. See *environ*(5) for a description of the **MANPATH** environment variable.

Before running **catman**, remove any existing **cat\*** directories. If the **-z** option is used, **cat\*.Z** directories should be removed instead. If both **cat\*.Z** and **cat\*** directories exist, *man*(1) updates both directories and more space is used.

Any command-line parameters not starting with **-** are interpreted as a list of manual sections (directories) to search. For example:

```
catman 123
```

restricts updating to manual sections 1, 2, and 3 (directories **man1**, **man2**, and **man3**).

**Options**

**catman** supports the following options:

- m** Create a merged **/usr/share/lib/whatis** database; i.e., information on new manual entries (added since the last time **catman** was run) is merged into the current database rather than overwriting it. Ignored if selected with the **-n** option.
- n** Prevents creation of **/usr/share/lib/whatis**.
- p** Prints what would be done instead of doing it.
- w** Causes only the **/usr/share/lib/whatis** database to be created. No manual reformatting is done.
- z** Puts the formatted entries in the **cat\*** directories rather than in the **cat\*.Z** directories.
- A *alt-path*** Perform actions based on the given alternate root. With this option, *alt-path* will be prepended to all directory paths, including default paths, the paths defined by **MANPATH**, and the path to **/usr/share/lib/whatis**.

**EXTERNAL INFLUENCES****Environment Variables**

**MANPATH** defines parent directories to be used when searching **man\*** and **man\*.Z** directories.

**WARNINGS**

If unformatted manual entries (those in the **../man\*** subdirectories) have been removed since the last time **catman** was run, information in the **/usr/share/lib/whatis** database may be lost. The **-m** option may be used to override this, but may result in repeated lines in the database for the same manual entry.

**EXAMPLES**

Create uncompressed **cat\*** files for sections 1 and 1m of the manual, but don't create the **/usr/share/lib/whatis** database:

```
catman -z -n 11m
```

Run **catman** from a server to create **cat\*** entries for a diskless client under the alternate root **/export/shared\_roots/OS\_700**:

```
catman -A /export/shared_roots/OS_700
```

This will create **cat\*** manpages under:

```
/export/shared_roots/OS_700/usr/share/man/
/export/shared_roots/OS_700/usr/contrib/man/
/export/shared_roots/OS_700/usr/local/man/
```

and a **whatis** file in:

```
/export/shared_roots/OS_700/usr/share/lib/whatis
```

Create **cat\*** entries for an application and merge the information with the **/usr/share/lib/whatis** database:

```
MANPATH=/opt/langtools/man
catman -m
```

Note that you may wish to save **MANPATH** before doing this, so as not to lose your current **MANPATH**.

**AUTHOR**

**catman** was developed by HP and the University of California, Berkeley.

**FILES**

<b>/usr/share/man/man*<i>[.Z]</i>/*</b>	Unformatted ( <i>nroff</i> (1)-compatible source) manual entry files [compressed].
<b>/usr/share/man/cat*<i>[.Z]</i>/*</b>	Formatted manual pages [compressed].
<b>/usr/local/man/man*<i>[.Z]</i>/*</b>	
<b>/usr/local/man/cat*<i>[.Z]</i>/*</b>	
<b>/usr/contrib/man/man*<i>[.Z]</i>/*</b>	
<b>/usr/contrib/man/cat*<i>[.Z]</i>/*</b>	
<b>/usr/share/lib/whatis</b>	Database of manpage entry summaries; utilized by the <b>man -k</b> command.
<b>/usr/sbin/mkwhatis</b>	Command to make <b>whatis</b> database.

**SEE ALSO**

**compress**(1), **fixman**(1M), **man**(1), **environ**(5).

**NAME**

cfl - configure a logical unit (LUN) on a SCSI disk array

**SYNOPSIS**

```
cfl [-L LUN_address] [-a -c list [,list] [-i]] [-b block_size] [-c list [,list]] [-d] [-f flag_word]
    [-k num_log_blocks] [-l sec_tenths] [-n num_log_blocks] [-p list] [-r RAID_level]
    [-s num_log_blocks] [-t reg|sub] [-z num_log_blocks] device_file
```

**DESCRIPTION**

**cfl** sets configuration parameters, and changes the status of a LUN on the HP SCSI disk array associated with *device\_file*.

**NOTE:** **newarray**, a front-end program for **cfl**, is recommended for doing array configuration (see **newarray(1M)**).

**Options**

**-L LUN\_address**

Specifies which SCSI unit address to affect.

**-a -c list [,list] [-i]**

*list* is a comma-separated drive list (**cX***i* **Y**,**cX***i* **Y**,...) describing drives on SCSI channel *X*, and SCSI ID *Y* (where *X* and *Y* are decimal numbers). Multiple *lists* are delimited by space characters.

Add a LUN to the set of LUNs known by the controller. If this option is used, the runstring must also contain a value for the **-c** parameter, and can contain values for all other applicable parameters except **-d** (the delete LUN option). If only the **-c** parameter is supplied, a default RAID-level 0 configuration is created with the drives specified in the parameter list. The user may thus specify all the LUN characteristics in one line; create a default configuration and change a few of the parameters to desired values in one line, or create a default configuration and iteratively change its parameters to the desired values. The **-i** option formats the newly added LUN after configuration. If multiple LUNs are to be added and configured, each LUN must be formatted before any other LUNs can be added and configured.

**-b block\_size** Set the logical block size of the LUN. *block\_size* is specified in bytes.

**-c list [,list2] device\_file**

Assign to the LUN a configuration table that describes which drives are associated with the LUN and specifies the order each drive appears in a data stripe. One, or more tables can be assigned to each LUN, depending on the RAID level. Each table can have a maximum of five drives.

**-d** Delete the LUN from the set of LUNs known by the controller. This option cannot be used simultaneously with the **-a** option.

**-f flag\_word** Assign the desired hexadecimal values, given in *flag\_word*, to the array's two LUN flag bytes. The default *flag\_word* is hex **0072**. User-changeable bits are in Mode Page 0x2b byte 25 (the lsb): bit 4, which enables AEN polling when set; bit 5, which enables parity verification when set, and bit 6, which enables writes with parity verification when set.

**-k num\_log\_blocks**

Set the reconstruction quantity in blocks. This represents the number of blocks reconstructed in a single reconstruction command. Reconstruction commands are issued at an adjustable interval until the LUN is reconstructed (see the **-l** option).

**-l sec\_tenths** Set the reconstruction frequency, the interval between successive reconstruction commands. It is expressed in tenths of a second.

**-n num\_log\_blocks**

Set the number of logical blocks in the LUN.

**-p list**

Create the LUN's disk bit map, which describes the drives associated with the LUN. Either a configuration table or a disk bit map, but not both, is required to configure a LUN; use of the configuration table is recommended.

**-r raid\_level** Set the RAID level of the LUN; valid RAID levels are 0, 1, 3 and 5.



- s num\_log\_blocks**  
Set the number of blocks in a LUN segment, the part of a data stripe residing on a single disk.
- t reg | sub**  
Set the LUN type, regular or sub-LUN. A sub-LUN is a LUN that can share its physical drive(s) with another LUN; usually, its data resides on more than one drive. Configurations involving data striping or mirroring should use sub-LUNs.
- z num\_log\_blocks**  
Set the number of blocks in the first segment of the LUN.

## RETURN VALUE

cfl returns the following values:

- 0** Successful completion.
- 1** Command failed (an error occurred).

## DIAGNOSTICS AND ERRORS

Errors can originate from problems with:

- cfl
- SCSI (device level) communications
- system calls

All error information is printed to stderr.

### Error messages generated by cfl:

```
usage: cfl -L <LUN_addr> -a <-c ...> [-i] <special> add LUN
cfl -L <LUN_addr> -b <n> <special> set logical block size
cfl -L <LUN_addr> -c <<cXiY,... [cXiY,...]> | none> <special> build
                                config table(s)
cfl -L <LUN_addr> -d <special> delete LUN
cfl -L <LUN_addr> -f <n> <special> set LUN flags
cfl -L <LUN_addr> -k <n> <special> set reconstruction amt in
                                blocks
cfl -L <LUN_addr> -l <n> <special> set reconstruction frequency
cfl -L <LUN_addr> -n <n> <special> set number of blocks in LUN
cfl -L <LUN_addr> -p <cXiY,...> <special> build disk bit map
cfl -L <LUN_addr> -r <n> <special> set RAID level
cfl -L <LUN_addr> -s <n> <special> set segment size in blocks
cfl -L <LUN_addr> -t <reg | sub> <special> set LUN type
cfl -L <LUN_addr> -z <n> <special> set segment 0 size in blocks
```

An error in command syntax has occurred. No valid tags were present, or an illegal tag was encountered. Re-enter the command with all required arguments. If a syntax error occurs in a runstring with a legal tag, only the template for that tag will be displayed.

#### cfl: Arg incompatible with other

One of the arguments is incompatible with another, for example, when the **-a** (add LUN) and **-d** (delete LUN) are both on the command line.

#### cfl: Arg out of range

One of the arguments is larger than its allowed maximum value (or smaller than its allowed minimum value), or is incorrect in form. Check the size, and form of each argument and make appropriate corrections.

#### cfl: device busy

To ensure that **cfl** does not modify a disk array that is being used by another process, **cfl** attempts to obtain exclusive access to the disk array. If the disk array is already opened by another process (for example, LVM — the Logical Volume Manager), a “**device busy**” error message is returned by the driver. To eliminate the “**device busy**” condition, determine what process has the device open. In the case of LVM, it is necessary to deactivate the volume group containing the array before configuring the array (see *vgchange(1M)*).

#### cfl: LUN does not exist

The addressed LUN is not known to the array controller. Only the **-a** option can operate on an

unconfigured LUN. The **-d** option ignores references unconfigured LUNs (and does nothing with them).

**cfl: LUN # too big**

The LUN number, which is derived from the device special file name, is out of range.

**cfl: Multiple args of same type**

An argument occurs more than once on the command line.

**cfl: Not a disk array**

The device being addressed did not identify itself as a SCSI disk array product that is supported by **cfl**.

**cfl: Not a raw file**

**cfl** must be able to open the device file for raw access (the character device file).

**cfl: Transfer length error**

The amount of data actually sent to or received from the device was not the expected amount.

#### SCSI (device level) communication errors:

Sense data associated with the failed operation is printed.

#### Error messages generated by system calls:

**cfl** uses the following system calls:

**malloc()**, **free()**, **stat()**, **open()**, **close()**, **read()**, **write()**, and **ioctl()**.

Documentation for these HP-UX system calls contains information about the specific error conditions associated with each call. **cfl** does not alter the value of **errno**. The interpretation of **errno** for printing purposes is performed by the system utility **strerror()**.

#### EXAMPLES

To delete LUN 5 associated with **/dev/rdisk/c2t0d0**:

```
cfl -L 5 -d /dev/rdisk/c2t0d0
```

To add the LUN 0 associated with **/dev/rdisk/c2t0d0**, which will have the following characteristics: logical block size 512 bytes, RAID level of 5, auto reconstruct disabled, reconstruction amount of 64 blocks, reconstruction frequency of .2 seconds, segment size of 64 blocks, type sub-LUN, segment zero size of 1, and drives with SCSI ID 1 on channels 1 through 5, to be striped in the channel order 3, 5, 1, 2 and 4:

```
cfl -L 0 -a -b 512 -r 5 -f 0072 -k 64 -l 2 -n 123456 -s 64
-t sub -z 1 -c c3i1,c5i1,c1i1,c2i1,c4i1 /dev/rdisk/c2t0d0
```

#### WARNING

Changing any configuration parameter except the reconstruction frequency and reconstruction quantity puts the affected LUN in an unusable ("dead") state. You must reformat the LUN before it can be used with the new configuration values. Formatting a LUN destroys all of its user data.

#### DEPENDENCIES

The HP C2425 and HP C2427 disk arrays are only supported on Series 700 systems running HP-UX version 9.0X.

The HP C2430 disk array is supported on Series 700 and 800 systems running HP-UX versions 9.0X and 10.0X.

#### AUTHOR

**cfl** was developed by HP.

#### SEE ALSO

**newarray(1M)**, **arraytab(4)**, **vgchange(1M)**.

**NAME**

cfsadmin - administer disk space used for caching file systems with the Cache File-System (CacheFS)

**SYNOPSIS**

```
cfsadmin -c [ -o cacheFS-parameters ] cache_directory
cfsadmin -d [ cache_ID | all ] cache_directory
cfsadmin -l cache_directory
cfsadmin -s [ mntpt1... | all ]
cfsadmin -u [ -o cacheFS-parameters ] cache_directory
```

**DESCRIPTION**

The **cfsadmin** command provides the following functions:

- cache creation
- deletion of cached file systems
- listing of cache contents and statistics
- resource parameter adjustment when the file system is unmounted.

For each form of the command, with the exception of the **-s** form, you must specify a cache directory, that is, the directory under which the cache is actually stored. A path name in the front file system identifies the cache directory. For the **-s** form of the command, you must specify a mount point.

You can specify a cache ID when you mount a file system with CacheFS, or you can let the system generate one for you. The **-l** option includes the cache ID in its listing of information. You must know the cache ID to delete a cached file system.

**Options**

- c** Create a cache under the directory specified by *cache\_directory*. This directory must not exist prior to cache creation.
- d** Remove the file system whose cache ID you specify and release its resources, or remove all file systems in the cache by specifying **all**. After deleting a file system from the cache, you must run the *fsck\_cacheFS*(1M) command to correct the resource counts for the cache.
- l** List file systems stored in the specified cache, as well as statistics about them. Each cached file system is listed by cache ID. The statistics document resource utilization and cache resource parameters.
- s** Request a consistency check on the specified file system (or all cacheFS mounted file systems). The **-s** option will only work if the cache file system was mounted with **demandconst** enabled (see *mount\_cacheFS*(1M)). Each file in the specified cache file system is checked for consistency with its corresponding file in the back file system. Note that the consistency check is performed file by file as files are accessed. If no files are accessed, no checks are performed. Use of this option will not result in a sudden "storm" of consistency checks.
- u** Update resource parameters of the specified cache directory. Parameter values can only be increased. To decrease the values, you must remove the cache and recreate it. All file systems in the cache directory must be unmounted when you use this option. Changes will take effect the next time you mount any file system in the specified cache directory. The **-u** option with no **-o** option sets all parameters to their default values.

**Operands**

*cache\_directory* The directory under which the cache is actually stored.

*mntpt1* The directory under which the CacheFS is mounted.

**CacheFS Resource Parameters**

You can specify the following cacheFS resource parameters as arguments to the **-o** option. Separate multiple parameters with commas.

**maxblocks**=*n* Maximum amount of storage space that CacheFS can use, expressed as a percentage of the total number of blocks in the front file system. If CacheFS does not have exclusive use of the front file system, there is no guarantee that all the space the **maxblocks** parameter allows will be available. The default is 90.

**minblocks=*n*** Minimum amount of storage space, expressed as a percentage of the total number of blocks in the front file system, that CacheFS is always allowed to use without limitation by its internal control mechanisms. If CacheFS does not have exclusive use of the front file system, there is no guarantee that all the space the **minblocks** parameter attempts to reserve will be available. The default is 0.

**threshblocks=*n*** A percentage of the total blocks in the front file system beyond which CacheFS cannot claim resources once its block usage has reached the level specified by **minblocks**. The default is 85.

**maxfiles=*n*** Maximum number of files that CacheFS can use, expressed as a percentage of the total number of inodes in the front file system. If CacheFS does not have exclusive use of the front file system, there is no guarantee that all the inodes the **maxfiles** parameter allows will be available. The default is 90.

**minfiles=*n*** Minimum number of files, expressed as a percentage of the total number of inodes in the front file system, that CacheFS is always allowed to use without limitation by its internal control mechanisms. If CacheFS does not have exclusive use of the front file system, there is no guarantee that all the inodes the **minfiles** parameter attempts to reserve will be available. The default is 0.

**threshfiles=*n*** A percentage of the total inodes in the front file system beyond which CacheFS cannot claim inodes once its usage has reached the level specified by **minfiles**. The default is 85.

**maxfilesize=*n*** Largest file size, expressed in megabytes, that CacheFS is allowed to cache. The default is 3.

Note: You cannot decrease the block or inode allotment for a cache. To decrease the size of a cache, you must remove it and create it again with different parameters.

## EXAMPLES

The following example creates a cache directory named **/cache**:

```
cfsadmin -c /cache
```

The following example creates a cache named **/cache1** that can claim a maximum of 60 percent of the blocks in the front file system, can use 40 percent of the front file system blocks without interference by CacheFS internal control mechanisms, and has a threshold value of 50 percent. The threshold value indicates that after CacheFS reaches its guaranteed minimum, it cannot claim more space if 50 percent of the blocks in the front file system are already used.

```
cfsadmin -c -o maxblocks=60,minblocks=40,threshblocks=50 /cache1
```

The following example changes the **maxfilesize** parameter for the cache directory **/cache2** to 2 megabytes:

```
cfsadmin -u -o maxfilesize=2 /cache2
```

The following example lists the contents of a cache directory named **/cache3** and provides statistics about resource utilization:

```
cfsadmin -l /cache3
```

The following example removes the cached file system with cache ID 23 from the cache directory **/cache3** and frees its resources (the cache ID is part of the information returned by **cfsadmin -l**):

```
cfsadmin -d 23 /cache3
```

The following example removes all cached file systems from the cache directory **/cache3**:

```
cfsadmin -d all /cache3
```

The following example checks for consistency all filesystems mounted with **demandconst** enabled. No errors will be reported if no **demandconst** filesystems were found.

```
cfsadmin -s all
```

**AUTHOR**

**cfsadmin** was developed by Sun Microsystems, Inc.

**SEE ALSO**

fsck\_cachefs(1M), mount\_cachefs(1M).

**NAME**

chroot - change root directory for a command

**SYNOPSIS**

`/usr/sbin/chroot newroot command`

**DESCRIPTION**

The **chroot** command executes *command* relative to the *newroot*. The meaning of any initial slashes (/) in path names is changed for *command* and any of its children to *newroot*. Furthermore, the initial working directory is *newroot*.

Note that command suffixes that affect input or output for the **chroot** command use the original root, not the new root. For example, the command:

```
chroot newroot command > x
```

locates file **x** relative to the original root, not the new one.

The *command* variable includes both the command name and any arguments.

The new root path name is always relative to the current root. Even if a **chroot** is currently in effect, the *newroot* argument is relative to the current root of the running process.

This command is restricted to users with appropriate privileges.

**EXTERNAL INFLUENCES****International Code Set Support**

Single- and multibyte character code sets are supported.

**WARNINGS**

*command* cannot be in a shell script.

Exercise extreme caution when referring to special files in the new root file system.

**chroot** does not search the **PATH** environment variable for the location of *command*, so the absolute path name of *command* must be given.

When using **chroot** to establish a new environment, all absolute path name references to the file system are lost, rendering shared libraries inaccessible. If continued access to shared libraries is needed for correct operation, the shared libraries and the dynamic loader *must* be copied into the new root environment.

**SEE ALSO**

chdir(2), chroot(2).

**STANDARDS CONFORMANCE**

**chroot**: SVID2, SVID3, XPG2, XPG3

**NAME**

ch\_rc - change system configuration file

**SYNOPSIS**

```
/usr/sbin/ch_rc -a|-r|-l [-v] [-A] [-R root] [-p {parameter|parameter=value}...] [file...]
```

**DESCRIPTION**

ch\_rc manages the addition, modification, removal, and retrieval of information stored in files having the format of those in the `/etc/rc.config.d` directory.

Parameter names are treated as strings. Thus, `X[0]` has no special meaning to ch\_rc in relation to other parameters named `X[1]` or `X`.

**Options**

*file* Specify the file(s) to be used as the configuration database. If no file is specified, the set of files used by ch\_rc defaults to `/etc/TIMEZONE` and all files in the `/etc/rc.config.d` directory.

Modification and deletion of configuration parameters occurs in the file where the parameter is found.

**-a** Add or modify a parameter definition. For each parameter specified on the command line, if the parameter is found in the specified (or default) files, it is modified to reflect the specified value. If the parameter is not found, it is added to the specified file(s).

If a new parameter is being defined, one or more files must be specified on the command line; the specified files are those in which the parameter will be defined.

**-r** Remove a parameter definition. For each parameter name specified on the command line, remove any occurrence of that parameter from the specified file(s).

**-l** List configuration values. For each parameter specified on the command line, output every definition of the parameter from the specified file(s). Output consists of only the values, one per line.

**-p** Specify a parameter name or name/value pair. If a name and value is expected, but only a name is specified, the value will be set to the empty string. For example, specifying `FOO` or `FOO=` will result in `FOO` and `FOO=` respectively.

Due to shell quoting rules, if you need a quoted parameter value, you must protect the quotes from the shell. For example,

```
ch_rc -a -p VALUE="a b c" <file>
```

yields:

```
VALUE=a b c
```

which is an error, whereas,

```
ch_rc -a -p VALUE='"a b c"' <file>
```

yields:

```
VALUE="a b c"
```

**-v** Verbose. When used with the **-l** option, the **-v** option causes a verbose listing to be output. This listing includes a filename followed by the entire line containing the specified parameter for each occurrence of the parameter.

**-A** The **-A** option is used to list all occurrences of array parameters matching the parameters specified on the command line.

For example,

```
ch_rc -l -A -v -p ZZZ file
```

may emit the following output:

```
file: ZZZ[0]=zero
file: ZZZ[5]=five
file: ZZZ[9]=fred
```

**-R root** Normally, the files specified on the command line are used as specified. By specifying a *root* directory with the **-R** option, all files (including the default files if none are specified) will be interpreted relative to *root*.

For example, if *root* is specified as */foo* and */etc/TIMEZONE* is specified on the command line, it will be interpreted as */foo/etc/TIMEZONE*.

## RETURN VALUE

**ch\_rc** exits with one of the following values:

- 0 add/delete/list successful
- 1 command line syntax/usage error
- 2 can not access one or more of the listed (or default) files
- 3 can not open/create/write file
- 4 memory error
- 5 no files specified on command line for add option

## EXAMPLES

Files in the */etc/rc.config.d* directory have the following format:

```
# Comments are preceded by pound signs and
# are always on a line of their own.
# Blank lines are allowed.

VARIABLE=value
VARIABLE_2=value2
VARIABLE_3[1]=value3
VARIABLE_3[2]=value4

# All parameters are defined on a single line
# Parameters must not be exported
```

## WARNINGS

**ch\_rc** does not interpret configuration files; it only does pattern matching. As a result, if comments appear on lines containing parameter definitions, the comments will also appear in output when using the **-l** option.

**ch\_rc** cannot parse multiple parameter definitions which occur on the same line of a file.

## AUTHOR

**ch\_rc** was developed by HP.

## FILES

```
/etc/rc.config    system configuration database driver file
/etc/rc.config.d  directory containing system configuration files
```

## SEE ALSO

rc.config(4).



**NAME**

clri - clear inode

**SYNOPSIS**

*/usr/sbin/clri special i-number ...*

**DESCRIPTION**

The **clri** command clears the inode *i-number* by filling it with zeros. *special* must be a special file name referring to a device containing a file system. For proper results, *special* should not be mounted (see WARNINGS below). After **clri** is executed, all blocks in the affected file show up as "missing" in an **fsck** of *special* (see *fsck(1M)*). This command should only be used in emergencies.

Read and write permission is required on the specified *special* device. The inode becomes allocatable.

**WARNINGS**

The primary purpose of this command is to remove a file that for some reason does not appear in any directory. If it is used to clear an inode that does appear in a directory, care should be taken to locate the entry and remove it. Otherwise, when the inode is reallocated to some new file, the old entry in the directory will still point to that file. At that point, removing the old entry destroys the new file, causing the new entry to point to an unallocated inode, so the whole cycle is likely to be repeated again.

If the file system is mounted, **clri** is likely to be ineffective.

**DEPENDENCIES**

**clri** operates only on file systems of type **hfs**.

**SEE ALSO**

*fsck(1M)*, *fsdb(1M)*, *ncheck(1M)*, *fs(4)*.

**STANDARDS CONFORMANCE**

*clri*: SVID2, SVID3

**NAME**

clrsvc - clear x25 switched virtual circuit

**SYNOPSIS**

**clrsvc** *line pad-type*

**DESCRIPTION**

**clrsvc** clears any virtual circuit that might be established on the specified *line*. *pad-type* indicates to **clrsvc** what **opx25** script to run from **/usr/lbin/uucp/X25**.

**DEPENDENCIES**

HP 2334A is the only PAD supported at this time, and results in an **opx25** execution of **HP2334A.clr**.

**EXAMPLES**

A typical invocation is:

```
/usr/lbin/uucp/X25/clrsvc /dev/x25.1 HP2334A
```

**AUTHOR**

**clrsvc** was developed by HP.

**SEE ALSO**

getx25(1M), opx25(1M), getty(1M), login(1), uucp(1).

**NAME**

config - configure and build an HP-UX system

**SYNOPSIS**

```
/usr/sbin/config [-c c_file] [-l m_file] [-m master] [-r path] [-s|-u] [-S] [-t] system_file
/usr/sbin/config -M module_name [[-M module_name]...] [-m master] [-u]
```

**DESCRIPTION**

**config** is used to configure the following parts of the operating system:

- device drivers
- swap and dump devices
- tunable system parameters
- kernel modules

**config** supports the following configurations:

- whole kernel configuration (first form)  
Both the static kernel (**vmunix**) and dynamically loadable modules are generated, and a system reboot is necessary.
- dynamically loadable module configuration (second form)  
Specified loadable modules are dynamically generated and registered with the current system. The newly configured services are available immediately, without requiring a system reboot.

Kernel modules can either be static modules or dynamically loadable modules.

The first form is used to configure the entire kernel; that is, the static kernel and all kernel modules. This type of configuration is called a whole kernel configuration. The second form is used to configure only the dynamically loadable modules.

Static modules are maintained in individual object files which are included or excluded from the static kernel (**vmunix**) based on whether the features they support are required in the system. Such modules are non-loadable and remain linked into the kernel.

Dynamically loadable modules are also maintained in individual object files but they are not statically linked into the kernel. Loadable modules can be configured to be included or excluded from the kernel dynamically, without having to relink the entire kernel or reboot the system. The loadable image generated during the configuration of such modules may be auto-loaded or unloaded by the kernel or demand-loaded or unloaded by the system administrator.

See the *Managing Systems and Workgroups* for information on how to include or remove a subsystem, file system, or kernel module, and how to boot the system.

**Whole Kernel Configuration (First Form)**

To configure a whole kernel, **config** reads the user-provided description of an HP-UX system (*system\_file*), the system description files for kernel modules, and the master kernel configuration table information.

Note that the system file and system description files for kernel modules should only be modified by using the **kmsystem** or **kmtune** system administration commands.

For all kernel modules to be configured, **config** checks the interface functions or symbols used by the modules. If modules rely on symbols not covered by the **\$INTERFACE** section of its master file, configuration fails. Otherwise, **config** generates the following output files and directories:

- C program source files (**conf.c** and **space.h**) that define the configuration tables for various parts of the system. Unless kernel modules are configured, these files will not be generated.
- C program header file (**tune.h**) that defines tunable parameters of the system required by kernel and kernel modules.
- C program source files (**mod\_conf.c**) that are required by kernel modules. If a **space.h** header file is provided with a module, it is included by the source file.
- a makefile (**config.mk**) to compile the C program produced and relink the newly configured system with statically linked kernel module object file (**vmunix\_test**), and to generate kernel

symbol table (**syntab**).

- another makefile (**config.mod**) to generate all dynamically loadable modules to be configured.
- a directory (**dlkm.vmunix\_test**) to store the generated dynamically loadable modules, kernel symbol table, and module registry file associated with the kernel being built (**vmunix\_test**). This directory here after will be referred to as the kernel function set directory. The files in this directory will be referred to as the kernel function set files.

Many header files are needed to compile **conf.c**. Also, archive library files containing the kernel objects are needed to link the kernel. These files are supplied with the system and are contained in the directories found under **/usr/conf**.

**config.mod** and the module registry file are not generated if there are no dynamically loadable modules being configured.

**config** executes the **make** command to compile **conf.c**, to link the kernel with the appropriate kernel libraries and statically linked modules, and to generate the kernel symbol table. It also executes the **make** command with **config.mod** to compile dynamically loadable modules.

The **make** command create several files in a working directory whose location depends on the name of the system file. If *system\_file* is **/stand/system**, the working directory is **/stand/build**; otherwise the working directory is the current directory. With successful completion of the **make** command, the following files are generated:

- kernel file  
The kernel file **vmunix\_test** is generated in the working directory.
- kernel function set directory  
The kernel function set directory (**dlkm.vmunix\_test**) is created in the working directory.
- kernel symbol table  
The kernel symbol table **syntab** is generated in the kernel function set directory.
- dynamically loadable modules  
Dynamically loadable modules are generated under a subdirectory (**mod.d**) of the kernel function set directory.

If the **-u** option is specified, the newly generated kernel file and its kernel function set directory are automatically copied to their default locations, **/stand/vmunix** and **/stand/dlkm**, respectively, on system shutdown or restart. The previous kernel file and its kernel function set directory will be saved as **/stand/vmunix.prev** and **/stand/dlkm.vmunix.prev**, respectively.

### Options for Whole Kernel Configuration

When configuring a whole kernel, the **config** command recognizes the following arguments:

- c c\_file**  
Specify the name of the C program source file produced by **config**. The default file name is **conf.c**.
- l m\_file**  
Specify the name of the makefile which is generated by **config**. This is the makefile which will be used by **config** to compile the C program source file and make the new kernel. The default file name is **config.mk**.
- m master**  
Specify the name of the master kernel configuration information file or directory that **config** should use in creating source files and makefiles. If *master* is a directory, **config** reads all files in that directory to create its data structures. If *master* is a file, only that file is read for creating data structures for **config**. By default, **config** reads the files in the directory **/usr/conf/master.d**. **/usr/conf/master.d** is supplied as part of the HP-UX operating system and should not be modified by anyone who does not fully understand its structure and purpose.
- r path**  
Search the directory *path* for the libraries and header files needed for making the kernel. By default, **config** uses the directory **/usr/conf**.

- S      Statically link all kernel modules into the kernel file. This option only takes effect if kernel modules are configured as loadable.
- s      Stop after generating source files and makefiles. **make** is not executed and no kernel (**vmunix\_test**) or kernel modules are created. The **-s** option cannot be used with the **-u** option.
- t      Give a short table of major device numbers for the character and block devices, the card drivers, the streams drivers and modules that require link routines, the streams devices and the streams modules named in *system\_file*. These tables may be useful when creating special device files.
- u      Invoke **kmupdate** after successfully configuring the new kernel environment. The **-u** option cannot be used together with the **-s** option.

#### *system\_file*

The file containing configuration information for the user's system. The default system file is **/stand/system** and when this file is used as input to **config**, the resulting output is placed in the directory **/stand/build**. If a file other than **/stand/system** is used for *system\_file*, **config** places its output files in the current directory. The system file is divided into two parts: the first part (mandatory) contains driver specifications; the second part (optional) contains system-dependent information.

### Constructing an HP-UX System File

The first part of *system\_file* is used to configure:

- device drivers
- pseudo-drivers
- subsystems

Each line has the following format:

*devname* where *devname* is the driver or subsystem name as it appears in the alias tables, driver install tables or the device tables in the files in the directory, **/usr/conf/master.d**. For example, **scsi** selects the driver for SCSI disk drives, **scsitape** selects the driver for SCSI tape drives, and **nfs** selects the NFS subsystem. Together, the files in **/usr/conf/master.d** contain a complete list of configurable devices, cards, subsystems, and pseudo-drivers.

The optional second part of *system\_file* is used to:

- define the swap device
- define the dump device(s)
- provide a mapping of a driver to a hardware path
- define status and values of selected system parameters.

Lines are constructed as indicated below for each category.

#### (1) *Swap device specification*

No more than one swap specification is allowed. If a swap specification is not given, the system will be configured to swap on the root device at the end of the filesystem.

**swap** *hw\_path* *offset* [*blocks*]

Configure the swap device location and its size as specified. Arguments are interpreted as follows:

*hw\_path*      The hardware path representing the device to configure as the swap device or the string default may be used to indicate using the root device.

*offset*      The swap area location. Boundaries are located at 1K-byte intervals. A negative value specifies that a file system is expected on the device. At boot-up, the super block is read to determine the exact size of the file system, and this value is put in *offset*. If the swap device is auto-configured, this is the mechanism used. If the super block is invalid, the entry will be skipped so that a corrupted super block will not later cause the entire file system to be corrupted by configuring the swap area on top of it. A positive or zero value for *offset* specifies the minimum area that must be reserved. Zero means to reserve no area at the head of the device. A zero value implies that there is no file system on the device.

C

*blocks* The number (in decimal) of 1K-byte disk blocks in the swap area. For this swap device specification, only the *blocks* parameter is optional. Zero is the default for auto-configuration. If *blocks* is zero, the entire remainder of the device is automatically configured in as swap area. If *blocks* is non-zero, its absolute value is treated as an upper bound for the size of the swap area. Then, if the swap area size has actually been cut back, the sign of *blocks* determines whether *blocks* remains as is, resulting in the swap area being adjacent to the reserved area, or whether *blocks* is bumped by the size of the unused area, resulting in the swap area being adjacent to the tail of the device.

**swap** *hw\_path options*

Configure the swap device at the location specified using the options specified. The *hw\_path* argument is interpreted as it is in the previous example.

The *options* field is used to specify a section. It is only offered for backwards compatibility purposes. For example, **s3** would put the swap area on section 3.

**swap** **lvol**

Configure swap on a logical volume.

**swap** **none**

Configure the kernel with no swap device.

(2) *Dump device(s) specification*

One or more dump specifications are allowed. If a dump specification is not given, then the primary swap area will be used.

**dump** *hw\_path [options]*

Configure the dump device location and its size as specified. Arguments are interpreted as follows:

*hw\_path* The hardware path representing the device to configure as a dump device or the string default may be used to indicate using the primary swap area.

*options* This field is used to specify a section. It is only offered for backwards compatibility purposes. For example **s3** would put the dump area at section 3.

**dump** **lvol**

Configure dump on a logical volume.

**dump** **none**

Configure the kernel with no dump device.

(3) *Device driver to hardware path*

One or more driver to hardware path specifications is allowed. If a driver statement is provided, the specified software module is forced into the kernel I/O system at the given hardware path. This can be used to make the system recognize a device that could not be recognized automatically.

**driver** *hw\_path driver\_name*

Bind the driver into the kernel I/O system at the given hardware path. Arguments are interpreted as follows:

*hw\_path* The hardware path representing the device to bind the software with.

*driver\_name*

The name of the software module to bind into the kernel at the specified hardware path.

(4) *System parameters*

These parameters should not be modified without a full understanding of the ramifications of doing so (see the *Managing Systems and Workgroups* manual).

Each line contains two fields. The first field can contain up to 20 characters; the second field up to 60 characters. Each line is independent, optional, and written in the following format:

*parameter\_name*    *number or formula*

Interprocess communication consists of messages (**mesg**), semaphores (**sema**) and shared memory (**shmem**) features. If **mesg**, **sema**, and/or **shmem** are specified as 0, the kernel code for these features is not included. If they are specified as 1, the kernel code is included; this is the

default. The features can be specified independent of each other. If the code is included, the parameters listed below can be modified:

<b>mesg</b>	1
<b>msgmap</b>	<i>number or formula</i>
<b>msgmax</b>	<i>number or formula</i>
<b>msgmnb</b>	<i>number or formula</i>
<b>msgmni</b>	<i>number or formula</i>
<b>msgseg</b>	<i>number or formula</i>
<b>msgssz</b>	<i>number or formula</i>
<b>msgtql</b>	<i>number or formula</i>
<b>sema</b>	1
<b>semaem</b>	<i>number or formula</i>
<b>semmap</b>	<i>number or formula</i>
<b>semnmi</b>	<i>number or formula</i>
<b>semmns</b>	<i>number or formula</i>
<b>semmnu</b>	<i>number or formula</i>
<b>semume</b>	<i>number or formula</i>
<b>semvmx</b>	<i>number or formula</i>
<b>shmem</b>	1
<b>shmall</b>	<i>number or formula</i>
<b>shbrk</b>	<i>number or formula</i>
<b>shmmax</b>	<i>number or formula</i>
<b>shmmni</b>	<i>number or formula</i>
<b>shmseg</b>	<i>number or formula</i>

### Dynamically Loadable Module Configuration (Second Form)

To configure loadable kernel modules, **config** builds components for the module specified by the **-M** option. If the **-M** option is specified in conjunction with the **-u** option, then **config** builds the loadable module and call upon **kmupdate** to update the loadable image of that module in memory. Updating the loadable image implies replacing the existing loadable image with the newly created loadable image, re-registering the module with the new information, if required, and performing any type-specific initialization; e.g. recreating the special device file, if needed.

When configuring loadable modules, **config** reads the running kernel's system description file, system description files for kernel modules, and the master kernel configuration information table.

Note that system description files for kernel modules should only be modified by using **kmsystem** or **kmtune** system administration commands.

To configure loadable modules, **config** checks the interface functions or symbols used by the modules. If the modules rely on symbols not covered by the **\$INTERFACE** section of its master file, configuration fails. **config** then generates the following output files:

- C program header file (**tune.h**) that defines tunable parameters of the system.
- C program source file (**mod\_conf.c**) that is required by each kernel module.
- makefile (**config.mod**) to generate specified dynamically loadable modules.
- module registry entry to register the specified modules.

After the above files have been generated, **config** executes the **make** command with **config.mod** to generate dynamically loadable module.

With a successful **make**, the object files of dynamically loadable modules are generated and placed under the kernel function set directory.

If the **-u** option is specified, **kmupdate** is executed by **config**.

All kernel module related files are needed to configure the module. See **kminstall(1M)** for details on kernel module files.

### Options for Loadable Module Configuration

When configuring a loadable module, **config** recognizes the following options:

**-M** *module\_name*

Configure the specified loadable module only. A kernel file is not generated in this case. If

successful, the loadable image of the module is generated.

If the specified module is a stub module (see *master(4)*), **config** prints a message and fails. An entire kernel build is required to configure stub modules.

**-m** *master*

Specify the name of the master kernel configuration information file or directory that **config** should use in creating source files and makefiles. If *master* is a directory, **config** reads all files in that directory to create its data structures. If *master* is a file, only that file is read for creating data structures for **config**. By default, **config** reads the files in the directory */usr/conf/master.d*. */usr/conf/master.d* is supplied as part of the HP-UX operating system and should not be modified by anyone who does not fully understand its structure and purpose.

**-u** Invoking **kmupdate** to update the module.

### Kernel Module System Description File

Kernel module description files are placed under */stand/system.d*. A system file for a module is named after the module name and is unique.

Each file consists of three mandatory and one optional sections.

#### **\$VERSION:**

The line starting with **\$VERSION** indicates the version number for the file format. Version is defined as a decimal number and starts from one.

Format is:

```
$VERSION version_number
```

Example:

```
$VERSION 1
```

#### **\$CONFIGURE:**

The line starting with **\$CONFIGURE** indicates whether the module needs to be configured. If the second field is either **Y** or **y**, the module will be configured on the next build. If the field is either **N** or **n**, the module will not be configured on the build.

Format is:

```
$CONFIGURE {Y|y|N|n}
```

Example:

```
$CONFIGURE Y
```

#### **\$LOADABLE:**

The line starting with **\$LOADABLE** indicates how the module will be configured. If the second field is either **Y** or **y**, the module will be configured as a dynamically loadable module.

If the field is either **N** or **n**, the module will be statically linked into the kernel.

If the master file for the module does not have a **\$LOADABLE** section, then the system file should not have one either.

Format is:

```
$LOADABLE {Y|y|N|n}
```

Example:

```
$LOADABLE Y
```

#### **\$TUNABLE** (Optional system parameter section)

The section between the lines starting with **\$TUNABLE**, and with **\$\$\$** indicates tunable parameters of the module.

The above mentioned keywords e.g. **\$VERSION**, **\$CONFIGURE** must start at the beginning of the line without white space or tabs. Field separators can be single white spaces, tabs, or a combination of both.



Lines starting with an asterisk (\*) are comment lines

## RETURN VALUE

**config** returns 0 upon successful completion. If an error occurs, a non-zero value is returned.

## DIAGNOSTICS

All error messages and warning messages of **config** are sent to stderr. Status report messages are sent to stdout. These messages are self explanatory. Some messages are generated by **make** or commands called from the makefiles.

## FILES

<code>/usr/conf/master.d/*</code>	Default input master configuration tables
<code>/usr/conf/interface.d/*</code>	Interface files
<code>/usr/conf/gen/config.sys</code>	Contains skeleton makefile
<code>/usr/conf/gen/config.lm</code>	Contains skeleton makefile for kernel modules
<code>/stand/system</code>	Default system file
<code>/stand/system.d/*</code>	Default kernel module description files
<code>/stand/build/conf.c</code>	Default output configuration table
<code>/stand/build/tune.h</code>	Default output system parameter table
<code>/stand/build/config.mk</code>	Default output <i>make</i> (1) script
<code>/stand/build/config.mod</code>	Default kernel module <i>make</i> (1) script
<code>/stand/build/vmunix_test</code>	Default kernel made by <b>config</b>
<code>/stand/build/dlkm.vmunix_test/symtab</code>	Default kernel symbol table
<code>/stand/build/dlkm.vmunix_test/mod.d/*</code>	Default kernel module loadable image
<code>/stand/build/dlkm.vmunix_test/mod_register</code>	Default module registry file

## SEE ALSO

kminstall(1M), kmmodreg(1M), kmsystem(1M), kmtune(1M), kmupdate(1M), make(1), interface(4), master(4).

**NAME**

convertfs - convert an HFS file system to allow long file names

**SYNOPSIS**

`/usr/sbin/convertfs [-q] [special-file]`

**DESCRIPTION**

The **convertfs** command converts an existing HFS file system supporting the default maximum file name length of 14 characters into one that supports file names up to 255 characters long. Once an HFS file system is converted to long file names, it cannot be restored to its original state, since the longer file names require a directory representation that is incompatible with the default HFS directory format. Since this is an irreversible operation, **convertfs** prompts for verification before it performs a conversion.

**convertfs** forces the system to reboot if the root file system is converted. When converting the root file system, the system should be in single-user mode, with all unnecessary processes terminated and all non-root file systems unmounted. Except for the root file system, **convertfs** requires that the file system to be converted be unmounted.

If invoked without arguments, **convertfs** interactively prompts the user with a list of the HFS file systems from `/etc/fstab`. One or more or all of the listed file systems can be selected for conversion. Typically, it is desirable to convert all of the file systems in `/etc/fstab` to avoid inconsistencies between two file systems mounted on the same system.

**convertfs** can also be invoked with an argument of either a block or character *special-file* of a file system to be converted. Only the block special file should be specified for a mounted root file system.

As part of the conversion process, **convertfs** performs an **fsck** on each file system (see *fsck*(1M)).

**Options**

**-q** Do quietly. **convertfs** will perform the conversions without querying the user. Normally **convertfs** prompts the user before converting a file system.

**RETURN VALUE**

**convertfs** returns the following values:

<b>0</b>	Success. Either <b>convertfs</b> successfully converted the file system, or the file system already allowed long file names.
<b>non-0</b>	Failure. <b>convertfs</b> was not able to convert the file system due to some failure in processing.

**AUTHOR**

**convertfs** was developed by HP.

**FILES**

`/etc/fstab` Default list of file systems to check.

**SEE ALSO**

*fsck*(1M), *mkfs*(1M), *newfs*(1M), *fs*(4), *fstab*(4).

**NAME**

convert\_awk - converts old sendmail.cf files to new format

**SYNOPSIS**

`/usr/newconfig/etc/mail/convert_awk`

**DESCRIPTION**

**convert\_awk** is an **awk** program that will convert pre-HP-UX 10.20 **sendmail.cf** files into the format required by sendmail 8.7 and up.

To run it, use:

```
awk -f convert_awk < old.cf > new.cf
```

Note that the new sendmail.cf files offer a wealth of new options and features. You should STRONGLY consider making a new **sendmail.cf** file from the distribution version or from the **m4** macros, which are provided in `/usr/newconfig/etc/mail/cf`.

**SEE ALSO**

sendmail(1M).

  
C

**NAME**

cpset - install object files in binary directories

**SYNOPSIS**

```
cpset [-o] object directory [-mode [-owner [-group]]]
```

**DESCRIPTION**

The **cpset** command installs the specified *object* file in the given *directory*. The *mode*, *owner*, and *group*, of the destination file can be specified on the command line. If this data is omitted, two results are possible:

- If you have administrative permissions (that is, your numerical ID is less than 100), the following defaults are provided:

```
mode      0555
owner     bin
group     bin
```

- If you do not have administrative permissions, the default *mode*, *owner*, and *group* of the destination file are the same as yours.

The **-o** option forces **cpset** to move *object* to **OLDobject** in the destination directory before installing the new object.

**cpset** reads the **/etc/src/destinations** file to determine the final destination of the file to be installed. The **destinations** file contains pairs of path names separated by spaces or tabs. The first name is the "official" destination (for example: **/usr/bin/echo**). The second name is the new destination. If **echo** is moved from **/usr/bin** to **/usr/local/bin**, the entry in **destinations** would be:

```
/usr/bin/echo    /usr/local/bin/echo
```

When the actual installation happens, **cpset** verifies that the "old" pathname does not exist. If a file exists at that location, **cpset** issues a warning and continues.

This file does not exist on a distribution tape; it is used by sites to track local command movement. The procedures used to build the source are responsible for defining the "official" locations of the source.

**Cross Generation**

The environment variable **ROOT** is used to locate the destination file (in the form **\$ROOT/etc/src/destinations**). This is necessary in the cases where cross generation is being done on a production system.

**EXAMPLES**

If you are an administrator, all of the following examples have the same effect. They copy file **echo** into **/usr/bin** with *mode*, *owner*, and *group* set to 0555, **bin**, **bin**, respectively:

```
cpset echo /usr/bin 0555 bin bin
cpset echo /usr/bin
cpset echo /usr/bin/echo
```

If you are not an administrator, the last two examples set *mode*, *owner*, and *group* to your current values.

**SEE ALSO**

chacl(1), make(1), install(1M), acl(5).

**NAME**

crashconf - configure system crash dumps

**SYNOPSIS**

```
/sbin/crashconf [-arv] [-i|-e class] ... [device...]
```

**DESCRIPTION**

**crashconf** displays and/or changes the current system crash dump configuration. The crash dump configuration consists of three lists:

- The *crash dump device* list. This list identifies all devices that can be used to store a crash dump.
- The *included class* list. This list identifies all system memory classes that *must* be included in any crash dump.
- The *excluded class* list. This list identifies all system memory classes that *should not* be included in a crash dump.

Most system memory classes are in neither the included class list nor the excluded class list. Instead, the system determines whether or not to dump those classes of memory based on the type of crash that occurs.

Note the system operator may request a full crash dump at the time the dump is taken. In this case, a full dump will be performed regardless of the contents of the excluded class list.

Any changes to the configuration take effect immediately and remain in effect until the next system reboot, or until changed with a subsequent invocation of **crashconf**.

*device* specifies a block device file name of a device that is a valid destination for crash dumps. All such devices listed on the command line will be added to the end of the current list of crash dump devices, or will replace the current list of crash dump devices, depending on whether **-r** is specified.

*class* is the name (or number) of a system memory class which should be added to the appropriate class list. The list of system memory classes can be obtained using **crashconf -v**. The memory page size is 4Kb.

*class* may also be the word **all**, in which case all classes are added to the appropriate list. (The effect of adding all classes to the included class list is to force full crash dumps under all circumstances. The effect of adding all classes to the excluded class list is to disable crash dumps.)

**Options**

- a The file **/etc/fstab** is read, and all dump devices identified in it will be added to (or will replace) the current list of crash dump devices. This is in addition to any crash dump *devices* specified on the command line. See *fstab(4)* for information on the format of **/etc/fstab**.
- e The *classes* specified with **-e** will be added to (or will replace) the list of excluded (i.e., should not dump) classes. If any of those classes are present in the current included class list, they will be removed from it.
- i The *classes* specified with **-i** will be added to (or will replace) the list of included (i.e., must dump) classes. If any of those classes are present in the current excluded class list, they will be removed from it.
- r Specifies that any changes should replace, rather than add to, the current configuration. Thus, if *devices* or **-a** are specified, the current crash dump device list is replaced with new contents; if *classes* are specified with **-e**, they replace the list of currently excluded classes, and if *classes* are specified with **-i**, they replace the list of currently included classes.
- v Displays the current crash dump configuration. This is the default option if no arguments are specified. If any changes to the current configuration are specified on the same command line as **-v**, the configuration will be displayed *after* the requested changes are made.

**RETURN VALUE**

Upon exit, **crashconf** returns the following values:

- 0 Success.
- 1 The requested configuration changes could not be made.

**WARNINGS**

The output of **crashconf** is not designed to be parsed by applications or scripts, but only to be read by humans. The output format may change without notice. Applications which require crash dump

configuration information should retrieve that information using *pstat(2)*.

Dump devices created by *lvcreate(1M)* must be contiguous (**-Cy** option) with bad block relocation turned off (**-rn** option).

**AUTHOR**

**crashconf** was developed by HP.

**SEE ALSO**

*lvcreate(1M)*, *crashconf(2)*, *pstat(2)*, *fstab(4)*.

C

**NAME**

crashutil - manipulate crash dump data

**SYNOPSIS**

`/usr/sbin/crashutil [-q] [-v version] source [destination]`

**DESCRIPTION**

**crashutil** copies and preserves crash dump data, and performs format conversions on it. Common uses of **crashutil** include:

- Copying portions of a dump that still reside on a raw dump device into a crash dump directory.
- Converting between different formats of crash dumps.
- Copying crash dumps from one directory, or medium, to another.

**crashutil** will write to its *destination* the crash dump it reads from its *source*. The crash dump format used to write the *destination* is specified with **-v**; if **-v** is not specified, the *destination* will have the same format as the *source*. If no *destination* is specified, *source* is used; the format conversion will be done in place in the *source*, without copying. When **crashutil** completes successfully, the entire contents of the crash dump will exist at *destination*; any portions that had still been on raw dump devices will have been copied to *destination*.

There are three known dump formats:

- COREFILE** (Version 0) This format, used up through HP-UX 10.01, consists of a single file containing the physical memory image, with a 1-to-1 correspondence between file offset and memory address. Normally there is an associated file containing the kernel image. *sources* or *destinations* of this type must be specified as two pathnames to plain files, separated by whitespace; the first is the core image file and the second is the kernel image file.
- COREDİR** (Version 1) This format, used in HP-UX 10.10, 10.20, and 10.30, consists of a **core.n** directory containing an **INDEX** file, the kernel (**vmunix**) file, and numerous **core.n.m** files, which contain portions of the physical memory image. *sources* or *destinations* of this type should be specified as the pathname to a core directory.
- CRASHDIR**  
**CURRENT** (Version 2 — the current version) This format, used in HP-UX 11.00 and later, consists of a **crash.n** directory containing an **INDEX** file, the kernel and all dynamically loaded kernel module files, and numerous **image.m.p** files, each of which contain portions of the physical memory image and metadata describing which memory pages were dumped and which were not. *sources* or *destinations* of this type should be specified as the pathname to a crash directory.

Other formats, for example tape archival formats, may be added in the future.

When the *source* and *destination* are different types of files — for example, when *source* is a directory and *destination* is a pair of plain files — both must be specified.

**Options**

- q** (Quiet) Disables the printing of progress messages. Warning and error messages are still printed.
- v *version*** Specifies the version of the destination format. Allowed values are **COREFILE**, **COREDİR**, **CRASHDIR**, 0, 1, or 2. Also allowed is the keyword **CURRENT**, which specifies that the destination format should be the same as the current source format. **CURRENT** is the default if **-v** is not specified.

**RETURN VALUE**

Upon exit, **crashutil** returns the following values:

- 0** The operation was successful.
- 1** The operation failed, and an appropriate error message was printed.

**EXAMPLES**

An HP-UX 11.00 crash dump was saved by **savecrash(1M)** to **/var/adm/crash/crash.2**. The **-p** flag was specified to **savecrash**, specifying that only those portions of the dump which were endangered by swap activity should be saved; the rest are still resident in the raw dump devices. To save the remainder of the dump into the crash dump directory, use:

```
crashutil /var/adm/crash/crash.2
```

If preferred, the completed crash dump directory could be in a different location — perhaps on another machine via NFS:

```
crashutil /var/adm/crash/crash.2 /nfs/remote/otherdir
```

To debug this crash dump using tools which do not understand the most current crash dump format, convert it to the older core directory format:

```
crashutil -v COREDIR /var/adm/crash/crash.2 /tmp/oldcoredir
```

or the even older "core file and kernel" format:

```
crashutil -v COREFILE /var/adm/crash/crash.2 /tmp/corefile  
/tmp/kernfile
```

**AUTHOR**

**crashutil** was developed by HP.

**SEE ALSO**

savecrash(1M).



**NAME**

create\_sysfile - create a kernel system file

**SYNOPSIS**

`/usr/sbin/sysadm/create_sysfile` [*outfile*]

**DESCRIPTION**

The **create\_sysfile** command creates a kernel generation description file (system file) which can be used as input to the command **config**. The system file is built according to the drivers required by the current system hardware. This command is intended for use during the install process when the system does not have a system file.

The **create\_sysfile** command first chooses a template file based on the CPU type of the machine, then it scans the system hardware and includes all drivers it can identify to run the existing hardware. If *outfile* is specified, the resulting system file is sent to *outfile*. If *outfile* is not specified, the output is placed in the file **/stand/system**.

**RETURN VALUE**

The **create\_sysfile** command returns zero upon normal completion or 1 if an error occurred.

**DIAGNOSTICS**

Errors are sent to stderr. Most of the diagnostic messages from **create\_sysfile** are self-explanatory. Errors cause **create\_sysfile** to halt immediately.

**AUTHOR**

**create\_sysfile** was developed by HP.

**FILES**

`/usr/conf/gen/templates/*`  
`/usr/conf/master.d/*`

**SEE ALSO**

**config(1M)**, **master(4)**.

NAME

cron - timed-job execution daemon

SYNOPSIS

/usr/sbin/cron

DESCRIPTION

**cron** executes commands at specified dates and times. Regularly scheduled commands can be specified according to instructions placed in crontab files. Users can submit their own crontab files with a **crontab** command (see *crontab(1)*). Users can submit commands that are to be executed only once with an **at** or **batch** command.

Since **cron** never exits, it should be executed only once. This is best done by running **cron** from the initialization process with the startup script **/sbin/init.d/cron** (see *init(1M)*).

**cron** only establishes a schedule for crontab files and **at/batch** command files during process initialization and when it is notified by **at**, **batch**, or **crontab** that a file has been added, deleted, or modified.

When **cron** executes a job, the job's user and group IDs are set to those of the user who submitted the job.

Spring and Autumn Time Transitions

On the days of daylight savings (summer) time transition (in time zones and countries where daylight savings time applies), **cron** schedules commands differently from normal.

In the following description, an **ambiguous time** refers to an hour and minute that occurs twice in the same day because of a daylight savings time transition (usually on a day during the Autumn season). A **nonexistent time** refers to an hour and minute that does not occur because of a daylight savings time transition (usually on a day during the Spring season). **DST-shift** refers to the offset that is applied to standard time to result in daylight savings time. This is normally one hour, but can be any combination of hours and minutes up to 23 hours and 59 minutes (see *tztab(4)*).

When a command is specified to run at an ambiguous time, the command is executed only once at the *first* occurrence of the ambiguous time.

When a command is specified to run at a nonexistent time, the command is executed after the specified time by an amount of time equal to the DST-shift. When such an adjustment would conflict with another time specified to run the command, the command is run only once rather than running the command twice at the same time.

Commands that are scheduled to run during all hours (there is a **\*** in the hour field of the crontab entry) are scheduled without any adjustment.

EXTERNAL INFLUENCES

Environment Variables

**LANG** determines the language in which messages are displayed.

If **LANG** is not specified or is set to the empty string, it defaults to "C" (see *lang(5)*). If any internationalization variable contains an invalid setting, all internationalization variables default to "C" (see *environ(5)*).

DIAGNOSTICS

A history of all actions taken by **cron** is recorded in **/var/adm/cron/log**.

EXAMPLES

The following examples assume that the time zone is **MST7MDT**. In this time zone, the DST transition occurs one second before 2:00 a.m. and the DST-shift is 1 hour.

Consider the following entries in a crontab file:

#	Minute	Hour	MonthDay	Month	Weekday	Command
#	-----	-----	-----	-----	-----	-----
	0	01	*	*	*	Job_1
	0	02	*	*	*	Job_2
	0	03	*	*	*	Job_3
	0	04	*	*	*	Job_4
	0	*	*	*	*	Job_hourly
	0	2,3,4	*	*	*	Multiple_1
	0	2,4	*	*	*	Multiple_2

For the period of 1:00 a.m. to 4:00 a.m. on the days of DST transition, the results will be:

Job	Times Run in Fall	Times Run in Spring
Job_1	01:00 MDT	01:00 MST
Job_2	02:00 MDT	03:00 MDT
Job_3	03:00 MST	03:00 MDT
Job_4	04:00 MST	04:00 MDT
Job_hourly	01:00 MDT	01:00 MST
	02:00 MDT	
	02:00 MST	
	03:00 MST	03:00 MDT
	04:00 MST	04:00 MDT
Multiple_1	02:00 MDT	
	03:00 MST	03:00 MDT
	04:00 MST	04:00 MDT
Multiple_2	02:00 MDT	03:00 MDT
	04:00 MST	04:00 MDT

WARNINGS

In the Spring, when there is a nonexistent hour because of daylight savings time, a command that is scheduled to run multiple times during the nonexistent hour will only be run once. For example, a command scheduled to run at 2:00 and 2:30 a.m. in the **MST7MDT** time zone will only run at 3:00 a.m. The command that was scheduled at 2:30 a.m. will not be run at all, instead of running at 3:30 a.m.

DEPENDENCIES

HP Process Resource Manager

If the optional HP Process Resource Management (PRM) software is installed and configured, jobs are launched in the initial process resource group of the user that scheduled the job. The user's initial group is determined at the time the job is started, not when the job is scheduled. If the user's initial group is not defined, the job runs in the user default group (**PRMID=1**). See *prmconfig(1)* for a description of how to configure HP PRM, and *prmconf(4)* for a description of how the user's initial process resource group is determined.

AUTHOR

**cron** was developed by AT&T and HP.

FILES

/var/adm/cron	Main <b>cron</b> directory
/var/spool/cron/atjobs	Directory containing <b>at</b> and <b>batch</b> job files
/var/spool/cron/crontabs	Directory containing crontab files
/var/adm/cron/log	Accounting information

SEE ALSO

at(1), crontab(1), sh(1), init(1M), queuedefs(4), tzt(4).

HP Process Resource Manager: *prmconfig(1)*, *prmconf(4)* in *HP Process Resource Manager User's Guide*.

STANDARDS CONFORMANCE

**cron**: SVID2, SVID3

**NAME**

cuegetty - set terminal type, modes, speed, and line discipline for *cue*(1)

**SYNOPSIS**

```
/usr/sbin/cuegetty [-L nls_language] [-T terminal_type] [-h] [-t timeout] line [speed]
```

**DESCRIPTION**

The **cuegetty** command, which is very similar to *getty*(1M), is the second process in the series, (*init-cuegetty-cue-work session*) that ultimately connects a user with the HP-UX CUE system. It is invoked by **init** to monitor the terminal lines configured on a system (see *init*(1M)). Each **cuegetty** process resets its process group using **setpgrp**, opens a particular terminal line, and usually sleeps in the **open( )** until the machine senses a hardware connection for the terminal. When **open( )** returns, **cuegetty** attempts to adapt the system to the terminal speed and type, and displays the contents of the */etc/issue* file, if it exists. Lastly, **cuegetty** invokes **cue** which displays the Login screen and performs user validation (see *cue*(1)).

To start **cuegetty**, an entry for **cuegetty** should be placed in the */etc/inittab* file. A typical CUE entry in the */etc/inittab* file resembles the following:

```
cue:2:respawn:/usr/sbin/cuegetty -L fr_FR.roman8 -h tty0p1
```

See */usr/newconfig/etc/cue.inittab* for an example */etc/inittab* file. See *cue*(1) for more details on the CUE system.

**Configuration Options and Arguments**

**cuegetty** recognizes the following arguments:

- |                   |   |
|-------------------|---|
| <i>line</i>       | Name of a tty line in <i>/dev</i> to which <b>cuegetty</b> is to attach itself. <b>cuegetty</b> uses this string as the name of a file in the <i>/dev</i> directory to open for reading and writing. By default <b>cuegetty</b> forces a hangup on the line by setting the speed to zero before setting the speed to the default or specified speed. However, when <b>cuegetty</b> is run on a direct port, <b>cuegetty</b> does not force a hangup on the line since the driver ignores changes to zero speed on ports open in direct mode (see <i>modem</i> (7)). |
| -L                | <b>nls_language</b> is used to set the language for the CUE login screens. If the message catalog, <b>cue.cat</b> , does not exist for <b>nls_language</b> , the default native language, C, is used.   |
| -T                | <b>terminal_type</b> is used to specify the type of terminal that <b>cuegetty</b> will be initiated on. Allowed values are <b>vt320</b> , <b>vt100</b> , <b>wy60</b> , and <b>hp</b> . The default is <b>hp</b> .   |
| -h                | Tells <b>cuegetty</b> not to force a hangup on the line before setting the speed to the default or specified speed.   |
| -t <i>timeout</i> | <b>cuegetty</b> exits if the open on the line succeeds and nothing is typed within <i>timeout</i> seconds.  |
| <i>speed</i>      | A label to a speed and tty definition in the file <i>/etc/gettydefs</i> . This definition tells <b>cuegetty</b> at what speed to initially run, what the login message should look like, what the initial tty settings are, and what speed to try next should the user indicate that the speed is inappropriate (by typing a <i>break</i> character). The default <i>speed</i> is 300 baud.   |

When no optional arguments appear on the command line, **cuegetty** sets the terminal interface as follows:

- Interface *speed*: 300 baud
- Raw mode (awaken on every character)
- Echo suppressed
- Parity: either
- New-line characters: convert to carriage-return, line-feed pair
- Expand tabs on the standard output
- Type login message then read user's name, one character at a time

- If a null character (or framing error) is received, assumed it to be the result of the user pushing the “break” key. This causes **cuegetty** to attempt the next *speed* in the series. The series that **cuegetty** tries is determined by what it finds in **/etc/gettydefs**.

After interface set-up is complete, **cue** is started to accept and validate the user name and password.

## WARNINGS

If a supported non-HP terminal (or an HP terminal such as HP 700/60 in VT320, VT100 or WYSE60 mode) is required to run **cuegetty**, make sure that a correct terminal type is specified using the **-T** option. For example, if you want to run **cuegetty** on a vt100 terminal, you should make an entry in the **/etc/inittab** file such as the following entry:

```
tty1:23:respawn:cuegetty -T vt100 -h ttylp1 9600
```

Absence of the **-T** option causes **cuegetty** to assume terminal to be a HP terminal which may then cause the terminal to behave incorrectly and may not even allow user to login.

## DEPENDENCIES

**cuegetty** is available only on Series 800 systems, and is compatible only with the following terminals:

HP 700/92   HP 700/94   HP 2392   HP 2394   VT 100   WYSE 60

See *WARNINGS* if you intend to use a non-HP terminal (or an HP terminal such as HP 700/60 in VT320, VT100, or WYSE60 mode).

## FILES

<b>/etc/gettydefs</b>	contains speed and terminal settings used by <b>cuegetty</b>
<b>/etc/inittab</b>	<b>init</b> reads this file to determine which processes to spawn
<b>/etc/issue</b>	contains issue identification data
<b>/usr/newconfig/etc/cue.inittab</b>	sample <b>inittab</b> file with <b>cuegetty</b> entry

## SEE ALSO

**cue**(1), **env**(1), **nlsinfo**(1), **getty**(1M), **init**(1M), **ioctl**(2), **gettydefs**(4), **inittab**(4), **environ**(5), **hpnls**(5), **lang**(5), **termio**(7).

## NAME

dcc - control read and write caching for HP SCSI disk array drives

## SYNOPSIS

dcc [*options*] [*drive\_list*] *device\_file*

## DESCRIPTION

dcc displays or changes the read-ahead caching status, and write-immediate reporting status of selected drives on the HP SCSI disk array referenced by *device\_file*.

### Options

- d Display only. Displays the read-ahead caching and write immediate reporting status of all selected drives on the HP SCSI disk array. For HP C2430 disk array devices, the number and size of cache segments is displayed. This option cannot be used with any other option.
- r on Read on. Enables read-ahead caching on all selected drives of the HP SCSI disk array. Can be used in combination with one of the write-immediate reporting options.
- r off Read off. Disables read-ahead caching on all selected drives of the HP SCSI disk array. Can be used in combination with one of the write-immediate reporting options.
- w on Write on. Enables write-immediate reporting on all selected drives of the HP SCSI disk array. Can be used in combination with one of the read-ahead caching options.
- w off Write off. Disables write immediate reporting on all selected drives of the HP SCSI disk array. Can be used in combination with one of the read-ahead caching options.
- s num\_segments Set the number of cache segments. This option is unique to the HP C2430 disk array. The disk mechanism cache can be segmented into 1, 2, 4, 8 or 16 segments. The default is 2 segments. This option cannot be used with other options.
- drive\_list Specify a set of drives. If this optional list is absent, the default set of affected drives is all drives attached to the controller. The list is in the form cXiY,... where X (a decimal number) represents SCSI channel number, and Y (a decimal number) represents the SCSI ID of the drive. Multiple drives in the list are separated by commas.

## RETURN VALUE

dcc returns the following values:

- 0 Successful completion.
- 1 Command failed (an error occurred).

## ERROR MESSAGES

Errors can originate from problems with:

- dcc
- SCSI (device level) communications
- system calls

### Error messages generated by dcc:

usage: dcc options [cXiY,...] <special>

An error in command syntax has occurred. Enter command again with the required arguments, in the order shown.

dcc: Arg out of range

One of the arguments is larger than its allowed maximum value (or smaller than its allowed minimum value), or is incorrect in form. Check the size, and form of each argument and make appropriate corrections.

dcc: device busy

To ensure that dcc does not modify a disk array that is being used by another process, dcc

attempts to obtain exclusive access to the disk array. If the disk array is already opened by another process (for example, LVM — the Logical Volume Manager), a “**device busy**” error message is returned by the driver. To eliminate the “**device busy**” condition, determine what process has the device open. In the case of LVM, it is necessary to deactivate the volume group containing the array before configuring the array (see *vgchange(1M)*).

**dcc: LUN does not exist**

The addressed LUN is not known to the array controller.

**dcc: LUN # too big**

The LUN number, which is derived from the device special file name, is out of range.

**dcc: Not a raw file**

Utilities must be able to open the device file for raw access.

**dcc: Not an HP SCSI disk array**

The device is not an HP SCSI disk array.

**dcc: Transfer length error**

The amount of data actually sent to (or received from) the device was not the expected amount.

**SCSI (device level) communication errors:**

Sense data associated with the failed operation is printed.

**Error messages generated by system calls:**

**dcc** uses the following system calls:

`malloc()`, `free()`, `stat()`, `open()`, `close()`, `read()`, `write()`, and `ioctl()`.

Documentation for these HP-UX system calls contains information about the specific error conditions associated with each call. **dcc** does not alter the value of `errno`. The interpretation of `errno` for printing purposes is performed by the system utility `strerror()`.

**EXAMPLES**

To display the status of read and write caching on all the drives of the disk array `/dev/rdisk/c2t2d0` on a Series 700:

```
dcc -d /dev/rdisk/c2t2d0
```

To enable write-immediate reporting on a list of drives on the disk array `/dev/rdisk/c2t2d0` on a Series 800:

```
dcc -won c2i0,c1i0,c5i0,c4i1 /dev/rdisk/c2t2d0
```

To disable read caching and write-immediate reporting on the drives of the disk array `/dev/rdisk/c2t4d0` on a Series 700:

```
dcc -roff -woff /dev/rdisk/c2t4d0
```

To set the number of cache segments on the HP C2430 disk array `/dev/rdisk/c2t2d0` to 4 on a Series 800:

```
dcc -s 4 /dev/rdisk/c2t2d0
```

**DEPENDENCIES**

The HP C2425 and HP C2427 disk arrays are only supported on Series 700 systems running HP-UX version 9.0X.

The HP C2430 disk array is supported on Series 700 and 800 systems running HP-UX versions 9.0X and 10.0X.

**AUTHOR**

**dcc** was developed by HP.

**NAME**

dcopy - copy HFS file system with compaction

**SYNOPSIS**

```
/usr/sbin/dcopy [-d] [-fsize[:isize]] [-F hfs] [-scyl:skip] [-v] [-V] source_fs destination_fs
```

**DESCRIPTION**

The **dcopy** command copies an existing HFS file system (*source\_fs*) to a new HFS file system (*destination\_fs*), appropriately sized to hold the reorganized results. For best results, the source file system should be a raw device, and the destination file system should be a block device. Always run **dcopy** on unmounted file systems. (In the case of the root file system, copy it to a new minidisk.)

If no options are specified, **dcopy** copies files from *source\_fs*, compressing directories by removing vacant entries and spacing consecutive blocks in a file by the optimal rotational gap. If options such as **-f** or **-s** are specified, the destination file system structure will be different from that of the source file system.

**dcopy** makes the destination file system identical to the source file system and preserves the pack and volume labels. Thus, to compress a file system without moving it, use **dcopy** to copy the files to another file system and the **dd** command to copy the file back (see *dd(1)*).

Directory compression is accomplished by running **dcopy** on the primary copy of the file system and allowing the modified directories to propagate to the other copies of the file system in the normal manner.

**Options**

**dcopy** recognizes the following options:

- d** Move subdirectories to the beginning of directories.
- fsize[:isize]** Specify the file system size (*fsize*) and inode-list size (*isize*) in blocks. If this option is not specified, the source file-system value is used.
- F hfs** Specify the HFS file system type. The type of a file system can be determined with the **fstyp** command (see *fstyp(1M)*). See *DEPENDENCIES*.
- scyl:skip** Supply device information for creating the best organization of blocks in a file. *cyl* is the number of block per cylinder; *skip* is the number of blocks to skip.
- v** Report size of source and destination file system.
- V** Echo the completed command line, but performs no other actions. The command line is generated by incorporating the user-specified options and other information derived from */etc/fstab*. This option allows you to verify the command line.

**EXAMPLES**

**dcopy** can be executed with or without options. If no options are specified as in this example, the source and destination file systems are identical. Any differences between the two file systems lie only in the available disk space.

```
dcopy /dev/rdisk/c2d0s4 /dev/dsk/c2d0s5
```

If options are specified, expect a major difference between the source and destination file system structure:

```
dcopy -F hfs -f40960:260 -s45:5 -d /dev/rdisk/c2d0s4 /dev/dsk/c2d0s5
```

**WARNINGS**

**dcopy** produces invalid results if run on a mounted file system.

The figures specified in option arguments cannot be smaller than corresponding figures in the source file system.

**DEPENDENCIES**

**dcopy** only operates on HFS file systems.

**AUTHOR**

**dcopy** was developed by HP.

**SEE ALSO**

*dd(1)*, *fstyp(1M)*.



**STANDARDS CONFORMANCE**  
**dcopy: SVID3**



d

**NAME**

devnm - device name

**SYNOPSIS**

`/usr/sbin/devnm [name ... ]`

**DESCRIPTION**

For each *name* specified, the **devnm** command identifies the special file associated with the mounted file system where the named file or directory resides.

**EXAMPLES**

The command:

`/usr/sbin/devnm /usr`

produces:

`/dev/dsk/c1d0s9 /usr`

if `/usr` is mounted on `/dev/dsk/c1d0s9`.

**FILES**

`/dev/dsk/*`

`/etc/mnttab` Mounted file system table.

**STANDARDS COMPLIANCE**

**devnm**: SVID2, SVID3

**NAME**

df - report number of free file system disk blocks

**SYNOPSIS**

`/usr/bin/df [-F FStype] [-befgiklnv] [-t|-P] [-o specific_options] [-V] [special|directory]...`

**DESCRIPTION**

The **df** command displays the number of free 512-byte blocks and free inodes available for file systems by examining the counts kept in the superblock or superblocks. If a *special* or a *directory* is not specified, the free space on all mounted file systems is displayed. If the arguments to **df** are path names, **df** reports on the file systems containing the named files. If the argument to **df** is a *special* of an unmounted file system, the free space in the unmounted file system is displayed.

**Options**

**df** recognizes the following options:

- b** Report only the number of kilobytes (KB) free.
- e** Report the number of files free.
- f** Report only the actual count of the blocks in the free list (free inodes are not reported).
- F *FStype*** Report only on the *FStype* file system type (see *fstyp*(1M)).
- g** Report the entire structure described in *statvfs*(2).
- i** Report the total number of inodes, the number of free inodes, number of used inodes, and the percentage of inodes in use.
- k** Report the allocation in kilobytes (KB).
- l** Report on local file systems only.
- n** Report the file system name. If used with no other options, display a list of mounted file system types.
- o *specific\_options*** Specify options specific to each file system type. *specific\_options* is a comma-separated list of suboptions intended for a specific *FStype* module of the command. See the file-system-specific manual entries for further details.
- P** Report the name of the file system, the size of the file system, the number of blocks used, the number of blocks free, the percentage of blocks used and the directory below which the file system hierarchy appears.
- t** Report the total allocated block figures and the number of free blocks.
- v** Report the percentage of blocks used, the number of blocks used, and the number of blocks free. This option cannot be used with other options.
- V** Echo the completed command line, but perform no other action. The command line is generated by incorporating the user-specified options and other information derived from */etc/fstab*. This option allows the user to verify the command line.

**EXTERNAL INFLUENCES****Environment Variables**

**LC\_MESSAGES** determines the language in which messages are displayed.

If **LC\_MESSAGES** is not specified in the environment or is set to the empty string, the value of **LANG** is used as a default for each unspecified or empty variable. If **LANG** is not specified or is set to the empty string, a default of "C" (see *lang*(5)) is used instead of **LANG**.

If any internationalization variable contains an invalid setting, **df** behaves as if all internationalization variables are set to "C". See *environ*(5).

**International Code Set Support**

Single-byte and multi-byte character code sets are supported.

**EXAMPLES**

Report the number of free disk blocks for all mounted file systems:

```
df
```

Report the number of free disk blocks for all mounted HFS file systems:

```
df -F hfs
```

Report the number of free files for all mounted NFS file systems:

```
df -F nfs -e
```

Report the total allocated block figures and the number of free blocks, for all mounted file systems:

```
df -t
```

Report the total allocated block figures and the number of free blocks, for the file system mounted as **/usr**:

```
df -t /usr
```

**FILES**

<b>/dev/dsk/*</b>	File system devices
<b>/etc/fstab</b>	Static information about the file systems
<b>/etc/mnttab</b>	Mounted file system table

**SEE ALSO**

du(1), df\_hfs(1M), df\_vxfs(1M), fsck(1M), fstab(4), fstyp(1M), statvfs(2), mnttab(4).

**STANDARDS CONFORMANCE**

**df**: SVID2, SVID3, XPG2, XPG3, XPG4

**NAME**

df - report number of free CDFS, HFS, or NFS file system disk blocks

**SYNOPSIS**

/usr/bin/df [-F *FStype*] [-befgiklnv] [-B] [-o *specific\_options*] [-V] [*special* | *directory*]...

**DESCRIPTION**

The **df** command displays the number of free 512-byte blocks and free inodes available for file systems by examining the counts kept in the superblock or superblocks. If a *special* or a *directory* is not specified, the free space on all mounted file systems is displayed. If the arguments to **df** are path names, **df** reports on the file systems containing the named files. If the argument to **df** is a *special* of an unmounted file system, the free space in the unmounted file system is displayed.

**Options**

**df** recognizes the following options:

- b Report only the number of kilobytes (KB) free.
- B Report the total number of blocks allocated for swapping to the file system as well as the number of blocks free for swapping to the file system. This option is supported on HFS file systems only.
- e Report the number of files free.
- f Report only the actual count of the blocks in the free list (free inodes are not reported). When this option is specified, **df** reports on raw devices.
- F *FStype* Report only on the *FStype* file system type (see *fstyp(1M)*). For the purposes of this manual entry, *FStype* can be one of **cdfs**, **hfs**, and **nfs**, for the CDFS, HFS, and NFS file systems, respectively.
- g Report the entire structure described in *statvfs(2)*.
- i Report the total number of inodes, the number of free inodes, number of used inodes, and the percentage of inodes in use.
- k Report the allocation in kilobytes (KB).
- l Report on local file systems only.
- n Report the file system name. If used with no other options, display a list of mounted file system types.
- o *specific\_options* Specify options specific to the HFS file system type. *specific\_options* is a comma-separated list of suboptions.  
The available suboption is:
  - i Report the number of used and free inodes.
- t Report the total allocated block figures and the number of free blocks.
- v Report the percentage of blocks used, the number of blocks used, and the number of blocks free. This option cannot be used with other options.
- V Echo the completed command line, but perform no other action. The command line is generated by incorporating the user-specified options and other information derived from */etc/fstab*. This option allows the user to verify the command line.

When **df** is used on an HFS file system, the file space reported is the space available to the ordinary user, and does not include the reserved file space specified by **fs\_minfree**.

Unreported reserved blocks are available only to users who have appropriate privileges. See *fs(4)* for information about **fs\_minfree**.

When **df** is used on NFS file systems, the number of inodes is displayed as -1. This is due to superuser access restrictions over NFS.

**EXAMPLES**

Report the number of free disk blocks for all mounted file systems:

**df**

Report the number of free disk blocks for all mounted HFS file systems:

```
df -F hfs
```

Report the number of free files for all mounted NFS file systems:

```
df -F nfs -e
```

Report the total allocated block figures and the number of free blocks, for all mounted file systems:

```
df -t
```

Report the total allocated block figures and the number of free blocks, for the file system mounted as /usr:

```
df -t /usr
```

**WARNINGS**

**df** does not account for:

- Disk space reserved for swap space,
- Space used for the HFS boot block (8K bytes, 1 per file system),
- HFS superblocks (8K bytes each, 1 per disk cylinder),
- HFS cylinder group blocks (1K-8K bytes each, 1 per cylinder group),
- Inodes (currently 128 bytes reserved for each inode).

Non-HFS file systems may have other items that this command does not account for.

The **-b** option, from prior releases, has been replaced by the **-B** option.

**FILES**

/dev/dsk/*	File system devices.
/etc/fstab	Static information about the file systems
/etc/mnttab	Mounted file system table

**SEE ALSO**

du(1), df(1M), fsck(1M), fstab(4), fstyp(1M), statvfs(2), fs(4), mnttab(4).

**STANDARDS CONFORMANCE**

**df**: SVID2, XPG2, XPG3

**NAME**

df - report number of free disk blocks on a VxFS file system

**SYNOPSIS**

```
/usr/bin/df [-F vxfs] [-V] [-egiklnvtfb] [-o s] [special | directory ...]
```

**DESCRIPTION**

**df** prints the number of free 512-byte blocks and free inodes available for file systems by examining the counts kept in the superblock or superblocks. If a *special* or a *directory* is not specified, the free space on all of the mounted file systems is printed. If the arguments to **df** are pathnames, **df** produces a report on the file system containing the named file. If the argument to **df** is a *special*, the file system can be an unmounted or mounted file system.

On a Version 2 disk layout extents smaller than 8 kilobytes may not be usable for all types of allocation, so **df** does not count free blocks in extents below 8 kilobytes when reporting the total number of free blocks.

On a Version 2 or greater disk layout, VxFS dynamically allocates inodes from the pool of free blocks, so the number of free inodes and blocks reported by **df** is an estimate based on the number of free extents and the current ratio of allocated inodes to allocated blocks. Allocating additional blocks may therefore decrease the count of free inodes, and vice versa.

**Options**

**df** recognizes the following options:

- b** Report only the number of kilobytes free.
- e** Report the number of files free.
- f** Report only an actual count of the blocks in the free list (free inodes are not reported). When this option is specified, **df** reports on raw devices.
- F vxfs** Specifies the file system type (**vxfs**).
- g** Report the entire statvfs(2) structure.
- i** Report the total number of inodes, the number of free inodes, number of used inodes and the percentage of inodes in use.
- k** Report the allocation in kilobytes.
- l** Report on local file systems only.
- n** Report the file system name. If invoked with no other options this option prints a list of mounted file system types.
- o s** Print the number of free extents of each size. Free extents are always an integral power of 2 in length, ranging from a minimum of 1 block to the maximum extent size supported by the file system. (VxFS-specific option).
- t** Report the total allocated block figures and the number of free blocks.
- v** Report the percentage of blocks used, the number of blocks used and the number of blocks free. This option cannot be used with other options.
- V** Echo the completed command line, but performs no other action. The command line is generated by incorporating the user-specified options and other information derived from **/etc/fstab**. This option allows the user to verify the command line.

There are a number of options that specify output formats, some combinations of which are incompatible. If an incompatible combination is specified, one of the options will override the other(s).

**EXAMPLES**

Report the number of free disk blocks for all mounted file systems:

```
df
```

Report the number of free extents of each size, for all mounted VxFS file systems:

```
df -F vxfs -o s
```

Report the number of free files for all mounted VxFS file systems:

```
df -F vxfs -e
```

Report the total allocated block figures and the number of free blocks, for all mounted file systems:

```
df -t
```

Report the total allocated block figures and the number of free blocks, for the file system mounted as **/usr**:

```
df -t /usr
```

**FILES**

<b>/dev/vg00/ *</b>	File-system devices.
<b>/dev/dsk/ *</b>	File-system devices.
<b>/etc/fstab</b>	Static information about the file systems.
<b>/etc/mnttab</b>	mounted-file-system table.

**SEE ALSO**

du(1), df(1M), fsck\_vxfs(1M), fs\_vxfs(4), mnttab(4), statvfs(2).

**STANDARDS CONFORMANCE**

**df** : SVID2, XPG2, XPG3



## NAME

dhcpcclient - Client for Dynamic Host Configuration Protocol Server

## SYNOPSIS

```
dhcpcclient [-G] [-N hostname] -b interface [-f tracefile] [-g class-id] [-l debug_level] [-n]
            [-p] [-s server_ipaddress] [-t log_type] [-x send_attempts] [-z time_to_wait]

dhcpcclient -d interface [-s server_ipaddress]

dhcpcclient [-N hostname] [-f tracefile] [-l debug_level] -m interface [-p] [-s server_ipaddress]
            [-t log_type] [-z time_to_wait]

dhcpcclient [-f tracefile] [-l debug_level] [-p] -r interface [-s server_ipaddress] [-t log_type]

dhcpcclient -v
```

## DESCRIPTION

**dhcpcclient** is used to obtain the configuration parameters from the Dynamic Host Configuration Protocol (DHCP) server (*bootpd*(1M)) to configure the host. Currently, **dhcpcclient** supports 10bt (10BaseT) and 100bt (100BaseT) **Ethernet** networks only. It does not support **FDDI** and **Token-Ring** networks.

**dhcpcclient** is invoked by the **auto\_parms** script (see *auto\_parms*(1M)) at boot up. The */etc/rc.config.d/netconf* file is used for the configuration of the interfaces. To invoke **dhcpcclient** using the **auto\_parms** script, set the **DHCP\_ENABLE**[*index*] variable to a non-zero value in the */etc/rc.config.d/netconf* file. **dhcpcclient** can also be invoked at the command prompt, when the interface is not configured.

Once the **dhcpcclient** obtains the configuration parameters from the server, it is stored in a configuration file named */etc/dhcpcclient.data*. The configuration parameters are valid for a fixed time, lease time.

The configuration information in */etc/dhcpcclient.data* file will be of the following format.

```
<code> <length> <data>
00    <length> IFname
01    <length> hostname
02    <length> sname
03    <length> bootfile
04    <length> NIS_domain
05    <length> domain_name
06    <length> lease_duration
07    <length> lease_expiration
08    <length> T1
09    <length> T2
10    <length> htype
11    <length> chaddr
12    <length> IP_addr
13    <length> subnet_mask
14    <length> broadcast_addr
15    <length> default_gateway
16    <length> server_addr
17    <length> boot_server
18    <length> dest_gateway
19    <length> DNS_server
20    <length> LPR_server
21    <length> swap_server
22    <length> NIS_server
23    <length> NTP_server
24    <length> raw_options
```

The length field denotes the length of the data field following it.

All these lines collectively indicate one record. The configuration file will have series of records.

Since this file contains all the configuration information, it can be used to identify the server from which the configuration parameters were obtained. The configuration file can also be used to obtain the same configuration parameters from the server. So, once the **dhcpcclient** is invoked, it will try to read the

configuration file, `/etc/dhcpcclient.data`.

While trying to read the `/etc/dhcpcclient.data` file, the following conditions are possible:

1. It may be possible that the configuration file had been deleted or it may not be available for reading.
2. The file may have got corrupted and contains irrelevant data for the particular interface.
3. There may be no configuration information for the particular interface.
4. User may have specified `-n` option meaning that a new lease is needed.

For all of the above cases, `dhcpcclient` has to first find the server which is ready for the service. For this purpose, it will send a `DHCPDISCOVER` message. If the user has specified server ip-address using the `-s` option, then this message will be sent to that ip-address, else, this message is broadcasted. The server in turn will reply with the configuration parameters that it can offer. This message is called `DHCPOFFER`. If the client has sent the `DHCPDISCOVER` to broadcast address, then it will receive `DHCPOFFER` from more than one server. The `dhcpcclient` selects one of the `DHCPOFFER` obtained. Then, it will send `DHCPREQUEST` to the server, requesting the configuration parameters. The server will reply with a positive acknowledgement called `DHCPACK`, if it is willing to send the configuration parameters to the host and with a negative acknowledgement, if it is not willing to do so. If the `dhcpcclient` has received `DHCPACK`, it will copy all the configuration parameters to the configuration file, else, it will send another `DHCPDISCOVER` message and repeat the whole process. This process is called "Obtaining a new lease".

If all the above 4 cases are not true, then the `dhcpcclient` can directly send the `DHCPREQUEST` message and obtain the configuration parameters as mentioned above. This process is called "Renewing the lease".

The various options supported by `dhcpcclient` are as described below.

Options

- G When this option is used, the client will accept the reply from the DHCP server only if the `class-id` in the request and reply packet matches.
- N *hostname* This option is used to send a hostname. The *hostname* should be specified as an ASCII string.
- b *interface* This option is used during bootup time to obtain a new lease. If the client is trying to obtain the configuration parameters from the server for the first time, then `/etc/dhcpcclient.data` does not exist. So, using this option, the client obtains a new lease for the interface specified.
- d *interface* This option is used for dumping the internal data of an interface. A sample dump is shown below.

Interface	PPA	MAC	Type	Physical Address
lan0	0	ETHER		0x080009709631
- f *tracefile* This option is used to specify the file name for the debugging output.
- g *class-id* This option is used for specifying the `class-id` of the device group. `class-id` should be specified as an ASCII string.
- l *debug\_level* This option is used for logging of debug information. *debug\_level* specifies the amount of debug information logged. The debugging messages are sent to `STDERR`.
- m *interface* This option is used for maintaining the lease. When invoked with this option, the `dhcpcclient` runs as a daemon. The `dhcpcclient` calculates how much time it needs to sleep before the lease needs to be renewed and then sleeps for that time. When it wakes up, it extends the lease by sending the request to the DHCP server.
- n This option is used for ignoring the current lease. When this option is used, the current configuration information is discarded and request for a new lease is sent.
- p This option is used for tracing the packets. It displays the packet contents in a readable format.
- r *interface* This option is used for releasing a lease. The `dhcpcclient` releases all the assigned configuration parameters. This is done by sending the `DHCPRELEASE` message to the server. The `dhcpcclient` then empties the record corresponding to that interface in the configuration file, `/etc/dhcpcclient.data`.

- s** *server\_ipaddress*  
This option is used to specify the ip address of the **DHCP** server to which the request should be sent. *server\_ipaddress* should be specified in dotted decimal form.
- t** *log\_type*  
If the *log\_type* is a non zero value and if the *debug\_level* is low enough for the process to detach from the tty (see *syslog(3C)*), then, the logging goes to **syslog**, else it goes to **STDERR**. This option is used when the **dhcpcclient** is running as a daemon.
- v**  
This option prints the various version strings of the **dhcpcclient** to stdout.
- x** *send\_attempts*  
This option is used to specify the number of attempts the packet should be sent before obtaining a reply. *send\_attempts* must be specified as an unsigned number.
- z** *time\_to\_wait*  
This option is used to specify the time interval between successive packet retransmission. *time\_to\_wait* will be in seconds.

## EXAMPLES

To **maintain** the lease for the interface `lan0` and invoke logging in **syslog**, the following set of commands is invoked at the command prompt:

```
dhcpcclient -b lan0 -N "dhcp1"
dhcpcclient -m lan0 -N "dhcp1" -l 3 -t 1
```

To trace the packet and get the debugging messages, use:

```
dhcpcclient -b lan0 -N "dhcp1" -p -l 8
```

This is an example of a `/etc/dhcpcclient.data` file.

```
00 4 lan0
01 5 dhcp
02 0
03 0
04 0
05 0
06 4 4294967295
07 4 4294967295
08 4 0
09 4 0
10 4 1
11 6 8 0 9 25 a5 b1
12 4 192.11.22.107
13 4 255.255.255.0
14 4 0.0.0.0
15 0
16 4 192.11.22.3
17 4 192.11.22.3
18 0
19 0
20 0
21 4 0.0.0.0
22 0
23 0
24 39 63 82 53 63 35 1 5 33 4 ff ff ff ff 36 4 c0 b8
16 3 1 4 ff ff ff 0 c 5 64 68 63 70 68 2 4 ff ff b2 a8 ff
```

## AUTHOR

**dhcpcclient** was developed by HP.

## FILES

`/etc/dhcpcclient.data` configuration file.

**SEE ALSO**

auto\_parms(1M), bootpd(1M), bootpquery(1M), dhcptools(1M), syslog(3C).

DARPA Internet Request For Comments: RFC1541, RFC1542, RFC1533, RFC1534, Assigned Numbers RFC.

  
d

**NAME**

dhcplib2conf - DHCP client database converter

**SYNOPSIS**

**dhcplib2conf** [*dhcplib2conf\_options*] [*lan\_interfaces*]

**DESCRIPTION**

**dhcplib2conf** provides a means of translating a client DHCP database into a set of standard configuration file variables. A DHCP client database can contain settings for such items as, IP address, hostname, and default gateway. Using **dhcplib2conf**, you can simply list the contents of the database to the screen, create a set of configuration staging files, or execute direct edits on existing configuration files using the values contained in the client database.

**Options**

**dhcplib2conf** allows you to specify a list of interfaces on the command line (e.g. lan0 lan1 ...). If no lan interface is specified, **dhcplib2conf** will process all entries referenced in the client database. The entries themselves are defined as a unique lan interface and a list of attributes which correspond to that interface. The attributes can be selected for processing by specifying one or more filter flags on the command line. Each filter flag may be combined with any other filter flag(s). If no filter flag is specified, all the attributes for a lan interface will be processed. The following options are supported:

- a** Using the results of the specified filter, directly apply the variable definitions to the existing configuration files (for example, `/etc/rc.config.d/netconf`).
- c** Create a set of staging files using the results of the selected filter(s). Each variable processed will be applied to its corresponding configuration file. Specifically, **dhcplib2conf** will generate a copy of the existing configuration file. As an example, `/etc/rc.config.d/netconf` will be copied to `/etc/rc.config.d/netconf.dhcp`. Once this staging file has been created, the variable that is being processed will be applied to the newly created file.  
*WARNING:* Using the **-c** option will override any existing values which are currently set in the system's configuration files.
- d** Process the DNS variable set: [domain, nameserver]
- h** Process HOSTNAME
- i** Process the INTERFACE variable set: [IP\_ADDRESS, SUBNET\_MASK, BROADCAST\_MASK, LANCONFIG\_ARGS]
- n** Process the NIS variable set: [NISDOMAIN, YPSET\_ADDR]
- p** Print results to the screen (stdout), this is the default action if neither **-c** or **-a** are specified
- r** Process the ROUTE variable set: [ROUTE\_DESTINATION, ROUTE\_GATEWAY, ROUTE\_COUNT]
- s set index** Specify the variable set index
- t** Process NTPDATE\_SERVER

**Configuration Files and Variable Names**

The files and variables which can be processed are the following:

```
/etc/rc.config.d/netconf
    HOSTNAME
    INTERFACE_NAME[index]
    IP_ADDRESS[index]
    SUBNET_MASK[index]
    BROADCAST_MASK[index]
    LANCONFIG_ARGS[index]
    ROUTE_DESTINATION[index]
    ROUTE_GATEWAY[index]
    ROUTE_COUNT[index]
```

```

/etc/rc.config.d/namesvrs
    NISDOMAIN
    YPSET_ADDR
/etc/rc.config.d/netdaemons
    NTPDATE_SERVER
/etc/resolv.conf
    domain
    nameserver

```

**d****EXAMPLES**

To list the entire contents of the DHCP client database type:

```
dhcplib2conf
```

To list only the INTERFACE variable set for lan0 type:

```
dhcplib2conf -i lan0
```

To list the INTERFACE and ROUTE variable sets for lan0 and lan1 type:

```
dhcplib2conf -ir lan0 lan1
```

To apply the INTERFACE and ROUTE variable sets for lan0 to the existing configuration files type:

```
dhcplib2conf -ira lan0
```

To apply all variable sets to the existing configuration files using lan0 and set index = 1 type:

```
dhcplib2conf -a -s 1 lan0
```

**WARNINGS**

Using the **-c** option will override any existing values which are currently set in the system's configuration files.

**FILES**

```

/usr/sbin/dhcplib2conf
/etc/dhclient.data

```

**SEE ALSO**

auto\_parms(1M).

**NAME**

dhcptools - command line tool for DHCP elements of bootpd

**SYNOPSIS**

```
dhcptools -d
dhcptools -h fip=first_IP_address no=number_of_entries_to_generate sm=subnet_mask
             hn=hostname_template [dn=domain_name]
dhcptools -p ht=hardware_type ha=hardware_address sn=subnet_identifier [lt=lease_time]
             [rip=requested_IP_address]
dhcptools -P ci=client_identifier sn=subnet_identifier [lt=lease_time]
             [rip=requested_IP_address]
dhcptools -C cl=class_identifier sn=subnet_identifier [lt=lease_time]
             [rip=requested_IP_address]
dhcptools -r ip=IP_address ht=hardware_type ha=hardware_address
dhcptools -R ip=IP_address ci=client_identifier
dhcptools -t [ct=count]
dhcptools -v [bt=bootptabfile] [dt=dhcptabfile]
```

**DESCRIPTION**

**dhcptools** is a command line tool that provides access to DHCP-related options for the bootpd server. The options provide control for dumping internal data structures, generating a hosts file, previewing client address assignment, reclaiming unused addresses, tracing packets, and validating configuration files.

**Options**

**dhcptools** supports the following options:

- d Dump internal bootpd data to output files. The dump output files are /tmp/dhcp.dump.bootptab, /tmp/dhcp.dump.dhcptab, and /tmp/dhcp.dump.other. The first file reports fixed address clients known to the currently active bootpd server. The second file reports bootpd global and group configuration. The third file reports miscellaneous bootpd internal data.
- h Generate a hosts file in /etc/hosts format; see hosts(4). The output file is /tmp/dhcphosts. The file can be incorporated into a name database in advance of bootpd server activation so that the server can automatically allocate a host name along with an IP address to a DHCP client. For IP address allocation to DHCP clients, the bootpd server uses gethostbyaddr(3N) to find the host name associated with a particular IP address. Each host entry in dhcphosts contains an IP address followed by a host-name. The IP address of the first entry is first\_IP\_address. The hostname of the first entry is derived from the hostname\_template. Each subsequent host entry contains a unique IP address and hostname derived from the first\_IP\_address, subnet\_mask, and hostname\_template. The wildcards permitted in the hostname\_template are \*#?. A \* means to use a character selected sequentially from the range [a-z,0-9]. A # means to use a digit selected sequentially from the range [0-9]. A ? means to use a letter selected sequentially from the range [a-z]. A maximum of 3 wildcards can be specified. If a domain\_name is specified, it will be appended to the hostname. The maximum number\_of\_entries\_to\_generate is 1000.
- p Preview a client's address assignment based on current conditions for the bootpd server. The output is written to stdout. The subnet-identifier tells bootpd the subnet for which the client is requesting an IP address. Optionally, the user may request a specific IP address and lease duration using the parameters lease-time and requested-IP-address. Use Internet address dot notation (see inet(3N) for the IP address and an integer number of seconds for the lease-time.
- P Preview a client's address assignment based on current conditions for the bootpd server. This option is the same as -p except that the client is identified by a unique client-identifier. See bootpd(1M).
- C Preview a client's address assignment based on current conditions of the bootpd server. This option is the same as -p except that the class identifier is used to identify the device

group from which the client is requesting an IP address. See *bootpd*(1M).

- r Reclaim a client's IP address for re-use by the **bootpd** server. This option is intended for limited use by the **bootpd** administrator to return an allocated but unused IP address to a DHCP allocation pool. The option may be useful to clear the bootpd database of old entries (e.g. for clients retired from service while holding an unexpired IP address lease). Do not reclaim an address that belongs to an active client. See *bootpd*(1M). The IP\_address, hardware\_address, and hardware\_type can be obtained from the bootpd database file.
- R Reclaim a client's IP address for re-use by the **bootpd** server. This option is the same as -r except that the client is identified by its unique client\_identifier. See *bootpd*(1M). The IP\_address and matching client\_identifier can be obtained from the bootpd database file.
- t Establish packet tracing for **bootpd**. This will trace the inbound and outbound BOOTP/DHCP packets for the local **bootpd** server. The output file is **/tmp/dhcptrace**. The packet trace count can be a value from 0 to 100. To query the current count, use **dhcptools -t**. To turn off packet tracing use **dhcptools -t ct=0**.
- v Validate **bootpd** configuration files. The default configuration files that will be validated are **/etc/bootptab** and **/etc/dhcptab**. When a bootptabfile or dhcptabfile is specified, the full pathname is required. The output file for validate is **/tmp/dhcpvalidate**.

Only one of the -d, -h, -t, -p, -P, -r, -R, or -v options is allowed per **dhcptools** command.

## RETURN VALUE

**dhcptools** returns zero upon successful completion or non-zero if the command failed, in which case an explanation is written to standard error.

## EXAMPLES

Dump the active **bootpd** server's internal data to the dump output files:

```
dhcptools -d
```

Generate a **/tmp/dhcphosts** file with 10 entries:

```
dhcptools -h fip=192.11.22.0 no=10 sm=255.255.255.0 hn=workstation#?
```

Query the active **bootpd** daemon for the the current packet trace count:

```
dhcptools -t
```

Set the count to 10 packets:

```
dhcptools -t ct=10
```

Preview two clients' address assignments by hardware address:

```
dhcptools -p ht=1 ha=080009000001 sn=192.11.22.0 lt=infinite
dhcptools -p ht=1 ha=080009000002 sn=192.11.22.0 lt=600 rip=192.11.22.105
```

To preview a client's address assignment by client identifier, a unique client identifier value is needed. This information can be obtained for actual DHCP clients (provided they support a client identifier) from the manufacturer's documentation. See *bootpd*(1M) for more information about the client identifier. Assuming that **serial\_number\_12345678** is a valid client identifier, the preview command is:

```
dhcptools -P ci="serial_number_12345678" sn=192.11.22.0
```

To reclaim an IP address by hardware address:

```
dhcptools -r ip=192.11.22.149 ht=1 ha=080009000006
```

The parameter values were obtained from this sample entry in the dhcpdb file:

```
C 192.11.22.0: 192.11.22.149 00 1 080009000006 FFFFFFFF 00
```

To reclaim an IP address by client identifier (see earlier example of preview by client identifier):

```
dhcptools -R ip=192.11.22.110 ci="serial_number_12345678"
```

To validate a bootptab and dhcptab file:



```
dhcptools -v bt=/home/mydir/bootptab dt=/home/mydir/dhcptab
```

**WARNINGS**

The **dhcptools** operations of dump, packet trace, preview, and reclaim depend on communication with the local **bootpd** server. If the server is not running, you may encounter an error.

**AUTHOR**

**dhcptools** was developed by HP.

**FILES**

/tmp/dhcp hosts	hostgen output file in /etc/hosts format
/tmp/dhcp trace	packet trace output file
/tmp/dhcp validate	validate output file
/tmp/libdhcp.sl	library file
/tmp/dhcp.dump.bootptab	dump output file
/tmp/dhcp.dump.dhcptab	dump output file
/tmp/dhcp.dump.other	dump output file
/etc/bootptab	default bootptab file for validate
/etc/dhcptab	default dhcptab file for validate
/tmp/dhcpfifo.root	FIFO file for dhcptools to bootpd(1M) communication
/tmp/dhcpfifo.any	FIFO file for dhcptools to bootpd(1M) communication
/tmp/dhcpfifo	FIFO file for bootpd(1M) to dhcptools communication

**SEE ALSO**

bootpd(1M), bootpquery(1M); DARPA Internet Request For Comments RFC1541, RFC1542, RFC1533, RFC1534, Assigned Numbers

**NAME**

diskinfo - describe characteristics of a disk device

**SYNOPSIS**

`/usr/sbin/diskinfo [-b|-v] character_devicefile`

**DESCRIPTION**

The **diskinfo** command determines whether the character special file named by *character\_devicefile* is associated with a SCSI or floppy disk drive. If so, **diskinfo** summarizes the disk's characteristics.

The **diskinfo** command displays information about the following characteristics of disk drives:

Vendor name	Manufacturer of the drive (SCSI only)
Product ID	Product identification number or ASCII name
Type	floppy or SCSI classification for the device
Disk	Size of disk specified in bytes
Sector	Specified as bytes per sector

Both the size of disk and bytes per sector represent formatted media.

**Options**

The **diskinfo** command recognizes the following options:

- b** Return the size of the disk in 1024-byte sectors.
- v** Display a verbose summary of all of the information available from the device. For floppy drives, this option has no effect.

SCSI disk devices return the following:

- Vendor and product ID
- Device type
- Size (in bytes and in logical blocks)
- Bytes per sector
- Revision level
- SCSI conformance level data

**WARNINGS**

As of release 10.20 of HP-UX, certain IDE devices, CD-ROMs in particular, will respond to **diskinfo** inquiries as if they were SCSI devices. Therefore, the text "SCSI describe" in the output of the **diskinfo** command does not definitively mean that the disk is in fact a SCSI drive (especially in the case of CD-ROMs). Use **ioscan(1M)**, **ioscan -fn**, and check which type of INTERFACE node, SCSI or IDE, the device's hardware path lies beneath, in order to definitively determine a drive's interface.

**DEPENDENCIES****General**

The **diskinfo** command supports floppy and HP SCSI disk devices.

**SCSI Devices**

The SCSI specification provides for a wide variety of device-dependent formats. For non-HP devices, **diskinfo** may be unable to interpret all of the data returned by the device. Refer to the drive operating manual accompanying the unit for more information.

**AUTHOR**

**diskinfo** was developed by HP.

**SEE ALSO**

**lsdev(1M)**, **ioscan(1M)**, **disktab(4)**, **disk(7)**.

NAME

disksecn - calculate default disk section sizes

SYNOPSIS

disksecn [-p] [-d] [-b block\_size] [-n disk\_name]

DESCRIPTION

disksecn is used to calculate the disk section sizes based on the Berkeley disk partitioning method.

disksecn recognizes the following options:

- p Produce tables suitable for inclusion in the device driver.
- d Produce tables suitable for generating the disk description file /etc/disktab.
- b block\_size When generating the above tables, use a sector size of block\_size bytes, where block\_size can be 256, 512, 1024, or 2048. Defaults to DEV\_BSIZE (defined in <sys/param.h>) if not specified.
- n disk\_name Specifies the disk name to be used in calculating sector sizes; for example, hp7912 or hp7945. If an unknown disk name is specified, disksecn prompts the user for the necessary disk information.

If neither -p nor -d table selection switches are specified a default table of the section sizes and range of cylinders used is output.

Disk section sizes are based on the total amount of space on the disk as given in the table below (all values are supplied in units of 256-byte sectors). If the disk is smaller than approximately 44 Mbytes, disksecn aborts and returns the message **disk too small, calculate by hand**.

Section	44-56MB	57-106MB	107-332MB	333+MB
0	97120	97120	97120	97120
1	39064	39064	143808	194240
3	39064	39064	78128	117192
4	unused	48560	110096	429704
6	7992	7992	7992	7992
10	unused	unused	unused	516096

NOTE:

It is important to note the difference between the block size passed into disksecn via the -b switch argument and the sector size the user is asked to input when an unknown disk name is passed to disksecn via the -n switch argument.

The block size is the sector size that disksecn assumes the disk to have when it prints the requested tables. All information printed in the tables is adjusted to reflect this assumed sector size (block size) passed in by the user. The sector size requested by disksecn when an unknown disk name is passed does not necessarily have to be the same as the assumed sector size (block size) passed in by the -b switch argument.

For example, a user wants to see the device driver tables for the disk named hp7945 with an assumed sector size (block size) of 256 bytes. The user has the following information about the hp7945 disk:

Disk type = winchester  
Sector size = 512  
Number of sectors per track (512 byte sectors) = 16  
Number of tracks = 7  
Number of cylinders = 968  
Revolutions per minute = 3600

The user invokes disksecn by typing the following command:

disksecn -p -b 256 -n hp7945

Assuming that hp7945 is an unknown disk name, disksecn prompts the user for the necessary disk information. The user should input the information as shown above, reflecting a sector size of 512 bytes. All the information will be adjusted within disksecn to reflect the assumed sector size (block size) of 256 bytes, passed as the argument of the -b switch, before the requested device driver table is output.

This adjustment also takes place when the disk name is known and an assumed sector size (block size) is passed in as the argument of the -b switch which is not DEV\_BSIZE bytes, the assumed sector size (block

size) used to create the **etc/disktab** file.

## RETURN VALUE

*disksecn* returns the following values:

- 0** Successful completion.
- 1** Usage error.
- 2** User did not input parameters for an unknown disk.
- 3** Disk too small or an invalid block size.

*disksecn* aborts and prints an error message under the following conditions:

- *disksecn* was invoked without specifying a disk name.
- Requested both **-p** and **-d** switch.
- Illegal block size requested.
- Unknown disk name was specified and user did not supply disk information.
- Disk's maximum storage space is less than approximately 44 MB.

## WARNINGS

Alternate names are not included in the output when the **-d** switch is used.

Blanks are required in the command line between each of the switches when invoking *disksecn*.

A blank is required between the **-n** switch and the disk name argument to that switch. For example:

**disksecn -p -b 1024 -n hp9712**

*disksecn* does not save the block size used to generate the **/etc/disktab** disk description file. The system assumes that the block size used was DEV\_BSIZE when it reads the information stored in the **etc/disktab** file.

## AUTHOR

*disksecn* was developed by the University of California, Berkeley.

## FILES

**/etc/disktab**

## SEE ALSO

disktab(4).

**NAME**

diskusg - generate disk accounting data by user ID

**SYNOPSIS**

`/usr/sbin/acct/diskusg` [*options*] [*files*]

**DESCRIPTION**

**diskusg** generates intermediate disk accounting information from data in *files*, or the standard input if omitted. **diskusg** outputs lines on the standard output, one per user, in the following format:

*uid login #blocks*

where:

*uid*            User's numerical user ID,

*login*        User's login name, and

*#blocks*      Total number of disk blocks allocated to this user.

**diskusg** normally reads only the inodes of file systems for disk accounting. In this case, *files* are the special filenames of these devices.

**Options**

**diskusg** recognizes the following options:

- s**            Input data is already in **diskusg** output format. **diskusg** combines all lines for a single user into a single line.
- v**            verbose. Print a list on standard error of all files that are charged to no one.
- i *fnmlist***   Ignore the data on those file systems whose file system name is in *fnmlist*. *fnmlist* is a list of file system names, separated by commas or enclosed within quotes. **diskusg** compares each name in this list with the file system name stored in the volume ID if it exists.
- p *file***      Use *file* as the name of the password file to generate login names. `/etc/passwd` is used by default.
- u *file***      Write records to *file* of files that are charged to no one. Records consist of the special file name, the inode number, and the user ID.

The output of **diskusg** is normally the input to **acctdisk** (see *acct(1M)*) which generates total accounting records that can be merged with other accounting records. **diskusg** is normally run in **dodisk** (see *acctsh(1M)*).

**EXAMPLES**

The following generates daily disk accounting information:

```
for i in /dev/rp00 /dev/rp01 /dev/rp10 /dev/rp11; do
    diskusg $i > dtmp.`basename $i` &
done
wait
diskusg -s dtmp.* | sort +0n +1 | acctdisk > diskacct
```

**FILES**

`/etc/passwd`      used for user-ID-to-login-name conversions

**SEE ALSO**

*acct(1M)*, *acctsh(1M)*, *volcopy(1M)*, *acct(4)*, *vxdiskusg(1M)*.

**STANDARDS CONFORMANCE**

**diskusg**: SVID2, SVID3

**NAME**

dlf - download firmware to an HP SCSI disk array

**SYNOPSIS**

**dlf -f** *firmware\_file device\_file*

**DESCRIPTION**

**dlf** downloads a new set of controller firmware to the HP SCSI disk array associated with device file *device\_file*. The *firmware\_file* must be a binary file with a special format.

**RETURN VALUE**

**dlf** returns the following values:

- 0 Successful completion.
- 1 Command failed (an error occurred).

**ERROR MESSAGES**

Errors can originate from problems with:

- **dlf**
- SCSI (device level) communications
- system calls

**Error messages generated by dlf:**

**usage: dlf -f <firmware file> <special>**

An error in command syntax has occurred. Enter command again with all required arguments, in the order shown.

**dlf: Binary file has bad format**

The binary file could not be read in properly by the utility.

**dlf: device busy**

To ensure that **dlf** does not modify a disk array that is being used by another process, **dlf** attempts to obtain exclusive access to the disk array. If the disk array is already opened by another process (for example, LVM — the Logical Volume Manager), a “**device busy**” error message is returned by the driver. To eliminate the “**device busy**” condition, determine what process has the device open. In the case of LVM, it is necessary to deactivate the volume group containing the array before configuring the array (see *vgchange(1M)*).

**dlf: LUN # too big**

The LUN number, which is derived from the device file name, is out of range.

**dlf: Not a raw file**

Utilities must be able to open the device file for raw access.

**dlf: Not an HP SCSI disk array**

The device being addressed is not an

**dlf: Transfer length error**

The amount of data actually sent to or received from the device was not the expected amount. HP SCSI disk array.

**SCSI (device level) communication errors:**

Sense data associated with the failed operation is printed.

**Error messages generated by system calls:**

**dlf** uses the following system calls:

**malloc()**, **free()**, **stat()**, **open()**, **close()**, **fopen()**, **fclose()**, **read()**, **write()**, and **ioctl()**.

Documentation for these HP-UX system calls contains information about the specific error conditions associated with each call. **dlf** does not alter the value of **errno**. The interpretation of **errno** for printing purposes is performed by the system utility **strerror()**.

**EXAMPLES**

To download the special-format binary file `new_firmware` to the HP SCSI disk array `/dev/rdisk/c2t0d0` on a series 800:

```
dlf -f new_firmware /dev/rdisk/c2t0d0
```

**DEPENDENCIES**

The HP C2425 and HP C2427 disk arrays are only supported on Series 700 systems running HP-UX version 9.0X.

The HP C2430 disk array is supported on Series 700 and 800 systems running HP-UX versions 9.0X and 10.0X.

**AUTHOR**

`dlf` was developed by HP.

  
d

**NAME**

**dmesg** - collect system diagnostic messages to form error log

**SYNOPSIS**

```
/usr/sbin/dmesg [-] [core] [system]
```

**DESCRIPTION**

**dmesg** looks in a system buffer for recently printed diagnostic messages and prints them on the standard output. The messages are those printed by the system when unusual events occur (such as when system tables overflow or the system crashes). If the **-** argument is specified, **dmesg** computes (incrementally) the new messages since the last time it was run and places these on the standard output. This is typically used with **cron** (see *cron(1)*) to produce the error log **/var/adm/messages** by running the command:

```
/usr/sbin/dmesg - >> /var/adm/messages
```

every 10 minutes.

The arguments **core** and **system** allow substitution for the defaults **/dev/kmem** and **/stand/vmunix** respectively, where **core** should be a file containing the image of the kernel virtual memory saved by the *savecore(1M)* command and **system** should be the corresponding kernel. If the system is booted with a kernel other than **/stand/vmunix** say **/stand/vmunix\_new**, **dmesg** must be passed this name, the command must be,

```
/usr/sbin/dmesg [-] /dev/kmem /stand/vmunix_new
```

**WARNINGS**

The system error message buffer is of small, finite size. **dmesg** is run only every few minutes, so there is no guarantee that all error messages will be logged.

**AUTHOR**

**dmesg** was developed by the University of California, Berkeley.

**FILES**

<b>/var/adm/messages</b>	error log (conventional location)
<b>/var/adm/msgbuf</b>	memory scratch file for <b>-</b> option
<b>/dev/kmem</b>	special file containing the image of kernel virtual memory
<b>/stand/vmunix</b>	the kernel, system name list

**SEE ALSO**

*savecore(1M)*.



**NAME**

download - download new controller or disk firmware

**SYNOPSIS**

```
download -D slot [-?] codefilename array-id
download -C [-?] codefilename array-id
download -M [-?] array-id
download -i [-?]
```

**DESCRIPTION**

**download** copies new firmware code to the controller or disks in the disk array identified by *array-id*. The new code is stored in the file identified by *codefilename*. Firmware can also be copied from a primary controller to a secondary controller.

The *array-id* used to address the disk array can be the disk array serial number, the character device file name of any LUN on the array (LUN 0 if no LUNs are created), or the alias text string assigned to the disk array.

When downloading new firmware code, the **arraymond** disk monitor and **ARMserver** can be turned off to avoid error messages generated while the disk array is not accessible. To turn off **arraymond** and **ARMserver** use the following command line:

```
/sbin/init.d/hparray stop
```

To restart **arraymond** and **ARMserver** use the following command line:

```
/sbin/init.d/hparray start
```

**Options**

**download** supports the following options:

- C** Download new controller firmware. If there are two controllers installed in the disk array, new firmware will be downloaded to both.
- D slot** Download firmware to the disk installed in the cabinet slot identified by *slot*. Slot numbers must be of the form *An* or *Bn*, where *A* or *B* correspond to a cabinet column, and *n* corresponds to a shelf position (1-6).
- i** Display the serial number and firmware revisions for all disk arrays currently connected to and recognized by the server. Because this option is device-independent, it does not use *array-id*.
- M** Copy the controller firmware from the primary controller to the secondary controller.
- ?** Display expanded usage message. This option overrides all others.

**EXTERNAL INFLUENCES****Environment Variables**

**LC\_MESSAGES** determines the language in which messages are displayed. The current language settings can be checked with *locale(1)*.

**RETURN VALUE**

**download** returns the following values:

- 0 Successful completion.
- 1 An error in execution (I/O, subsystem, security, etc.) occurred.
- 2 An error in command syntax occurred: for example, an unknown command-line option was passed.
- 3 Timeout in communication to server. May indicate **ARMserver** is not running.

**DIAGNOSTICS**

The following messages can be generated by **download**:

**Usage:** **download** -D slot | -C | -M [-?] codefilename array-id

An error in command syntax has occurred. Re-enter the command with all necessary arguments.

**download:** Arg out of range

One of the arguments has exceeded its maximum or minimum size, or is incorrect in form. Check the size and form of each argument.

**download: Unknown argument**

An invalid argument was specified. Check command usage.

**download: No such file or directory**

The specified *array-id* does not exist or does not identify a disk array that is communicating with the system. Verify the correct *array-id* with *ioscan*(1M) or *arraydsp*(1M).

**download: Error in command execution, <Additional Error Info>:  
<Error Info Decode>**

The command failed due to a device error, an internal error, or a system error. The *Additional Error Info* and *Error Info Decode* fields will hold specifics about the failure and its cause.

**download: Array was just powered-on or reset.**

The command failed because the disk array has been powered-on or reset. Reissue the command and it should succeed.

## EXAMPLES

Download new firmware from file *coderev2.1* to the controllers on the disk array identified by device file */dev/rdsd/c2t0d0*:

```
download -C coderev2.1 /dev/rdsd/c2t0d0
```

Download new firmware from file *diskrev3.2* to the disk installed in slot B2 of disk array serial number 00786b5c0000:

```
download -D B2 diskrev3.2 00786b5c0000
```

Copy the current firmware from the primary controller to the secondary controller on disk array identified by alias *AUTORAID4*:

```
download -M AUTORAID4
```

## SECURITY CONFIGURATION

This command is modified for all security configurations.

### Security Behavior/Restrictions

Use of this command is restricted to authorized users only.

### Command Authorizations

This command requires the **sysadmin** authorization to successfully execute.

### Privileges

The command has been modified to support least privilege. The potential privileges possessed by the command and their uses include:

**allowdacread** This privilege is raised to provide discretionary read access to the devices.

**allowdacwrite** This privilege is raised to provide discretionary write access to the devices.

**allowmacread** This privilege is raised to provide mandatory read access to the devices.

**allowmacwrite** This privilege is raised to provide mandatory write access to the devices.

**filesysops** This privilege is raised to allow the *mknod*(2) system call to succeed.

**writeaudit** The command generates its own audit records and submits these directly to the system audit trail. This privilege is raised whenever the command needs to write an audit record.

## AUTHOR

**download** was developed by HP.

## SEE ALSO

*ARMServer*(1M), *arraycfg*(1M), *arraydsp*(1M), *arrayfmt*(1M), *arraylog*(1M), *arraymgr*(1M), *arrayrbld*(1M), *arrayrecover*(1M), *drivetest*(1M), *dteststat*(1M), *logprint*(1M).

**NAME**

dpp - dedicated ports parser used by DDFA software

**SYNOPSIS**

**dpp** *dp\_file* [-c] [-k] [-l *log\_file*] [-p *ocd\_program*]

**DESCRIPTION**

The Dedicated Ports Parser command (**dpp**) is part of the Data Communications and Terminal Controller (DTC) Device File Access (DDFA) software. It parses the Dedicated Ports file (**dp**) and spawns an Outbound Connection Daemon (**ocd**) for each valid entry in the **dp** file.

**dpp** can be run from the shell or it can be included in a system initialization script to automatically run the DDFA software each time the system is booted.

See *ddfa(7)* for more information on how to configure the DDFA software and for an explanation of how it works.

**Options and Arguments**

**dpp** recognizes the following options and arguments:

- |                       |   |
|-----------------------|---|
| <i>dp_file</i>        | It must be the first argument. The <b>dp</b> file ( <i>dp_file</i> ) defines the link between a terminal server port and the device file used by applications to access the port. Its contents must meet the specifications given in <i>dp(4)</i> . If it is modified, <b>dpp</b> must be run again to activate the changes.  |
| -c                    | Specify that the <b>dp</b> file should be parsed and that all incorrect entries should be logged without invoking any <b>ocd</b> processes. This option is useful for debugging the <b>dp</b> file before running it properly. The -p option is ignored if the -c option is used.   |
| -k                    | Specify that the device file corresponding to each valid entry in the <b>dp</b> file should be removed before launching <b>ocd</b> for each valid entry. Removing the device file eventually causes an <b>ocd</b> process (if any is running) to shutdown. If this option is omitted, no device files will be removed and, therefore, only newly added valid entries in the <b>dp</b> file will have <b>ocd</b> launched.<br><br><b>ocd</b> normally creates and removes device files. However, if the process is killed incorrectly, such as with <b>kill -9</b> , the device file may remain. If the system is rebooted, the -k option can be specified to restart all <b>dp</b> file entries correctly.<br><br>If a corresponding <b>ocd</b> no longer exists, the device file is removed by any following invocation of <b>ocd</b> that requires the same device file.<br><br>In order to shutdown every <b>ocd</b> running without restarting them, the following command can be executed:<br><br><pre>kill -15 `ps -e   grep ocd   awk '{print \$1}'`</pre> |
| -l <i>log_file</i>    | Specify where to log error messages. If this option is omitted, all error messages are logged to standard output.<br><br>If the specified file does not already exist, it is created. The file must be nonexecutable and readable by <b>dpp</b> .   |
| -p <i>ocd_program</i> | Specify the path for an outbound connection daemon. The default path for is <b>/usr/sbin/ocd</b> . The daemon must be executable.   |

**DIAGNOSTICS**

Error messages are logged for bad arguments, bad file entries, and **ocd** creation errors. By default, they are logged to standard output. If the -l option is used, they are appended to the specified log file.

- (0) **ERROR: dp file is mandatory**
- (1) **ERROR: dp file must be the first argument**
- (2) **ERROR: Cannot read dp file (*filename*)**

The **dp** file either does not exist or cannot be accessed with the current access privileges.

- (3) **ERROR: No log file defined (-l option)**
- (4) **ERROR: Cannot create log file (-l *filename*)**

The log file cannot be created, either because of an invalid path or because of insufficient access privileges.

(5) **ERROR: Cannot access log file (-l filename)**

The log file cannot be accessed, either because of an invalid path or because of insufficient access privileges. The log file must be readable by everyone.

(6) **ERROR: No ocd file defined in program option**

(7) **ERROR: Cannot execute ocd program (-p pathname)**

The ocd program specified in the -p option either does not exist or is not an executable file with the current access privileges.

(8) **ERROR: Cannot purge device file (/dev/ filename)**

The -k option has been specified and the device file exists, but it cannot be purged because of insufficient access privileges.

(9) **ERROR: Cannot execute default program (/usr/sbin/ocd)**

The default ocd cannot be executed, either because of insufficient access privileges or because it has not been correctly installed.

(10) **ERROR: Entry ignored (Bad IP address)**

The dp file entry specified does not have a valid IP address.

(11) **ERROR: Entry ignored (no port/board info)**

(12) **ERROR: Entry ignored (Bad port number)**

The port specified is either not a decimal value or a string composed of x or X characters.

(13) **ERROR: Entry ignored (Bad board number)**

The board specified is either not a decimal value or a string composed of x or X characters.

(14) **ERROR: No more processes available on system**

The ocd program specified cannot be started because there are no processes available on the system.

(15) **ERROR: Entry ignored (no device\_name)**

(16) **ERROR: Entry ignored (Bad device\_name)**

The device file specified cannot be created, either because of an invalid path or because of insufficient access privileges.

(17) **ERROR: Entry ignored (Bad config name)**

The specified configuration file cannot be read, either because of an invalid path or because of insufficient access privileges.

(18) **ERROR: Entry ignored (Invalid log level)**

The specified logging level is not in the range 0 to 3.

(19) **ERROR: Entry ignored (Bad node name)**

The specified node name does not exist or does not have an entry in a name database.

## WARNINGS

To ensure that commands (such as *ps*) display the correct device file name (that is, the *pseudonym*), all pseudonyms should be placed into the directory */dev/telnet*. If pseudonyms are not specified for placement in this directory, the correct display of device file names with many commands is not guaranteed.

In addition, to ensure that commands (such as *w*, *passwd*, *finger*, and *wall*) work correctly, each pseudonym must be unique in its first 17 characters (including the directory prefix */dev/telnet/*). If pseudonyms are not unique in their first 17 characters, the correct functioning of many commands is not guaranteed.

Also, in order to reliably handle timing mark negotiations (and ensure that files printing on a printer attached to a terminal server have been completely flushed to that printer), the following line must be added near the end of each printer interface script for printers attached to a terminal server:

```
stty exta <&1 2>/dev/null
```

The printer interface scripts reside in the directory `/etc/lp/interface`. The line must be added just prior to the final 'exit' command in each printer interface script.

If this line is not added as specified, the printing reliability of printers attached to a terminal server is not guaranteed.

Finally, `ocd` should be killed using `kill -15`. Do not use `kill -9` for this purpose as it does not remove the device file. `ocd` verifies the validity of an existing pseudonym before trying to use it. `dpp` and `ocd` use data stored in the file `/var/adm/utmp.dfa` to verify whether a process still owns a pseudonym before taking it over. If `ocd` finds an unowned pseudonym, it uses it.

**FILES**

```
/usr/examples/ddfa/dp
/usr/examples/ddfa/pcf
/usr/sbin/dpp
/usr/sbin/ocd
/usr/sbin/ocdebug
/var/adm/dpp_login.bin
/var/adm/utmp.dfa
```

**SEE ALSO**

`ocd(1M)`, `ocdebug(1M)`, `dp(4)`, `pcf(4)`, `ddfa(7)`.

  
d

**NAME**

drivetest - test a disk in the array

**SYNOPSIS**

```
drivetest -D slot { -w percent | -r percent | -s } [-?] array-id
```

**DESCRIPTION**

**drivetest** initiates testing of disks in the disk array identified by *array-id*. Three different types of testing can be performed to verify disk operation and integrity. The amount of the disk surface tested can be controlled depending on how thorough a test is desired. Results of the testing are retrieved using the **dteststat** command.

The *array-id* used to address the disk array can be the disk array serial number, the character device file name of any LUN on the array (LUN 0 if no LUNs are created), or the alias text string assigned to the disk array.

**Options**

**drivetest** supports the following options:

- D slot** Identifies which disk to test. The disk installed in the cabinet slot identified by *slot* will be tested. Slot numbers must be of the form *An* or *Bn*, where *A* or *B* correspond to a cabinet column, and *n* corresponds to a shelf position (1-6).
- r percent** Perform a read/verify test on the percentage of the disk media indicated by *percent* (0-100). This is a nondestructive test that will not alter any data on the disk being tested. It is not necessary to down the disk before performing a read/verify test.
- s** Initiate a selftest of the disk. An internal selftest will check the operation of the disk. Results of the testing are retrieved using the **dteststat** command.
- w percent** Perform a write/read/verify test on the percentage of the disk media indicated by *percent* (0-100). This is a destructive test that will destroy data on the disk being tested. The **arraycfg** utility must be used to down the disk before beginning the test.
- ?** Display expanded usage message. This option overrides all others.

**EXTERNAL INFLUENCES****Environment Variables**

**LC\_MESSAGES** determines the language in which messages are displayed. The current language settings can be checked with *locale(1)*.

**RETURN VALUE**

**drivetest** returns the following values:

- 0 Successful completion.
- 1 An error in execution (I/O, subsystem, security, etc.) occurred.
- 2 An error in command syntax occurred: for example, an unknown command-line option was passed.
- 3 Timeout in communication to server. May indicate the **ARMServer** is not running.

**DIAGNOSTICS**

The following messages can be generated by **drivetest**:

**Usage: drivetest -D slot -w percent | -r percent | -s [-?] array-id**

An error in command syntax has occurred. Re-enter the command with all necessary arguments.

**drivetest: Arg out of range**

One of the arguments has exceeded its maximum or minimum size, or is incorrect in form. Check the size and form of each argument.

**drivetest: Unknown argument**

An invalid argument was specified. Check command usage.

**drivetest: No such file or directory**

The specified *array-id* does not exist or does not identify a disk array that is communicating with the system. Verify the correct *array-id* with *ioscan(1M)* or *arraydsp(1M)*.

**drivetest: Error in command execution, <Additional Error Info>:  
<Error Info Decode>**

The command failed due to a device error, an internal error, or a system error. The *Additional Error Info* and *Error Info Decode* fields will hold specifics about the failure and its cause.

**drivetest: Disk did not start up following reset.**

The disk did not start up following a reset. Check the status of the disk.

**drivetest: A diagnostic is currently running on this disk.**

A diagnostic test is already running on the specified disk.

**drivetest: Array was just powered-on or reset.**

The command failed because the disk array has been powered-on or reset. Reissue the command and it should succeed.

## EXAMPLES

Perform a read/verify test on the disk installed in slot B3 of the disk array identified by device file `/dev/rdisk/c2t0d0`. Perform the test over 50 percent of the disk surface.

```
drivetest -D B3 -r 50 /dev/rdisk/c2t0d0
```

Perform a write/read/verify test on the disk installed in slot A5 of disk array serial number `00786b5c0000`. Perform the test over 100 percent of the disk surface. Note that before issuing this command the disk must be downed (destructively) using the `arraycfg` utility.

```
drivetest -D A5 -w 100 00786b5c0000
```

Initiate a selftest on the disk installed in slot A1 of the disk array identified by alias `AUTORAID1`.

```
drivetest -D A1 -s AUTORAID1
```

## DEPENDENCIES

**ARMServer** must be running to execute this command. See *ARMServer(1M)*.

## SECURITY CONFIGURATION

This command is modified for all security configurations.

### Security Behavior/Restrictions

Use of this command is restricted to authorized users only.

### Command Authorizations

This command requires the **sysadmin** authorization to successfully execute.

### Privileges

The command has been modified to support least privilege. The potential privileges possessed by the command and their uses include:

<b>allowdacread</b>	This privilege is raised to provide discretionary read access to the devices.
<b>allowdacwrite</b>	This privilege is raised to provide discretionary write access to the devices.
<b>allowmacread</b>	This privilege is raised to provide mandatory read access to the devices.
<b>allowmacwrite</b>	This privilege is raised to provide mandatory write access to the devices.
<b>filesysops</b>	This privilege is raised to allow the <code>mknod(2)</code> system call to succeed.
<b>writeaudit</b>	The command generates its own audit records and submits these directly to the system audit trail. This privilege is raised whenever the command needs to write an audit record.

## AUTHOR

**drivetest** was developed by HP.

## SEE ALSO

*ARMServer(1M)*, *arraycfg(1M)*, *arraydsp(1M)*, *arrayfmt(1M)*, *arraylog(1M)*, *arraymgr(1M)*, *arrayrld(1M)*, *arrayrecover(1M)*, *dteststat(1M)*, *download(1M)*, *logprint(1M)*.

**NAME**

dsp - display status of an HP SCSI disk array

**SYNOPSIS**

**dsp -p** [-h | -d] *device\_file*

**dsp -l** [-h | -d] *device\_file*

**DESCRIPTION**

**dsp** displays the status of the LUN (in an HP SCSI disk array) that is associated with the device file *device\_file*. **dsp** displays the status of physical drives in an array (when the **-p** option is specified), or the status of LUNs in an array (when the **-l** option is specified). This information can be displayed in interpreted form, or in raw hexadecimal or raw decimal format.

**Options**

- p** Display physical drive status. The **-p** option displays the status of a LUN's physical drives, regardless of their LUN ownership. This information is retrieved the array physical page (Mode Page 2A), and inquiry data.
- l** Display LUN status. The **-l** option displays information about the state of the LUN including it's RAID level, block and segment sizes, reconstruction information, and so on. This information is retrieved from the array logical page (Mode Page 2B), and inquiry data.

By default, data is displayed in interpreted form; if raw data is desired, one of the following options can be used:

- h** Raw hex format. Displays the data in raw hex format in rows, each of which contains the ASCII representation of 16 hexadecimal data bytes, separated by spaces.
- d** Raw decimal format. Displays the data in raw decimal format in rows, each of which contains the ASCII representation of 16 decimal data bytes, separated by spaces.

**RETURN VALUE**

**dsp** returns the following values:

- 0** Successful completion.
- 1** Command failed (an error occurred).

**DIAGNOSTICS AND ERRORS**

Errors can originate from problems with:

- **dsp**
- SCSI (device level) communications
- system calls

**Error messages generated by dsp:**

**usage: dsp <-p | -l> [-h | -d] <special>**

An error in command syntax has occurred. Enter the command again with all required arguments.

**dsp: Arg out of range**

One of the arguments is larger than its allowed maximum value (or smaller than its allowed minimum value), or is incorrect in form. Check the size and form of each argument and make appropriate corrections.

**dsp: LUN # too big**

The LUN number, which is derived from the device special file name, is out of range.

**dsp: Not a raw file**

Utilities must be able to open the device file for raw access.

**dsp: Transfer length error**

The amount of data actually sent to or received from the device was not the expected amount.

**dsp: LUN does not exist**

The requested LUN is not among those known to the controller.

**dsp: Not an HP SCSI disk array**

The device being addressed is not an HP SCSI disk array.



**SCSI (device level) communication errors:**

Sense data associated with the failed operation is printed.

**Error messages generated by system calls:**

**dsp** uses the following system calls:

`stat()`, `open()`, `close()`, `read()`, `write()`, and `ioctl()`.

Documentation for these HP-UX system calls contains information about the specific error conditions associated with each call. **dsp** does not alter the value of `errno`. The interpretation of `errno` for printing purposes is performed by the system utility `strerror()`.

**EXAMPLES**

To display the status of the drives on the HP SCSI disk array `/dev/rdisk/c2t4d0` on a Series 700:

```
dsp -p /dev/rdisk/c2t4d0
```

To display the status of the LUN associated with the HP SCSI disk array `/dev/rdisk/c2t0d0` on a Series 800 in raw hex format:

```
dsp -l -h /dev/rdisk/c2t0d0
```

To display the status of the drives on the HP SCSI disk array `/dev/rdisk/c2t5d0` in raw decimal format on a Series 700:

```
dsp -p -d /dev/rdisk/c2t5d0
```

**DEPENDENCIES**

The HP C2425 and HP C2427 disk arrays are only supported on Series 700 systems running HP-UX version 9.0X.

The HP C2430 disk array is supported on Series 700 and 800 systems running HP-UX versions 9.0X and 10.0X.

**AUTHOR**

**dsp** was developed by HP.

**NAME**

dteststat - display the results of disk testing, or cancel the test

**SYNOPSIS**

```
dteststat [-D slot] [-?] array-id
dteststat -D slot -c [-?] array-id
```

**DESCRIPTION**

**dteststat** displays the results of testing performed on the indicated disk in the disk array identified by *array-id*. A test in progress can also be canceled using **dteststat**. Disk testing is initiated using the **drivetest** command.

The *array-id* used to address the disk array can be the disk array serial number, the character device file name of any LUN on the array (LUN 0 if no LUNs are created), or the alias text string assigned to the disk array.

**Options**

**dteststat** supports the following options:

- none** Display results for all disks being tested. Tests results are shown for disks that have completed testing, and progress is shown for tests still underway.
- c** Cancel the test currently in progress on the indicated disk. A test can be canceled on only one disk at a time, so the **-D** option must be included when canceling a test.
- D slot** Display test results for the disk installed in the cabinet slot identified by *slot*. Slot numbers must be of the form *An* or *Bn*, where *A* or *B* correspond to a cabinet column, and *n* corresponds to a shelf position (1-6).
- ?** Display expanded usage message. This option overrides all others.

**EXTERNAL INFLUENCES****Environment Variables**

**LC\_MESSAGES** determines the language in which messages are displayed. The current language settings can be checked with *locale(1)*.

**RETURN VALUE**

**dteststat** returns the following values:

- 0 Successful completion.
- 1 An error in execution (I/O, subsystem, security, etc.) occurred.
- 2 An error in command syntax occurred: for example, an unknown command-line option was passed.
- 3 Timeout in communication to server. May indicate the **ARMServer** is not running.

**DIAGNOSTICS**

The following messages can be generated by **dteststat**:

**Usage: dteststat [-D slot] [-c] array-id**

An error in command syntax has occurred. Re-enter the command with all necessary arguments.

**dteststat: Arg out of range**

One of the arguments has exceeded its maximum or minimum size, or is incorrect in form. Check the size and form of each argument.

**dteststat: Unknown argument**

An invalid argument was specified. Check command usage.

**dteststat: No such file or directory**

The specified *array-id* does not exist or does not identify a disk array that is communicating with the system. Verify the correct *array-id* with *ioscan(1M)* or *arraydsp(1M)*.

**dteststat: Error in command execution, <Additional Error Info>:**

**<Error Info Decode>**

The command failed due to a device error, an internal error, or a system error. The *Additional Error Info* and *Error Info Decode* fields will hold specifics about the failure and its cause.

**dteststat: Array was just powered-on or reset.**

The command failed because the disk array has been powered-on or reset. Reissue the command and it should succeed.

### EXAMPLES

Display the results of the test performed on the disk installed in slot B3 of the disk array identified by device file `/dev/rdisk/c2t0d0`:

```
dteststat -D B3 /dev/rdisk/c2t0d0
```

Display the results of the tests performed on all disks installed in the disk array identified by alias `AUTORAID1`:

```
dteststat AUTORAID1
```

Cancel a test currently in progress on the disk installed in slot A1 of disk array serial number `00786b5c0000`:

```
dteststat -D A1 -c 00786b5c0000
```

### DEPENDENCIES

**ARMServer** must be running to execute this command. See *ARMServer(1M)*.

### SECURITY CONFIGURATION

This command is modified for all security configurations.

#### Security Behavior/Restrictions

Use of this command is restricted to authorized users only.

#### Command Authorizations

This command requires the **sysadmin** authorization to successfully execute.

#### Privileges

The command has been modified to support least privilege. The potential privileges possessed by the command and their uses include:

<b>allowdacread</b>	This privilege is raised to provide discretionary read access to the devices.
<b>allowdacwrite</b>	This privilege is raised to provide discretionary write access to the devices.
<b>allowmacread</b>	This privilege is raised to provide mandatory read access to the devices.
<b>allowmacwrite</b>	This privilege is raised to provide mandatory write access to the devices.
<b>filesysops</b>	This privilege is raised to allow the <i>mknod(2)</i> system call to succeed.
<b>writeaudit</b>	The command generates its own audit records and submits these directly to the system audit trail. This privilege is raised whenever the command needs to write an audit record.

### AUTHOR

**dteststat** was developed by HP.

### SEE ALSO

*ARMServer(1M)*, *arraycfg(1M)*, *arraydsp(1M)*, *arrayfmt(1M)*, *arraylog(1M)*, *arraymgr(1M)*, *arrayrbld(1M)*, *arrayrecover(1M)*, *download(1M)*, *drivetest(1M)*, *logprint(1M)*.

**NAME**

dump, rdump - incremental file system dump, local or across network

**SYNOPSIS**

`/usr/sbin/dump` [*option* [*argument* ...] *filesystem*]

`/usr/sbin/rdump` [*option* [*argument* ...] *filesystem*]

**DESCRIPTION**

The **dump** and **rdump** commands copy to magnetic tape all files in the *filesystem* that have been changed after a certain date. This information is derived from the files `/var/adm/dumpdates` and `/etc/fstab`. *option* specifies the date and other options about the dump. *option* consists of characters from the set `0123456789bdfnsuw`. The **dump** and **rdump** commands work only on file systems of type **hfs**. If the given file system is not of type **hfs**, **dump** and **rdump** will abort after printing an error message.

**Options**

- 0-9** This number is the "dump level". All files modified since the last date stored in file `/var/adm/dumpdates` for the same file system at lesser levels will be dumped. If no date is determined by the level, the beginning of time is assumed. Thus, the option **0** causes the entire file system to be dumped.
- b** The blocking factor is taken from the next argument (default is 10 if not specified). Block size is defined as the logical record size times the blocking factor. **dump** writes logical records of 1024 bytes. When dumping to tapes with densities of 6250 BPI or greater without using the **b** option, the default blocking factor is 32.
- d** The density of the tape (expressed in BPIs) is taken from the next *argument*. This is used in calculating the amount of tape used per reel. The default value of 1600 assumes a reel tape.
- f** Place the dump on the next *argument* file instead of the tape. If the name of the file is `-`, **dump** writes to the standard output. When using **rdump**, this option should be specified, and the next argument supplied should be of the form *machine:device*.
- n** Whenever **dump** and **rdump** require operator attention, notify all users in group **operator** by means similar to that described by *wall(1)*.
- s** The size of the dump tape is specified in feet. The number of feet is taken from the next *argument*. When the specified size is reached, **dump** and **rdump** wait for reels to be changed. The default tape size value of 2300 feet assumes a reel tape.
- u** If the dump completes successfully, write on file `/var/adm/dumpdates` the date when the dump started. This file records a separate date for each file system and each dump level. The format of `/var/adm/dumpdates` is user-readable and consists of one free-format record per line: file system name, increment level, and dump date in *ctime(3C)* format. The file `/var/adm/dumpdates` can be edited to change any of the fields if necessary.
- W** For each file system in `/var/adm/dumpdates`, print the most recent dump date and level, indicating which file systems should be dumped. If the **W** option is set, all other options are ignored and **dump** exits immediately.
- w** Operates like **W**, but prints only file systems that need to be dumped.

If no arguments are given, *option* is assumed to be **0u** and a default file system is dumped to the default tape.

Sizes are based on 1600-BPI blocked tape; the raw magnetic tape device must be used to approach these densities. Up to 32 read errors on the file system are ignored. Each reel requires a new process; thus parent processes for reels already written remain until the entire tape is written.

The **rdump** command creates a server, `/usr/sbin/rmt` or `/etc/rmt`, on the remote machine to access the tape device.

**dump** and **rdump** require operator intervention for any of the following conditions:

- end of tape,
- end of dump,
- tape-write error,
- tape-open error, or
- disk-read error (if errors exceed threshold of 32).

In addition to alerting all operators implied by the **n** option, **dump** and **rdump** interact with the control terminal operator by posing questions requiring **yes** or **no** answers when it can no longer proceed or if something is grossly wrong.

Since making a full dump involves considerable time and effort, **dump** and **rdump** each establish a checkpoint at the start of each tape volume. If, for any reason, writing that volume fails, **dump** and **rdump** will, with operator permission, restart from the checkpoint after the old tape has been rewound and removed and a new tape has been mounted.

**dump** and **rdump** periodically report information to the operator, including typically low estimates of the number of blocks to write, the number of tapes it will require, the time needed for completion, and the time remaining until tape change. The output is verbose to inform other users that the terminal controlling **dump** and **rdump** is busy and will be for some time.

### Access Control Lists (ACLs)

The optional entries of a file's access control list (ACL) are not backed up with **dump** and **rdump**. Instead, the file's permission bits are backed up and any information contained in its optional ACL entries is lost (see *acl(5)*).

### EXAMPLES

In the following example, assume that the file system **/mnt** is to be attached to the file tree at the root directory, **(/)**. This example causes the entire file system (**/mnt**) to be dumped on **/dev/rmt/c0t0d0BEST** and specifies that the density of the tape is 6250 BPI.

```
/usr/sbin/dump 0df 6250 /dev/rmt/c0t0d0BEST /mnt
```

### WARNINGS

**dump** will not backup a file system containing large files.

Tapes created from file systems containing files with UID/GIDs greater than 60,000 will have a new magic number in the header to prevent older versions of *restore(1M)* from incorrectly restoring ownerships for these files.

### AUTHOR

**dump** and **rdump** were developed by the University of California, Berkeley.

### FILES

<b>/dev/rdisk/c0d0s0</b>	Default file system to dump from.
<b>/dev/rmt/0m</b>	Default tape unit to dump to.
<b>/var/adm/dumpdates</b>	New format-dump-date record.
<b>/etc/fstab</b>	Dump table: file systems and frequency.
<b>/etc/group</b>	Used to find group <b>operator</b> .

### SEE ALSO

*restore(1M)*, *rmt(1M)*, *fstab(4)*, *acl(5)*.

**NAME**

dumpfs - dump file system information

**SYNOPSIS**

*/usr/sbin/dumpfs rootdir | special*

**DESCRIPTION**

The **dumpfs** command prints the super block and cylinder group information for an HFS file system to the standard output. The file system may be specified by its root directory or the name of the device special file on which it resides. The information is very long and detailed. This command can be used to find file system information such as the file system block size or the minimum free space percentage.

**DEPENDENCIES**

The **dumpfs** command can only be used on HFS file systems.

**AUTHOR**

**dumpfs** was developed by the University of California, Berkeley.

**SEE ALSO**

fsck(1M), mkfs(1M), newfs(1M), tuneufs(1M), disktab(4), fs(4).

**NAME**

edquota - edit user disk quotas

**SYNOPSIS**

```
/usr/sbin/edquota [-p proto-user] username ...
/usr/sbin/edquota -t
```

**DESCRIPTION**

The **edquota** command is the quota editor. One or more user names can be specified on the command line. For each *username*, a temporary file is created with a textual representation of the current disk quotas for that user, and an editor is invoked on the file. The quotas can then be modified, new quotas added, etc. Upon leaving the editor, **edquota** reads the temporary file and modifies the binary quota files to reflect the changes made.

The editor invoked is specified by the **EDITOR** environment variable. It defaults to **vi** (see *vi*(1)).

In order for quotas to be established on a file system, the root directory of the file system must contain a file named **quotas**. See *quota*(5) for details.

Quotas can be established only for users whose user ID is less than 67,000,000. Attempts to establish quotas for other users will result in an error message. This restriction will be removed in a future version of HP-UX.

Only users who have appropriate privileges can edit quotas.

**Options**

- p** *proto\_user* Duplicate the quotas of the user name *proto\_user* for each *username*. This is the normal mechanism used to initialize quotas for groups of users.
- t** Edit the time limits for each file system. Time limits are set for file systems, not users. When a user exceeds the *soft* limit for blocks or inodes on a file system, a countdown timer is started and the user has an amount of time equal to the time limit in which to reduce usage to below the soft limit (the required action is given by the **quota** command). If the time limit expires before corrective action is taken, the quota system enforces policy as if the *hard* limit had been exceeded. The default time limit of 0 is interpreted to mean the value in **<sys/quota.h>**, or one week (7 days). Time units of sec(onds), min(utes), hour(s), day(s), week(s), and month(s) are understood. Time limits are printed in the greatest possible time unit such that the value is greater than or equal to one.

**Temporary File Formats**

Here is an example of the temporary file created for editing user block and inode quotas:

```
fs /mnt blocks (soft = 100, hard = 120) inodes (soft = 0, hard = 0)
fs / blocks (soft = 1000, hard = 1200) inodes (soft = 200, hard = 200)
```

Here is the format for editing quota time limits:

```
fs /mnt blocks time limit = 10.00 days, files time limit = 20.00 days
fs / blocks time limit = 0 (default), files time limit = 0 (default)
```

When editing **(default)** values, it is not necessary to remove the **(default)** string. For example, to change the **blocks time limit** for **/**, changing the 0 to **4 days** is sufficient.

**WARNINGS**

When establishing quotas for a user who has had none before, (for either blocks or inodes), the quota statistics for that user do not include any currently occupied file system resources. Therefore, it is necessary to run **quotacheck** (see *quotacheck*(1M)) to collect statistics for that user's current usage of that file system. See *quota*(5) for a detailed discussion of this topic.

**edquota** will only edit quotas on local file systems.

**AUTHOR**

**edquota** was developed by the University of California, Berkeley, and by Sun Microsystems, Inc.

**FILES**

**/etc/fstab** Static information about the file systems.

<code>/etc/mnttab</code>	Mounted file system table
<code>directory/quotas</code>	Quota statistics static storage for a file system, where <i>directory</i> is the root of the file system as specified to the <b>mount</b> command (see <i>mount(1M)</i> ).

**SEE ALSO**

`vi(1)`, `quota(1)`, `quotacheck(1M)`, `quotacheck_hfs(1M)`, `quota(5)`.

e



**NAME**

`eisa_config` - EISA configuration tool

**SYNOPSIS**

```
eisa_config
eisa_config [-a]
eisa_config [-c cfgfile]
eisa_config [-n scifile]
```

**DESCRIPTION**

`eisa_config` is a specialized program for configuring EISA and ISA (referred to collectively as E/ISA) I/O boards on HP-UX workstations equipped with EISA backplanes. It is used each time the E/ISA configuration is to be changed in any way; i.e., whenever an EISA or ISA board is added to the system, removed from the system, or moved to a different location in the system. `eisa_config` should be run before any physical board configuration or installation changes are made. (This is not necessary in some cases -- see automatic mode below.)

`eisa_config` interprets information stored in configuration files and uses it to configure system resources needed to properly interact with E/ISA boards. Even though they may be physically present in the computer, E/ISA boards cannot be used by the HP-UX operating system until configuration by `eisa_config` is complete.

The `eisa_config` command takes one of four forms:

<code>eisa_config</code>	Use interactive commands to examine or modify configuration. <code>eisa_config</code> prompts for a command, executes it, reports the results of command execution, then prompts for the next command.
<code>eisa_config -a</code>	Attempt to automatically add new EISA boards to the configuration. This option is used by <code>/sbin/bcheckrc</code> but should not be used elsewhere. ISA boards cannot be added with this option.
<code>eisa_config -c <i>cfgfile</i></code>	Check configuration (CFG) file (discussed below). This option is used mostly by E/ISA board developers. It simply checks the specified CFG file to verify that it follows correct grammar and can be used by <code>eisa_config</code> . This option does not affect current configuration in any way.
<code>eisa_config -n <i>scifile</i></code>	Non-target mode. This option uses the contents of <i>scifile</i> instead of non-volatile memory (NVM) to set up E/ISA configuration, and is most commonly used for creating identical configurations on multiple workstations.

**Assigning Resources**

Depending on their design, internal capabilities, and their role in system operation, E/ISA boards use various combinations of one or more system resources such as DMA channels, interrupt lines, memory, etc. Also, given boards do not always use a full set of system resources; for example, EISA provides 11 interrupt lines, but a given board might be able to use only lines 3, 5, and 6. Thus a means for the board to determine what resources are to be used must be provided.

ISA boards use physical switches or jumpers on the board to specify what resources are to be used. The person installing the board sets the switches or jumpers as specified by the board's manufacturer and based on system needs. There are thousands of different kinds of ISA boards, but unfortunately there are no standard conventions for switch and jumper usage. This results in much confusion and numerous configuration problems. For example, it is easy to inadvertently assign a given resource to two different boards, but often very difficult to diagnose the problem.

EISA boards usually have no switches or jumpers for resource assignment. Instead, each EISA board has a corresponding configuration (CFG) file that tells the system how the board can be used and what resources it needs. `eisa_config` is the HP-UX system program that interprets the various CFG files for all boards in the system, then builds a conflict-free configuration.

**Configuration Files**

All EISA boards have a corresponding CFG file. ISA boards, when used in HP-UX systems, must also have a corresponding CFG file. Although `eisa_config` cannot automatically configure an ISA board, it can use

the contents of the CFG file to determine what switch or jumper settings on an ISA board can be used to prevent resource conflicts.

**eisa\_config** expects to find a CFG file for each E/ISA board connected to the workstation. The administrator is responsible for making sure that these CFG files are present in directory `/sbin/lib/eisa`. CFG files corresponding to boards being used should always be kept in this directory. Do not remove them after **eisa\_config** is run the first time, because they will be needed every time the configuration is changed, such as when a new board is added or one is removed. Do not change the file names of the CFG files. The file name has a specific format which is used by **eisa\_config** to automatically match a board with its CFG file.

CFG files are normally supplied by the E/ISA board manufacturer. Two scenarios apply:

- If the E/ISA board is supplied by HP, the CFG file corresponding to the board is loaded into `/sbin/lib/eisa` as part of normal operating system installation. It should never be removed.
- If the E/ISA board is not supplied by HP, install both the CFG file and the software driver for the board from HP-UX-readable media supplied by the board manufacturer. Copy the CFG file to directory `/sbin/lib/eisa` where it must remain as long as the card is present in the system.

All CFG files must follow a grammar specified in the EISA bus specification. The most basic building block in the CFG grammar is the *board*. Each board has several attributes including board ID (to match with a board's ID register), manufacturer, ASCII text describing what the board does, what kinds of slots the board can go in, whether the board has a readable ID register, and various other capability attributes.

Each file can also contain lists of board-wide resources (such as I/O registers, switches, and jumpers) and how they should be initialized.

A board can be treated as a set of one or more *functions* where a given board contains a single function or multiple functions. An example of a two-function board is one having both a serial port and a parallel printer port. Each function has a separate block in that board's CFG file. Each function has a name, a type, and a set of configuration *choices*.

Each *choice* block has a name and a set of attributes. These attributes include what resources the choice requires and whether the function is enabled or disabled by that choice. Initialization is also usually specified within a choice. A given choice might require that certain registers be initialized to a specified value and that switches be set in a certain way.

## Configuration Processing

E/ISA configuration is handled as follows:

- **eisa\_config** builds a conflict-free configuration, then saves the configuration in EISA non-volatile memory (NVM).
- Appropriate drivers and device files must be installed before rebooting the system.
- Next time the operating system is rebooted, the HP-UX kernel initializes the specified E/ISA boards according to the contents of NVM.

If a board is currently present in the system, but has no corresponding configuration data in NVM, the EISA board cannot be used until the **eisa\_config** program is run again and the new board is accounted for in NVM. A newly installed or existing E/ISA board is not usable until **eisa\_config** has added it and the system has been rebooted with the necessary drivers and device special files installed. See EXAMPLES for an illustration of how to add a new board to the system.

It is possible to add EISA boards that do not have switches or jumpers to the configuration without running **eisa\_config** interactively. The `/sbin/bcheckrc` script invokes **eisa\_config** with automatic mode during each system initialization. If a board has been added since the last time **eisa\_config** was executed, **eisa\_config** attempts to add the new board to the configuration. If the new board is successfully added, the system may need to be rebooted (`/sbin/bcheckrc` does this automatically). If the new board could not be added to the configuration, a warning is written to the system console and `/etc/eisa/config.err`.

In addition to writing to NVM, **eisa\_config** also automatically saves the current configuration to an SCI file called `/etc/eisa/system.sci`. SCI files can also be created by the interactive **save** command (see below). The E/ISA subsystem can also be initialized from an SCI file, rather than from NVM by using the **eisa\_config -n** command form discussed earlier. SCI files are quite useful when a site has several identically-configured workstations. Run **eisa\_config** on one system and save the configuration in an SCI file. Copy this file to other systems, then use it to initialize those systems. Remember that the

configuration must be saved to NVM and the system rebooted before the E/ISA boards can be used.

### Drivers and Device Files

Running **eisa\_config** is not the only task necessary when adding an E/ISA board to a system. Corresponding I/O drivers must be added to the kernel and appropriate device files must be created. These steps are the same as is required for any I/O card, and can be performed either before or after running **eisa\_config**. The important thing to remember is that the E/ISA board cannot be used until *all* necessary tasks are complete.

### Interactive Commands

If the command form **eisa\_config** is used, **eisa\_config** runs in interactive mode. Interactive mode conducts configuration changes by using a series of keyboard commands. **eisa\_config** prompts for a command, executes it, displays the results of executing the command, then prompts for the next command. Interactive commands are broadly grouped into five categories:

<i>action</i>	Alter the configuration in some way.
<i>display</i>	Show current configuration.
<i>cfg</i>	Manage CFG files.
<i>comments</i>	Display help and comments information found in CFG files.
<i>help</i>	Help for using <b>eisa_config</b> interactive commands

The *action* commands are:

<b>add</b> <i>cfgfile slotnum</i>	Adds a board to the current configuration. <i>cfgfile</i> specifies which CFG file corresponds to the board and <i>slotnum</i> identifies the slot where the board resides.
<b>remove</b> <i>slotnum</i>	Remove a board from the current configuration. <i>slotnum</i> identifies the slot where the board currently resides.

<b>move</b> <i>curslotnum newslotnum</i>	Move a board that is currently configured in one slot to a different slot. <i>curslotnum</i> and <i>newslotnum</i> specify the current and new slot numbers, respectively.
--	--

<b>change</b> <i>slotnum functionnum choicenum</i>	Change the choice used for a given function. All three arguments, <i>slotnum</i> , <i>functionnum</i> , and <i>choicenum</i> are required. The function number ( <i>functionnum</i> ) and choice number ( <i>choicenum</i> ) can be obtained by using the <b>show board</b> command on the slot in question. Function numbers are of the format <b>Fnum</b> and choice numbers are of the format <b>CHnum</b> . Note that a board must already be part of the configuration before the change command can be used.
--	--

When **eisa\_config** adds a board, it selects a choice for each function. Generally, the first choice for each function is selected (the default). However, in order to resolve conflicts, **eisa\_config** may select a different choice for a given function. When specifying a choice for a particular function by use of the **change** command, **eisa\_config** always uses that choice; it does not select a different one, even when a conflict needs to be resolved.

<b>save</b> [ <i>filename</i> ]	Save the current configuration. If the current configuration is not conflict-free, a warning is produced and the save is not done. If you specify a file name, the save is done to that file; otherwise, the save is done to NVM (and the <b>/etc/eisa/system.sci</b> file). Note that the <b>quit</b> command also (optionally) saves the configuration to NVM (and file <b>/etc/eisa/system.sci</b> ).
---------------------------------	--

When the configuration is saved to NVM, a log file is created that provides a brief description of the new configuration. The log file is named **/etc/eisa/config.log**, and contains information generated by a **show** command, followed by a **show board** command, followed by a **show switch** command.

<b>init</b> [ <i>filename</i> ]	Initialize the configuration. The initial configuration is retrieved from a file if one has been specified. Otherwise, it is retrieved from NVM. Note that an implicit <b>init</b> is done when <b>eisa_config</b> is first started. This command should only be used when the current configuration <b>eisa_config</b> is dealing with is incorrect. For example, if you make some changes that you decide you do not
---------------------------------	--

want, you can use this command to start over.

**quit**

Leave **eisa\_config**. If the configuration is conflict-free and has been changed, you are asked if you want to save the configuration (to NVM). If any switches or jumpers have to be changed as a result of this new configuration, you are notified of these changes prior to saving the configuration. Be sure that all switches and jumpers match what **eisa\_config** has specified before booting the system.

When the configuration is saved to NVM, a log file is created that provides a brief description of the new configuration. The log file is named **/etc/eisa/config.log**, and contains information generated by a **show** command, followed by a **show board** command, followed by a **show switch** command.

The *show* (display) commands are:

**show** List all slots and their current status; i.e., whether occupied by a particular board, or empty.

**show slots** *cfgfile*

List all of the slots that could accept the board corresponding to the CFG file *cfgfile*.

**show board** [*cfgfile* | *slotnum*]

List the basic attributes for the selected board or boards. Includes a list of all the functions on the board and a list of all available choices for each function. If the board is currently part of the configuration, the currently selected choice is marked. The default choice is the first choice listed for each function. If a board is not specified (either by CFG file name or slot number), information is displayed for each of board installed and configured in the system.

**show switch** [**changed**] [*slotnum*]

List the switch and jumper settings (both default and required) for the boards in the configuration. If the keyword **changed** is used, only those switches and jumpers that were changed from the previous configuration are displayed. If a slot number is specified, only switches and jumpers on the board in that slot are displayed. Note that **show switch** supports all combinations of **changed** and *slotnum*.

There are two kinds of *cfg* commands:

**cfgtypes** List the types of boards that have CFG files in directory **/sbin/lib/eisa** and how many CFG files in **/sbin/lib/eisa** are of each type.

**cfgfiles** [*type*] List all CFG files that are currently available for use in the **/sbin/lib/eisa** directory. If a specific board *type* is specified, only CFG files of that type are displayed.

*comment* commands extract the help and comments text provided in the specified CFG file or files. Both help and comments are displayed if they are available. Each command form accepts as an argument either a CFG file or a slot number identifying which board you want help for.

**comment board** [*cfgfile* | *slotnum*]

Display board-level help and comments.

**comment function** [*cfgfile* | *slotnum*]

Display function-level help and comments.

**comment choice** [*cfgfile* | *slotnum*]

Display choice-level help.

**comment switch** [*cfgfile* | *slotnum*]

Display help and comments for switches and/or jumpers as appropriate.

Note that all arguments (except the type of comments requested) are optional. If no optional argument is specified, all available comments for the specified file or board are extracted. For example:

**comment board** 1

Display help and comments available for the board currently configured in slot 1.

**comment board** Display help and comments available for *all* currently configured boards.

The *help* commands explain how to use the **eisa\_config** interactive commands. If no other arguments are given, help is displayed for all of the interactive commands. Alternatively, any valid command can be used as a argument to the help command. Help is then given for the specified command only.

**help** Display a brief explanation of all valid **eisa\_config** interactive commands.

**help [cmdname]** Display an explanation of the command specified.

## EXAMPLES

Add a new EISA board to the system:

1. Load the CFG file (from media provided by the manufacturer) into directory `/sbin/lib/eisa` if the file is not already present.
2. Run **eisa\_config**. **eisa\_config** reads the contents of NVM to obtain current system configuration.
3. Use the interactive **add** command to add the new board. **eisa\_config** reads the corresponding CFG file to obtain needed configuration information.
4. Exit **eisa\_config**, noting any required switch or jumper settings. **eisa\_config** generates a new configuration and writes it to NVM. The required switch and jumper settings are also saved in the log file `/etc/eisa/config.log`.
5. Add the correct software drivers for the board (and board devices) to the kernel, and use **mknod(1M)** to create any needed device special files.
6. Shut down and disconnect power to the system.
7. Install the EISA board after changing any switch or jumper settings required by **eisa\_config**.
8. Reboot the system. When the system is running again, the contents of NVM will match the EISA boards present in the system, and the newly added board can be used immediately.

This procedure can also be used to add multiple new boards at the same time. Simply use the **add** command once for each board and alter the other steps as appropriate.

If the board to be added is an EISA board that does not have switches or jumpers, the board can be added via automatic mode; that is, steps 2-4 above can be skipped.

## AUTHOR

**eisa\_config** was developed by HP and Compaq.

## FILES

<code>/sbin/lib/eisa/!XXX0000.CFG</code>	CFG files
<code>/etc/eisa/config.err</code>	errors encountered in automatic mode
<code>/etc/eisa/config.log</code>	log file containing current EISA configuration
<code>/etc/eisa/system.sci</code>	mirror image of configuration saved to NVM

## SEE ALSO

**config(1M)**, **mknod(1M)**.

**NAME**

envd - system physical environment daemon

**SYNOPSIS**

```
/usr/sbin/envd [-f configfile]
```

**DESCRIPTION**

The **envd** daemon provides a means for the system to respond to environmental conditions detected by hardware. Such responses are typically designed to maintain file system integrity and prevent data loss. The environmental conditions currently recognized by **envd** are over-temperature and chassis fan failure.

**envd** logs messages and then executes actions when a supported environmental event is detected. Whether to do message logging and what actions to perform for a given environmental event are determined by *configfile* (default is */etc/envd.conf*). If no **-f** option was specified and the default *configfile* */etc/envd.conf* does not exist, **envd** fails. A recommended default *configfile* is available in */usr/newconfig/etc/envd.conf*. The *configfile* (or */etc/envd.conf*) is only examined when the daemon is started or when it receives a **SIGHUP** signal to restart and re-initialize the daemon itself.

**envd** uses the **syslog** message logging facility to log warning messages. If *configfile* specifies messages to be logged, the destination of the warning messages is determined by the configuration of the **LOG\_DAEMON** facility of the **syslogd** daemon (see *syslogd(1M)* and *syslog(3C)* for details) and various **syslog** priorities defined below for the corresponding environmental events. Warning messages are written to the console if **envd** is unable to send to **syslogd**.

The *configfile* is composed of event lines, each of which followed by zero or more action lines. Comment lines can be interspersed at any point. No more than one event line can be specified for a given event.

Event	<p>Event lines consist of an event keyword and a message indicator, separated by a colon (:). Valid event keywords are <b>OVERTEMP_CRIT</b>, <b>OVERTEMP_EMERG</b>, <b>FANFAIL_CRIT</b>, and <b>FANFAIL_EMERG</b>. Valid message indicators are <b>y</b> and <b>n</b>. An example is <b>OVERTEMP_EMERG:y</b>, indicating that warning messages are to be sent for the <b>OVERTEMP_EMERG</b> event.</p> <p>Event keywords must start in the first column, and only one event and one message indicator are allowed on a given line.</p>
Action	<p>Action lines can consist of a sequence of any valid <i>/usr/bin/sh</i> commands or pipe-lines. Lines from one event line to the next event line, or to the end of the file, are part of the action lines for the preceding event, and are passed intact to the shell to execute upon detecting the event. The action for an event can span across several lines, but the syntax of every line must be understood by <i>/usr/bin/sh</i>. There are no default actions for any events if no action lines are specified.</p> <p>No parsing or syntax checking is performed on the action lines; system administrators are responsible for verifying the correctness of the action syntax.</p>
Comments	<p>Lines beginning with the <b>#</b> character in the first column are comment lines, and all characters up to the subsequent new-line character are ignored.</p> <p>Blank lines are ignored as comment lines.</p>

Here is an example */etc/envd.conf* file:

```
# The example below configures envd to log the warning message and
# to rcp critical applications to a remote machine at OVERTEMP_CRIT
# or FANFAIL_CRIT. It configures envd to log emergency messages
# and to perform system shutdown at OVERTEMP_EMERG or FANFAIL_CRIT,
# in order to reserve data integrity.
OVERTEMP_CRIT:y
    /usr/bin/rcp critical_appl_files \
        remote_machine:/backup
OVERTEMP_EMERG:y
    /usr/sbin/reboot -qh
FANFAIL_CRIT:y
    /usr/bin/rcp critical_appl_files \
        remote_machine:/backup
```

```
FANFAIL_EMERG:y
/usr/sbin/reboot -qh
```

Only users with appropriate privileges can invoke **envd**.

**Over-temperature and Fan Failure Handling**

Over-temperature and fan failure handling is supported only on systems equipped with appropriate sensing hardware. Over-temperature and fan failure limits vary, depending on the hardware. Each system processor defines its own thresholds for supported equipment combinations. The table below shows temperature and fan failure states. For the temperature ranges and fan states specific to your system configuration, refer to any of the following documents for your system: *Site Planning and Preparation Guide*, *Installation and Configuration Guide*, or *Operator Handbook*.

State	State Description
NORMAL	Within normal operating temperature range
OVERTEMP_CRIT	Temperature has exceeded the normal operating range of the system, but is still within the operating limit of the hardware media.
OVERTEMP_EMERG	Temperature has exceeded the maximum specified operating limit of hardware media; power loss is imminent. A minimum of about 60 seconds is guaranteed between the OVERTEMP_MID state and the OVERTEMP_POWERLOSS (power loss) state.
OVERTEMP_POWERLOSS	Hardware will disconnect all power from all cards in the system chassis.
FAN_NORMAL	All chassis fans are operating normally.
FANFAIL_CRIT	One or more chassis fans have failed, but the system has enough redundant fans to allow continued operation while the failed fans are replaced.
FANFAIL_EMERG	Chassis fan failures prevent continued operation of the system; power loss is imminent.
FANFAIL_POWERLOSS	Hardware will disconnect all power from all cards in the system chassis.

The **syslog** priorities mapped to the environmental events are: **LOG\_EMERG** (for **OVERTEMP\_EMERG** and **FANFAIL\_EMERG**) and **LOG\_CRIT** (for **OVERTEMP\_CRIT** and **FANFAIL\_CRIT**).

Any non-shutdown activities (e.g. file transfer) should be performed at **OVERTEMP\_CRIT** and **FANFAIL\_CRIT**. It is important to configure only critical activities for **OVERTEMP\_CRIT** because the over-temperature might rise dramatically fast to **OVERTEMP\_EMERG**. It is recommended to perform a quick shutdown using **/usr/sbin/reboot -qh** at **OVERTEMP\_EMERG** and **FANFAIL\_EMERG** to preserve file system data integrity. If the hardware enters the **OVERTEMP\_POWERLOSS** or **FANFAIL\_POWERLOSS** state and the system has not been shut down, the sudden loss of power could result in data loss. Note that power-fail recovery functionality is not available in this case. When the hardware powers down, no warning messages are produced, and no action is taken by the system.

Whenever an environmental state changes from one level to another (such as from **NORMAL** to **OVERTEMP\_CRIT** or from **FANFAIL\_CRIT** to **FANFAIL\_EMERG**), the warning message, if specified, is logged, and the corresponding action is executed once, and only once, per state change.

**AUTHOR**

**envd** was developed by HP.

**FILES**

/usr/sbin/envd	<b>envd</b> executable file
/etc/envd.conf	default <b>envd</b> configuration file
/etc/syslog.conf	default <b>syslog</b> configuration file

`/var/tmp/envd.action[123]`      `envd` work files

**SEE ALSO**

`reboot(1M)`, `shutdown(1M)`, `syslogd(1M)`, `syslog(3C)`.

HP-UX System Administration manuals.

e



**NAME**

exportfs - export and unexport directories to NFS clients

**SYNOPSIS**

```
/usr/sbin/exportfs [-auv]
/usr/sbin/exportfs [-uv] [dir ...]
/usr/sbin/exportfs -i [-o options] [-v] [dir ...]
```

**DESCRIPTION**

The **exportfs** command makes a local directory or file available to NFS clients for mounting over the network. Directories and files cannot be NFS-mounted unless they are first exported by **exportfs**.

**exportfs** is normally invoked at boot time by the `/sbin/init.d/nfs.server` script, and uses information contained in the `/etc/exports` file to export the file or file system named by each *dir*, which must be specified as a full path name.

If no options or arguments are specified in the command line, **exportfs** displays a list of the currently exported directories and files on standard output.

A superuser can run **exportfs** at any time to alter the list or characteristics of exported directories and files.

**Options**

**exportfs** recognizes the following options:

- a Export all directories listed in `/etc/exports`. If **-u** is also specified, unexport all of the currently exported directories.
- i Ignore the options in `/etc/exports`. Normally, **exportfs** consults `/etc/exports` for the options associated with the exported directory.
- u Unexport the indicated directories.
- v Verbose. Print each directory or file name as it is exported or unexported.
- o *options*  
Specify a comma-separated list of optional characteristics for the directory being exported. The list of *options* can include any of the following:

**async**

All NFS Protocol Version 2 mounts will be asynchronous. This option is ignored for NFS PV3. Refer to *exports(4)* for warnings when using this option.

- ro** Export the directory read-only. If not specified, the directory is exported read-write. The **ro** and **rw** options cannot be used on the same **exportfs** command line.

**rw=hostname[:hostname]...**

Export the directory read-mostly. Read-mostly means read-only to most machines, but read-write to those specified. If neither **ro** nor **rw** is specified, the directory is exported read-write to all. The **ro** and **rw** options cannot be used on the same **exportfs** command line. Up to 256 *hostnames* can be specified. With a server configured for DNS naming in the `nsswitch` "hosts" entry, any hostname must be represented as a fully qualified DNS name. Currently HP-UX will attempt to match a non-fully qualified hostname; this HP-only feature will be obsoleted in a later release of HP-UX.

**anon=uid**

If a request comes from an unknown user, use *uid* as the effective user ID.

Root users (user ID 0) are always treated as user **unknown** by the NFS server unless they are included in the **root** option below.

If the client is a UNIX system, only root users are considered **unknown**. All other users are recognized even if they are not in `/etc/passwd`.

The default value for *uid* is the user ID of user **nobody**. If user **nobody** does not exist, the value `-2` is used. Setting the value of **anon** to `-1` disables anonymous access.

**root=hostname[:hostname]...**

Give root access only to the root users from a specified *hostname*. The default is for no

hosts to be granted root access. Up to 256 *hostnames* can be specified. *hostnames* on this list are not guaranteed to successfully mount the specified file system. If a non-empty access list is specified, the *hostname* must also meet one of the *access\_list* criteria for **access=** or be on the **rw=** list. With a server configured for DNS naming in the *nsswitch* "hosts" entry, any hostname must be represented as a fully qualified DNS name. Currently HP-UX will attempt to match a non-fully qualified hostname; this HP-only feature will be obsoleted in a later release of HP-UX.

**access=**[*access\_list*][:*access\_list*]...

Give mount access to each *access\_list* listed. See the "access\_list" subsection below. An empty **access=** list allows all machines to mount the specified mount point. *hostnames* on the **rw=** list do not have to exist on the access list in order to successfully mount the exported file system. *hostnames* on the **root=** list must either appear on the **rw=** list or **access=** list in order to successfully mount the file system.

*access\_list*

The *access\_list* argument is a colon-separated list whose components may be one or more of the following:

*hostname*

The name of a host. With a server configured for DNS naming in the *nsswitch* "hosts" entry, any hostname must be represented as a fully qualified DNS name. Currently HP-UX will allow a match for a non-fully qualified hostname; this HP-only feature will be obsoleted in a later release of HP-UX.

*netgroup*

A netgroup contains a number of hostnames. With a server configured for DNS naming in the *nsswitch* "hosts" entry, any hostname in a netgroup must be represented as a fully qualified DNS name.

*DNS suffix*

To use domain membership, the server must use DNS to resolve hostnames to IP addresses. That is, the "hosts" entry in the */etc/nsswitch.conf* file must specify "dns" ahead of "nis" or "nisplus", since only DNS returns the full domain name of the host. Other name services like NIS or NIS+ cannot be used to resolve hostnames on the server, because when mapping an IP address to a hostname, they do not return domain information. For example,

NIS or NIS+

```
129.144.45.9 --> "myhost"
```

DNS

```
129.144.45.9 --> "myhost.myd.myc.com"
```

The DNS suffix is distinguished from hostnames and netgroups by a prefixed dot. A dot by itself will match "myhost" but not "myhost.myd.myc.com". This single dot feature can be used to match hosts resolved from NIS and NIS+ rather than DNS.

*network*

The network or subnet component is preceded by an at-sign (@). It can be either a name or a dotted address. If a name, it will be converted to a dotted address by *getnetbyname* (see *getnetent*(3N)). Entries in */etc/networks* must contain all four octets in order to be valid.

The network prefix assumes an octet aligned netmask determined from the zero octets in the low order part of the address. In the case where network prefixes are not byte-aligned, the syntax will allow a mask length to be specified explicitly following a slash (/) delimiter. The mask is the number of leftmost contiguous significant bits in the corresponding IP address.

- A prefixed minus sign (-) denies access to that component of *access\_list*. The list is searched sequentially until a match is found that either grants or denies access, or until the end of the list is reached. This option is valid only in conjunction with hostname, network and DNS Suffix. If prefixing a hostname and you are configured for DNS naming, you must fully qualify the hostname.

**DIAGNOSTICS**

If an NFS-mounted directory is unexported by **exportfs**, any access by the client to the directory causes an **NFS stale file handle** error. However, if **exportfs** is used to remove a client from the access list of an exported directory, an **NFS stale file handle** error does not result from any access by the client to the directory.

**EXAMPLES**

The following invocation of **exportfs** lists currently exported directories and files:

```
exportfs
```

Export entries in **/etc/exports**:

```
exportfs -a
```

Unexport all exported files and directories:

```
exportfs -ua
```

Unexport all exported files and directories and print each directory or file name as it is unexported:

```
exportfs -uav
```

Export **/usr** to the world, ignoring options in **/etc/exports**:

```
exportfs -i /usr
```

or

```
exportfs -i -o access= /usr
```

Export **/usr/bin** and **/var/adm** read-only to the world:

```
exportfs -i -o ro /usr/bin /var/adm
```

Export **/usr/bin** read-write only to systems, **polk** and **vanness**, when using DNS as the name service:

```
exportfs -i -o rw=polk.myd.myc.com:vanness.myd.myc.com /usr/bin
```

Export **/usr/bin** read-write only to systems, **polk** and **vanness**, when using NIS or NIS+ as the name services:

```
exportfs -i -o rw=polk:vanness /usr/bin
```

Export root access on **/var/adm** only to the system named **pine**, and mount access to both **pine** and **geary** when using DNS as the name services:

```
exportfs -i -o \  
root=pine.myd.myc.com,access=pine.myd.myc.com:geary.myd.myc.com \  
/var/adm
```

Export access to **/var/adm** for all hosts in the **myd.myc.com** domain.

```
exportfs -i -o access=.myd.myc.com /var/adm
```

Export access to **/var/adm** for all hosts in the same NIS domain, but deny access to all hosts in the DNS name space:

```
exportfs -i -o access=. /var/adm
```

Export access to **/var/adm** using a network submask that is a dotted address:

```
exportfs -i -o access=@192.144 /var/adm
```

or

```
exportfs -i -o access=@192.144.0.0 /var/adm
```

or using a name where **mynetwork** is defined in **/etc/networks** as follows:

```
mynetwork 192.144.0.0 mount_144 #allow mounts using this mask
```

```
exportfs -i -o access=@mynetwork /var/adm
```

Export access to **/var/adm** where the network prefixes are not byte aligned:

```
exportfs -i -o access=@192.144.132/17 /var/adm
```

or

```
exportfs -i -o access=@mynetwork/17 /var/adm
```

Export access to `/var/adm` where the hostname, `terra`, in the netgroup engineering is denied access:

```
exportfs -i -o access=-terra:engineering /var/adm
```

Export access to `/var/adm` where the hostname, `terra`, is granted access because it is part of the netgroup engineering.

```
exportfs -i -o access=engineering:-terra /var/adm
```

## WARNINGS

e

You cannot export a directory that resides within the same file system and is either a parent or sub-directory of a directory that is currently exported. For example, `/usr` and `/usr/local` cannot both be exported if they reside in the same disk partition.

If you unexport a directory, remove a client from the access list, then export again, the client still has access to the directory until the client unmounts the directory. Removing a client from the `root` or `rw` list takes effect immediately.

`/etc/xtab` is a system file that contains a list of currently exported directories and files. This file is maintained by `exportfs`. To ensure that this file is always synchronous with current system data structures, do not attempt to edit `/etc/xtab` by hand.

## FILES

<code>/etc/exports</code>	Static export information
<code>/etc/hosts</code>	List of hostnames
<code>/etc/netgroup</code>	List of network groups
<code>/etc/xtab</code>	Current state of exported directories
<code>/etc/networks</code>	Network information

## SEE ALSO

`showmount(1M)`, `exports(4)`, `hosts(4)`, `netgroup(4)`, `networks(4)`.

**NAME**

extendfs (generic) - extend a file system size

**SYNOPSIS**

```
/usr/sbin/extendfs [-F FStype] [-q] [-v] [-s size] special
```

**DESCRIPTION**

If the original file system image created on *special* does not make use of all of the available space, **extendfs** can be used to increase the capacity of a file system by updating the file system structure to include the extra space.

The command-line parameter *special* specifies the device special file of either a logical volume or a disk partition. The *special* must be un-mounted before **extendfs** can be run (see *mount(1M)*).

**Options**

**extendfs** recognizes the following options:

**-F FStype**

Specify the file system type on which to operate (see *fstyp(1M)* and *fs\_wrapper(5)*). If this option is not included on the command line, then the file system type is determined from the file `/etc/default/fs`.

**-q**

Query the size of *special*. No file system extension will be done.

**-v**

Verbose flag.

**-s size**

Specifies the number of **DEV\_BSIZE** blocks to be added to the file system. If *size* is not specified, the maximum possible size is used.

**EXAMPLES**

To increase the capacity of a file system created on a logical volume, enter:

```
umount /dev/vg00/lvol1
lvextend -L larger_size /dev/vg00/lvol1
extendfs -F hfs /dev/vg00/rlvol1
mount /dev/vg00/lvol1 mount_directory
```

**SEE ALSO**

*fstyp(1M)*, *lvextend(1M)*, *mkfs(1M)*, *mount(1M)*, *umount(1M)*, *fs(4)*, *fs\_wrapper(5)*.

**NAME**

extendfs (hfs) - extend an HFS file system size

**SYNOPSIS**

```
/usr/sbin/extendfs [-F hfs] [-q] [-v] [-s size] special
```

**DESCRIPTION**

If the original HFS file system image created on *special* does not make use of all of the available space, the **extendfs** command can be used to increase the capacity of an HFS file system by updating the file system structure to include the extra space.

The command-line parameter *special* specifies the character device special file of either a logical volume or a disk partition. The *special* must be unmounted before the **extendfs** command can be run (see *mount(1M)*).

**Options**

**extendfs** recognizes the following options:

- F hfs Specify the HFS file system type.
- q Query the size of *special*. No file system extension will be done.
- v Verbose flag.
- s size Specifies the number of **DEV\_BSIZE** blocks to be added to the file system. If the number of blocks is not specified, the maximum possible size is used.

**EXAMPLES**

To increase the capacity of a file system created on a logical volume, enter:

```
umount /dev/vg00/lvol1
lvextend -L larger_size /dev/vg00/lvol1
extendfs -F hfs /dev/vg00/rlvol1
mount /dev/vg00/lvol1 mount_directory
```

**WARNINGS**

The root file system cannot be extended using the **extendfs** command because the root file system is always mounted, and the **extendfs** command only works on unmounted file systems.

**extendfs** will fail if used on a file system, on a logical volume, where the logical block size of the logical volume is greater than the file system's fragment size. The logical block size, of a logical volume changes, when additional disks with larger sector size are added.

**RETURN VALUE**

**extendfs** returns the following values:

- 0 No errors were detected and file system was successfully extended.
- 1 Command aborted.

**SEE ALSO**

extendfs(1M), lvextend(1M), mkfs(1M), mount(1M), umount(1M), fs(4).

**NAME**

extendfs (vxfs) - extend a VxFS file system size

**SYNOPSIS**

```
/usr/sbin/extendfs [-F vxfs] [-q] [-v] [-s size] special
```

**DESCRIPTION**

If the VxFS file system image created on *special* does not use all of the available space, **extendfs** increases the capacity of a VxFS file system by updating the file system structure to include the extra space.

*special* specifies the device special file of either a logical volume or a disk partition. If *special* refers to a mounted file system, you must unmount *special* before running **extendfs** (see *mount(1M)*).

**Options**

**extendfs** recognizes the following options:

- F vxfs** Specify the VxFS file system type.
- q** Query *special* to determine the size. The file system is not extended.
- v** Specify verbose mode, which displays the resulting size of file system along with **fsck** output of the newly sized file system. Without **-v** there is no output.
- s size** Specify the number of **DEV\_BSIZE** blocks to add to the file system. If *size* is omitted, the maximum possible size is used.

**EXAMPLES**

This example shows how to increase the capacity of a file system created on a logical volume.

```
umount /dev/vg00/lvol1
lvextend -L larger_size /dev/vg00/lvol1
extendfs -F vxfs /dev/vg00/rlvol1
mount /dev/vg00/lvol1 mount_directory
```

**SEE ALSO**

extendfs(1M), lvextend(1M), mkfs(1M), mount(1M), umount(1M), fs(4).

## NAME

fbackup - selectively back up files

## SYNOPSIS

```
/usr/sbin/fbackup -f device [-f device] ... [-0-9] [-nsuvyAEl] [-i path] [-e path]
[-g graph] [-d path] [-I path] [-V path] [-c config]
```

```
/usr/sbin/fbackup -f device [-f device] ... [-R restart] [-nsuvyAEl] [-d path] [-I path]
[-V path] [-c config]
```

## DESCRIPTION

**fbackup** combines features of **dump** and **ftio** to provide a flexible, high-speed file system backup mechanism (see *dump(1M)* and *ftio(1)*). **fbackup** selectively transfers files to an output device. For each file transferred, the file's contents and all the relevant information necessary to restore it to an equivalent state are copied to the output device. The output device can be a raw magnetic tape drive (for example, a DLT tape drive), the standard output, a rewritable magneto-optical disk, or a file.

The selection of files to back up is done by explicitly specifying trees of files to be included or excluded from an **fbackup** session. The user can construct an arbitrary graph of files by using the **-i** or **-e** options on the command line, or by using the **-g** option with a graph file. For backups being done on a regular basis, the **-g** option provides an easier interface for controlling the backup graph. **fbackup** selects files in this graph, and attempts to transfer them to the output device. The selectivity depends on the mode (full or incremental) in which **fbackup** is being used.

When doing full backups, all files in the graph are selected. When doing incremental backups, only files in the graph that have been modified since a previous backup of that graph are selected. If an incremental backup is being done at level 4 and the **-g** option is used, the database file is searched for the most recent previous backup at levels 0-3. If a file's modification time is before the time when the last appropriate session began and the i-node change time is before the time that same session ended, the file is not backed up. All directories lying on the path to a file that qualifies for the incremental backup will also be on the backup media, even if the directories do not qualify on their own status.

If **fbackup** is used for incremental backups, a database of past backups must be kept. **fbackup** maintains this data in the text file **/var/adm/fbackupfiles/dates**, by default. Note that the directory **/var/adm/fbackupfiles** must be created prior to the first time **fbackup** is used for incremental backups. The **-d** option can be used to specify an alternate database file. The user can specify to update this file when an **fbackup** session completes successfully. Entries for each session are recorded on separate pairs of lines. The following four items appear on the first line of each pair: the graph file name, backup level, starting time, and ending time (both in *time(2)* format). The second line of each pair contains the same two times, but in *strtime(3C)* format. These lines contain the local equivalent of **STARTED:**, the start time, the local equivalent of **ENDED:**, and the ending time. These second lines serve only to make the dates file more readable; **fbackup** does not use them. All fields are separated by white space. Graph file names are compared character-by-character when checking the previous-backup database file to ascertain when a previous session was run for that graph. Caution must be exercised to ensure that, for example, **graph** and **./graph** are not used to specify the same graph file because **fbackup** treats them as two different graph files.

The general structure of an **fbackup** volume is the same, no matter what type of device is used. There are some small specific differences due to differing capabilities of devices. The general structure is as follows:

- reserved space for ASCII tape label (1024 bytes)
- **fbackup** volume header (2048 bytes)
- session index (size in field of volume header)
- data

Each file entry in the index contains the file size, the volume number and the pathname of the file. At the beginning of every volume, **fbackup** assumes that all files not already backed up will fit on that volume, an erroneous assumption for all but the last volume. Indices are accurate only for the previous volumes in the same set. Hence, the index on the last volume may indicate that a file resides on that volume, but it may not have actually been backed up (for example, if it was removed after the index was created, but before **fbackup** attempted to back it up). The only index guaranteed to be correct in all cases is the on-line index (**-I** option), which is produced after the last volume has been written.

Specific differences in the structure of **fbackup** volumes are listed below:



- When using magnetic tape devices, the main blocks of information (tape label, volume header, index, data) are separated by EOF marks. **fbackup** also checkpoints the media periodically to enhance error recovery. If a write error is detected, the user normally has two options: (1) a new volume can be mounted and that volume rewritten from the beginning; or, (2) if the volume is not too severely damaged, the good data before the error can be saved, and the write error is treated as a normal end-of-media condition. The blocks of data with their checkpoint records are also separated by EOF marks. In addition, for DDS tape drives, if **fast search marks** are supported, these will be used to enhance selective recovery speed by placing them between blocks of files. Similarly on DLT tape drives, faster selective recovery is achieved using the EOF marks used for checkpointing in conjunction with the file sizes given in the index.
- For a magneto-optical device, a disk, a file, or standard output, there are no special marks separating the information pieces; the backup is always a single file (volume).

**fbackup** provides the ability to use UCB-mode tape drives. This makes it possible to overlap the tape rewind times if two or more tape drives are connected to the system.

### Set-up

There are several things the user will want to consider when setting up **fbackup** for regular use. These include type of device and media, full versus incremental frequency, amount of logging information to keep on-line, structure of the graph file, and on-line versus off-line backup.

The type of device used for backups can affect such things as media expenses, ability to do unattended backups, and speed of the backup. Using 36-track tapes will probably result in the highest performance, but require user intervention for changing tapes. Both DLT and DDS autochangers and libraries can provide unattended backups. A magneto-optical autochanger can also provide an unattended backup for a large system and long life media, however the media cost is high. Lower cost and good performance can be achieved with a single DLT tape drive, but multi-volume backups must be attended.

It is also important to consider how often full backups should be made, and how many incremental backups to make between full backups. Time periods can be used, such as a full backup every Friday and incrementals on all other days. Media capacities can be used if incremental backups need to run unattended. The availability of personnel to change media can also be an important factor as well as the length of time needed for the backup. Other factors may affect the need for full and incremental backup combinations such as contractual or legal requirements.

If backup information (output from the **-V** or **-I** options) is kept on-line, the required storage space must also be considered. Index file sizes are hard to predict in advance because they depend on system configuration. Each volume header file takes less than 1536 bytes. Of course the more information that is kept on-line, the faster locating a backup media for a recovery will be.

There are several ways to structure the graph file or files used in a system backup. The first decision involves whether to use one or more than one graph file for the backup. Using one file is simpler, but less flexible. Using two or more graph files simplifies splitting backups into logical sets. For example, one graph file can be used for system disks where changes tend to be less frequent, and another graph file for the users area. Thus two different policies can be implemented for full and incremental backups.

**fbackup** was designed to allow backups while the system is in use by providing the capability to retry an active file. When absolute consistency on a full backup is important, the system should probably be in single-user mode. However, incremental backups can be made while the system is in normal use, thus improving system up-time.

### Options

**-c config** *config* is the name of the configuration file, and can contain values for the following parameters:

- Number of 1024-byte blocks per record.
- Number of records of shared memory to allocate.
- Number of records between checkpoints. Since the EOF marks between checkpoints are also used for fast searching on DLT tape drives, changing the checkpoint frequency may also affect selective recovery speed (see WARNINGS section).
- Number of file-reader processes.
- Maximum number of times **fbackup** is to retry an active file.
- Maximum number of bytes of media to use while retrying the backup of an active file.

- Maximum number of times a magnetic tape volume can be used.
- Name of a file to be executed when a volume change occurs. This file must exist and be executable.
- Name of a file to be executed when a fatal error occurs. This file must exist and be executable.
- The number of files between the **fast search marks** on DDS tapes. The cost of these marks are negligible in terms of space on the DDS tape. Not all DDS tape devices support **fast search marks**.

Each entry in the configuration file consists of one line of text in the following format: identifier, white space, argument. In the following sample configuration file, the number of blocks per record is set to 16; the number of shared memory records is set to 16; the checkpoint frequency is set to 256; the number of file reader processes is set to 2; the maximum number of retries of an active file is set to 5; the maximum retry space for active files is set to 5,000,000 bytes; the maximum number of times a magnetic tape volume can be used is set to 100; the file to be executed at volume change time is `/var/adm/fbackupfiles/chgvol`; the file to be executed when a fatal error occurs is `/var/adm/fbackupfiles/error`; and the number of files between **fast search marks** on DDS tapes is set to 200.

```

blocksperrecord  16
records          16
checkpointfreq   256
readerprocesses  2 (maximum of 6)
maxretries       5
retrylimit       5000000
maxvoluses       100
chgvol           /var/adm/fbackupfiles/chgvol
error            /var/adm/fbackupfiles/error
filesperfsm      200

```

Each value listed is also the default value, except **chgvol** and **error**, which default to null values.

- d path** This specifies a path to a database for use with incremental backups. It overrides the default database file `/var/adm/fbackupfiles/dates`.
- e path** *path* specifies a tree to be excluded from the backup graph. This tree must be a subtree of part of the backup graph. Otherwise, specifying it will not exclude any files from the graph. There is no limit on how many times the **-e** option can be specified.
- f device** *device* specifies the name of an output file. If the name of the file is `-`, **fbackup** writes to the standard output. There is no default output file; at least one must be specified. If more than one output file is specified, **fbackup** uses each one successively and then repeats in a cyclical pattern. Patterns can be used in the device name in a manner resembling file name expansion as done by the shell (see *sh-bourne*(1) and other shell manual entries). The patterns must be protected from expansion by the shell by quoting them. The expansion of the pattern results in all matching names being in the list of devices used.

There is slightly different behavior if remote devices are used. A device on the remote machine can be specified in the form *machine:device*. **fbackup** creates a server process from `/usr/sbin/rmt` on the remote machine to access the tape device. If `/usr/sbin/rmt` does not exist on the remote system, **fbackup** creates a server process from `/etc/rmt` on the remote machine to access the tape device. Only magnetic tapes can be remote devices. When remote DDS tape devices are used, the **fast search marks** capability is not used.

- g graph** *graph* defines the graph file. The graph file is a text file containing the list of file names of trees to be included or excluded from the backup graph. These trees are interpreted in the same manner as when they are specified with the **-i** and **-e** options. Graph file entries consist of a line beginning with either **i** or **e**, followed by white space, and then the path name of a tree. Lines not beginning with **i** or **e** are treated as an error. There is no default graph file. For example, to back up all of `/usr` except for the subtree `/usr/lib`, a file could be created with the following two records:

- ```

i /usr
e /usr/lib

```
- i path** *path* specifies a tree to be included in the backup graph. There is no limit on how many times the **-i** option can be specified.
- n** Cross NFS mount points. By default, **fbackup** does not cross NFS mount points, regardless of paths specified by the **-i** or **-g** options.
- l** Includes LOFS files specified by the backup graph. By default, **fbackup** does not cross LOFS mount points. If **-l** is specified, and the backup graph includes files which are also in an LOFS directory that is in the backup graph, then those files will be backed up twice.
- s** Back up the object that a symbolic link refers to. The default behavior is to back up the symbolic link.
- u** Update the database of past backups so that it contains the backup level, the time of the beginning and end of the session, and the graph file used for this **fbackup** session. For this update to take place, the following conditions must exist: Neither the **-i** nor the **-e** option can be used; the **-g** option must be specified exactly once (see below); the **fbackup** must complete successfully.
- v** Run in verbose mode. Generates status messages that are otherwise not seen.
- y** Automatically answer **yes** to any inquiries.
- A** Do not back up optional entries of access control lists (ACLs) for files. Normally, all mode information is backed up including the optional ACL entries. With the **-A** option, the summary mode information (as returned by **stat( )**) is backed up. Use this option when backing up files from a system that contains ACLs to be recovered on a system that does not understand ACLs (see *acl(5)*).
- E** Do not back up extent attributes. Normally, all extent attributes that have been set are included with the file. This option only applies to file systems which support extent attributes.
- I path** *path* specifies the name of the on-line index file to be generated. It consists of one line for each file backed up during the session. Each line contains the file size, the volume number on which that file resides, and the file name. If the **-I** option is omitted, no index file is generated.
- V path** The volume header information is written to *path* at the end of a successful **fbackup** session. The following fields from the header are written in the format *label:value* with one pair per line.
- |                               |                                                                                                                                                                                        |
|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Magic Field</b>            | On a valid <b>fbackup</b> media it contains the value <b>FBACKUP_LABEL</b> (HP-UX release 10.20 and beyond). Before HP-UX release 10.20, it contained the value <b>FBACKUP LABEL</b> . |
| <b>Machine Identification</b> | This field contains the result of <b>uname -m</b> .                                                                                                                                    |
| <b>System Identification</b>  | This field contains the result of <b>uname -s</b> .                                                                                                                                    |
| <b>Release Identification</b> | This field contains the result of <b>uname -r</b> .                                                                                                                                    |
| <b>Node Identification</b>    | This field contains the result of <b>uname -n</b> .                                                                                                                                    |
| <b>User Identification</b>    | This field contains the result of <b>cuserid( )</b> (see <i>cuserid(3S)</i> ).                                                                                                         |
| <b>Record Size</b>            | This field contains the maximum length in bytes of a data record.                                                                                                                      |
| <b>Time</b>                   | This field contains the clock time when <b>fbackup</b> was started.                                                                                                                    |
| <b>Media Use</b>              | This field contains the number of times the media has been used for backup. Since the information is actually on the media, this field will always contain the value 0.                |
| <b>Volume Number</b>          | This field contains a # character followed by 3 digits, and identifies the number of volumes in the backup.                                                                            |
| <b>Checkpoint Frequency</b>   | This field contains the number of data records between checkpoints.                                                                                                                    |

|                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                                                                               |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|
|                   | <b>Index Size</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | This field contains the size of the index.                                                    |
|                   | <b>Backup Identification Tag</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | This field is composed of two items: the process ID (pid) and the start time of that process. |
|                   | <b>Language</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | This field contains the language used to make the backup.                                     |
| <b>-R restart</b> | Restart an <b>fbackup</b> session from where it was previously interrupted. The <i>restart</i> file contains all the information necessary to restart the interrupted session. None of the <b>-[ieg0-9]</b> options can be used together with the restart option.                                                                                                                                                                                                                                              |                                                                                               |
| <b>-0-9</b>       | This single-digit number is the backup level. Level 0 indicates a full backup. Higher levels are generally used to perform incremental backups. When doing an incremental backup of a particular graph at a particular level, the database of past backups is searched to find the date of the most recent backup of the same graph that was done at a lower level. If no such entry is found, the beginning of time is assumed. All files in the graph that have been modified since this date are backed up. |                                                                                               |

### Access Control Lists (ACLs)

If a file has optional ACL entries, the **-A** option is required to enable its recovery on a system where the ACL capability is not present.

## EXTERNAL INFLUENCES

### Environment Variables

**LC\_COLLATE** determines the order in which files are stored on the backup device and the order of output by the **-I** option.

**LC\_TIME** determines the format and contents of date and time strings.

**LC\_MESSAGES** determines the language in which messages are displayed.

If **LC\_COLLATE**, **LC\_TIME**, and **LC\_MESSAGES** are not all specified in the environment, or if either is set to the empty string, the value of **LANG** is used as a default for each unspecified or empty variable. If **LANG** is not specified or is set to the empty string, a default of "C" (see *lang(5)*) is used instead of **LANG**. If any internationalization variable contains an invalid setting, **fbackup** behaves as if all internationalization variables are set to "C". See *environ(5)*.

### International Code Set Support

Single- and multi-byte character code sets are supported.

## RETURN VALUE

**fbackup** returns one of the following values:

- 0 upon normal completion.
- 1 if it is interrupted but allowed to save its state for possible restart.
- 2 if any error conditions prevent the session from completing.
- 4 if any warning conditions are encountered.

If warnings occur, the operator should check the **fbackup** logs to verify the sanity of the backup.

## EXAMPLES

In the following two examples, assume the graph of interest specifies all of **/usr** except **/usr/lib** (as described for the **-g** option above).

The first example is a simple case where a full backup is done but the database file is not updated. This can be invoked as follows:

```
/usr/sbin/fbackup -0i /usr -e /usr/lib -f /dev/rmt/c0t0d0BEST
```

The second example is more complicated, and assumes the user wants to maintain a database of past **fbackup** sessions so that incremental backups are possible.

If sufficient on-line storage is available, it may be desirable to keep several of the most recent index files on disk. This eliminates the need to recover the index from the backup media to determine if the files to be recovered are on that set. One method of maintaining on-line index files is outlined below. The system administrator must do the following once before **fbackup** is run for the first time (creating intermediate level directories where necessary):

- Create a suitable configuration file called **config** in the directory **/var/adm/fbackupfiles**.
- Create a graph file called **usr-usrlib** in the directory **/var/adm/fbackupfiles/graphs**.
- Create a directory called **usr-usrlib** in the directory **/var/adm/fbackupfiles/indices**.

A shell script that performs the following tasks could be run for each **fbackup** session:

- Build an index file path name based on both the graph file used (passed as a parameter to the script) and the start time of the session (obtained from the system). For example:

```
/var/adm/fbackupfiles/indices/usr-usrlib/871128.15:17
(for Nov 28, 1987 at 3:17 PM)
```

- Invoke **fbackup** with this path name as its index file name. For example:

```
cd /var/adm/fbackupfiles
/usr/sbin/fbackup -0uc config -g graphs/usr-usrlib\
-I indices/usr-usrlib/871128.15:17\
-f /dev/rmt/c0t0d0BEST
```

When the session completes successfully, the index is automatically placed in the proper location.

## WARNINGS

**fbackup** consists of multiple executable objects, all of which are expected to reside in directory **/usr/sbin**.

**fbackup** does not require special privileges. However, if the user does not have access to a given file, the file is not backed up.

For security reasons, configuration files and the **chgvol** and **error** executable files should only be writable by their owners.

With release 10.20, HP-UX supports large files (greater than 2GB) and increased UID/GIDs (greater than 60,000). Archives containing files with these attributes would cause severe problems on systems that do not support the increased sizes. For this reason, **fbackup** creates tapes with a new magic number ("FBACKUP\_LABEL"). This prevents **fbackup** tape archives from being restored on pre-10.20 HP-UX systems. **frecover** still reads both tape formats so that **fbackup** tape archives created on pre-10.20 HP-UX systems can be restored.

EOF marks are used for checkpointing on all magnetic tape devices. On DLT tape devices, these EOF marks are also used for fast searching on a selective recovery; "fast searching" in this case means spacing to the nearest checkpoint before the desired file, and then reading until the file is found. With this dual purpose for checkpoints, caution should be used when changing the checkpoint frequency parameter.

Starting with HP-UX Release 8.0, **fbackup** does not back up network special files because RFA networking is obsolete. A warning message is issued if a network special file is encountered in the backup graph and the file is skipped.

The use of **fbackup** for backing up NFS mounted file systems is not guaranteed to work as expected if the backup is done as a privileged user. This is due to the manner in which NFS handles privileged-user access by mapping user **root** and uid 0 to user **nobody**, usually uid -2, thus disallowing root privileges on the remote system to a root user on the local system.

The utility set comprised of **fbackup** and **frecover** was originally designed for use on systems equipped with not more than one gigabyte of total file system storage. Although the utilities have no programming limitations that restrict users to this size, complete backups and recoveries of substantially larger systems can cause a large amount of system activity due to the amount of virtual memory (swap space) used to store the indices. Users who want to use these utilities, but are noticing poor system-wide performance due to the size of the backup, are encouraged to back up their systems in multiple smaller sessions, rather than attempting to back up the entire system at one time.

Due to present file-system limitations, files whose inode data, but not their contents, are modified while a backup is in progress might be omitted from the next incremental backup of the same graph. Also, **fbackup** does not reset the inode change times of files to their original values.

**fbackup** should not be used with no-rewind devices, for example, **/dev/rmt/0mn**.

**fbackup** allocates resources that are not returned to the system if it is killed in an ungraceful manner. If it is necessary to kill **fbackup**, send it a **SIGTERM**, not a **SIGKILL**.

If sparse files are backed up without using data compression, a very large amount of media can be consumed.

**fbackup** creates volumes with a format that makes duplication of volumes by **dd** impossible (see *dd(1)*). Copying an **fbackup** volume created on one media type to another media type does not produce a valid **fbackup** volume on the new media because the formats of volumes on raw magnetic tape, on a regular file, and on rewritable optical disks are not identical.

When configuring the parameter **blockssperrecord** (see **-c** option), the record size is limited by the maximum allowed for the tape drive. Common record sizes include 128 blocks for DLT and DDS tape drives, and 60 blocks for the HP 7980. Note also that the *blocksize* used in earlier releases (7.0 and before) was 512 bytes, whereas it is now 1024 bytes. This means that the same value specified in *blockssperrecord* in an earlier release creates blocks twice their earlier size in the current release; for example, a *blockssperrecord* parameter of 32 would create 16-Kbyte blocks at Release 7.0, but now creates 32-Kbyte blocks. If *blockssperrecord* exceeds the byte count allowed by the tape drive, the tape drive rejects the write, causing an error to be communicated to **fbackup** which **fbackup** interprets as a bad tape. The resulting write error message resembles the following:

```
fbackup (3013): Write error while writing backup at tape block 0.
Diagnostic error from tape 11..... SW_PROBLEM (printed by driver on console)
fbackup (3102): Attempting to make this volume salvageable.
etc.
```

## DEPENDENCIES

### NFS

Access control lists of networked files are summarized (as returned in **st\_mode** by **stat()**), but not copied to the new file (see *stat(2)*).

**fbackup** does not support QIC-120 and QIC-150 formats on QIC devices. If **fbackup** is attempted for these formats, **fbackup** fails and the following message is displayed:

```
mt lu X: Write must be a multiple of 512 bytes in QIC 120 or QIC 150
```

## AUTHOR

**fbackup** was developed by HP.

## FILES

```
/var/adm/fbackupfiles/dates      database of past backups
```

## SEE ALSO

*cpio(1)*, *ftio(1)*, *dump(1M)*, *frecover(1M)*, *restore(1M)*, *rmt(1M)*, *stat(2)*, *acl(5)*, *mt(7)*.

**NAME**

fcmsutil - Fibre Channel Mass Storage Utility Command for the TACHYON A3740A, A3591A, A3404A, A3636A, and TACHYON TL A5158A Fibre Channel Host Bus Adapters.

**SYNOPSIS**

```
/opt/fcms/bin/fcmsutil device_file
/opt/fcms/bin/fcmsutil device_file echo remote-AL_PA [data-size] [count]
/opt/fcms/bin/fcmsutil device_file rls remote-AL_PA
/opt/fcms/bin/fcmsutil device_file test remote-AL_PA [data-size] [count]
/opt/fcms/bin/fcmsutil device_file read offset [pci]
/opt/fcms/bin/fcmsutil device_file write offset [pci]
/opt/fcms/bin/fcmsutil device_file [-f] lb [plm|tachyon]
/opt/fcms/bin/fcmsutil device_file get local|fabric
/opt/fcms/bin/fcmsutil device_file get remote {all|remote-AL_PA}
/opt/fcms/bin/fcmsutil device_file get_lgn AL_PA
/opt/fcms/bin/fcmsutil device_file reset
/opt/fcms/bin/fcmsutil device_file [-f] bdr target-device_file
/opt/fcms/bin/fcmsutil device_file read_cr
/opt/fcms/bin/fcmsutil device_file lgninfo_all
/opt/fcms/bin/fcmsutil device_file stat [-s]
/opt/fcms/bin/fcmsutil device_file clear_stat
/opt/fcms/bin/fcmsutil device_file nsstat
/opt/fcms/bin/fcmsutil device_file clear_nsstat
/opt/fcms/bin/fcmsutil device_file devstat {all|remote-AL_PA}
/opt/fcms/bin/fcmsutil device_file clear_devstat {all|remote-AL_PA}
/opt/fcms/bin/fcmsutil device_file replace_dsk remote-AL_PA
/opt/fcms/bin/fcmsutil device_file [-f] disable
/opt/fcms/bin/fcmsutil device_file enable
```

**NOTE:** *AL\_PA* is Arbitrated Loop Physical Address. It is the one-byte value used to identify a port in an Arbitrated Loop topology. The value of *AL\_PA* corresponds to bits 7:0 of the 24-bit Native Address Identifier.

**NOTE:** For the TACHYON TL A5158A card, *remote-AL\_PA* can be substituted with *-l loop\_id* (in Private Loop), or *-w wwn* for all options that take *remote-AL\_PA* as an argument. The *devstat all* option displays the *loop\_id* (in Private Loop) along with the statistics for all *N\_Ports*, the initiator was able to communicate with.

**DESCRIPTION**

The **fcmsutil** command is a diagnostic tool to be used for the TACHYON A3740A, A3591A, A3404A, A3636A and TACHYON TL A5158A Fibre Channel Host Bus Adapters. This command provides the ability to perform Fibre Channel Test and Echo functionality, read the card's registers, etc. This command requires the use of a device file to indicate the interface over which the requested command needs to be performed. The options that are available only for the TACHYON A3740A, A3591A, A3404A and A3636A HBAs are indicated by a "T" in the description for the option. The options that are available only for the TACHYON TL A5158A HBA are indicated by a "TL" in the description for the option. All other options (which do not have the "T" or "TL" indication) are supported (completely or partially) by all the above HBAs. **fcmsutil** can be used only by users who have an effective user ID of 0. Some of the options require detailed knowledge of the device specific adapter.

**Options**

**fcmsutil** recognizes the following options as indicated in SYNOPSIS. All keywords are case-insensitive and are position dependent.

*device\_file* Can be used alone or with other options.

When used without any options it provides information such as the N\_Port ID, Node World Wide Name and Port World Wide Name, Topology of the Fabric, the Speed of the Link, the Hard Physical Address of the Card, the Driver State, the number of Active Outbound Exchanges and number of Active Logins.

For the TACHYON TL A5158A card, it provides information such as the N\_port\_ID, Node World Wide Name and Port World Wide Name, Topology, Vendor ID, Device ID, the Driver State, number of Assisted IOs and number of Active Logins. As per the FC protocol, Node WWN will be numerically one more than the Port WWN.

**The following topologies are defined:**

**UNKNOWN.** The host adaptor card is not attached to a loop/loopback hood or has connection problems and could not come up in a valid topology.

**IN\_LOOP/PRIVATE\_LOOP.** The card is attached to a loop/loopback hood.

**IN\_LOOP\_FL/PUBLIC\_LOOP.** Supported by the TACHYON TL card. The card is attached to a loop connected to a Fibre Channel switch.

**IN\_PTTOPT\_NPORT.** The card has come up in a point to point topology. This topology can be an error if the the card was expected to come up in loop topology. Not legal in most cases.

**IN\_PTTOPT\_FABRIC.** The card has come up in a point to point topology when connected through a Fibre Channel switch. This topology can be an error if the card was expected to come up in loop topology. Not legal in most cases.

**IN\_PTTOPT** This is a transition topology. Not legal in most cases.

**The following are the driver states:**

**DEAD.** The driver couldn't come up due to some problem. Check the connectivity with the loop/loopback hood.

**LOOPBACK\_STATE.** The host bus adaptor is in the loop back test phase.

**OFFLINE.** The host bus adaptor card is not participating on the loop.

**READY/ONLINE.** The driver is up and functional.

**RESETTING.** The host bus adaptor card is being reset.

All other states are only "transient" and should not continue for long. If the "transient" state persists, there might be a problem in the hardware connectivity or configuration.

**echo** This option requires two parameters, the *remote-AL\_PA* and *data-size* (size of packet to send). An optional third argument (*count*) can be specified for the number of echo packets to be sent. If the *count* option is not specified, one packet will be sent. The *count* option is supported only by the TACHYON TL A5158A card.

Fibre Channel Echo packet(s) of the specified size is sent to the remote node. The command completes successfully when an echo response is received from the remote node and matches the data sent, for all packets sent. The command times out if a response is not received in twice RA\_TOV time. Echo packets cannot be sent in a PUBLIC\_LOOP (FABRIC) topology.

**Note:** Packet size specified must be a multiple of 4.

**rls** This option requires one parameter, the *remote-AL\_PA*. A Fibre Channel Request Link Status (RLS) ELS is sent to the *remote-AL\_PA* and the response data is displayed.

**test** This option requires two parameters, the *remote-AL\_PA* and *data-size* (size of packet to send). An optional third argument (*count*) can be specified for the number of echo packets to be sent. If the *count* option is not specified, one packet will be sent. The *count* option is supported only by the TACHYON TL A5158A card.



A Fibre Channel Test packet of the specified size is sent to the remote node. The command completes successfully and immediately on sending all the test packets.

**Note:** Packet size specified must be a multiple of 4.

- read** This option requires one parameter, the *offset* of the register to read from. The *offset* can be specified in either hex or in decimal format. The *offset* specified is an offset from the base of the Memory Map. The user of this command is therefore expected to have internal knowledge of the chip. Reading from the Tachyon frame manager status register (0x01c8) is restricted.
- An optional second argument (*pci*) can be specified for the TACHYON TL A5158A card, to read from the PCI config space. If no second argument is specified, it reads from the chip register space.
- write** This option requires two parameters, the offset of the register to write to and the value to be written.
- An optional third argument (*pci*) can be specified for the TACHYON TL A5158A card, to write into the PCI config space. If no third argument is specified, it writes into the chip register space.
- lb** WARNING: This is a DESTRUCTIVE test and DATA LOSS during the execution of this test may occur. The **-f** option can be used to suppress the warning message displayed by the TACHYON TL A5158A card.
- This option requires one parameter, **tachyon** or **plm**. The TACHYON TL A5158A card supports only the **plm** option. An optional third argument (*count*) can be specified for the number of loopback packets to be sent. If the *count* option is not specified, one packet will be sent. The *count* option is supported only by the TACHYON TL A5158A card.
- This command performs an internal loopback test when the **tachyon** option is specified and performs an external loopback test when **plm** option is specified. The fibre channel chip is programmed in either internal loopback mode (**tachyon**) or external loopback mode (**plm**) based on the parameter specified. Here **plm** refers to physical link module or gigabit link module. The self test then involves sending a packet and receiving back the packet within the adapter and checking its integrity. Since this self test is at the adapter level, no packet goes on the fibre link.
- get** The **get** option is used to obtain Fibre Channel login parameters of either the **local** port, the **fabric** port or of a **remote** port. The TACHYON TL A5158A card does not support the **local** option. If the **all** argument is specified for the **remote** option, login parameters and current states of all N\_ports that the initiator is aware of, are displayed. The **all** option is supported only by the TACHYON TL A5158A card.
- get\_lgn** (T) The **get\_lgn** option is used to obtain detailed information maintained in the login block associated with each N\_Port that this N\_Port has communicated with. The *remote-AL\_PA* is a required parameter for this option. This option is not supported by the TACHYON TL A5158A card.
- reset** resets the fibre channel card. This is a destructive test and communication to all nodes will be terminated till the reset process is completed. This option only results in a soft reset for the TACHYON TL A5158A card and does not result in termination of communication.
- bdr** (TL) WARNING: This is a DESTRUCTIVE test. The **-f** option can be used to suppress the warning message displayed by the TACHYON TL A5158A card.
- This option resets the target, clearing all commands, without doing any checks. This option is supported only by the TACHYON TL A5158A card.
- read\_cr** This option can be used to read all of the readable registers on the card and format the detailed information.
- lgninfo\_all** (T) This option is used to obtain a comprehensive list of nodes to which a successful login has been established. This option is not supported by the TACHYON TL A5158A card.
- stat** This option is used to obtain detailed statistics maintained by the driver. An optional argument (**-s**) can be specified for the TACHYON TL A5158A card, to obtain a shortened version of the statistics maintained by the driver.

- clear\_stat** (TL)  
This option is used to clear the statistics maintained by the driver. This option is supported only by the TACHYON TL A5158A card.
- nsstat** (TL) This option is used to obtain detailed nameserver statistics maintained by the driver. This option is supported only by the TACHYON TL A5158A card.
- clear\_nsstat** (TL)  
This option is used to clear the nameserver statistics maintained by the driver. This option is supported only by the TACHYON TL A5158A card.
- devstat** (TL)  
The **devstat** option is used to obtain detailed statistics associated with each AL\_PA that this AL\_PA has communicated with.  
If the *remote-AL\_PA* is specified, then the statistics associated with that AL\_PA are displayed. If the **all** option is specified, statistics associated with all N\_ports that the initiator has been able to communicate with, are displayed.  
This option is supported only by the TACHYON TL A5158A card.
- clear\_devstat** (TL)  
This option is used to clear the statistics associated with a target.  
If the *remote-AL\_PA* is specified, then the statistics associated with that AL\_PA are cleared. If the **all** option is specified, statistics associated with all valid AL\_PAs are cleared.  
This option is supported only by the TACHYON TL A5158A card.
- replace\_dsk** (TL)  
This option is used to specify that no authentication should be performed the next time we communicate with the device. This option is to be used by system administrators for replacing a disk, with another (with the same N\_Port\_ID). Note that this is not required, if the new disk acquires a new N\_Port\_ID.  
This option is supported only by the TACHYON TL A5158A card.
- disable** WARNING: This is a DESTRUCTIVE test and communication to all nodes will be terminated. The **-f** option can be used to suppress the warning message displayed by the TACHYON TL A5158A card.  
This option is used to disable a card, typically because of a hardware problem which cannot be resolved and is interfering with system performance.
- enable** This option is used to enable a card, typically when a previous hardware problem has been resolved.

## EXAMPLES

Print the remote port parameters using the get remote option if the driver is idle. */dev/fcms1* is the device file and */dev/rdisk/c27t0d0* is the respective raw disk file.

```
fcmsutil /dev/fcms1 get remote 0x98 < /dev/rdisk/c27t0d0
```

Print a short listing of the statistics maintained by the driver, with */dev/td1* as the device file.

```
fcmsutil /dev/td1 stat -s
```

Send 5 echo packets of 200 bytes each to a remote AL\_PA with *loop\_id* 4, with */dev/td1* as the device file

```
fcmsutil /dev/td1 echo -l 4 200 5
```

## AUTHOR

*/opt/fcms/bin/fcmsutil* was developed by HP.

**NAME**

fdetach - detach a STREAMS-based file descriptor from a filename

**SYNOPSIS**

**fdetach** *path*

**DESCRIPTION**

The **fdetach** command detaches or disassociates a file descriptor for an open STREAMS device or pipe from its filename in the file system. The *path* argument is the *path* that was previously associated with the file descriptor by the **fattach( )** function.

Operations on *path* will subsequently affect the file system node, not the STREAMS device or pipe. The permissions and status of the node are returned to the state that they were in before the STREAMS device or pipe was attached. Any other paths that the STREAMS device or pipe may be attached to are not affected.

To successfully issue the **fdetach** command, the user must be superuser or must be the owner of the file and have write permission.

**RETURN VALUE**

**fdetach** returns 0 (zero) on success. If **fdetach** fails, it returns 1 and prints a message to **stderr**.

**EXAMPLES**

To detach the file descriptor for the STREAMS file **/tmp/streamfile** from its associated file system node, enter:

```
fdetach /tmp/streamfile
```

**FILES**

**/usr/lib/nls/C/fdetach.cat** NLS catalog for **fdetach**.

**SEE ALSO**

fattach(3C), fdetach(3C), streamio(7).

**NAME**

ff - list file names and statistics for a file system

**SYNOPSIS**

`/usr/sbin/ff [-F FStype] [-o specific_options] [-V] special ...`

**DESCRIPTION**

The **ff** command reads the i-list and directories of each *special* file, assuming it to be a file system, saving i-node data for files that match the selection criteria. Output consists of the path name for each saved i-node, plus any other file information requested with the **-o** option. Output fields are positional. The output is produced in i-node order; fields are separated by tabs. The default line produced by **ff** includes the path name and i-number fields.

**Options and Arguments**

**ff** recognizes the following options and arguments:

- F *FStype*** Specify the file system type on which to operate (see *fstyp*(1M) and *fs\_wrapper*(5)). If this option is not included on the command line, then the file system type is determined from the file `/etc/fstab` by matching each *special* with an entry in that file. If there is no entry in `/etc/fstab`, then the file system type is determined from the file `/etc/default/fs`.
- o *specific\_options*** Specify options specific to each file system type. *specific\_options* is a list of suboptions and/or keyword/attribute pairs intended for a specific *FStype*-specific module of the command. See the file-system-specific man pages for a description of the *specific\_options* supported, if any.
- V** Echo the completed command line, but perform no other action. The command line is generated by incorporating the user-specified options and other information derived from `/etc/fstab`. This option allows the user to verify the command line.

**EXAMPLES**

List the path names and i-numbers of all files in the file system `/dev/dsk/c1d2s0`:

```
ff /dev/dsk/c1d2s0
```

Execute the **ff** command on HFS file system `/dev/dsk/c1d2s0`:

```
ff -F hfs /dev/dsk/c1d2s0
```

Display a completed command line without executing the command:

```
ff -V /dev/dsk/c1d2s0
```

**FILES**

|                              |                                              |
|------------------------------|----------------------------------------------|
| <code>/etc/default/fs</code> | File that specifies the default system type. |
| <code>/etc/fstab</code>      | Static information about the file systems.   |

**SEE ALSO**

`find`(1), `ff_hfs`(1M), `ff_vxfs`(1M), `fstyp`(1M), `ncheck`(1M), `fstab`(4), `fs_wrapper`(5).

**NAME**

ff - list file names and statistics for HFS file system

**SYNOPSIS**

```
/usr/sbin/ff [-F hfs] [-a num] [-c num] [-i inode-list] [-I] [-l] [-m num] [-n file]
[-p prefix] [-s] [-u] [-V] special ...
```

**DESCRIPTION**

The **ff** command reads the i-list and directories of each special file *special*, assuming it to be an HFS file system, saving i-node data for files that match the selection criteria. Output consists of the path name for each saved i-node, plus any other file information requested using the print options below. Output fields are positional. The output is produced in i-node order; fields are separated by tabs. The default line produced by **ff** contains the path name and i-number fields. With all options specified, the output fields include path name, i-number, size, and user ID.

The *num* parameter in the options descriptions is a decimal number, where *+num* means more than *num*, *-num* means less than *num*, and *num* means exactly *num*. A day is defined as a 24-hour period.

**ff** lists only a single path name out of many possible ones for an i-node with more than one link, unless you specify the **-l** option. With **-l**, **ff** applies no selection criteria to the names listed. All possible names for every linked file on the file system are included in the output. On very large file systems, memory may run out before **ff** completes execution.

**Options and Arguments**

**ff** recognizes the following options and arguments:

- a num** Select a file if the i-node has been accessed in *num* days.
- c num** Select a file if the i-node has been changed in *num* days.
- F hfs** Specify the HFS file system type.
- i inode-list** Generate names for any i-node specified in the *inode-list*.
- I** Do not display the i-node number after each path name.
- l** Generate a list of all path names for files with more than one link.
- m num** Select a file associated with an i-node if it has been modified in *num* days.
- n file** Select a file associated with an i-node if it has been modified more recently than the specified *file*.
- p prefix** Add the specified *prefix* to each path name. The default prefix is . (dot).
- s** Write the file size, in bytes, after each path name.
- u** Write the owner's login name after each path name.
- V** Echo the completed command line, but performs no other action. The command line is generated by incorporating the user-specified options and other information derived from */etc/fstab*. This option allows the user to verify the command line.

**EXAMPLES**

List the path names and i-numbers of all files in the file system */dev/dsk/c1d2s0*:

```
ff /dev/dsk/c1d2s0
```

Same as above, but suppress the printing of i-numbers:

```
ff -I /dev/dsk/c1d2s0
```

List files on the same file system that have been modified recently, displaying the path name, i-number, and owner's user name (the **-u** option). List only files that have been modified within the last two days (the **-m -2** option):

```
ff -m -2 -u /dev/dsk/c1d2s0
```

List all files on the same file system, including the path name and i-number of each file, that was last accessed more than 30 days ago (**-a +30**):

```
ff -a +30 /dev/dsk/c1d2s0
```

Find all path names associated with i-nodes **451** and **76** (the **-l** option):

```
ff -l -i 451,76 /dev/dsk/c1d2s0
```

Execute the **ff** command on an HFS file system **/dev/dsk/c1d2s0**:

```
ff -F hfs /dev/dsk/c1d2s0
```

**FILES**

**/etc/fstab**        Static information about the file systems.

**SEE ALSO**

**find(1)**, **ff(1M)**, **ff\_vxfs(1M)**, **ncheck(1M)**, **fstab(4)**.

f

**NAME**

ff - fast find: list file names and statistics for a VxFS file system

**SYNOPSIS**

```
/usr/sbin/ff [-F vxfs] [-Vil su] [-a num] [-c num] [-i inode-list] [-m num] [-n file]
              [-o s] [-p prefix] special...
```

**DESCRIPTION**

**ff** reads the inode list and directories of each *special* file, assuming it to be a VxFS file system, and prints inode data for files that match the selection criteria. Output consists of the pathname for each saved inode, plus any other file information requested using the print options below. Output fields are positional. The output is produced in inode order; fields are separated by tabs. The default line produced by **ff** is:

*pathname inumber*

The maximum information **ff** displays is:

*pathname inumber size owner*

The *num* parameter in the options descriptions is a decimal number, where *+num* means more than *num* days, *-num* means less than *num* days, and *num* means exactly *num* days. A day is defined as a 24-hour period.

**Options**

**ff** recognizes the following options:

- a num** Select a file if the inode has been accessed in *num* days.
- c num** Select a file if the inode has been changed in *num* days.
- F vxfs** Specify the VxFS file system type.
- i inode-list** Generate names for any inodes specified in the *inode-list*.
- I** Do not display the inode number after each pathname.
- l** Generate a list of all pathnames for files with more than one link.
- m num** Select a file associated with the inode if it has been modified in *num* days.
- n file** Select a file associated with an inode if it has been modified more recently than the specified *file*.
- p prefix** Add the specified *prefix* to each pathname. The default prefix is **.** (dot).
- o s** Print only special files and files with set-user-ID mode (VxFS-specific option).
- s** Write the file size, in bytes, after each pathname.
- u** Write the owner's login name after each pathname.
- v** Echo the completed command line, but performs no other action. The command line is generated by incorporating the user specified options and other information derived from **/etc/fstab**. This option allows the user to verify the command line.

**EXAMPLES**

List the pathnames and inumber of all files in the file system **/dev/vg01/r1vol1**:

```
ff /dev/vg01/r1vol1
```

Same as above, but suppress the printing of inumber:

```
ff -I /dev/vg01/r1vol1
```

List files on the same file system that have been modified in the last two days (**-m -2**), displaying the pathname, inumber, and owner's user name (**-u**).

```
ff -m -2 -u /dev/vg01/r1vol1
```

List all files on the same file system, including the pathname and inumber of each file, that were last accessed more than 30 days ago (**-a +30**):

```
ff -a +30 /dev/vg01/r1vol1
```

Find all pathnames associated with inodes 451 and 76 (-l):

```
ff -l -i 451,76 /dev/vg01/rlvol1
```

Execute the **ff** command on a VxFS file system **/dev/vg01/rlvol1**:

```
ff -F vxfs /dev/vg01/rlvol1
```

**FILES**

**/etc/fstab**        Static information about the file systems.

**SEE ALSO**

ff(1M), find(1), fstab(4), ncheck\_vxfs(1M).

f



**NAME**

fingerd - remote user information server

**SYNOPSIS**

`/usr/sbin/fingerd [-r]`

**DESCRIPTION**

**fingerd** is the server for the RFC 742 Name/Finger protocol. It provides a network interface to **finger**, which gives a status report of users currently logged in on the system or a detailed report about a specific user (see *finger(1)*). The Internet daemon executes **fingerd** when it receives a service request at the port listed in the services data base for “finger” using “tcp” protocol; see *inetd(1M)* and *services(4)*.

To start **fingerd** from **inetd**, the configuration file `/etc/inetd.conf` must contain an entry as follows:

```
finger stream tcp nowait bin /usr/sbin/fingerd fingerd
```

Once a remote host is connected, **fingerd** reads a single “command line” terminated by a carriage-return and line-feed. It uses this command line as the arguments to an invocation of **finger**. **fingerd** sends the output of **finger** to the remote host and closes the connection.

If the command line is null (contains only a carriage-return and line-feed pair), **finger** returns a report that lists all users logged in on the system at that moment.

If a user name is specified on the command line (for example, `user<CR><LF>`), the response lists more extended information for only that particular user, whether logged in or not. See *finger(1)* for the details of this extended information.

If **fingerd** is run with the **-r** option, it allows remote user names on the command line (for example, `user@host<CR><LF>`). Otherwise, if the command line contains a remote user name, **fingerd** prints the error message **Remote finger not allowed** and closes the connection.

**AUTHOR**

**fingerd** was developed by the University of California, Berkeley and HP.

**SEE ALSO**

*finger(1)*, *inetd(1M)*, *services(4)*,  
RFC 742 for the Name/Finger protocol.

f

**NAME**

fixman - fix manual pages for faster viewing with man(1)

**SYNOPSIS**

```
/usr/sbin/fixman [-A alt-path]
```

**DESCRIPTION**

The **fixman** command is a shell script that processes man pages in the **cat\*** directories to unexpand spaces to tabs where possible, and to remove all character-backspace pairs (which usually exist to cause overstriking or underscoring for printer output). Removal of unnecessary character sequences improves the speed of *man(1)*, and reduces disk space consumption. The **fixman** command should be run after using **catman** to create formatted, **cat**-able manual entries from unformatted, *nroff(1)*-compatible source files (see *catman(1M)*).

By default, **fixman** searches for **cat\*** subdirectories in the following parent directories in the order indicated:

- /usr/share/man
- /usr/contrib/man
- /usr/local/man

If the **MANPATH** environment variable is set, the directory paths specified by **MANPATH** are searched instead of the default. See *environ(5)* for a description of the **MANPATH** environment variable.

The **fixman** command does not remove duplicate blank lines. Thus, all files remain a multiple of one page (66 lines) long and can still be passed directly to **lp** (see *lp(1)*). (Note that *man(1)* normally uses **more -s** to accomplish this removal.)

To ensure success, **fixman** should be run by a user who has appropriate privileges. It will take awhile to complete depending on system speed, load, memory size, etc. As a side-effect, file ownerships and permissions may be changed.

**Options**

**-A *alt-path***

Perform actions based on the given alternate root. With this option, *alt-path* will be prepended to all directory paths, including default paths or the paths defined by **MANPATH**.

**EXTERNAL INFLUENCES****Environment Variables**

**MANPATH**, if set, defines the directories to be searched for **cat**-able manual entries.

**WARNING**

If the value of **MANPATH** is not the same while **fixman** is running as it was when **catman** was run or when manpage files were installed, some files may be missed and not processed (see *catman(1M)*).

**EXAMPLES**

Run **fixman** from a server to fix the manual pages on a diskless under the alternate root **/export/shared\_roots/OS\_700**:

```
fixman -A /export/shared_roots/OS_700
```

This will fix manpages in **cat\*** directories under:

```
/export/shared_roots/OS_700/usr/share/man/
/export/shared_roots/OS_700/usr/contrib/man/
/export/shared_roots/OS_700/usr/local/man/
```

**FILES**

**/usr/share/man/cat\**[.Z]*** Directories containing [compressed] *nroff(1)*-formatted versions of manual entries

```
/usr/local/man/cat*[.Z]
/usr/contrib/man/cat*[.Z]
```

**AUTHOR**

**fixman** was developed by HP.

**SEE ALSO**

catman(1M), chmod(1), expand(1), lp(1), man(1), mv(1), sed(1), environ(5).

  
f

**NAME**

format - format an HP SCSI disk array LUN

**SYNOPSIS**

**format** *device\_file*

**DESCRIPTION**

**format** formats one LUN of the HP SCSI disk array associated with device file, *device\_file*. The format will usually be a soft or zeroing format, in which the controller writes zeroes to the data area and parity area, if any, of the LUN.

**NOTE:** The above should always be true of a sub-LUN, but the controller might decide, based on certain conditions, to do a full format of a regular LUN, which consists of sending a mode select and a media initialization command to the physical drive(s) in question, followed by zeroing the data and parity area, if any. The conditions which will cause a full format to be done are as follows:

1. The controller received a Mode Select command which requires a drive sector size change.
2. The controller received a Mode Select command which changed a parameter in the Format Device Page (0x03).
3. The LUN contains one or more failed drives. In this case only a certain subset of the drives containing the failed drives will be formatted.
4. Either the FmtData or the CmpLst bit in the Format Unit CDB is set.

**RETURN VALUE**

**format** returns the following values:

- 0 Successful completion.
- 1 Command failed.

**DIAGNOSTICS AND ERRORS**

Errors can originate from problems with:

- **format**
- SCSI (device level) communications
- system calls

**Error messages generated by format:**

**usage: format <special>**

An error in command syntax has occurred. Enter command again with all required arguments, in the order shown.

**format: device busy**

To ensure that **format** does not modify a disk array that is being used by another process, **format** attempts to obtain exclusive access to the disk array. If the disk array is already opened by another process (for example, LVM — the Logical Volume Manager), a "**device busy**" error message is returned by the driver. To eliminate the "**device busy**" condition, determine what process has the device open. In the case of LVM, it is necessary to deactivate the volume group containing the array before formatting array LUNs (see *vgchange(1M)*).

**format: LUN # too big**

The LUN number, which is derived from the device file name, is out of range.

**format: LUN does not exist**

The addressed LUN is not configured, and thus is not known to the array controller.

**format: Not a raw file**

Utilities must be able to open the device file for raw access.

**format: Not an HP SCSI disk array**

The device being addressed is not an HP SCSI disk array.

**SCSI (device level) communication errors:**

Sense data associated with the failed operation is printed.

**Error messages generated by system calls:**

**format** uses the following system calls:

**malloc()**, **free()**, **stat()**, **open()**, **close()**, **read()**, **write()**, and **ioctl()**.

Documentation for these HP-UX system calls contains information about the specific error conditions associated with each call. **format** does not alter the value of **errno**. The interpretation of **errno** for printing purposes is performed by the system utility **strerror()**.

**EXAMPLES**

To format the HP SCSI disk array LUN **/dev/rdisk/c2t0d0** on a Series 800:

```
format /dev/rdisk/c2t0d0
```

**WARNING**

The **format** command will destroy all user data on the addressed LUN.

**DEPENDENCIES**

The HP C2425 and HP C2427 disk arrays are only supported on Series 700 systems running HP-UX version 9.0X.

The HP C2430 disk array is supported on Series 700 and 800 systems running HP-UX versions 9.0X and 10.0X.

**AUTHOR**

**format** was developed by HP.

**f**

## NAME

frecover - selectively recover files

## SYNOPSIS

```
/usr/sbin/frecover -r [-hmosvyAFNOX] [-c config] [-f device] [-S skip] [-E extarg]
/usr/sbin/frecover -R path [-f device]
/usr/sbin/frecover -x [-hmosvyAFNOX] [-c config] [-e path] [-f device] [-g graph]
[-i path] [-S skip] [-E extarg]
/usr/sbin/frecover -I path [-vy] [-f device] [-c config]
/usr/sbin/frecover -V path [-vy] [-f device] [-c config]
```

## DESCRIPTION

**frecover** reads media written by the **fbackup** command. Its actions are controlled by the selected function **-r**, **-R**, **-x**, **-V**, or **-I**.

The function performed by **frecover** is specified by one of the following options:

- r** The backup media is read and the contents are loaded into the directories from which they were backed up. This option should only be used to recover a complete backup onto a clear directory or to recover an incremental backup after a full level-zero recovery (see *fbackup(1M)*). This is the default behavior.
- x** The files identified by the **-i**, **-e**, and **-g** options (see below) are extracted or not extracted from the backup media. If a file to be extracted matches a directory whose contents have been written to the backup media, and the **-h** option is not specified, the directory is recursively extracted. The owner, modification time, and access control list (including optional entries, unless the **-A** option is specified) are recovered. If no file argument is given (including an empty graph file), all files on the backup media are extracted, unless the **-h** option is specified.
- I path** The index on the current volume is extracted from the backup media and is written to *path*.
- V path** The volume header on the current volume is extracted from the backup media and is written to *path*. The following fields from the header are extracted in the format *label:value* with one pair per line.

|                            |                                                                                                                                                   |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| Magic Field                | On valid <b>fbackup</b> media, it contains the value <b>FBACKUP_LABEL</b> . On pre-10.20 <b>fbackup</b> media, it contains <b>FBACKUP LABEL</b> . |
| Machine Identification     | This field contains the result of <b>uname -m</b> .                                                                                               |
| System Identification      | This field contains the result of <b>uname -s</b> .                                                                                               |
| Release Identification     | This field contains the result of <b>uname -r</b> .                                                                                               |
| Node Identification        | This field contains the result of <b>uname -n</b> .                                                                                               |
| User Identification        | This field contains the result of <i>cuserid(3S)</i> .                                                                                            |
| Record Size                | This field contains the maximum length in bytes of a data record.                                                                                 |
| Time                       | This field contains the time <b>fbackup</b> was started.                                                                                          |
| Media Use                  | This field contains the number of times the media has been used for backup.                                                                       |
| Volume Number              | This field contains a # character followed by 3 digits, and identifies the current volume in the backup.                                          |
| Checkpoint Frequency       | This field contains the number of data records between checkpoints.                                                                               |
| Fast Search Mark Frequency | This field contains the number of files between <b>fast search marks</b> for backups made with DDS tape drives.                                   |
| Index Size                 | This field contains the size of the index.                                                                                                        |
| Backup Identification Tag  | This field is composed of 2 items: the process ID (pid), and the start time of that process.                                                      |

**Language**

This field contains the language used to make the backup.

- R path** An interrupted full recovery can be continued using this option. **frecover** uses the information in file *path* to continue the recovery from where it was interrupted. The only command line option used by **frecover** with this option is **-f**. The values in *path* override all other options to **frecover**. Note also that only full recoveries are restarted with this option, because no history of include or exclude lists is stored in the restart file. If a partial recovery (i.e., using the **-x** option) is interrupted then restarted with this option, **frecover** continues recovering where the partial recovery left off, but restores all files on the backup media beyond this point.

The following options can be used in addition to the option above that selects the desired function:

- c config** *config* specifies the name of a configuration file to be used to alter the behavior of **frecover**. The configuration file allows the user to specify the action to be taken on all errors, the maximum number of attempts at resynchronizing on media errors (**-S** option), and the action to be taken on media errors. Each entry of a configuration file consists of an action identifier followed by a separator followed by the specified action. Valid action identifiers are **error**, **chgvol**, and **sync**. Separators can be either tabs or spaces. In the following sample configuration file, each time an error is encountered, the script `/var/adm/fbackupfiles/frecovererror` is executed. The script `/var/adm/fbackupfiles/frecoverchgvol` is executed each time the backup media is to be changed. The maximum number of resynchronization attempts is five.

```
error /var/adm/fbackupfiles/frecovererror
chgvol /var/adm/fbackupfiles/frecoverchgvol
sync 5
```

- e path** *path* is interpreted as a graph to be excluded from the recovery. There is no limit on how many times the **-e** option can be specified.
- f device** *device* identifies the backup device to be used instead of the default `/dev/rmt/0m`. If *device* is **-**, **frecover** reads from standard input. Thus **fbackup** and **frecover** can be used in a pipeline to backup and recover a file system as follows:

```
fbackup -i /usr -f - | (cd /mnt; frecover -Xrf -)
```

If more than one output file is specified, **frecover** uses each one successively and then repeats in a cyclical pattern. Patterns can be used in the device name in a way similar to file name expansion as done by *sh*(1). The expansion of the pattern results in all matching names being in the list of devices used. A device on the remote machine can be specified in the form *machine:device*. **frecover** creates a server process, `/usr/sbin/rmt`, on the remote machine to access the tape device. If `/usr/sbin/rmt` does not exist on the remote system, **frecover** creates a server process from `/etc/rmt` on the remote machine to access the tape device. The pattern matching capability does not apply to remote devices. Only raw magnetic tapes can be remote devices. The **fast search marks** capability is not used when accessing remote DDS devices.

- g graph** *graph* defines a graph file. Graph files are text files and contain the list of file names (graphs) to be recovered or skipped. Files are recovered using the **-i** option; so, for example, if the user wants to recover all of `/usr`, the graph file contains one entry:

```
i /usr
```

It is also possible to skip files by using the **-e** option. For example, if a user wants to recover all of `/usr` except for the subgraph `/usr/lib`, the graph file contains two entries:

```
i /usr
e /usr/lib
```

If the graph file is missing, **frecover** exits with an error message. An empty graph file results in recovering all files on the media.

- h** Extract the actual directory, rather than the files that it references. This prevents hierarchical restoration of complete subtrees from the backup media.
- i path** *path* is interpreted as a graph to be included in the recovery. There is no limit on how many times the **-i** option can be specified.

- m** Print a message each time a file marker is encountered. Using this option, **frecover** prints a message each time either a DDS **fast search mark**, a filemark (EOF), or a checkpoint record is read. Although useful primarily for troubleshooting, these messages can also be used to reassure the user that the backup is progressing during long, and otherwise silent, periods during the recovery.
- o** Recover the file from the backup media irrespective of age. Normally **frecover** does not overwrite an existing file with an older version of the file.
- s** Attempt to optimize disk usage by not writing null blocks of data to sparse files.
- v** Normally **frecover** works silently. Verbose option. Displays the file type and name of each file processed.
- y** Automatically answer **yes** to any inquiries.
- A** Do not recover any optional entries in access control lists (ACLs). Normally, all access control information, including optional ACL entries, is recovered. This option drops any optional entries and sets the permissions of the recovered file to the permissions of the backed up file. Use this option when recovering files backed up from a system with ACLs on a system where ACLs are not present (see *acl(5)*).
- F** Recover files without recovering leading directories. For example, this option would be used if a user wants to recover `/usr/bin/vi`, `/usr/bin/sh`, and `/etc/passwd` to a local directory without creating each of the graph structures.
- E extarg** Specifies the handling of any extent attributes backed up by **fbackup**. The **-E** option takes the following keywords as arguments:
- warn** Issue a warning message if extent attributes cannot be restored, but restore the file anyway.
  - ignore** Do not restore extent attributes.
  - force** Issue an error message and do not restore the file if extent attributes cannot be restored.
- Extent attributes cannot be restored if the files are being restored to a file system which does not support extent attributes or if the file system's block size is incompatible with the extent attributes. If **-E** is not specified, *extarg* defaults to **warn**.
- N** (no recovery) Prevent **frecover** from actually recovering any files onto disk, but read the backup as if it was, in fact, recovering the data from the backup, producing the same output that it would on a normal recovery. This option is useful for verifying backup media contents in terms of validity (block checksum errors are reported), and contents (a listing of files can be produced by using the **-N** and **-v** options together). Note that the listing of files produced with the **-N** and **-v** options requires the reading of the entire backup, but is therefore a more accurate reflection of the backup's contents than the index stored at the beginning of the backup (which was created at the start of the backup session, and is not changed during the course of the backup).
- O** Use the effective uid and gid for the owner and group of the recovered file instead of the values on the backup media.
- S skip** **frecover** does not ask whether it should abort the recovery if it gets a media error. It tries to skip the bad block or blocks and continue. Residual or lost data is written to the file named by *skip*. The user can then edit this file and recover otherwise irretrievable data.
- X** Recover files relative to the current working directory. Normally **frecover** recovers files to their absolute path name.

## EXTERNAL INFLUENCES

### Environment Variables

**LC\_COLLATE** determines the order in which **frecover** expects files to be stored on the backup device and the order in which file names are output by the **-I** option.

**LC\_MESSAGES** determines the language in which messages are displayed.

If **LC\_COLLATE** and **LC\_MESSAGES** are not specified in the environment or are set to the empty string, the value of **LANG** is used as a default for each unspecified or empty variable. If **LANG** is not specified or is set to the empty string, a default of "C" (see *lang(5)*) is used instead of **LANG**. If any internationalization



variable contains an invalid setting, **frecover** behaves as if all internationalization variables are set to "C". See *environ*(5).

### International Code Set Support

Single- and multi-byte character code sets are supported.

### WARNINGS

For incremental backups created prior to installing HP-UX Release 8.0, or for recoveries that do not begin with the first volume (such as when reading tape 3 first), it is possible for the preceding directories to a recoverable file to not be on the media. This can happen, for example, if the directories did not change since the last full backup. If **frecover** encounters a file on the backup that should be recovered, but it has not recovered the file's parent directories from the backup, it prints a message stating that the recovery will continue with that file, and attempts to create the file's parent directories as needed.

Use of **frecover** does not require special privileges. However, if a user does not have access permission to a given file, the file is not recovered.

The **fbackup** index format now includes the file size in the first field; the previous format simply had the '#' character in that field. The implementation provides both forward and backward compatibility between the old and new index formats. However, the file sizes are used in conjunction with the checkpoints to increase selective recovery speed on DLT devices, so recovery of an **fbackup** volume that does not have the new index format will not see that performance gain.

When using a DDS tape written with the current release of **fbackup** to do a partial recovery, **frecover** attempts to use the DDS fast-search capability to find files on the tape more quickly. In order to do this, however, **frecover** needs to create an in-memory copy of the index, and mark the files on that index which it needs to recover before actually reading through the tape to find the files. This is done when the first index is read from the tape, and accounts for a period of time just after recovery is begun where the tape is inactive while this in-memory index is constructed. The larger the index is, the longer this period lasts.

The utility set comprised of **fbackup** and **frecover** was originally designed for use on systems equipped with not more than one gigabyte of total file system storage. Although the utilities have no programming limitations that restrict users to this size, complete backups and recoveries of substantially larger systems can cause a large amount of system activity due to the amount of virtual memory (swap space) used to store the indices. Users who want to use these utilities, but are noticing poor system-wide performance due to the size of the backup, are encouraged to back up their systems in multiple smaller sessions, rather than attempting to back up the entire system at one time. However, if the entire backup must be done with a single session, the user may encounter an error in **frecover** if there is not enough virtual memory available. If this happens, the user might consider adjusting the *maxdsiz* parameter or the swap space; both of these require a reboot.

Note that when recovering files with access control lists, the ACL entries are stored on the backup as user login names. If a login name cannot be found in the password file, the file is recovered without its ACL, and an error is printed. In order to fully recover files backed up with ACLs, the password file (*/etc/passwd*) must be recovered before attempting to recover any desired ACLs.

Network special files are obsolete. Therefore, **frecover** cannot restore these files. A warning message is issued if an attempt is made to recover a network special file, and the file is skipped.

Care should be taken to match the names specified by the include and exclude options with the names in the index on the tape. Since the files are stored on the backup in lexicographic order as defined by the **LANG** or **LC\_COLLATE** environment variable, **frecover** uses the exact path names to determine when a partial recovery is complete, and when an earlier tape needs to be loaded. If a user's specification of a file to be recovered is misspelled, this may cause confusing messages, such as **frecover** asking for the previous volume, when volume one is mounted.

### DEPENDENCIES

**frecover** does not support QIC-120 and QIC-150 formats on QIC devices. If **frecover** is attempted for these formats, **frecover** fails and the following message is displayed:

```
mt lu X:Read must be a multiple of 512 bytes in QIC 120 and QIC 150
```

### AUTHOR

**frecover** was developed by HP.

**FILES**

`/dev/rmt/0m`      Default backup device.

**SEE ALSO**

`cpio(1)`, `dump(1M)`, `fbackup(1M)`, `restore(1M)`, `rmt(1M)`, `acl(5)`.

  
f

**NAME**

freedisk - recover disk space

**SYNOPSIS**

**freedisk** [-a *n*] [-v]

**DESCRIPTION**

The **freedisk** command is an interactive script that finds and optionally removes filesets that do not appear to have been used since they were originally installed by **swinstall** (see *swinstall(1M)*). NOTE: Familiarity with **swremove** (see *swremove(1M)*) is required for successful use of this tool.

The **freedisk** command has two phases, any combination of which can be executed or skipped.

The first phase analyzes the regular files in all filesets to discover filesets that have remained unused since installation. Use the **-a** option to specify a usage time other than “since installation.”

Filesets that appear to be entirely unused, but which are dependencies of other filesets that are in use, are treated by **freedisk** as though they were “in use” and are not presented as candidates for removal.

At the end of the first phase, the **swremove** command is invoked interactively with the filesets that are candidates for removal already selected. During the **swremove** session any, all, or none of the pre-selected filesets can be removed.

The second phase of **freedisk** optionally removes filesets that are used only for building kernels. These filesets are identified by containing a control file named **freedisk\_rmdbl**. This removal occurs regardless of when the filesets were last used. This phase should be executed only if you are sure you will not need to rebuild a kernel for any reason. The interactive interface provides more information on this capability.

You can reload kernel build filesets removed during this phase by using **/var/adm/sw/krn\_rmvd.log** as the argument to the **-f** option of **swinstall**.

**Options**

**freedisk** supports the following options:

- a *n*** Check access of files only in the previous *n* days instead of the default of checking access since the fileset installation date. The *n* value should be a positive integer. It is passed to **find** (see *find(1)*) as **-atime -n**.
- v** Provide very verbose output. Useful when detailed information is required as to which specific files have been used in each fileset.  
  
If you prefer to track the operation of the utility in a scrollable and easily viewable form, redirect the output to a file (see the example below) and use an editor on that file.

**RETURN VALUE**

The following are exit values of **freedisk**:

- 0 Successful completion.
- 1 One or more critical errors occurred.

**DIAGNOSTICS**

Error messages are self-explanatory.

**EXAMPLES**

Use the verbose option of **freedisk** to identify individual files used in each fileset and keep a copy of the output in a file for later use:

```
/opt/contrib/bin/freedisk -v 2>&1 | tee filename
```

Find filesets that have not been used in the past 90 days:

```
/opt/contrib/bin/freedisk -a 90
```

**WARNINGS**

Removing the kernel build filesets in phase two can result in unresolved fileset dependencies. This means that **swverify** (see *swverify(1M)*) will indicate errors, unless the appropriate options are used to ignore missing dependencies.

Be careful when using the **-a n** option. Small values of *n* might cause infrequently used filesets to be discovered as unused.

**AUTHOR**

**freedisk** was developed by the Hewlett-Packard Company.

**FILES**

**/var/adm/sw/krn\_rmvd.log** log of removed kernel-build filesets  
**/var/adm/sw/swremove.log** log of **swremove** actions  
**/var/adm/sw/swagent.log** log of **swagent** actions

**SEE ALSO**

**find(1)**, **swinstall(1M)**, **swmodify(1M)**, **swremove(1M)**, **swverify(1M)**, and the manual *Software Distributor Administration Guide*.

f

**NAME**

frupower - turn on/off or display current status of power for cells and I/O chassis

**SYNOPSIS**

```
frupower [ -d | -o | -f | -c cell [-c...]
frupower [ -d | -o | -f | -i I/Ochassis [-i...]
frupower [-d] -C [-l cabinet] [-l...]
frupower [-d] -I [-l cabinet] [-l...]
```

**DESCRIPTION**

The **frupower** command turns on/off or displays the current status of power for cells and I/O chassis.

This command only powers on a component which is off and vice versa. An attempt to power on an object which is already on, or power off an object that is already off, is ignored. When powering on a cell, firmware will automatically power on and activate the I/O chassis that is attached to the cell. When powering off a cell, firmware will automatically power off and deactivate (in terms of the hardware) the I/O chassis that is attached to the cell. Also when powering on a root cell without a GSP in the complex, the partition will boot.

**Options and Arguments**

**frupower** recognizes the following command line options and arguments:

- d Display power status of specified cells or I/O chassis. This is the default.
- o Power on specified cells or I/O chassis.  
The -o and -f options are mutually exclusive. The -o and -f options are unavailable with -C and -I.
- f Power off specified cells or I/O chassis.
- c cell The specified *cell* is powered on/off or the power status is displayed.  
A *cell* can be specified either in the local (*cabinet#/slot#*) or global (*cell\_ID*) format. For example, the cell located in cabinet 2, slot 4 is locally identified as 2/4 or globally as simply 20.
- i I/Ochassis The specified *I/Ochassis* is powered on/off or the power status is displayed.  
An I/O chassis id can be specified in the form of *cabinet#/enclosure#/chassis#*. For example, the I/O chassis located in cabinet 1, enclosure 4 and I/O chassis slot 1 is identified as 1/4/1.
- C Display power status of all cells. By default the scope is the entire complex if the -l option is not specified.
- I Display power status of all I/O chassis. By default the scope is the entire complex if the -l option is not specified.
- l cabinet Limit the scope of the -C or -I option to a given *cabinet*.

**frupower** only powers off a cell that is not active in a partition. Also, an I/O chassis connected to active cells cannot be powered off.

Root permissions are required to run this command.

**EXIT STATUS**

The **frupower** utility exits with one of the following values:

- 0 Successful completion.
- 1 Error condition occurred.

**EXAMPLES**

Power on the cell located in cabinet 2 slot 4:

```
frupower -o -c 2/4
```

Power off 2 cells located in cabinet 2, slot 4 and cabinet 2, slot 6.

```
frupower -f -c 2/4 -c 2/6
```

**WARNINGS**

The presence of ? in the command output indicates a problem in address translation. If you see ? (example: "I/O chassis 0/?/3") in the command output, please contact your Hewlett-Packard representative.

**AUTHOR**

**frupower** was developed by the Hewlett-Packard Company.

**SEE ALSO**

fruled(1), parstatus(1), partition(1), parcreate(1M), parmodify(1M), parremove(1M), parunlock(1M).

  
f

**NAME**

fsadm - a file system administration command

**SYNOPSIS**

```
/usr/sbin/fsadm [-F FStype] [-V] [-o specific_options] special
```

**DESCRIPTION**

The **fsadm** command is designed to perform selected administration tasks on file systems. These tasks may differ between file system types. *special* is a device file containing an unmounted file system. However, if the file system is of the type that provides online administration capabilities the *special* could be a *directory*. *directory* must be the root of a mounted file system.

Only a superuser can invoke **fsadm**.

**Options**

- F *FStype*** Specify the file system type on which to operate (see *fstyp*(1M) and *fs\_wrapper*(5)). If this option is not included on the command line, then the file system type is determined from the file **/etc/fstab** by matching each *special* with an entry in that file. If there is no entry in **/etc/fstab**, then the file system type is determined from the file **/etc/default/fs**.
- o *specific\_options*** Specify options specific to each file system type. *specific\_options* is a list of comma separated suboptions and/or keyword/attribute pairs intended for a specific *FStype*-specific module of the command. See the file system specific manual entries for a description of the *specific\_options* supported, if any.
- V** Echo the completed command line, but perform no other action. The command line is generated by incorporating the user-specified options and other information derived from **/etc/fstab**. This option allows the user to verify the command line.

**EXAMPLES**

Convert a HFS file system from a **nolargefiles** file system to a **largefiles** file system:

```
fsadm -F hfs -o largefiles /dev/vg02/lvol1
```

Display HFS relevant file system statistics:

```
fsadm -F hfs /dev/vg02/lvol1
```

**FILES**

**/etc/fstab** Static information about the systems

**SEE ALSO**

fsadm\_hfs(1M), fsadm\_vxfs(1M), fsck(1M), fstab(4), fs\_wrapper(5).

**NAME**

fsadm - an HFS file system administration command

**SYNOPSIS**

`/usr/sbin/fsadm [-F hfs] [-V] [-o specific_options] special`

**DESCRIPTION**

The **fsadm** command is designed to perform selected administration tasks on a HFS file systems. *special* is a device file containing an unmounted file system.

Only a superuser can invoke **fsadm**.

**Options**

**-F hfs** Specify the HFS file system type.

**-o *specific\_options***

Specify a list of comma separated suboptions and/or keyword/attribute pairs from the list below. The following *specific\_options* are valid on HFS file systems.

**largefiles** Converts a **nolargefiles** file system to a **largefiles** file system. The file system should be unmounted and must be in a clean state (see *fsck*(1M)). A **largefiles** file system supports file sizes greater than 2 gigabytes.

**nolargefiles**

Converts a **largefiles** file system to a **nolargefiles** file system. The file system should be unmounted and must be in a clean state (see *fsck*(1M)). All **largefiles** should be purged from the file system for the conversion to succeed.

**-V** Echo the completed command line, but perform no other action. The command line is generated by incorporating the user-specified options and other information derived from */etc/fstab*. This option allows the user to verify the command line.

**DIAGNOSTICS**

Error and warning messages may originate from **fsadm** and **fsck**. See *fsadm*(1M) or *fsck*(1M) to interpret the error and warning messages.

**EXAMPLES**

Convert a **nolargefiles** HFS file system to a **largefiles** HFS file system:

```
fsadm -F hfs -o largefiles /dev/vg02/rlvol1
```

Convert a **largefiles** HFS file system to a **nolargefiles** file system:

```
fsadm -F hfs -o nolargefiles /dev/vg02/rlvol1
```

Display relevant HFS file system statistics:

```
fsadm -F hfs /dev/vg02/rlvol1
```

**WARNINGS**

The size of a file system will impact the performance of the **fsadm** command.

During conversion from largefiles file system to a nolargefiles file system **fsadm** scans the entire file system for a large file. This functionality degrades the performance of the **fsadm** command.

**FILES**

*/etc/fstab* Static information about the systems

**SEE ALSO**

*fsadm*(1M), *fsadm\_vxfs*(1M), *fsck*(1M), *fstab*(4), *fs\_wrapper*(5).



**NAME**

fsadm - resize or reorganize a VxFS file system

**SYNOPSIS**

```
/usr/sbin/fsadm [-F vxfs] [-V] [-d] [-D] [-e] [-E] [-s] [-v] [-a days]
                 [-l largesize] [-p passes] [-r rawdev] [-t time] mount_point

/usr/sbin/fsadm [-F vxfs] [-V] [-b newsize] [-r rawdev] mount_point

/usr/sbin/fsadm [-F vxfs] [-V] [-o largefiles | nolargefiles] mount_point | special

/usr/sbin/fsadm [-F vxfs] [-V] [-c] mount_point
```

**DESCRIPTION**

**fsadm** performs online administration functions on VxFS file systems. **fsadm** supports file-system resizing, extent reorganization, directory reorganization, and querying or changing the **largefiles** flag. **fsadm** operates on file systems mounted for read/write access, however, the **-o** option can also operate on a special device containing a clean, unmounted file system. Only a privileged user can change the **largefiles** flag on a mounted file system, or resize or reorganize a file system.

VxFS file systems running on HP-UX 10.20 and later contain features that are incompatible with earlier versions of HP-UX and applications. These features are large files (file sizes greater than two gigabytes), and hierarchical storage management via the DMAPI (Data Management Applications Programming Interface).

**Options**

- a days** Consider files not accessed within the specified number of *days* as aged files. The default is 14 days. The **-d** option moves aged files to the end of the directory. The **-a days** option applies only with **-d**.
- b newsize** Resize the file system to *newsize* sectors.
- c** Convert the inode format of files upgraded from the Version 2 disk layout so that they can grow beyond a two-gigabyte offset. This option is required because some file systems upgraded from the Version 2 disk layout could not be extended past two gigabytes or contain more than 8 million inodes.
- D** Report on directory fragmentation. If specified in conjunction with the **-d** option, the fragmentation report is produced both before and after the directory reorganization.
- d** Reorganize directories. Directory entries are reordered to place subdirectory entries first, then all other entries in decreasing order of time of last access. The directory is also compacted to remove free space.
- E** Report on extent fragmentation. If specified in conjunction with the **-e** option, the fragmentation report is produced both before and after the extent reorganization.
- e** Reorganize extents. Minimize file system fragmentation. Files are reorganized to have the minimum number of extents.
- F vxfs** Specify the VxFS file system type.
- l largesize** Large extent size in file system blocks. Indicates the size of extents to be considered as large extents, that is, extents that are immovable while performing an extent defragmentation. The value must be between 8 and 2048 blocks. The default is 64 blocks. The **-l largesize** option applies only with **-E** or **-e**.
- o specific\_options** Specify options specific to the vxfs file system type.  
The following *specific\_options* are valid on a VxFS file system:
  - largefiles**  
Set the **largefiles** flag for the file system. When this flag is set, large files (greater than two gigabytes) can be created on the file system.
  - nolargefiles**  
Clear the **largefiles** flag for the file system. When this flag is not set, large files cannot be created on the file system. Any attempt to clear the flag fails if a

large file exists on the file system.

NOTE: Large files are supported on HP-UX 10.20 systems and above. Be careful when implementing large file system capability. System administration utilities such as backup may not operate correctly if they are not large file aware.

- p passes** Maximum number of *passes* to run. The default is 5 passes. Reorganizations are processed until reorganization is complete, or the specified number of *passes* are run.
- r rawdev** Pathname of raw device to read to determine file layout and fragmentation. This option can be used when **fsadm** cannot determine the raw device.
- s** Print a summary of activity at the end of each pass.
- t time** Maximum time to run. Reorganizations are processed until reorganization is complete, or the time limit has expired. *time* is specified in seconds.
- v** Echo the completed command line, but do not execute the command. The command line is generated by incorporating the user-specified options. This option allows the user to verify the command line.
- v** Specify verbose mode. Report reorganization activity.

If no options are specified, **fsadm** prints the current **largefiles** flag setting, then exits. The **-b**, **-o largefiles**, and **-o nolargefiles** options cannot be specified if any other options are given. If both **-e** and **-d** are specified, **fsadm** first completes the directory reorganization, then does the extent reorganization.

### Largefiles Flag

Files larger than two gigabytes are called large files. The **-o largefiles** and **-o nolargefiles** options change the **largefiles** flag, allowing or disallowing large files in the file system.

Large files can be created only on file systems with disk layout Version 3 or above. A file system with large files cannot be mounted on an HP-UX system older than HP-UX 10.20. Many existing applications cannot operate on large files.

Setting the flag with the **-o largefiles** option succeeds only if the file system has the Version 3 disk layout or above. See **vxupgrade(1M)** for information on how to upgrade a file system from an older disk layout to the current version. Clearing the flag with the **-o nolargefiles** option succeeds only if the flag is set and there are no large files present on the file system. See **mkfs\_vxfs(1M)** and **mount\_vxfs(1M)** for information on creating and mounting file systems with large files.

The **-o largefiles** and **-o nolargefiles** options are the only **fsadm** options that can be used on an unmounted file system. An unmounted file system can be specified by invoking **fsadm** with a special device rather than a mount point. If an unmounted file system is specified, it must be clean.

Changing the **largefiles** flag may require changes to **/etc/fstab**. For example, if **fsadm** is used to set the **largefiles** flag, but **nolargefiles** is specified as a mount option in **/etc/fstab**, the file system is not mountable.

### Defragmentation

For optimal performance, the kernel-extent allocator must be able to find large extents when it wants them. To maintain file-system performance, run **fsadm** periodically against all VxFS file systems to reduce fragmentation. The frequency depends on file system usage and activity patterns, and the importance of performance; typically between once a day and once a month against each file system. The **-v** option can be used to examine the amount of work performed by **fsadm**. You can adjust the frequency of reorganization based on the rate of file system fragmentation.

There are two options that are available to control the amount of work done by **fsadm**. The **-t** option specifies a maximum length of time to run. The **-p** option specifies a maximum number of passes to run. If both are specified, **fsadm** exits if either of the terminating conditions is reached. By default, **fsadm** runs 5 passes. If both the **-e** and **-d** options are specified, **fsadm** runs all the directory reorganization passes before any extent reorganization passes.

**fsadm** uses the file **.fsadm** in the **lost+found** directory as a lock file. When **fsadm** is invoked, it opens the file **lost+found/.fsadm** in the root of the file system specified by *mount\_point*. If the file does not exist, it is created. The **fcntl(2)** system call obtains a write lock on the file. If the write lock fails, **fsadm** assumes that another instance of **fsadm** is running and fails. **fsadm** reports the process ID of the process holding the write lock on the **.fsadm** file.

### File System Resizing

If the **-b** option is specified, **fsadm** resizes the file system whose mount point is *mount\_point*. If *newsize* is larger than the current size of the file system, the file system is expanded to *newsize* sectors. Similarly, if *newsize* is smaller than the current size of the file system, **fsadm** shrinks the file system to *newsize* sectors.

Increasing the size of a file system requires that the file system contain enough free space, prior to the expansion, for the growth of the structural files. In the case where a file system has no free blocks available, the attempt to increase the size of the file system will fail (see *extendfs*(1M) for an alternate method to increase file system size).

In a Version 3 or above disk layout, if there are file system resources in use in the sectors being removed, **fsadm** relocates those resources to sectors staying within the resized file system. The time needed for relocation depends on the number of blocks being moved.

In Version 2 disk layouts, file system structural components are fixed, so reducing the size of a file system fails if there are file system resources in use in the sectors being removed. In that case, a reorganization (using **fsadm -e**) can free busy resources and allow shrinking the file system. If there are still file system structural components within the area to be removed, you must upgrade the file system to a Version 3 or above disk layout to do a resize (see *vxupgrade*(1M)).

### Reporting on Directory Fragmentation

As files are allocated and freed, directories tend to grow and become sparse. In general, a directory is as large as the largest number of files it ever contained, even if some files were subsequently removed.

To obtain a directory fragmentation report, use the command syntax:

```
fsadm -D [-r rawdev] mount_point
```

The following is some example output from the **fsadm -D** command:

```
# fsadm -F vxfs -D /lhome
```

#### Directory Fragmentation Report

|       | Dirs<br>Searched | Total<br>Blocks | Immed<br>Dirs | Immeds<br>to Add | Dirs to<br>Reduce | Blocks to<br>Reduce |
|-------|------------------|-----------------|---------------|------------------|-------------------|---------------------|
| total | 15               | 3               | 12            | 0                | 0                 | 0                   |

The column labeled "Dirs Searched" contains the total number of directories. A directory is associated with the extent-allocation unit containing the extent in which the directory's inode is located. The column labeled "Total Blocks" contains the total number of blocks used by directory extents.

The column labeled "Immed Dirs" contains the number of directories that are immediate, meaning that the directory data is in the inode itself, as opposed to being in an extent. Immediate directories save space and speed up pathname resolution. The column labeled "Immeds to Add" contains the number of directories that currently have a data extent, but that could be reduced in size and contained entirely in the inode.

The column labeled "Dirs to Reduce" contains the number of directories for which one or more blocks could be freed if the entries in the directory are compressed to make the free space in the directory contiguous. Since directory entries vary in length, it is possible that some large directories may contain a block or more of total free space, but with the entries arranged in such a way that the space cannot be made contiguous. As a result, it is possible to have a non-zero "Dirs to Reduce" calculation immediately after running a directory reorganization. The **-v** (verbose) option of directory reorganization reports occurrences of failure to compress free space.

The column labeled "Blocks to Reduce" contains the number of blocks that could be freed if the entries in the directory are compressed.

### Measuring Directory Fragmentation

If the totals in the columns labeled "Dirs to Reduce" are substantial, a directory reorganization can improve performance of pathname resolution. The directories that fragment tend to be the directories with the most activity. A small number of fragmented directories may account for a large percentage of name look-ups in the file system.

### Directory Reorganization

If the **-d** option is specified, **fsadm** reorganizes the directories on the file system whose mount point is *mount\_point*. Directories are reorganized in two ways: compression and sorting.

For compression, **fsadm** moves valid entries to the front of the directory and groups the free space at the end of the directory. If there are no entries in the last block of the directory, the block is released and the directory size is reduced.

If the total space used by all directory entries is small enough, **fsadm** puts the directory in the inode immediate data area.

**fsadm** also sorts directory entries to improve pathname lookup performance. Entries are sorted based on the last access time of the entry. The **-a** option specifies a time interval; 14 days is the default if **-a** is not specified. The time interval is broken up into 128 buckets, and all times within the same bucket are considered equal. All access times older than the time interval are considered equal, and those entries are placed last. Subdirectory entries are placed at the front of the directory and symbolic links are placed after subdirectories, followed by the most-recently-accessed files.

The command syntax for reorganizing directories in a file system is:

```
fsadm -d [-D] [-v] [-s] [-a days] [-p passes] [-r rawdev] [-t time] mount_point
```

The following example shows the output of the **fsadm -d -D** command:

```
#fsadm -F vxfs -d -D -s /opt
```

#### Directory Fragmentation Report

|              | Dirs Searched | Total Blocks | Immed Dirs   | Immeds to Add | Dirs to Reduce | Blocks to Reduce |
|--------------|---------------|--------------|--------------|---------------|----------------|------------------|
| <b>total</b> | <b>34663</b>  | <b>8800</b>  | <b>26655</b> | <b>2569</b>   | <b>2716</b>    | <b>2836</b>      |

#### Directory Reorganization Statistics (pass 1 of 2)

|                 | Dirs Searched | Dirs Changed | Total Ioctl's | Failed Ioctl's | Blocks Reduced | Blocks Changed | Immeds Added |
|-----------------|---------------|--------------|---------------|----------------|----------------|----------------|--------------|
| <b>fset 999</b> | <b>8008</b>   | <b>3121</b>  | <b>5017</b>   | <b>0</b>       | <b>3037</b>    | <b>4428</b>    | <b>2569</b>  |
| <b>total</b>    | <b>8008</b>   | <b>3121</b>  | <b>5017</b>   | <b>0</b>       | <b>3037</b>    | <b>4428</b>    | <b>2569</b>  |

#### Directory Reorganization Statistics (pass 2 of 2)

|                 | Dirs Searched | Dirs Changed | Total Ioctl's | Failed Ioctl's | Blocks Reduced | Blocks Changed | Immeds Added |
|-----------------|---------------|--------------|---------------|----------------|----------------|----------------|--------------|
| <b>fset 999</b> | <b>5439</b>   | <b>552</b>   | <b>2448</b>   | <b>0</b>       | <b>708</b>     | <b>4188</b>    | <b>0</b>     |
| <b>total</b>    | <b>5439</b>   | <b>552</b>   | <b>2448</b>   | <b>0</b>       | <b>708</b>     | <b>4188</b>    | <b>0</b>     |

#### Directory Fragmentation Report

|              | Dirs Searched | Total Blocks | Immed Dirs   | Immeds to Add | Dirs to Reduce | Blocks to Reduce |
|--------------|---------------|--------------|--------------|---------------|----------------|------------------|
| <b>total</b> | <b>34663</b>  | <b>6231</b>  | <b>29224</b> | <b>0</b>      | <b>147</b>     | <b>267</b>       |

The column labeled "Dirs Searched" contains the number of directories searched. Only directories with data extents are reorganized. Immediate directories are skipped. The column labeled "Dirs Changed" contains the number of directories for which a change was made.

The column labeled "Total Ioctl's" contains the total number of VX\_DIRSORT ioctls performed. Reorganization of directory extents is performed using this ioctl.

The column labeled "Failed Ioctl's" contains the number of requests that failed for some reason. The reason for failure is usually that the directory being reorganized is active. A few failures should be no cause for alarm. If the **-v** option is used, all ioctl calls and status returns are recorded.

The column labeled "Blocks Reduced" contains the total number of directory blocks freed by compressing entries. The column labeled "Blocks Changed" contains the total number of directory blocks updated while sorting and compressing entries.

The column labeled "Immeds Added" contains the total number of directories with data extents that were compressed into immediate directories.

### Reporting on Extent Fragmentation

As files are created and removed over time, the free extent map for an allocation unit changes from having one large free area to having many smaller free areas. This process is known as fragmentation. Also, when files increase in size (particularly when growth occurs in small increments) small files can be allocated in multiple extents. In the best case, each file that is not sparse would have exactly one extent (containing the entire file), and the free-extent map is one continuous range of free blocks.

Conversely, in a case of extreme fragmentation, there can be free space in the file system, none of which can be allocated. For example, on Version 2 disk layouts, the indirect-address extent size is always 8K long.

This means that to allocate an indirect-address extent to a file, an 8K extent must be available. If no extent of 8K bytes or larger is available, even though more than 8K of free space is available, an attempt to allocate a file into indirect extents fails and returns ENOSPC.

### Determining Fragmentation

To determine whether a file system is fragmented, the free extents for that file system must be examined. If a large number of small extents are free, then there is fragmentation. If more than half of the amount of free space is taken up by small extents (smaller than 64 blocks), or there is less than 5 percent of total file system space available in large extents, then there is serious fragmentation.

### Running the Extent-Fragmentation Report

The extent-fragmentation report provides detailed information about the degree of fragmentation in a given file system.

The command syntax for an extent-fragmentation report is:

```
fsadm -E [-l largesize] [-r rawdev] mount_point
```

The extent reorganization facility considers some extents to be immovable: that is, if reallocating and consolidating extents does not improve performance, those extents are considered immovable. For example, if a file already contains large extents, reallocating and consolidating these extents does not improve performance. The -l option controls when **fsadm** considers an extent as immovable. By default, *largesize* is 64 blocks, meaning that any extent larger than 64 blocks is considered to be immovable. For the extent-fragmentation report, the value for *largesize* affects which extents are reported as being immovable extents.

The following is an example of the output generated by the **fsadm -E** command:

```
# fsadm -F vxfs -E /home

Extent Fragmentation Report
      Total      Average      Average      Total
      Files     File Blks    # Extents   Free Blks
      9293         115          1       149352

blocks used for indirects: 48
% Free blocks in extents smaller than 64 blks: 10.40
% Free blocks in extents smaller than 8 blks: 0.56
% blks allocated to extents 64 blks or larger: 91.67

Free Extents By Size
      1:         156          2:         140          4:         101
      8:         292         16:         290         32:         241
      64:        155        128:         94        256:         43
      512:        33       1024:        20       2048:         1
      4096:         1       8192:         1       16384:         1
      32768:         1      65536:         0      131072:         0
      262144:         0     524288:         0     1048576:         0
      2097152:         0    4194304:         0     8388608:         0
      16777216:         0    33554432:         0     67108864:         0
      134217728:         0   268435456:         0    536870912:         0
      1073741824:         0  2147483648:         0
```

The numbers in the column "Total Files" indicate the total number of files that have data extents. The column "Average File Blks" contains the average number of blocks belonging to all files. The column "Average # Extents" contains the average number of extents used by files in the file system. The column "Total Free Blks" contains the total number of free blocks in the file system. The total number of blocks used for indirect address extent are reported as "blocks used for indirects".

The general shape of free extent map is also reported. There are two percentages reported: % free extents smaller than 64 blocks and % free extents smaller than 8 blocks. These numbers are typically near zero on an unfragmented file system.

Another metric reported is the percentage of blocks that are part of extents 64 blocks or larger. Files with a single small extent are not included in this calculation. This number is generally large on file systems that contain many large files, and is small on file systems that contain many small files.

The figures under the heading "Free Extents By Size" indicate the totals for free extents of each size. The totals are for free extents of size 1, 2, 4, 8, 16, ... up to a maximum of the number of data blocks in an allocation unit. The totals are similar to the output of the **df -o** command unless there was recent allocation

or deallocation activity (because **fsadm** acts on mounted file systems). These figures provide an indication of fragmentation and extent availability on a file system.

### Extent Reorganization

If the **-e** option is specified, **fsadm** reorganizes the data extents on the file system whose mount point is *mount\_point*. The primary goal of extent reorganization is to defragment the file system.

To reduce fragmentation, extent reorganization tries to place all small files in one contiguous extent. The **-l** option specifies the size of a file that is considered large. The default is 64 blocks. Extent reorganization also tries to group large files into large extents of at least 64 blocks. Extent reorganization can improve performance. Small files can be read or written in one I/O operation. Large files can approach raw-disk performance for sequential I/O operations.

**fsadm** performs extent reorganization on all inodes on the file system. Each pass through the inodes will move the file system closer to optimal organization.

**fsadm** reduces both file fragmentation and free extent fragmentation in each pass. In older versions of VxFS, considerable effort was made to obtain an optimal file system layout. In current versions, **fsadm** relies on VxFS kernel allocation mechanisms to reallocate files in a more favorable extent geometry. At the same time, the kernel allocation mechanism is prevented from using blocks in areas of the free list that **fsadm** tries to make more contiguous.

The command syntax to perform extent reorganization is

```
fsadm -e [-E] [-v] [-s] [-l largesize] [-p passes] [-r rawdev] [-t time] mount_point
```

The following example shows the output from the **fsadm -F vxfs -e -s -E** command:

```
# fsadm -F vxfs -e -s -E /home
```

#### Extent Fragmentation Report

| Total Files | Average File Blks | Average # Extents | Total Free Blks |
|-------------|-------------------|-------------------|-----------------|
| 9293        | 115               | 1                 | 149352          |

blocks used for indirects: 48  
 % Free blocks in extents smaller than 64 blks: 10.40  
 % Free blocks in extents smaller than 8 blks: 0.56  
 % blks allocated to extents 64 blks or larger: 91.67

#### Free Extents By Size

|             |     |             |     |            |     |
|-------------|-----|-------------|-----|------------|-----|
| 1:          | 156 | 2:          | 140 | 4:         | 101 |
| 8:          | 292 | 16:         | 290 | 32:        | 241 |
| 64:         | 155 | 128:        | 94  | 256:       | 43  |
| 512:        | 33  | 1024:       | 20  | 2048:      | 1   |
| 4096:       | 1   | 8192:       | 1   | 16384:     | 1   |
| 32768:      | 1   | 65536:      | 0   | 131072:    | 0   |
| 262144:     | 0   | 524288:     | 0   | 1048576:   | 0   |
| 2097152:    | 0   | 4194304:    | 0   | 8388608:   | 0   |
| 16777216:   | 0   | 33554432:   | 0   | 67108864:  | 0   |
| 134217728:  | 0   | 268435456:  | 0   | 536870912: | 0   |
| 1073741824: | 0   | 2147483648: | 0   |            |     |

#### Pass 1 Statistics

| Extents Searched | Reallocations Attempted | Ioctls Issued | FileBusy | Errors NoSpace | Total |
|------------------|-------------------------|---------------|----------|----------------|-------|
| total 12547      | 287                     | 158           | 0        | 0              | 0     |

#### Pass 2 Statistics

| Extents Searched | Reallocations Attempted | Ioctls Issued | FileBusy | Errors NoSpace | Total |
|------------------|-------------------------|---------------|----------|----------------|-------|
| total 13157      | 148                     | 72            | 0        | 0              | 0     |

#### Extent Fragmentation Report

| Total Files | Average File Blks | Average # Extents | Total Free Blks |
|-------------|-------------------|-------------------|-----------------|
| 9294        | 123               | 1                 | 70400           |

blocks used for indirects: 64  
 % Free blocks in extents smaller than 64 blks: 14.00  
 % Free blocks in extents smaller than 8 blks: 0.70

```

% blks allocated to extents 64 blks or larger: 92.09
Free Extents By Size
      1:      102      2:      79      4:      59
      8:      192     16:     189     32:     150
     64:      64     128:      37     256:      22
    512:       2    1024:       4    2048:       4
   4096:       4    8192:       2   16384:       0
  32768:       0   65536:       0   131072:       0
 262144:       0  524288:       0  1048576:       0
2097152:       0  4194304:       0  8388608:       0
16777216:       0 33554432:       0 67108864:       0
134217728:       0 268435456:       0 536870912:       0
1073741824:       0 2147483648:       0

```

Note that the default five passes were scheduled, but the reorganization finished in two passes.

This file system had a significant amount of free space although there were several free small extents. The situation was corrected by reallocating one or more of the extents on many of the files. The files selected for reallocation in this case are those with extents in the heavily fragmented section of the allocation units. The time it takes to complete extent reorganization varies, depending on the degree of fragmentation, disk speed, and the number of inodes in the file system. In general, extent reorganization takes approximately one minute for every 100 megabytes of disk space.

In the preceding example, the column "Extents Searched" contains the total number of extents examined. The column "Reallocations Attempted" contains the total number of consolidations or merging of extents performed. The column "Ioctl's Issued" contains the total number of reorganization request calls made during the pass. This corresponds closely to the number of files that are being operated on in that pass as most files can be reorganized with a single ioctl. (More than one extent may be consolidated in one operation.)

The column "FileBusy" (located under the heading "Errors") shows the total number of reorganization requests that failed because the file was active during reorganization. The column "NoSpace" (located under the heading "Errors") contains the total number of reorganization requests that failed because an extent presumed free was allocated during the reorganization. The column "Total" (located under the heading "Errors") is the total number of errors encountered during the reorganization and may include errors that were not included with "FileBusy" or "NoSpace."

#### Notes

The online reorganization and online resize features of **fsadm** are available only with the HP OnLineJFS product.

#### FILES

```

lost+found/.fsadm  lock file
/dev/rds*/*        file system devices

```

#### SEE ALSO

fcntl(2), fsadm(1M), mkfs\_vxfs(1M), mount\_vxfs(1M), vxfsio(7), vxupgrade(1M).

f

**NAME**

fscat - cat a VxFS file system

**SYNOPSIS**

```
/usr/sbin/fscat [-F vxfs] [-f output-file] [-o offset] [-l length] [-b block_size] special
```

**DESCRIPTION**

**fscat** provides an interface to a VxFS snapshot file system similar to that provided by **dd** invoked on the block or character special file of regular VxFS file systems.

On most VxFS file systems, the block or character special file for the file system provides access to a raw image of the file system to back up the file system to tape. On a snapshot file system, access to the corresponding block or character special file provides little useful information. **fscat** provides a stream of bytes representing the file system snapshot. This datastream is written by default to standard output, although the **-f output-file** option can be used to specify another destination. The datastream on standard output can be processed several ways, such as in a pipeline, or written to a tape. **fscat** works when executed on the special device of any VxFS file system.

By default, the output is a stream of bytes that starts at the beginning of the file system and continues to the last byte. On a snapshot file system, data is read from the file system using special ioctls on the mount point. On other VxFS file systems, data is read from the specified *special* file. Unless otherwise specified, data is written to standard output.

All numbers entered as option arguments may have **0** as a prefix to indicate octal, or **0x** as a prefix to indicate hexadecimal. A **b** or **B** may be appended to indicate the value is in 512-byte blocks, a **k** or **K** to indicate the value is in kilobytes, an **m** or **M** to indicate the value is in megabytes, or a **g** or **G** to indicate the value is in gigabytes. An appended letter may be separated from the number by a space, in which case the letter and number should be enclosed in a set of quotes. For example:

```
"512 b"
```

All numbers entered as options must be in multiples of 512 bytes. For example, a value of 5713 as an offset is rejected.

**Options**

- b block\_size** Specify the output block size, in bytes. *block\_size* must be less than or equal to 1 megabyte.
- F vxfs** Specify the VxFS file system type.
- f output-file** Specify an output file in which to write the datastream.
- l length** Specify the transfer length, in bytes. A *length* of **0** includes the remainder of the file system after the specified offset.
- o offset** Specify the starting offset in bytes.

**NOTES**

Snapshot file systems are only available with the HP OnLineJFS product.

A snapshot file system cannot be written to. A snapshot file system exists only as long as it is mounted; once unmounted, the special file no longer contains a snapshot file system.

**SEE ALSO**

**dd**(1), **fs\_vxfs**(4), **vxfsio**(7).



**NAME**

fsck - file system consistency check and interactive repair

**SYNOPSIS**

`/usr/sbin/fsck [-F FSType] [-m] [-V] [special ...]`

`/usr/sbin/fsck [-F FSType] [-o FSspecific-options] [-V] [special ...]`

**DESCRIPTION**

The **fsck** command audits and interactively repairs inconsistent conditions for HP-UX file systems on mass storage device files identified by *special*. If the file system is consistent, the number of files on that file system and the number of used and free blocks are reported. If the file system is inconsistent, **fsck** provides a mechanism to fix these inconsistencies, depending on which form of the **fsck** command is used.

*special* represents a special device (e.g., `/dev/rdisk/c1d0s8`).

**Options**

**fsck** recognizes the following options:

- F FSType** Specify the file system type on which to operate (see *fstyp*(1M) and *fs\_wrapper*(5)). If this option is not included on the command line, then the file system type is determined from the file `/etc/fstab` by matching *special* with an entry in that file. If there is no entry in `/etc/fstab`, then the file system type is determined from the file `/etc/default/fs`.
- m** Perform a sanity check only. **fsck** will return 0 if the file system is suitable for mounting. If the file system needs additional checking, the return code is 32. If the file system is mounted, the return code is 33. Error codes larger than 33 indicate that the file system is badly damaged.
- o FSspecific-options** Specify options specific to each file system type. *FSspecific-options* is a list of suboptions and/or keyword/attribute pairs intended for a file-system-specific version of the command. See the file-system-specific manual entries for a description of the *specific\_options* supported, if any.
- V** Echo the completed command line, but perform no other action. The command line is generated by incorporating the user-specified options and other information derived from `/etc/fstab`. This option allows the user to verify the command line.

**RETURN VALUES**

The following values are returned by the **-m** option to **fsck**:

- 0 Either no errors were detected or all errors were corrected.
- 32 The file system needs additional checking.
- 33 The file system is mounted.

Return values greater than 33 indicate that file system is badly corrupted. File system specific versions of **fsck** will have their own additional return values (see *fsck\_cachefs*(1M), *fsck\_hfs*(1M), or *fsck\_vxfs*(1M)).

**WARNINGS**

This command may not be supported for all file system types.

**FILES**

`/etc/default/fs` Specifies the default file system type  
`/etc/fstab` Default list of file systems to check

**SEE ALSO**

*fsck\_cachefs*(1M), *fsck\_hfs*(1M), *fsck\_vxfs*(1M), *mkfs*(1M), *newfs*(1M), *fstab*(4), *fs\_wrapper*(5).

**STANDARDS CONFORMANCE**

**fsck**: SVID3

**NAME**

fsck\_cachefs - check integrity of data cached with CacheFS

**SYNOPSIS**

**fsck -F cachefs** [ **-m** | **-o noclean** ] *cache\_directory*

**DESCRIPTION**

The CacheFS version of the **fsck** command checks the integrity of a cache directory. By default it corrects any CacheFS problems it finds. There is no interactive mode. The most likely invocation of **fsck** for CacheFS file systems is at boot time from an entry in the **/etc/fstab** file.

**Options**

Two command line options are available:

**-m** Check, but do not repair.

**-o noclean** Force a check on the cache even if there is no reason to suspect there is a problem.

**EXAMPLES**

The following example forces a check on the cache directory **/cache3**:

**fsck -F cachefs -o noclean /cache3**

**AUTHOR**

**fsck\_cachefs** was developed by Sun Microsystems, Inc.

**SEE ALSO**

cfsadmin(1M), fsck(1M), mount\_cachefs(1M).

**NAME**

fsck - HFS file system consistency check and interactive repair

**SYNOPSIS**

```
/usr/sbin/fsck [-F hfs] [-m] [-v] [-b blocknum] [special ...]
/usr/sbin/fsck [-F hfs] [-c size] [-f] [-p|-P] [-v] [special ...]
/usr/sbin/fsck [-F hfs] [-b blocknum] [-c size] [-f] [-n|-N|-y|-Y]
               [-q] [-v] [special ...]
```

**DESCRIPTION**

The **fsck** command audits and repairs inconsistent conditions for HFS file systems on mass storage device files identified by *special*. If the file system is consistent, the number of files on that file system and the number of used and free blocks are reported. If the file system is inconsistent, **fsck** provides a mechanism to fix these inconsistencies, depending on which form of the **fsck** command is used.

*special* represents a special device (e.g., `/dev/rdisk/c1d0s8`).

If the target device is a swap device, **fsck** does not continue to process. **fsck** also checks the target device to ensure a mounted file system is not being checked. If a mounted device is specified but the **-f** option is omitted, **fsck** prompts the user for a response.

If the **-p|-P** option is used and *special* is not specified, **fsck** reads the pass numbers in `/etc/fstab` to determine which groups of disks to inspect in parallel, taking maximum advantage of I/O overlap to process the file systems as quickly as possible. The **-p|-P** option is normally used in the script `/sbin/bcheckrc` during automatic reboot.

Normally, the root file system is checked on pass 1, and other "root" (section 0) file systems on pass 2. Other small file systems are checked on separate passes (such as the section 4 file systems on pass 3 and the section 7 file systems on pass 4), and finally the large user file systems are checked on the last pass (for example, pass 5). A pass number of 0 in `/etc/fstab` causes a file system not to be checked. If the optional fields are not present on a line in `/etc/fstab`, **fsck** processes the file system on such lines sequentially after all eligible file systems with positive pass numbers have been processed.

The inconsistencies that **fsck** with the **-p|-P** option corrects are shown below. These are inconsistencies that are correctable without data loss. If it encounters other inconsistencies, it exits with an abnormal return status. For each corrected inconsistency, one or more lines are printed identifying the file system on which the correction will take place and the nature of the correction. Correctable inconsistencies are limited to the following:

- Unreferenced inodes
- Unreferenced continuation inodes (see *inode(4)*)
- Unreferenced pipes and FIFOs
- Link counts in inodes too large
- Missing blocks in the free list
- Blocks in the free list also in files
- Counts in the superblock wrong.

The **-P** option operates in the same manner as the **-p** option except that cleanly unmounted file systems are not checked (see *fsckclean(1M)*). This can greatly decrease the amount of time required to reboot a system that was brought down cleanly.

If the **-p|-P** option is not specified, the pass numbers are ignored and the file systems are checked interactively in the order they are listed in `/etc/fstab`.

Without the **-p|-P** option, **fsck** prompts for concurrence before each correction is attempted when the file system is inconsistent. It should be noted that some corrective actions result in a loss of data. The amount and severity of data loss can be determined from the diagnostic output. The default action for each consistency correction is to wait for the operator to respond **yes** or **no**. If the operator does not have write permission, **fsck** defaults to a **-n** action.

**Options**

**fsck** recognizes the following options:

- F hfs** Specify the HFS file system.
- c size** Set the size of the buffer cache which **fsck** uses to cache disk blocks. *size* is the number of cache blocks, and is between 0 and 100 inclusive. The most common use of this option is

-c 0 to disable all caches, thus reducing memory usage.

-b *blocknum*

Use the specified *blocknum* as the superblock for the file system. An alternate superblock can usually be found at block  $((SBSIZE+BBSIZE)/DEV\_BSIZE)$ , typically block 16. `DEV_BSIZE` is defined in `<sys/param.h>`. You can also find a list of alternate superblocks in `/var/adm/sbtab` (see *mkfs(1M)*).

-f Force **fsck** to check a mounted file system.

-m Perform a sanity check only. Verify whether *special* is mounted, or needs additional checking. Refer to the RETURN VALUE section for more information.

-n|-N Assume a **no** response to all questions asked by **fsck** about repairing a file system. Do not open the file system for writing.

-p "Preen" the file system. Proceed to process and repair file systems without user interaction, as described above. Exit immediately if there is a problem requiring intervention.

-P Same as -p except that cleanly unmounted file systems are not checked.

-q Quiet. Do not print size-check messages in Phase 1. Unreferenced fifos are silently removed. If **fsck** requires it, counts in the superblock and cylinder groups are automatically fixed.

-v Echo the completed command line, but perform no other actions. The command line is generated by incorporating the user-specified options and other information derived from `/etc/fstab`. This option allows the user to verify the command line.

-y|-Y Assume a **yes** response to all questions asked by **fsck** about repairing a file system. This should be used with great caution, because this is a free license to continue after essentially unlimited trouble has been encountered.

In all cases, **fsck** checks the following inconsistencies:

- Blocks claimed by more than one inode or the free list.
- Blocks claimed by an inode or the free list outside the range of the file system.
- Incorrect link counts.
- Size checks:
  - Directory size not of proper format.
- Bad inode format.
- Blocks not accounted for anywhere.
- Directory checks:
  - File pointing to unallocated inode.
  - Inode number out of range.
- Superblock checks:
  - More blocks for inodes than there are in the file system.
- Bad free block list format.
- Total free block and/or free inode count incorrect.
- Invalid continuation inode number in a primary inode.

Orphaned files and directories (allocated but unreferenced) are, with the operator's concurrence, reconnected by placing them in the **lost+found** directory. The name assigned is the inode number. The only restriction is that the directory **lost+found** must have empty slots in which entries can be made. This is accomplished by copying a number of files to the directory, then removing them before **fsck** is executed.

Unreferenced continuation inodes are removed with the -p option, since they do not refer back to the primary inode. When a primary inode contains an invalid continuation inode number, the continuation inode number should be cleared (that is, set to 0). This is not done automatically (with the -p option), because access control list information may have been lost and should be corrected.

After **fsck** has checked and fixed the file system, it stores the correct **fs\_clean** flag in the superblock if it is not already there. For a nonroot file system, **FS\_CLEAN** is stored there. For the root file system, which is mounted at the time of the **fsck**, no changes are required to the superblock if no problems were found and **FS\_OK** was already set.

Checking the raw device is almost always faster.

**RETURN VALUE**

**fsck** returns the following values:

- 0 Either no errors were detected or all errors were corrected.
- 1 A syntax error or other operational error occurred when invoked with the **-V** option.
- 4 Root file system errors were corrected. The system must be rebooted.
- 8 Some uncorrected errors exist on one or more of the file systems checked, there was a syntax error, or some other operational error occurred.
- 12 A signal was caught during processing.
- 32 The file system is unmounted and needs additional checking.
- 33 The file system is mounted.
- 34 The file system is damaged.

**WARNINGS**

**fsck** should not be run on mounted file systems or on the root device. If you do run on mounted file systems, be sure the system is in single-user state (see *shutdown(1M)*).

The special case of the **-c** option, **-c 0**, will disable all internal caches, which will reduce memory usage but may impact performance.

The **-F** option, from prior releases, has been replaced by the **-f** option.

**AUTHOR**

**fsck** was developed by HP, AT&T, the University of California, Berkeley.

**FILES**

- /etc/fstab** Default list of file systems to check.
- /var/adm/sbtabs** List of locations of the superblocks for file systems. The **mkfs** command appends entries to this file.

**STANDARDS CONFORMANCE**

**fsck**: SVID3

**SEE ALSO**

*fsck(1M)*, *fsck\_vxfs(1M)*, *dumpfs(1M)*, *fsckclean(1M)*, *mkfs(1M)*, *newfs(1M)*, *shutdown(1M)*, *fstab(4)*, *fs(4)*, *inode(4)*, *fs\_wrapper(5)*, *acl(5)*.

**NAME**

fsck - check and repair a VxFS file system

**SYNOPSIS**

```
/usr/sbin/fsck [-F vxfs] [-V] [-mnNpPyY] [-o full,nolog] [special... ]
```

**DESCRIPTION**

**fsck** checks VxFS file systems for consistency. Because VxFS records pending file system updates in an intent log, **fsck** typically replays the intent log instead of doing a full structural file system check. You can use options (**-o full** or **-y**) to force a full structural file system check.

If *special* is not specified, **fsck** prompts you with each file system listed in **/etc/fstab** to determine which file system to check, unless you specify **-y** or **-Y**, to automatically answer yes to the prompts.

**Options**

- F vxfs** Specify the VxFS file system type.
- m** Check whether or not the file system is marked clean. This option does not validate the file system. The file system could have been corrupted since it was marked clean (for example, by a system crash), and if so, a mount could fail. In that case, a full **fsck** would be required to clean it. Use **fsck -n** to test for file system corruption.
- n|N** Assume a "no" response to all prompts by **fsck**; do not open the file system for writing, do not replay the intent log. A full file system check is performed.
- p** Produce messages that identify the device being checked.
- P** With VxFS, **-P** is used by **fsck** by default; it does not provide any functionality.
- V** Echo the completed command line, but do not execute the command. The command line is generated by incorporating the user specified options and other information derived from **/etc/fstab**. This option allows the user to verify the command line.
- y|Y** Assume a "yes" response to all prompts by **fsck**. Additionally, if the file system requires a full file system check after the log replay, or if the **nolog** suboption causes the log replay to be skipped and the file system is not clean, then a full file system check is performed.
- o** Specify VxFS file system specific options. These options can be a combination of the following in a comma-separated list:
- full** Perform a full file system check.
- nolog** Do not perform log replay. This option may be used if the log area was physically damaged.

Note: Use **fsck -o full,nolog** on clean file systems. If **fsck -o full** without **nolog** is run on a clean file system, it first replays the intent log. Since the **VX\_FULLFSCK** flag is set, an **fsck** does not update the inode and extent maps before performing the full check, so it reports inconsistencies. Use the **-n** option to verify file system inconsistencies.

Because VxFS maintains an intent log, a complete check is generally not required; the default is to replay the intent log only. If **fsck\_vxfs** detects file system damage or the log replay operation detects damage, an indication that a complete check is required is placed in the super-block. In this case, if the **-y** option was specified, the full check will be run after the log replay. If the **-y** option was not used, **fsck** must be run again, with the **-o full** option to perform the full structural check.

A full check looks for the following inconsistencies:

- Blocks claimed by more than one inode or the free list.
- Blocks claimed by an inode outside the range of the file system.
- Incorrect link counts.
- Size checks:
  - Incorrect number of blocks.
  - Directory entry format.
- Bad inode format.
- Blocks not accounted for anywhere.
- Directory checks:

- File pointing to unallocated inode.
- Inode number out of range.
- Linkage to parent directory.
- Hash chain linkage.
- Free space count.
- Super-block checks:
  - Checksum mismatch.
  - More blocks for inodes than there are in the file system.
- Structural Files:
  - Fileset headers.
  - Object Location Table (OLT).
  - Inode list files.
  - Inode allocation summary files.
  - Attribute files (including Access Control Lists).
  - Attribute link counts.
- Bad free block list format.
- Total free block and/or free inode count incorrect.

Orphaned files and directories (allocated but unreferenced) are, with the user's agreement, reconnected by placing them in the **lost+found** directory. The name assigned is the inode number. The only restriction is that the directory **lost+found** must already exist in the file system's root directory.

## OUTPUT

Structural errors discovered during a full check are displayed on standard output. Responses required during a full check are read from standard input.

The following return codes are used for the **-m** option for all devices other than the one used by the root file system:

- 0 The file system is unmounted and clean.
- 32 The file system is unmounted and needs checking.
- 33 The file system is mounted.
- 34 The stat of the device failed.
- Other The state could not be determined because of an error.

The following return codes are used for the **-m** option for the device used by the root file system:

- 0 The root file system is mounted read-only and is clean, or the root file system is mounted read/write and therefore is clean.
- 32 The root file system is mounted read-only and needs checking.
- 34 The stat of the device failed.
- Other The state could not be determined because of an error.

In most cases, **fsck** prints the following messages:

```
log replay in progress
replay complete - marking super-block as CLEAN
```

If the file system is already clean, **fsck** prints the following message instead:

```
file system is clean - log replay is not required
```

If **fsck** prints any other messages, a full structural check is needed.

## ERROR/DIAGNOSTICS

All error messages that relate to the contents of a file system produced during a log replay are displayed on standard output. All I/O failures and exit messages are displayed on standard error output.

## NOTES

Checking the raw device is almost always faster.

Unlike VxFS running on HP-UX 10.01 and 10.10, a full file system check does not always perform pending extended inode operations. Some extended operations can only be processed when the file system is mounted. A file system that has been marked **CLEAN** can still contain extended operations.

If a structural flaw is detected, the **VX\_FULLFCK** flag will be set on the file system, without operator interaction. If **fsck** was not invoked with the **-y** option, it must be reinvoked with the **-y** or **-o full** option to perform a full **fsck**.

Large files (over two gigabytes) are supported on HP-UX 10.20 systems and above. If **fsck** encounters a large file on an older OS version, it stops without completing the file system check.

**FILES**

**/etc/fstab**          Default list of file systems to check.

**SEE ALSO**

**fsck(1M)**, **mkfs\_vxfs(1M)**, **ncheck\_vxfs(1M)**.

f



**NAME**

fsclean - determine the shutdown status of HFS file systems

**SYNOPSIS**

`/sbin/fsclean [-q] [-v] [special ...]`

**DESCRIPTION**

The **fsclean** command determines the shutdown status of the HFS file system specified by *special* or, in the absence of *special*, the file systems listed in `/etc/fstab` of type **hfs** with the **rw**, **default**, or **ro** options set. All optional fields in `/etc/fstab` must be present for **fsclean** to be able to check each file system.

**fsclean** reads the superblock to determine whether the file system's last shutdown was done correctly, and returns one of the following values:

- 0** All of the checked file systems were shut down correctly.
- 1** One or more checked file systems were not shutdown correctly, implying that **fsck** should be run (see *fsck(1M)*).
- 2** Other error (such as **cannot open the specified device file**).

The **fsclean** command is usually silent.

**Options:**

- q** Check quotas. Instead of checking the file system shutdown status, **fsclean** checks the validity of disk quota statistics. This option is useful for determining whether **quotacheck** should be run (see *quotacheck(1M)*). If *special* is not provided, then all file systems in `/etc/fstab` of type **hfs** with the **rw** (or **default**) and **quota** options are checked.
- v** Be verbose. Prints the status of each file system checked.

**DEPENDENCIES**

**fsclean** only operates on HFS file systems.

**AUTHOR**

**fsclean** was developed by HP.

**FILES**

`/etc/fstab` Default list of file systems to check

**SEE ALSO**

*dumpfs(1M)*, *fsck(1M)*, *fsck\_hfs(1M)*, *mount(1M)*, *quotacheck(1M)*, *quotacheck\_hfs(1M)*, *reboot(1M)*, *fstab(4)*.

**NAME**

fsdb - file system debugger (generic)

**SYNOPSIS**

**/usr/sbin/fsdb** [-F *FStype*] [-o *specific\_options*] [-V] *special*

**Remarks**

Always execute the **fsck** command (see *fsck(1M)*) after running **fsdb**.

**DESCRIPTION**

The **fsdb** command can be used to patch up a damaged file system after a crash. It is intended for experienced users only. The file system type to be debugged is specified as *FStype*. Each file system type has a unique structure requiring different debugging capabilities. The manual entries for the file-system-specific **fsdb** should be consulted before attempting any debugging or modifications.

**Options and Arguments**

**fsdb** recognizes the following options and arguments:

- |                                   |                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>special</i>                    | The file name of the special file containing the file system.                                                                                                                                                                                                                                                                                                                                                    |
| <b>-F</b> <i>FStype</i>           | Specify the file system type on which to operate (see <i>fstyp(1M)</i> and <i>fs_wrapper(5)</i> ). If this option is not included on the command line, then the file system type is determined from the file <b>/etc/fstab</b> by matching <i>special</i> with an entry in that file. If there is no entry in <b>/etc/fstab</b> , then the file system type is determined from the file <b>/etc/default/fs</b> . |
| <b>-o</b> <i>specific_options</i> | Specify suboptions specific to each file system type. <i>specific_options</i> is a comma-separated list of suboptions and/or keyword/attribute pairs supported by the specific <i>FStype</i> .                                                                                                                                                                                                                   |
| <b>-V</b>                         | Echo the completed command line, but perform no other action. The command line is generated by incorporating the user-specified options and other information derived from the <b>/etc/fstab</b> file. This option allows the user to verify the command line.                                                                                                                                                   |

**EXAMPLES**

Invoke the file system debugger on HFS file system **/dev/dsk/c1d2s0**:

```
fsdb -F hfs /dev/dsk/c1d2s0
```

Display a completed command line without executing the debugger:

```
fsdb -V /dev/dsk/c1d2s0
```

The previous command might display:

```
fsdb -F hfs /dev/dsk/c1d2s0
```

**WARNINGS**

Only experienced users should use **fsdb**. The failure to fully understand the usage of **fsdb** and the file system's internal organization can lead to complete destruction of the file system and total loss of data.

**AUTHORS**

**fsdb** was developed by HP and AT&T.

**FILES**

|                        |                                           |
|------------------------|-------------------------------------------|
| <b>/etc/default/fs</b> | Specifies the default file system type    |
| <b>/etc/fstab</b>      | Static information about the file systems |

**SEE ALSO**

*fsck(1M)*, *fsdb\_hfs(1M)*, *fsdb\_vxfs(1M)*, *fstyp(1M)*, *stat(2)*, *fs\_wrapper(5)*.

**STANDARDS CONFORMANCE**

**fsdb**: SVID3

**NAME**

fsdb - HFS file system debugger

**SYNOPSIS**

/usr/sbin/fsdb [-F hfs] [-V] *special* [-b *blocknum*] [-]

**Remarks**

Always execute the **fsck** command (see *fsck*(1M)) after running **fsdb**.

**DESCRIPTION**

The **fsdb** command can be used to patch up a damaged file system after a crash.

**Options and Arguments**

**fsdb** recognizes the following options and arguments.

- |                    |                                                                                                                                                                                                                                                                |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>special</i>     | The file name of the special file containing the file system.                                                                                                                                                                                                  |
| -                  | Initially disable the error-checking routines that are used to verify the inode and fragment addresses. See the <b>O</b> symbol. If used, this option must follow <i>special</i> on the command line.                                                          |
| -b <i>blocknum</i> | Use <i>blocknum</i> as the superblock for the file system. If used, this option must follow <i>special</i> on the command line.                                                                                                                                |
| -F hfs             | Specify the HFS file system type.                                                                                                                                                                                                                              |
| -V                 | Echo the completed command line, but perform no other action. The command line is generated by incorporating the user-specified options and other information derived from the <b>/etc/fstab</b> file. This option allows the user to verify the command line. |

**Operation**

**fsdb** normally uses the first superblock for the file system, located at the beginning of the disk section, as the effective superblock. An alternate superblock can always be found at block  $((\text{SBSIZE} + \text{BBSIZE}) / \text{DEV\_BSIZE})$ , typically block 16. The **-b** option can be used to specify the superblock location.

**fsdb** deals with the file system in terms of block fragments, which are the unit of addressing in the file system and the minimum unit of space allocation. To avoid possible confusion, *fragment* is used to mean that, and *block* is reserved for the larger true block. **fsdb** has conversions to translate fragment numbers and i-numbers into their corresponding disk addresses. Also included are mnemonic offsets to access different parts of an inode. These greatly simplify the process of correcting control block entries or descending the file system tree.

**fsdb** contains several error-checking routines to verify inode and fragment addresses. These can be disabled if necessary by invoking **fsdb** with the optional **-** argument, or by using the **O** symbol.

Numbers are considered decimal by default. Octal numbers must be prefixed with a zero. Hexadecimal numbers must be prefixed with **0x**. During any assignment operation, numbers are checked for a possible truncation error due to a size mismatch between source and destination.

**fsdb** reads a fragment at a time. A buffer management routine is used to retain commonly used fragments of data in order to reduce the number of read system calls. All assignment operations result in an immediate write-through of the corresponding fragment.

**Symbols**

The following symbols are recognized by **fsdb**:

- |    |                        |
|----|------------------------|
| !  | Escape to shell        |
| #  | Absolute address       |
| +  | Address arithmetic     |
| -  | Address arithmetic     |
| <  | Restore an address     |
| >  | Save an address        |
| =  | Numerical assignment   |
| += | Incremental assignment |
| -= | Decremental assignment |

|   |                                                                                                                  |
|---|------------------------------------------------------------------------------------------------------------------|
| = | Character string assignment                                                                                      |
| b | Convert from fragment number to disk address (historically "block")                                              |
| d | Directory slot offset                                                                                            |
| f | File print facility                                                                                              |
| i | Convert from i-number to inode address; for continuation inodes as well as primary inodes (see <i>inode(4)</i> ) |
| p | General print facility                                                                                           |
| q | Quit                                                                                                             |
| B | Byte mode                                                                                                        |
| D | Double-word mode                                                                                                 |
| O | Error checking flip-flop                                                                                         |
| W | Word mode                                                                                                        |
| X | Hexadecimal flip-flop                                                                                            |

Dots, tabs, and spaces can be used as function delimiters, but are not necessary. A line with just a newline character increments the current address by the size of the data type last printed. That is, the address is set to the next byte, word, double word, directory entry, or inode, allowing the user to step through a region of a file system.

Information is printed in a format appropriate to the data type. If the **X** toggle is off, bytes, words, and double words are printed in the form:

*octal-address* : *octal-value* (*decimal-value*)

If the **X** toggle is on, bytes, words, and double words are printed in the form:

*hex-address* : *hex-value*

If the **B** (byte) or **D** (double-word) mode is in effect, the colon (:) shown above is preceded by **.B** or **.D**, respectively.

Directories are printed as a directory slot offset followed by the decimal i-number and the character representation of the entry name.

Inodes are printed with labeled fields describing each element.

### Print Facilities

The print facilities generate a formatted output in various styles. Octal numbers are prefixed with a zero. Hexadecimal numbers are prefixed with **0x**. The current address is normalized to an appropriate boundary before printing begins. It advances with the printing and is left at the address of the last item printed. The output can be terminated at any time by typing the interrupt character. If a number follows the **p** symbol, that many entries are printed. A check is made to detect fragment boundary overflows since logically sequential blocks are generally not physically sequential. If a count of zero is used, all entries to the end of the current fragment are printed. The print options available are:

|   |                                           |
|---|-------------------------------------------|
| b | Print as octal bytes                      |
| c | Print as characters                       |
| d | Print as directories                      |
| e | Print as decimal words                    |
| i | Print as inodes (primary or continuation) |
| o | Print as octal words                      |
| x | Print as hexadecimal words                |

The **f** symbol prints data fragments associated with the current inode. If followed by a number, that fragment of the file is printed. (Fragments are numbered from zero). The desired print option letter follows the fragment number, if present, or the **f** symbol. This print facility works for small as well as large files except for special files such as FIFOs, and device special files.

### Inode and Directory Mnemonics

The following mnemonics are used for inode examination and refer to the current working inode:

|                     |                                                         |
|---------------------|---------------------------------------------------------|
| <b>a</b> <i>num</i> | Data block numbers ( <i>num</i> is in the range 0 – 14) |
| <b>at</b>           | Time last accessed                                      |
| <b>ci</b>           | Continuation inode number                               |
| <b>ct</b>           | Last time inode changed                                 |
| <b>gid</b>          | Group ID number                                         |
| <b>ln</b>           | Link count                                              |

|            |                        |
|------------|------------------------|
| <b>maj</b> | Major device number    |
| <b>md</b>  | Mode                   |
| <b>min</b> | Minor device number    |
| <b>mt</b>  | Time last modified     |
| <b>sz</b>  | File size in byte unit |
| <b>uid</b> | User ID number         |

The following mnemonics are used for directory examination:

|           |                                            |
|-----------|--------------------------------------------|
| <b>di</b> | I-number of the associated directory entry |
| <b>nm</b> | Name of the associated directory entry     |

### EXAMPLES

|                        |                                                                                                                                                                                                          |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>386i</b>            | Print i-number 386 in an inode format. This now becomes the current working inode.                                                                                                                       |
| <b>ln=4</b>            | Change the link count for the working inode to 4.                                                                                                                                                        |
| <b>ln+=1</b>           | Increment the link count by 1.                                                                                                                                                                           |
| <b>fc</b>              | Print in ASCII fragment zero of the file associated with the working inode.                                                                                                                              |
| <b>2i.fd</b>           | Print the first fragment-size piece of directory entries for the root inode of this file system.                                                                                                         |
| <b>d5i.fc</b>          | Change the current inode to that associated with the fifth directory entry (numbered from zero) found from the above command. The first fragment's worth of bytes of the file are then printed in ASCII. |
| <b>1b.px</b>           | Print the first fragment of the superblock of this file system in hexadecimal.                                                                                                                           |
| <b>2i.a0b.d7=3</b>     | Change the i-number for the seventh directory slot in the root directory to 3. This example also shows how several operations can be combined on one command line.                                       |
| <b>d7.nm="newname"</b> | Change the name field in the directory slot to the given string. Quotes are optional if the first character of the name field is alphabetic.                                                             |
| <b>a2b.p0d</b>         | Print the third fragment of the current inode as directory entries.                                                                                                                                      |

### WARNINGS

Only experienced users should use **fsdb**. The failure to fully understand the usage of **fsdb** and the file system's internal organization can lead to complete destruction of the file system and total loss of data.

### AUTHOR

**fsdb** was developed by HP and AT&T.

### FILES

**/etc/fstab** Static information about the file systems

### SEE ALSO

**dumpfs(1M)**, **fsck(1M)**, **fsdb(1M)**, **stat(2)**, **dir(4)**, **fs(4)**.

### STANDARDS CONFORMANCE

**fsdb**: SVID3

**NAME**

fsdb - VxFS file system debugger

**SYNOPSIS**

`/usr/sbin/fsdb [-F vxfs] [-z inumber] special`

**DESCRIPTION**

**fsdb** can analyze VxFS file system problems or repair a damaged VxFS file system after a crash. A special device *special* specifies the file system to debug. The **fsdb** command is intended for experienced users only.

**fsdb** can convert block and inumbers into their corresponding disk addresses. In addition, mnemonic offsets allow access to different parts of an inode. These greatly simplify the process of correcting control block entries or descending the file system tree.

By default, numbers are considered decimal. You must prefix octal numbers with 0, and prefix hexadecimal numbers with 0x. When using hexadecimal numbers, it is preferable to follow the number with a space, because several commands are letters that are also hexadecimal digits. In this document a pound sign (#) indicates that a number is to be specified.

**fsdb** reads a block at a time and works with raw and block I/O. All I/O is unbuffered, so changes made to the file system are immediate and changes made by other processes or by the kernel are immediately seen by **fsdb**.

**Options**

- F vxfs** Specify the VxFS file-system type.
- z inumber** Clear the inode identified by *inumber* (non-interactive). Multiple **-z** options accumulate.

The following symbols are recognized by the **fsdb** command:

- h[mod|print]** Print summary of commands that display [modify|format] the file system.
- ?[mod|print]** Print summary of commands that display [modify|format] the file system.
- help[mod|print]** Print summary of commands that display [modify|format] the file system.
- !** Escape to shell.
- |** Pipe output of **fsdb** to a shell command.
- q** Quit.
- string* A character string. Inside a character string, a NULL character may be specified with "\0"; a double quote may be specified with "\""; and a backslash may be specified with "\\ .
- + - \* / %** Add, subtract, multiply, divide, and modulus.
- =** Assignment
- i** An inode in the primary inode list.
- ai** An inode in the attribute inode list.
- au** An allocation unit.
- b** A block.
- im** The immediate data area of an inode. Small directories and symbolic link files (96 bytes or less) are stored directly in the inode itself, in the area normally occupied by data block numbers and extent sizes.
- attr** An attribute inode.
- cdb** Current directory block.
- d** A directory entry.

|              |                                                       |
|--------------|-------------------------------------------------------|
| <b>a</b>     | An inode address entry.                               |
| <b>B</b>     | A byte.                                               |
| <b>H</b>     | A half-word (2 bytes)                                 |
| <b>W</b>     | A word (4 bytes)                                      |
| <b>D</b>     | A double-word (8 bytes)                               |
| <b>p</b>     | General print facility                                |
| <b>calc</b>  | Simple calculator and base converter                  |
| <b>find</b>  | Find a matching pattern in the file system            |
| <b>fset</b>  | A fileset.                                            |
| <b>iau</b>   | An inode allocation unit in the primary inode list.   |
| <b>aiau</b>  | An inode allocation unit in the attribute inode list. |
| <b>cut</b>   | The current usage table.                              |
| <b>olt</b>   | The object location table.                            |
| <b>mapi</b>  | Map logical file offset to an inode extent.           |
| <b>reset</b> | Reset device.                                         |

The print facility recognizes the following print formats:

|                    |                                                                 |
|--------------------|-----------------------------------------------------------------|
| <b>S</b>           | Print as a super-block.                                         |
| <b>A</b>           | Print as an allocation-unit header.                             |
| <b>AS</b>          | Print as an auxiliary super-block.                              |
| <b>L</b>           | Print as intent-log records.                                    |
| <b>I</b>           | Print as inodes.                                                |
| <b>T</b>           | Print as typed extent descriptors.                              |
| <b>dent</b>        | Print as directory entries.                                     |
| <b>db</b>          | Print as a directory block.                                     |
| <b>dh</b>          | Print as a directory header.                                    |
| <b>o</b>           | Print as octal words.                                           |
| <b>oB oH oW oD</b> | Print as octal bytes, half-words, words, or double-words.       |
| <b>x</b>           | Print as hexadecimal words.                                     |
| <b>xB xH xW xD</b> | Print as hexadecimal bytes, half-words, words, or double-words. |
| <b>e</b>           | Print as decimal words.                                         |
| <b>eB eH eW eD</b> | Print as decimal bytes, half-words, words, or double-words.     |
| <b>c</b>           | Print as characters.                                            |
| <b>F</b>           | Print as fileset headers.                                       |
| <b>C</b>           | Print as current usage table entries.                           |
| <b>IA</b>          | Print as an inode allocation unit header.                       |
| <b>oltext</b>      | Print as an object location table extent.                       |
| <b>Q</b>           | Print as a BSD quota record.                                    |
| <b>DV</b>          | Print as a device record.                                       |

Changes to inode fields may be made symbolically. The following symbols represent inode fields:

|                 |                                                  |
|-----------------|--------------------------------------------------|
| <b>md</b>       | Inode mode field                                 |
| <b>ln</b>       | Inode link count field                           |
| <b>uid</b>      | Inode user ID Number field                       |
| <b>gid</b>      | Inode group ID Number field                      |
| <b>szlo</b>     | Low-order word of inode file size field          |
| <b>szhi</b>     | High-order word of inode file size field         |
| <b>sz</b>       | Inode file size field                            |
| <b>de#</b>      | Inode direct extent data block numbers (0 - 9)   |
| <b>des#</b>     | Inode direct extent sizes (0 - 9)                |
| <b>ie#</b>      | Inode indirect extent data block numbers (0 - 1) |
| <b>ies</b>      | Inode indirect extent size                       |
| <b>at</b>       | Inode access time field (seconds)                |
| <b>ats</b>      | Inode access time field (microseconds).          |
| <b>ct</b>       | Inode change time field (seconds).               |
| <b>cts</b>      | Inode change time field (microseconds).          |
| <b>mt</b>       | Inode modification time field (seconds).         |
| <b>mts</b>      | Inode modification time field (microseconds).    |
| <b>af</b>       | Inode allocation flags field.                    |
| <b>gen</b>      | Inode generation count field.                    |
| <b>org</b>      | Inode mapping type field.                        |
| <b>fe</b>       | Inode fixed extent size field.                   |
| <b>bl</b>       | Inode blocks held field.                         |
| <b>eo_flg</b>   | Inode extended operation flag field.             |
| <b>eo_dat</b>   | Inode extended operation data field.             |
| <b>rdev</b>     | If device, inode device number.                  |
| <b>maj</b>      | If device, inode major number.                   |
| <b>min</b>      | If device, inode minor number.                   |
| <b>pd</b>       | If directory, inode parent directory.            |
| <b>res</b>      | If regular file, inode reservation.              |
| <b>verhi</b>    | Inode high-order word of serial number.          |
| <b>verlo</b>    | Inode low-order word of serial number.           |
| <b>fsindex</b>  | Referencing fileset ID.                          |
| <b>matching</b> | Inode number of matching inode.                  |
| <b>iano</b>     | Indirect attribute inode.                        |

Changes to directory block fields may be made symbolically. The following symbols represent directory block fields:

|              |                                                           |
|--------------|-----------------------------------------------------------|
| <b>tfree</b> | Total free space (only if in a data block).               |
| <b>hash#</b> | Hash chain start (0 through 31, only if in a data block). |
| <b>d#</b>    | Directory entry (variable number of entries).             |
| <b>nhash</b> | Number of hash chains.                                    |

Changes to directory entry fields may be made symbolically. The following symbols represent directory entry fields:



|               |                                          |
|---------------|------------------------------------------|
| <b>ino</b>    | Inode number                             |
| <b>nm</b>     | Entry name                               |
| <b>nmlen</b>  | Name length                              |
| <b>reclen</b> | Record length (only if in a data block)  |
| <b>hnext</b>  | Name hash next (only if in a data block) |

It is best to separate each token on a command line with a space. Although the command parser does not require element separation, there is no ambiguity in the command language if each token is separated with a space. For example, the command **0x23b b** sets the current position to block 0x23b hexadecimal. The command **0x23bb** is invalid, because the command is parsed as simply a hexadecimal number. The command **23b** positions to block 23 decimal, since the command is not ambiguous.

You can separate commands with new lines. You can put multiple commands on one line and separate them with a dot (.) or a semicolon (;). If multiple commands are placed on one line, generally only the last command displays results. This allows positioning commands to be followed by printing commands or change commands without intermediate printing.

**fsdb** maintains several positions in the file system:

- current position
- current primary-inode position (i)
- current attribute-inode position (ai)
- current inode type (i or ai)
- current fileset-header position (fset)
- current allocation-unit position (au)
- current primary-inode allocation-unit (iau) position
- current inode allocation-unit type (iau or aiau)
- current attribute-inode allocation-unit (aiau) position

These are used by various **fsdb** commands. (The au positions are only supported on the Version 2 disk layout.)

The following commands are supported:

- # B|H|W|D** Set the current position in the file system to the specified offset in bytes, half-words, words, or double-words. If it is the last command on a line, print the byte, half-word, word, or double-words in hexadecimal.
- +|- # B|H|W|D** Set the current position to the specified relative offset in bytes, half-words, words, or double-words. If it is the last command on a line, print the byte, half-word, word, or double-words in hexadecimal.
- # au** Set the current position in the file system to the specified allocation unit (au) position. Set the current allocation unit position to the resulting offset. If it is the last command on a line, print the allocation unit header.
- +|- # au** Set the current position in the file system to the specified position relative to the current allocation unit (au) position. Set the current allocation unit position to the resulting offset. If it is the last command on a line, print the allocation unit header.
- au** Set the current position in the file system to the current allocation unit position. If it is the last command on a line, print the allocation unit header.
- # b** Set the current position in the file system to the specified offset in blocks. Set the current block position to the resulting offset. The block size is the block size of the file system. If it is the last command on a line, print the first word in the block in hexadecimal.
- +|- # b** Set the current position to specified relative offset in blocks. Set the current block position to the resulting offset. If it is the last command on a line, print the first word in the block in hexadecimal.

|                   |                                                                                                                                                                                                                                                                                                                         |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>b</b>          | Set the current position to current block position (the block specified by the last [+ -] # b operation). If it is the last command on a line, print the first word in the block in hexadecimal.                                                                                                                        |
| <b>cut</b>        | Set the current position to the current usage table (cut). If it is the last command on a line, print the first current usage table entry.                                                                                                                                                                              |
| <b>dev</b>        | Set the current position to the primary device's configuration record. If it is the last command on a line, print the device-configuration record.                                                                                                                                                                      |
| <b># fset</b>     | Set the current position in the file system to the fileset header entry for the specified fileset index. Set the current fileset position to the resulting offset. If it is the last command on a line, print the specified fileset header.                                                                             |
| <b>+ - # fset</b> | Set the current position in the file system to the fileset header entry for the specified position relative to the current fileset position. Set the current fileset position to resulting offset. If it is the last command on a line, print the specified fileset header.                                             |
| <b>fset</b>       | Set the current position in the file system to the current fileset position. If it is the last command on a line, print the fileset header for the current fileset.                                                                                                                                                     |
| <b># aiau</b>     | Set the current position in the file system to the specified attribute inode allocation unit (aiau) in a fileset. Set the current attribute inode allocation unit position to the resulting offset. If it is the last command on a line, print the attribute inode allocation unit header.                              |
| <b>+ - # aiau</b> | Set the current position in the file system to the specified position relative to the current attribute inode allocation unit (aiau) position. Set the current attribute inode allocation unit position to the resulting offset. If it is the last command on a line, print the attribute inode allocation unit header. |
| <b>aiau</b>       | Set the current position in the file system to the current attribute inode allocation unit (aiau) position. If it is the last command on a line, print the attribute inode allocation unit header.                                                                                                                      |
| <b># iau</b>      | Set the current position in the file system to the specified inode allocation unit (iau) in a fileset. Set the current inode allocation unit position to the resulting offset. If it is the last command on a line, print the inode allocation unit header.                                                             |
| <b>+ - # iau</b>  | Set the current position in the file system to the specified position relative to the current inode allocation unit (iau) position. Set the current inode allocation unit position to the resulting offset. If it is the last command on a line, print the inode allocation unit header.                                |
| <b>iau</b>        | Set the current position in the file system to the current inode allocation unit (iau) position. If it is the last command on a line, print the inode allocation unit header.                                                                                                                                           |
| <b># ai</b>       | Set the current position in the current fileset to the ilist entry for the specified attribute inode. Set the current attribute inode position to the resulting offset. If it is the last command on a line, print the ilist entry for the inode.                                                                       |
| <b>+ - # ai</b>   | Set the current position in the current fileset to the ilist entry for the specified relative attribute inode. Set the current attribute inode position to the resulting offset. If it is the last command on a line, print the ilist entry for the inode.                                                              |
| <b>ai</b>         | Set the current position in the current fileset to the current attribute inode position. If it is the last command on a line, print the ilist entry for the inode.                                                                                                                                                      |
| <b># i</b>        | Set the current position in the current fileset to the ilist entry for the specified inode. Set the current inode position to the resulting offset. If it is the last command on a line, print the ilist entry for the inode.                                                                                           |
| <b>+ - # i</b>    | Set the current position in the current fileset to the ilist entry for the specified relative inode. Set the current inode position to the resulting offset. If it is the last command on a line, print the ilist entry for the inode.                                                                                  |
| <b>i</b>          | Set the current position in the current fileset to the current inode position. If it is the last command on a line, print the ilist entry for the inode.                                                                                                                                                                |
| <b>a#</b>         | Set the current position to specified offset in blocks specified by the inode address #. Addresses 0 through 9 are for direct extents ( de ). Addresses 10-11 are for indirect                                                                                                                                          |

|                                  |                                                                                                                                                                                                                                                                                                                                            |
|----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                  | extents ( ie ). The addresses are displayed when printing an ilist entry. Set the current block position to the resulting offset. If it is the last command on a line, print the first word in the block in hexadecimal.                                                                                                                   |
| <b>im</b>                        | Set the current position to immediate data area of the current inode. Set the current block position to the resulting offset. If it is the last command on a line, print the first word of the area in hexadecimal.                                                                                                                        |
| <b>attr</b>                      | Set the current position to attribute data area of the current inode. Set the current block position to the resulting offset. If it is the last command on a line, print the first word in the block in hexadecimal.                                                                                                                       |
| <b># B H W D =# [#]</b>          | Set the current position and change the number at the specified offset to the given number. If a double-word offset is specified, then two numbers separated by a space are required. The resulting value is printed in hexadecimal.                                                                                                       |
| <b>+ -# B H W D =# [#]</b>       | Set the current position and change the number at the specified relative offset to the given number. If a double-word offset is specified, then two numbers separated by a space are required. The resulting value is printed in hexadecimal.                                                                                              |
| <b># B H W D = string</b>        | Set the current position and change the characters at the specified offset to the given string. The resulting value is printed as a character string.                                                                                                                                                                                      |
| <b>+ - # B H W D = string</b>    | Set the current position and change the characters at the specified relative offset to the given string. The resulting value is printed as a character string.                                                                                                                                                                             |
| <b>olt</b>                       | Set the current position to the object location table (olt). If it is the last command on a line, print the object location table.                                                                                                                                                                                                         |
| <b>p [#] format</b>              | Print the contents of the file system at the current offset as the specified number of entries of a given format. The allowable print formats are specified above. If a number of entries to print is not specified, one entry is printed.                                                                                                 |
| <b>inode_field = #</b>           | Set the contents of the given inode field to the specified number. The current inode specifies the inode list entry to be modified. The symbols representing inode fields are listed above.                                                                                                                                                |
| <b>directory_block_field = #</b> | Set the contents of the given directory block field to the specified number. The current block is treated as a directory block and the offset in that block which is represented by the given field is changed. The symbols representing directory block fields are listed above.                                                          |
| <b>d#</b>                        | Set the current directory entry to the specified number. The current block is treated as a directory block. If the current block is an immediate data area for an inode, then the block is treated as containing immediate directory entries. If it is the last command on a line, the directory entry at the resulting offset is printed. |
| <b>directory_entry_field = #</b> | Set the contents of the given directory field to the specified number. The current directory entry specifies where the directory entry is located. The resulting value is printed in hexadecimal.                                                                                                                                          |
| <b>nm = string</b>               | Set the directory name field of the current directory entry to the specified string. The resulting value is printed as a character string.                                                                                                                                                                                                 |
| <b>calc # [+ - * / #]</b>        | Take a number or the sum, difference, product or dividend of two numbers and print in decimal, octal, hexadecimal and character format.                                                                                                                                                                                                    |
| <b>find # B H W D [#]</b>        | Search for the given numeric pattern in the file system. The size of the object to match is specified. If a double-word is specified, then two numbers must be given. The search is performed forward from the current offset. A maximum number of                                                                                         |

blocks to search may be specified. If found, the location and value are printed in hexadecimal.

**find** *string* [#]

Search for the given character string in the file system. The search is performed forward from the current offset. A maximum number of blocks to search may be specified. If found the location and string are printed.

**fmtlog**

Format all intent log entries. A completely formatted intent log can be quite lengthy. It is a good idea use the **fsdb** command as a filter and redirect the output to a file or pager to look at a complete log format.

**listfset**

List all filesets by their indexes and names.

**mapi** #

Treat the number as a logical offset in the file described by the current inode, and print the extent that it maps to.

**reset**

Does the equivalent of exiting **fsdb** and restarting on same device.

The following help commands are supported:

**h|help**

Display primary help screen.

**h mod**

Display modification-commands help screen.

**h print**

Display print-commands help screen.

## EXAMPLES

**386i** Prints inumber 386 in an inode format. This now becomes the current working inode.

**ln=4** Changes the link count for the working inode to 4.

**8192B.p s** Prints the super-block of this file system symbolically.

**d7.nm = foo** Changes the name field in the directory slot to *foo*.

**23i.im.d5** Prints the sixth directory entry in the immediate area of inode 23.

## WARNINGS

Always execute *fsck*(1M) after using the **fsdb** command to modify a file system (use **fsck -o full,nolog**).

## SEE ALSO

*fsck\_vxfs*(1M), *fsdb*(1M).

**NAME**

fsirand - install random inode generation numbers

**SYNOPSIS**

`/usr/sbin/fsirand [-p] special`

**DESCRIPTION**

**fsirand** installs random inode generation numbers on all the inodes on device *special*, and also installs a filesystem ID in the superblock. This process increases the security of filesystems exported by NFS.

Use **fsirand** only on an unmounted filesystem that was checked with **fsck** (see *fsck(1M)*). The only exception is that it can be used on the root filesystem in single-user mode if the system is immediately re-booted afterwards using **reboot -n**.

The **-p** option prints the generation numbers for all inodes.

**WARNINGS**

**fsirand** should not be run on mounted filesystems. If executing **fsirand** on the root filesystem, the system should be in single-user mode and should be re-booted immediately afterwards using **reboot -n**.

**AUTHOR**

**fsirand** was developed by Sun Microsystems, Inc.

**SEE ALSO**

statfs(2).

**NAME**

**fstyp** - determine file system type

**SYNOPSIS**

**/usr/sbin/fstyp** [-v] *special*

**DESCRIPTION**

The **fstyp** command allows the user to determine the file system type of a mounted or unmounted file system. *special* represents a device special file (for example: **/dev/dsk/c1t6d0**).

The file system type is determined by reading the superblock of the supplied *special* file. If the superblock is read successfully, the command prints the file system type identifier on the standard output and exits with an exit status of 0. If the type of the file system cannot be identified, the error message **unknown\_fstyp** (no matches) is printed and the exit status is 1. Exit status 2 is not currently returned, but is reserved for the situation where the file system matches more than one file system type. Any other error will cause exit status 3 to be returned.

The file system type is determined by reading the superblock of the supplied *special* file.

**Options**

**-v** Produce verbose output. The output contains information about the file system's superblock.

**RETURN VALUE**

**fstyp** returns the following values:

|   |                                         |
|---|-----------------------------------------|
| 0 | Successful completion.                  |
| 1 | Unknown file system type.               |
| 2 | File system matches more than one type. |
| 3 | Usage error or access problem.          |

**EXAMPLES**

Find the type of the file system on a disk, **/dev/dsk/c1t6d0**:

```
fstyp /dev/dsk/c1t6d0
```

Find the type of the file system on a logical volume, **/dev/vg00/lvol6**:

```
fstyp /dev/vg00/lvol6
```

Find the file system type for a particular device file and also information about its super block:

```
fstyp -v /dev/dsk/c1t6d0
```

**SEE ALSO**

stat(2), statvfsdev(2).

**NAME**

ftpd - DARPA Internet File Transfer Protocol server

**SYNOPSIS**

```
/usr/sbin/ftpd [-l] [-p] [-v] [-t timeout] [-P] [-T maxtimeout] [-u umask] [-B size] [-a]
[-A] [-L] [-i] [-o] [-m number_of_tries]
```

**DESCRIPTION**

**ftpd** is the DARPA Internet File Transfer Protocol server. It expects to be run by the Internet daemon (see *inetd*(1M) and *inetd.conf*(4)). *inetd* runs **ftpd** when a service request is received at the port indicated in the **ftp** service specification in */etc/services* (see *services*(4)). **ftpd** recognizes the following options and command-line arguments.

- l** Causes each FTP session to be logged in the syslog file.
- p** The default action of **ftpd** does not allow usage of reserved ports as the originating port on the client's system i.e., the PORT command cannot specify a reserved port. This option allows the client to specify a reserved port. Note, allowing usage of reserved ports can result in the misuse of **ftpd**. The security ramifications should be understood before the option is turned on.
- v** The debugging information is written to the syslog file.
- t timeout** Causes **ftpd** to timeout inactive sessions after *timeout* seconds. By default, **ftpd** terminates an inactive session after 15 minutes.
- P** Enables third party transfer.
- T maxtimeout** A client can also request a different timeout period. The **-T** option sets to *maxtimeout* the maximum timeout that client can request, in seconds. By default, the maximum timeout is 2 hours.
- u umask** Change default **ftpd** umask from 027 to *umask*.
- B size** Sets the buffer size of the data socket to *size* blocks of 1024 bytes. The valid range for *size* is from 1 to 64 (default is 56). **NOTE:** A large buffer size will improve the performance of **ftpd** on fast links (e.g. FDDI), but may cause long connection times on slow links (e.g. X.25).
- a** Enables the use of the configuration file */etc/ftpd/ftpaccess*. (see *ftpaccess*(4)).
- A** Disables the use of the configuration file */etc/ftpd/ftpaccess*. (see *ftpaccess*(4)).
- L** Logs all commands sent to the *ftpd*(1M) server to be logged to the syslog. The **-L** option is overridden by */etc/ftpd/ftpaccess* file. (see *ftpaccess*(4)). If the **-L** option is used, commands will be logged to syslog by default.
- i** Logs all the files received by **ftpd** server to **xferlog** (see *xferlog*(5)). This option is overridden by the */etc/ftpd/ftpaccess* file. (see *ftpaccess*(4)).
- o** Logs all files transmitted by **ftpd** to **xferlog** (see *xferlog*(5)). It logs outgoing files from the **ftpd** server. This option is overridden by the */etc/ftpd/ftpaccess* file. (see *ftpaccess*(4)).
- m number\_of\_tries** Specifies the number of tries for a **bind()** socket call.

**ftpd** currently supports the following commands (uppercase and lowercase are interpreted as equivalent):

| Command | Description                                   |
|---------|-----------------------------------------------|
| ABOR    | Abort previous command                        |
| ACCT    | Specify account (ignored)                     |
| ALLO    | Allocate storage (vacuously)                  |
| APPE    | Append to a file                              |
| CDUP    | Change to parent of current working directory |
| CWD     | Change working directory                      |

|              |                                                 |
|--------------|-------------------------------------------------|
| <b>DELE</b>  | Delete a file                                   |
| <b>HELP</b>  | Give help information                           |
| <b>LIST</b>  | Give list files in a directory ( <b>ls -l</b> ) |
| <b>MKD</b>   | Make a directory                                |
| <b>MDTM</b>  | Show last modification time of file             |
| <b>MODE</b>  | Specify data transfer <i>mode</i>               |
| <b>NLIST</b> | Give name list of files in directory            |
| <b>NOOP</b>  | Do nothing                                      |
| <b>PASS</b>  | Specify password                                |
| <b>PASV</b>  | Prepare for server-to-server transfer           |
| <b>PORT</b>  | Specify data connection port                    |
| <b>PWD</b>   | Print the current working directory             |
| <b>QUIT</b>  | Terminate session                               |
| <b>REST</b>  | Restart incomplete transfer                     |
| <b>RETR</b>  | Retrieve a file                                 |
| <b>RMD</b>   | Remove a directory                              |
| <b>RNFR</b>  | Specify rename-from file name                   |
| <b>RNTO</b>  | Specify rename-to file name                     |
| <b>SITE</b>  | Non-standard commands (see next section)        |
| <b>SIZE</b>  | Return size of file                             |
| <b>STAT</b>  | Return status of server                         |
| <b>STOR</b>  | Store a file                                    |
| <b>STOU</b>  | Store a file with a unique name                 |
| <b>STRU</b>  | Specify data transfer <i>structure</i>          |
| <b>SYST</b>  | Show operating system type of server system     |
| <b>TYPE</b>  | Specify data transfer <i>type</i>               |
| <b>USER</b>  | Specify user name                               |
| <b>XCUP</b>  | Change to parent of current working directory   |
| <b>XCWD</b>  | Change working directory                        |
| <b>XMKD</b>  | Make a directory                                |
| <b>XPWD</b>  | Print the current working directory             |
| <b>XRMD</b>  | Remove a directory                              |

The following non-standard or HP-UX specific commands are supported by the **SITE** command:

| Command      | Description                                                        |
|--------------|--------------------------------------------------------------------|
| <b>UMASK</b> | Change umask. (e.g., <b>SITE UMASK 002</b> )                       |
| <b>IDLE</b>  | Set idle-timer. (e.g., <b>SITE IDLE 60</b> )                       |
| <b>CHMOD</b> | Change mode of a file. (e.g., <b>SITE CHMOD 755 filename</b> )     |
| <b>HELP</b>  | Give help information. (e.g., <b>SITE HELP</b> )                   |
| <b>NEWER</b> | List files newer than a particular date.                           |
| <b>MINFO</b> | Works like <b>SITE NEWER</b> , but gives extra information.        |
| <b>GROUP</b> | Request for special group access. (e.g., <b>SITE GROUP foo</b> )   |
| <b>GPASS</b> | Give special group access password. (e.g., <b>SITE GPASS bar</b> ) |
| <b>EXEC</b>  | Execute a program. (e.g., <b>SITE EXEC program params</b> )        |

The remaining FTP requests specified in Internet RFC 959 are recognized, but not implemented. **MDTM** and **SIZE** are not specified in RFC 959, but are expected in the next updated **FTP RFC**.

The FTP server aborts an active file transfer only when the **ABOR** command is preceded by a Telnet "Interrupt Process" (IP) signal and a Telnet "Synch" signal in the command Telnet stream, as described in Internet RFC 959. If **ftpd** receives a **STAT** command during a data transfer, preceded by a Telnet IP and Synch, it returns the status of the transfer.

**ftpd** interprets file names according to the "globbing" conventions used by *csh*(1). This allows users to utilize the metacharacters \*, ., [, ], {, }, ~, and ?.

**ftpd** authenticates users according to three rules:

- The user name must be in the password data base, **/etc/passwd**, and not have a null password. The client must provide the correct password for the user before any file operations can be performed.
- The user name must not appear in the file **/etc/ftpd/ftpusers** (see *ftpusers*(4)).
- The user must have a standard shell returned by **getusershell()**.



Optionally, a system administrator can permit public access or “anonymous FTP.” If this has been set up, users can access the anonymous FTP account with the user name **anonymous** or **ftp** and any non-null password (by convention, the client host’s name). **ftpd** does a **chroot()** to the home directory of user **ftp**, thus limiting anonymous FTP users’ access to the system. If the user name is **anonymous** or **ftp**, an anonymous FTP account must be present in the password file (user **ftp**). In this case the user is allowed to log in by specifying any password (by convention this is given as the user’s e-mail address).

In order to permit anonymous FTP, there must be an entry in the *passwd(4)* database for an account named **ftp**. The password field should be **\***, the group membership should be **guest**, and the login shell should be **/usr/bin/false**. For example (assuming the **guest** group ID is 10):

```
ftp:*:500:10:anonymous ftp:/home/ftp:/usr/bin/false
```

The anonymous FTP directory should be set up as follows:

**~ftp** The home directory of the FTP account should be owned by user **root** and mode 555 (not writable). Since **ftpd** does a **chroot()** to this directory, it must have the following subdirectories and files:

**~ftp/usr/bin**

This directory must be owned by root and mode 555 (not writable). The file **/sbin/ls** should be copied to **~ftp/usr/bin**. This is needed to support directory listing by **ftpd**. The command should be mode 111 (executable only). If the FTP account is on the same file system as **/sbin**, **~ftp/usr/bin/ls** can be hard link, but it may not be a symbolic link, because of the **chroot()**. The command must be replaced when the system is updated.

**~ftp/etc**

This directory must be owned by root and mode 555 (not writable). It should contain versions of the files *passwd* and *group*. See *passwd(4)* and *group(4)*. These files must be owned by root and mode 444 (readable only). These files must be present for the **LIST** command to be able to produce owner names rather than numbers.

**~ftp/etc/passwd**

This file should contain entries for the **ftp** user and any other users who own files under the anonymous **ftp** directory. Such entries should have **\*** for passwords. Group IDs must be listed in the anonymous FTP group file, **~ftp/etc/group**. The path names of home directories in **~ftp/etc/passwd** must be with respect to the anonymous FTP home directory.

**~ftp/etc/group**

This file should contain the group names associated with any group IDs in file **~ftp/etc/passwd** and any group IDs of files in the anonymous FTP subdirectories.

**~ftp/pub** (optional)

This directory is used by anonymous FTP users to deposit files on the system. It should be owned by user **ftp** and should be mode 777 (readable and writable by all).

**~ftp/dist** (optional)

Directories used to make files available to anonymous ftp users should be mode 555 (not writable), and any files to be distributed should be owned by root and mode 444 (readable only) so that they cannot be modified or removed by anonymous FTP users.

**Note:** The steps that are followed to create an anonymous account is used to create a guest account also.

## DIAGNOSTICS

**ftpd** replies to FTP commands to ensure synchronization of requests and actions during file transfers, and to indicate the status of **ftpd**. Every command produces at least one reply, although there may be more than one. A reply consists of a three-digit number, a space, some text, and an end of line. The number is useful for programs; the text is useful for users. The number must conform to this standard, but the text can vary.

The first digit of the message indicates whether the reply is good, bad, or incomplete. Five values exist for the first digit. The values and the interpretations of the values are:

- 1 The requested action is being initiated; expect another reply before proceeding with a new command.

- 2 The requested action is complete. The server is ready for a new request.
- 3 The command has been accepted, but the requested action requires more information.
- 4 The command was not accepted, the requested action failed, but the error condition is temporary and the action can be requested again.
- 5 The command was not accepted, the requested action failed, and the error condition would most likely occur again if the same command sequence is repeated.

The second digit indicates the functional area that the message addresses. The values of the second digit and the interpretations of these values are:

- 0 Syntax. A message with a 0 for the second digit indicates that a syntax error occurred.
- 1 Information. A message with a 1 as the second digit indicates that the message is in reply to a request for information.
- 2 Connections. A message with a 2 as the second digit indicates that the message is a reply to a request for control and data connection information.
- 3 Authentication and accounting. A message with a 3 as the second digit indicates that the message is a reply to a login or accounting procedure.
- 4 Not currently specified.
- 5 File system. A message with a 5 as the second digit indicates that the text following the number contains information concerning the status of the server file system.

The third digit provides a further clarification of the information supplied by the second digit. Following are several examples of messages. Note that **ftpd**'s replies match the number but not the text.

- 110 Restart marker reply. MARK *yyyy=mmmm* where *yyyy* is a user process data stream marker, and *mmmm* is **ftpd**'s equivalent marker
- 120 Service ready in *mmn* minutes
- 200 Command okay
- 211 System status, or system help reply
- 212 Directory status
- 230 User logged in, proceed
- 250 Requested file action okay, completed
- 331 User name okay, need password
- 350 Requested file action pending further information
- 425 Cannot open data connection
- 451 Requested action aborted: local error in processing
- 500 Syntax error, command unrecognized or command line too long
- 530 Not logged in
- 550 Requested action not taken; file unavailable, not found, no access

## WARNINGS

The password is sent unencrypted through the socket connection.

Anonymous FTP is inherently dangerous to system security.

## DEPENDENCIES

### Pluggable Authentication Modules (PAM)

PAM is an Open Group standard for user authentication, password modification, and validation of accounts. In particular, **pam\_authenticate()** is invoked to perform all functions related to login. This includes retrieving the password, validating the account, and displaying error messages.

## AUTHOR

**ftpd** was developed by the University of California, Berkeley and the Washington University, St. Louis, Missouri.

## SEE ALSO

ftp(1), inetd(1M), chroot(2), getusershell(3C), pam\_authenticate(3), inetd.conf(4), ftpaccess(4), ftpusers(4), group(4), passwd(4), xferlog(5).

**NAME**

ftpd - DARPA Internet File Transfer Protocol server

**SYNOPSIS**

```
/usr/sbin/ftpd [-l] [-p] [-v] [-t timeout] [-P] [-T maxtimeout] [-u umask] [-K] [-B size]
[-a] [-A] [-L] [-i] [-o] [-m number_of_tries]
```

**DESCRIPTION**

**ftpd** is the DARPA Internet File Transfer Protocol server. It expects to be run by the Internet daemon (see *inetd*(1M) and *inetd.conf*(4)). *inetd* runs **ftpd** when a service request is received at the port indicated in the **ftp** service specification in */etc/services* (see *services*(4)).

**Options**

**ftpd** recognizes the following options and command-line arguments.

- l** Causes each FTP session to be logged in the syslog file.
- p** The default action of **ftpd** does not allow usage of reserved ports as the originating port on the client's system i.e., the PORT command cannot specify a reserved port. This option allows the client to specify a reserved port. Note, allowing usage of reserved ports can result in the misuse of **ftpd**. The security ramifications should be understood before the option is turned on.
- v** The debugging information is written to the syslog file.
- t *timeout*** Causes **ftpd** to timeout inactive sessions after *timeout* seconds. By default, **ftpd** terminates an inactive session after 15 minutes.
- P** Enables third party transfer.
- T *maxtimeout***  
A client can also request a different timeout period. The **-T** option sets to *maxtimeout* the maximum timeout that client can request, in seconds. By default, the maximum timeout is 2 hours.
- u *umask*** Change default **ftpd** umask from 027 to *umask*.
- K** Applicable only in a secure environment based on Kerberos V5. Causes access to be denied if network authentication fails. See *sis*(5).
- B *size*** Sets the buffer size of the data socket to *size* blocks of 1024 bytes. The valid range for *size* is from 1 to 64 (default is 56). **NOTE:** A large buffer size will improve the performance of **ftpd** on fast links (e.g. FDDI), but may cause long connection times on slow links (e.g. X.25).
- a** Enables the use of the configuration file */etc/ftpd/ftpaccess* (see *ftpaccess*(4)).
- A** Disables the use of the configuration file */etc/ftpd/ftpaccess* (see *ftpaccess*(4)).
- L** Logs all commands sent to the **ftpd** server to be logged to the syslog. The **-L** option is overridden by */etc/ftpd/ftpaccess* file (see *ftpaccess*(4)). If the **-L** option is used, commands will be logged to syslog by default.
- i** Logs all the files received by ftpd server to **xferlog**. This option is overridden by the */etc/ftpd/ftpaccess* file (see *ftpaccess*(4)).
- o** Logs all files transmitted by ftpd to **xferlog**. This option logs outgoing files from the ftpd server. This option is overridden by the */etc/ftpd/ftpaccess* file (see *ftpaccess*(4)).
- m *number\_of\_tries***  
Specifies the number of tries for a **bind()** socket call.

**ftpd** currently supports the following commands (uppercase and lowercase are interpreted as equivalent):

| Command | Description                  |
|---------|------------------------------|
| ABOR    | Abort previous command       |
| ACCT    | Specify account (ignored)    |
| ALLO    | Allocate storage (vacuously) |
| APPE    | Append to a file             |

|       |                                                 |
|-------|-------------------------------------------------|
| CDUP  | Change to parent of current working directory   |
| CWD   | Change working directory                        |
| DELE  | Delete a file                                   |
| HELP  | Give help information                           |
| LIST  | Give list files in a directory ( <b>ls -l</b> ) |
| MKD   | Make a directory                                |
| MDTM  | Show last modification time of file             |
| MODE  | Specify data transfer <i>mode</i>               |
| NLIST | Give name list of files in directory            |
| NOOP  | Do nothing                                      |
| PASS  | Specify password                                |
| PASV  | Prepare for server-to-server transfer           |
| PORT  | Specify data connection port                    |
| PWD   | Print the current working directory             |
| QUIT  | Terminate session                               |
| REST  | Restart incomplete transfer                     |
| RETR  | Retrieve a file                                 |
| RMD   | Remove a directory                              |
| RNFR  | Specify rename-from file name                   |
| RNTO  | Specify rename-to file name                     |
| SITE  | Non-standard commands (see next section)        |
| SIZE  | Return size of file                             |
| STAT  | Return status of server                         |
| STOR  | Store a file                                    |
| STOU  | Store a file with a unique name                 |
| STRU  | Specify data transfer <i>structure</i>          |
| SYST  | Show operating system type of server system     |
| TYPE  | Specify data transfer <i>type</i>               |
| USER  | Specify user name                               |
| XCUP  | Change to parent of current working directory   |
| XCWD  | Change working directory                        |
| XMKD  | Make a directory                                |
| XPWD  | Print the current working directory             |
| XRMD  | Remove a directory                              |

The following commands are supported when **ftpd** is operating in a secure environment which is based on Kerberos V5 (see *sis(5)*).

| Command | Description                                    |
|---------|------------------------------------------------|
| AUTH    | Authentication/security mechanism              |
| ADAT    | Authentication/security data                   |
| CCC     | Clear command channel                          |
| ENC     | Privacy protected command                      |
| MIC     | Integrity protected command                    |
| PROT    | Data channel protection level (level 'C' only) |
| PBSZ    | Protection buffer size (has no effect)         |

These commands are described in draft 8 of the FTP security extensions.

The following non-standard or HP-UX specific commands are supported by the **SITE** command:

| Command | Description                                                        |
|---------|--------------------------------------------------------------------|
| UMASK   | Change umask. (e.g., <b>SITE UMASK 002</b> )                       |
| IDLE    | Set idle-timer. (e.g., <b>SITE IDLE 60</b> )                       |
| CHMOD   | Change mode of a file. (e.g., <b>SITE CHMOD 755 filename</b> )     |
| HELP    | Give help information. (e.g., <b>SITE HELP</b> )                   |
| NEWER   | List files newer than a particular date.                           |
| MINFO   | Works like <b>SITE NEWER</b> , but gives extra information.        |
| GROUP   | Request for special group access. (e.g., <b>SITE GROUP foo</b> )   |
| GPASS   | Give special group access password. (e.g., <b>SITE GPASS bar</b> ) |
| EXEC    | Execute a program. (e.g., <b>SITE EXEC program params</b> )        |

The remaining FTP requests specified in Internet RFC 959 are recognized, but not implemented. **MDTM** and **SIZE** are not specified in RFC 959, but are expected in the next updated **FTP RFC**.

The FTP server aborts an active file transfer only when the **ABOR** command is preceded by a Telnet "Interrupt Process" (IP) signal and a Telnet "Synch" signal in the command Telnet stream, as described in Internet RFC 959. If **ftpd** receives a **STAT** command during a data transfer, preceded by a Telnet IP and Synch, it returns the status of the transfer.

**ftpd** interprets file names according to the "globbing" conventions used by **cs**. This allows users to utilize the metacharacters **\***, **.**, **[**, **]**, **{**, **}**, **~**, and **?**.

**ftpd** authenticates users according to three rules:

- The user name must be in the password data base, **/etc/passwd**, and not have a null password. The client must provide the correct password for the user before any file operations can be performed.
- The user name must not appear in the file **/etc/ftpd/ftpusers** (see *ftpusers(4)*).
- The user must have a standard shell returned by **getusershell()**.

Optionally, a system administrator can permit public access or "anonymous FTP." If this has been set up, users can access the anonymous FTP account with the user name **anonymous** or **ftp** and any non-null password (by convention, the client host's name). **ftpd** does a **chroot()** to the home directory of user **ftp**, thus limiting anonymous FTP users' access to the system. If the user name is **anonymous** or **ftp**, an anonymous FTP account must be present in the password file (user **ftp**). In this case the user is allowed to log in by specifying any password (by convention this is given as the user's e-mail address).

In order to permit anonymous FTP, there must be an entry in the *passwd(4)* database for an account named **ftp**. The password field should be **\***, the group membership should be **guest**, and the login shell should be **/usr/bin/false**. For example (assuming the **guest** group ID is 10):

```
ftp:*:500:10:anonymous ftp:/home/ftp:/usr/bin/false
```

The anonymous FTP directory should be set up as follows:

**~ftp** The home directory of the FTP account should be owned by user **root** and mode 555 (not writable). Since **ftpd** does a **chroot()** to this directory, it must have the following subdirectories and files:

**~ftp/usr/bin**

This directory must be owned by root and mode 555 (not writable). The file **/sbin/ls** should be copied to **~ftp/usr/bin**. This is needed to support directory listing by **ftpd**. The command should be mode 111 (executable only). If the FTP account is on the same file system as **/sbin**, **~ftp/usr/bin/ls** can be hard link, but it may not be a symbolic link, because of the **chroot()**. The command must be replaced when the system is updated.

**~ftp/etc**

This directory must be owned by root and mode 555 (not writable). It should contain versions of the files *passwd* and *group*. See *passwd(4)* and *group(4)*. These files must be owned by root and mode 444 (readable only). These files must be present for the **LIST** command to be able to produce owner names rather than numbers.

**~ftp/etc/passwd**

This file should contain entries for the **ftp** user and any other users who own files under the anonymous **ftp** directory. Such entries should have **\*** for passwords. Group IDs must be listed in the anonymous FTP group file, **~ftp/etc/group**. The path names of home directories in **~ftp/etc/passwd** must be with respect to the anonymous FTP home directory.

**~ftp/etc/group**

This file should contain the group names associated with any group IDs in file **~ftp/etc/passwd** and any group IDs of files in the anonymous FTP subdirectories.

**~ftp/pub** (optional)

This directory is used by anonymous FTP users to deposit files on the system. It should be owned by user **ftp** and should be mode 777 (readable and writable by all).

**~ftp/dist** (optional)

Directories used to make files available to anonymous ftp users should be mode 555 (not writable), and any files to be distributed should be owned by root and mode 444 (readable only) so that they cannot be modified or removed by anonymous FTP users.

**Note:** The steps that are followed to create an anonymous account is used to create a guest account also.

## DIAGNOSTICS

**ftpd** replies to FTP commands to ensure synchronization of requests and actions during file transfers, and to indicate the status of **ftpd**. Every command produces at least one reply, although there may be more than one. A reply consists of a three-digit number, a space, some text, and an end of line. The number is useful for programs; the text is useful for users. The number must conform to this standard, but the text can vary.

The first digit of the message indicates whether the reply is good, bad, or incomplete. Five values exist for the first digit. The values and the interpretations of the values are:

- 1 The requested action is being initiated; expect another reply before proceeding with a new command.
- 2 The requested action is complete. The server is ready for a new request.
- 3 The command has been accepted, but the requested action requires more information.
- 4 The command was not accepted, the requested action failed, but the error condition is temporary and the action can be requested again.
- 5 The command was not accepted, the requested action failed, and the error condition would most likely occur again if the same command sequence is repeated.

The second digit indicates the functional area that the message addresses. The values of the second digit and the interpretations of these values are:

- 0 Syntax. A message with a 0 for the second digit indicates that a syntax error occurred.
- 1 Information. A message with a 1 as the second digit indicates that the message is in reply to a request for information.
- 2 Connections. A message with a 2 as the second digit indicates that the message is a reply to a request for control and data connection information.
- 3 Authentication and accounting. A message with a 3 as the second digit indicates that the message is a reply to a login or accounting procedure.
- 4 Not currently specified.
- 5 File system. A message with a 5 as the second digit indicates that the text following the number contains information concerning the status of the server file system.

The third digit provides a further clarification of the information supplied by the second digit. Following are several examples of messages. Note that **ftpd**'s replies match the number but not the text.

- 110 Restart marker reply. MARK *yyyy=mmmm* where *yyyy* is a user process data stream marker, and *mmmm* is **ftpd**'s equivalent marker
- 120 Service ready in *nnn* minutes
- 200 Command okay
- 211 System status, or system help reply
- 212 Directory status
- 230 User logged in, proceed
- 250 Requested file action okay, completed
- 331 User name okay, need password
- 350 Requested file action pending further information
- 425 Cannot open data connection
- 451 Requested action aborted: local error in processing
- 500 Syntax error, command unrecognized or command line too long
- 530 Not logged in
- 550 Requested action not taken; file unavailable, not found, no access

## WARNINGS

The password is sent unencrypted through the socket connection.

Anonymous FTP is inherently dangerous to system security.

**DEPENDENCIES****Pluggable Authentication Modules (PAM)**

PAM is an Open Group standard for user authentication, password modification, and validation of accounts. In particular, `pam_authenticate()` is invoked to perform all functions related to login. This includes retrieving the password, validating the account, and displaying error messages.

**AUTHOR**

`ftpd` was developed by the University of California, Berkeley and the Washington University, St. Louis, Missouri.

**SEE ALSO**

`ftp(1)`, `inetd(1M)`, `chroot(2)`, `getusershell(3C)`, `pam_authenticate(3)`, `ftpaccess(4)`, `ftpusers(4)`, `group(4)`, `inetd.conf(4)`, `passwd(4)`, `sis(5)`, `xferlog(5)`.

f

**NAME**

fuser - list processes using a file or file structure

**SYNOPSIS**

```
/usr/sbin/fuser [-c|-f] [-ku] file ... [[-] [-c|-f] [-ku] file ...] ...
```

**DESCRIPTION**

The **fuser** command lists the process IDs of processes that have each specified *file* open. For block special devices, all processes using any file on that device are listed. The process ID can be followed by a letter, identifying how the *file* is being used.

- c** *file* is its current directory.
- r** *file* is its root directory, as set up by the **chroot** command (see *chroot(1M)*).
- o** It has *file* open.
- m** It has *file* memory mapped.
- t** *file* is its text file.

**Options**

You can specify the following options:

- c** Display the use of a mount point and any file beneath that mount point. Each *file* must be a file system mount point.
- f** Display the use of the named file only, not the files beneath it if it is a mounted file system.
- u** Display the login user name in parentheses following each process ID.
- k** Send the **SIGKILL** signal to each process using each *file*.

You can re-specify options between groups of files. The new set of options replaces the old set. A dash (–) by itself cancels all options currently in force.

The process IDs associated with each file are printed to standard output as a single line separated by spaces and terminated with a single newline. All other output — the file name, the letter, and the user name — is written to standard error.

You must be superuser to use **fuser**.

**NETWORKING FEATURES**

You can use **fuser** with NFS file systems or files. If the file name is in the format used in **/etc/mnttab** to identify an NFS file system, **fuser** will treat the NFS file system as a block special device and identify any process using that file system.

If contact with an NFS file system is lost, **fuser** will fail, since contact is required to obtain the file system identification. Once the NFS file system is re-contacted, stale file handles from the previous contact can be identified, provided that the NFS file system has the same file system identification.

**EXAMPLES**

Terminate all processes that are preventing disk drive 1 from being unmounted, listing the process ID and login name of each process being killed.

```
fuser -ku /dev/dsk/c201d1s?
```

List process IDs and login names of processes that have the password file open.

```
fuser -u /etc/passwd
```

Combine both the above examples into a single command line.

```
fuser -ku /dev/dsk/c201d1s? - -u /etc/passwd
```

If the device **/dev/dsk/c201d1s7** is mounted on directory **/home**, list the process IDs and login names of processes using the device. Alternately, if **/home** is the mount point for an NFS file system, list process IDs and login names of processes using that NFS file system.

```
fuser -cu /home
```

If **machine1:/filesystem/2mount** is an NFS file system, list all processes using any file on that file



system. If it is not an NFS file system, treat it as a regular file.

```
fuser machine1:/filesystem/2mount
```

**SEE ALSO**

ps(1), mount(1M), kill(2), signal(2).

**STANDARDS CONFORMANCE**

**fuser**: SVID2, SVID3

**NAME**

fwtmp, wtmpfix - manipulate connect accounting records

**SYNOPSIS**

/usr/sbin/acct/fwtmp [-ic]

/usr/sbin/acct/wtmpfix [files]

**DESCRIPTION****fwtmp**

*fwtmp* reads from the standard input and writes to the standard output, converting binary records of the type found in **wtmp** to formatted ASCII records. The ASCII version is useful to enable editing, via *ed*(1), bad records or general purpose maintenance of the file.

The argument **-ic** is used to denote that input is in ASCII form, and output is to be written in binary form. (The arguments **i** and **c** are independent, respectively specifying ASCII input and binary output, thus **-i** is an ASCII to ASCII copy and **-c** is a binary to binary copy).

**wtmpfix**

*wtmpfix* examines the standard input or named files in **wtmp** format, corrects the time/date stamps to make the entries consistent, and writes to the standard output. A - can be used in place of *files* to indicate the standard input. If time/date corrections are not performed, *acctcon1* will fault when it encounters certain date-change records.

Each time the date is set, a pair of date change records is written to **/var/adm/wtmp**. The first record is the old date denoted by the string **old time** placed in the line field and the flag **OLD\_TIME** placed in the type field of the **<utmp.h>** structure. The second record specifies the new date, and is denoted by the string **new time** placed in the line field and the flag **NEW\_TIME** placed in the type field. *wtmpfix* uses these records to synchronize all time stamps in the file. *wtmpfix* nullifies date change records when writing to the standard output by setting the time field of the **<utmp.h>** structure in the old date change record equal to the time field in the new date change record. This prevents *wtmpfix* and *acctcon1* from factoring in a date change record pair more than once.

In addition to correcting time/date stamps, *wtmpfix* checks the validity of the name field to ensure that it consists solely of alphanumeric characters or spaces. If it encounters a name that is considered invalid, it changes the login name to **INVALID** and write a diagnostic to the standard error. This minimizes the risk that *acctcon1* will fail when processing connect accounting records.

**DIAGNOSTICS**

*wtmpfix* generates the following diagnostics messages:

Cannot make temporary: xxx failed to make temp file  
 Input truncated at offset: xxx missing half of date pair  
 New date expected at offset: xxx missing half of date pair  
 Cannot read from temp: xxx some error reading  
 Bad file at offset: xxx ut\_line entry not digit, alpha, nor | or { (first character only checked)  
 Out of core: *malloc* fails. (Saves table of date changes)  
 No dtab: software error (rarely seen, if ever)

**FILES**

/usr/include/utmp.h  
 /var/adm/wtmp

**SEE ALSO**

acct(1M), acctcms(1M), acctcom(1M), acctcon(1M), acctmerg(1M), acctprc(1M), acctsh(1M), ed(1), runacct(1M), acct(2), acct(4), utmp(4).

**BUGS**

*fwtmp* generates no errors, even on garbage input.

**STANDARDS CONFORMANCE**

**fwtmp**: SVID2, SVID3

**wtmpfix**: SVID2, SVID3

**NAME**

gated - gateway routing daemon

**SYNOPSIS**

**gated** [-b *buffer\_size*] [-c] [-C] [-n] [-N] [-t *trace\_options*] [-f *config\_file*] [*trace\_file*]

**DESCRIPTION**

**gated** is a routing daemon that handles multiple routing protocols and replaces routed, egpup, and any routing daemon that speaks the HELLO routing protocol. **gated** currently handles the RIP, BGP, EGP, HELLO, and OSPF routing protocols. The **gated** process can be configured to perform all routing protocols or any subset of them (see *WARNINGS* below).

**Options**

The command-line options are:

- b *buffer\_size*  
Specifies a buffer size for the socket read/write buffer. The buffer size should not be less than one MB (megabyte) and should not exceed the available system memory. If this option is not specified, the buffer size will be set to the available system memory.
- c  
Specifies that the configuration file will be parsed for syntax errors and then **gated** will exit. **gated** will leave a dump file in `/var/tmp/gated_dump` if there were no errors. **gated** does not need to be run as the superuser to use the **-c** option but it may not be possible to read the kernel forwarding table and interface configuration if not run as superuser. The **-c** option implies **-tgeneral**. All *trace\_option* clauses in the configuration file will be ignored.
- C  
Specifies that the configuration file will just be parsed for syntax errors. **gated** will exit with a status 1 if there were any errors and 0 (zero) if there were not. **gated** does not need to be run as the superuser to use the **-C** option but it may not be possible to read the kernel forwarding table and interface configuration if not run as the superuser.
- n  
Specifies that **gated** will not modify the kernel forwarding table. This is used for testing **gated** configurations with actual routing data.
- N  
Specifies that **gated** will not daemonize. Normally, if tracing to stderr is not specified **gated** will daemonize if the parent process ID is not 1. This allows the use of an `/etc/inittab`-like method of invoking **gated** that does not have a PID of 1.
- t *trace\_options*  
Specifies a comma separated list of trace options to be enabled on startup. If no flags are specified, **general** is assumed. No space is allowed between this option and its arguments.  
  
This option must be used to trace events that take place before the configuration file is parsed, such as determining the interface configuration and reading routes from the kernel.  
  
See the *GateD Configuration Guide* for valid trace options and a more detailed explanation of tracing.
- f *config\_file*  
Use an alternate config file. By default, **gated** uses `/etc/gated.conf`.
- trace\_file*  
Trace file in which to place trace information.  
  
If a trace file is specified on the command line, or no trace flags are specified on the command line, **gated** detaches from the terminal and runs in the background. If trace flags are specified without specifying a trace file, **gated** assumes that tracing is desired to stderr and remains in the foreground.

**Signal Processing**

The following signals may be used to control **gated**:

- SIGHUP** Re-read configuration. A **SIGHUP** causes **gated** to reread the configuration file. **gated** first performs a clean-up of all allocated policy structures. All BGP and EGP peers are flagged for deletion and the configuration file is re-parsed.

If the re-parse is successful, any BGP and EGP peers that are no longer in the configuration are shut down, and new peers are started. **gated** attempts to determine if changes to existing peers require a shutdown and restart. OSPF is not capable of reconfiguring; it is shutdown and restarted during a reconfiguration. This may have an adverse impact on the routing system.

It should also be possible to enable/disable any protocol without restarting **gated**.

**SIGINT** Snap-shot of current state.

The current state of all **gated** tasks, timers, protocols and tables are written to `/var/tmp/gated_dump`.

On systems supporting `fork()`, this is done by forking a subprocess to dump the table information so as not to impact **gated**'s routing functions. On systems where memory management does not support copy-on-write, this will cause the **gated** address space to be duplicated; this may cause a noticeable impact on the system. On system not supporting `fork()`, the main process immediately processes the dump, which may impact **gated**'s routing functions.

**SIGTERM** Graceful shutdown.

On receipt of a **SIGTERM**, **gated** attempts a graceful shutdown. All tasks and protocols are asked to shutdown. Most will terminate immediately, the exception being EGP peers which wait for confirmation. It may be necessary to repeat the **SIGTERM** once or twice if this process takes too long.

All protocol routes are removed from the kernel's routing table on receipt of a **SIGTERM**. Interface routes, routes with `RTF_STATIC` set (from the route command where supported) and static routes specifying **retain** will remain. To terminate **gated** with the exterior routes intact, use **SIGKILL**.

**SIGUSR1** Toggle tracing.

On receipt of a **SIGUSR1**, **gated** will close the trace file. A subsequent **SIGUSR1** will cause it to be reopened. This will allow the file to be moved regularly.

It is not possible to use **SIGUSR1** if a trace file has not been specified, or tracing is being performed to `stderr`.

**SIGUSR2** Check for interface changes.

On receipt of a **SIGUSR2**, **gated** will rescan the kernel interface list looking for changes.

## WARNINGS

**gated** contains provisions for BGP protocol, but it is not officially supported by HP at the present time. Some RIP version 2 features (RFC1388) are not currently supported: MIB and route tag. The optional OSPF version 2 (RFC1247) feature of TOS (type of service) based routing is not supported. The route aggregation, generating a more general route from compressing the specific routes through the explicit configuration, is not supported in this release.

## AUTHORS

**gated** was primarily developed by Cornell University which includes code from the Regents of the University of California and the University of Maryland.

This software and associated documentation is Copyright 1990, 1991, 1992 by Cornell University.

## SEE ALSO

`gated.conf(4)`, `arp(1M)`, `fork(2)`, `gdc(1M)`, `ifconfig(1M)`, `netstat(1)`, `ospf_monitor(1M)`, `ripquery(1M)`, *GateD Documentation*, *GateD Configuration Guide*.

|          |                                                            |
|----------|------------------------------------------------------------|
| RFC 891  | DCN Local-Network Protocols (HELLO)                        |
| RFC 904  | Exterior Gateway Protocol Formal Specification             |
| RFC 1058 | Routing Information Protocol                               |
| RFC 1163 | A Border Gateway Protocol (BGP)                            |
| RFC 1164 | Application of the Border Gateway Protocol in the Internet |
| RFC 1247 | OSPF Specification, Version 2.                             |

**NAME**

gdc - operational user interface for gated

**SYNOPSIS**

**gdc** [-q] [-n] [-c *coresize*] [-f *filesize*] [-m *datasize*] [-s *stacksize*] [-t *seconds*] *command*

**DESCRIPTION**

**gdc** provides a user-oriented interface for the operation of the *gated*(1M) routing daemon. It provides support for starting and stopping the daemon, for the delivery of signals to manipulate the daemon when it is operating, for the maintenance and syntax checking of configuration files, and for the production and removal of state dumps and core dumps.

**gdc** can reliably determine **gated**'s running state and produces a reliable exit status when errors occur, making it advantageous for use in shell scripts which manipulate **gated**. Commands executed using **gdc** and, optionally, error messages produced by the execution of those commands, are logged via the same *syslogd*(1M) facility which **gated** itself uses, providing an audit trail of operations performed on the daemon.

If installed as a setuid root program **gdc** will allow non-root users who are members of a trusted group (by default the **gdmaint** group) to manipulate the routing daemon while denying access to others. The name of the user is logged along via *syslogd*(1M) along with an indication of each command executed, for audit purposes.

The command-line options are:

- n Run without changing the kernel forwarding table. Useful for testing, and when operating as a route server which does no forwarding.
- q Run quietly. With this option informational messages which are normally printed to the standard output are suppressed and error messages are logged via *syslogd*(1M) instead of being printed to the standard error output. This is often convenient when running **gdc** from a shell script.
- t *seconds* Specifies the time in seconds which **gdc** will spend waiting for **gated** to complete certain operations, in particular at termination and startup. By default this value is set to 10 seconds.

These additional command-line options may be present, depending on the options used to compile **gdc**:

- c *coresize* Sets the maximum size of a core dump a **gated** started with **gdc** will produce. Useful on systems where the default maximum core dump size is too small for **gated** to produce a full core dump on errors.
- f *filesize* Sets the maximum file size a **gated** started with **gdc** will produce. Useful on systems where the default maximum file dump size is too small for **gated** to produce a full state dump when requested.
- m *datasize* Sets the maximum size of the data segment of a **gated** started with **gdc**. Useful on systems where the default data segment size is too small for **gated** to run.
- s *stacksize* Sets the maximum size of stack of a **gated** started with **gdc**. Useful on systems where the default maximum stack size is too small for **gated** to run.

The following commands cause signals to be delivered to **gated** for various purpose:

- COREDUMP** Sends an abort signal to **gated**, causing it to terminate with a core dump.
- dump** Signal **gated** to dump its current state into the file */usr/tmp/gated\_dump*.
- interface** Signal **gated** to recheck the interface configuration. **gated** normally does this periodically in any event, but the facility can be used to force the daemon to check interface status immediately when changes are known to have occurred.
- KILL** Cause **gated** to terminate ungracefully. Normally useful when the daemon has hung.
- reconfig** Signal **gated** to reread its configuration file, reconfiguring its current state as appropriate.
- term** Signal **gated** to terminate after shutting down all operating routing protocols gracefully. Executing this command a second time should cause **gated** to terminate even if some protocols have not yet fully shut down.

**toggletrace**

If **gated** is currently tracing to a file, cause tracing to be suspended and the trace file to be closed. If **gated** tracing is current suspended, cause the trace file to be reopened and tracing initiated. This is useful for moving trace files.

By default **gated** obtains its configuration from a file normally named **/etc/gated.config**. The **gdc** program also maintains several other versions of the configuration file, in particular named:

- /etc/gated.conf+** The *new* configuration file. When **gdc** is requested to install a new configuration file, this file is renamed **/etc/gated.conf**.
- /etc/gated.conf-** The *old* configuration file. When **gdc** is requested to install a new configuration file, the previous **/etc/gated.conf** is renamed to this name.
- /etc/gated.conf--** The *really old* configuration file. **gdc** retains the previous *old* configuration file under this name.

The following commands perform operations related to configuration files:

- checkconf** Check **/etc/gated.conf** for syntax errors. This is usefully done after changes to the configuration file but before sending a **reconfig** signal to the currently running **gated**, to ensure that there are no errors in the configuration which would cause the running **gated** to terminate on reconfiguration. When this command is used, **gdc** issues an informational message indicating whether there were parse errors or not, and if so saves the error output in a file for inspection.
- checknew** Like **checkconf** except that the *new* configuration file, **/etc/gated.conf+**, is checked instead.
- newconf** Move the **/etc/gated.conf+** file into place as **/etc/gated.conf**, retaining the older versions of the file as described above. **gdc** will decline to do anything when given this command if the *new* configuration file doesn't exist or otherwise looks suspect.
- backout** Rotate the configuration files in the *newer* direction, in effect moving the *old* configuration file to **/etc/gated.conf**. The command will decline to perform the operation if **/etc/gated.conf-** doesn't exist or is zero length, or if the operation would delete an existing, non-zero length **/etc/gated.conf+** file.
- BACKOUT** Perform a **backout** operation even if **/etc/gated.conf+** exists and is of non-zero length.
- modeconf** Set all configuration files to mode 664, owner root, group gdmaint. This allows a trusted non-root user to modify the configuration files.
- createconf** If **/etc/gated.conf+** does not exist, create a zero length file with the file mode set to 664, owner root, group gdmaint. This allows a trusted non-root user to install a new configuration file.

The following commands provide support for starting and stopping **gated**, and for determining its running state:

- running** Determine if **gated** is currently running. This is done by checking to see if **gated** has a lock on the file containing its pid, if the pid in the file is sensible and if there is a running process with that pid. Exits with zero status if **gated** is running, non-zero otherwise.
- start** Start **gated**. The command returns an error if **gated** is already running. Otherwise it executes the **gated** binary and waits for up to the delay interval (10 seconds by default, as set with the **-t** option otherwise) until the newly started process obtains a lock on the pid file. A non-zero exit status is returned if an error is detected while executing the binary, or if a lock is not obtained on the pid file within the specified wait time.
- stop** Stop **gated**, gracefully if possible, ungracefully if not. The command returns an error (with non-zero exit status) if **gated** is not currently running. Otherwise it sends a terminate signal to **gated** and waits for up to the delay interval (10 seconds by default, as specified with the **-t** option otherwise) for the process to exit. Should **gated** fail to exit within the delay interval it is then signaled again with a second terminate signal. Should it fail to exit by the end of the second delay interval it is signaled for a third time with a kill signal. This should force immediate termination unless something is very broken. The command terminates with zero exit status when it detects that **gated** has terminated, non-zero otherwise.

**restart** If **gated** is running it is terminated via the same procedure as is used for the **stop** command above. When the previous **gated** terminates, or if it was not running prior to command execution, a new **gated** process is executed using the procedures described for the **start** command above. A non-zero exit status is returned if any step in this procedure appears to have failed.

The following commands allow the removal of files created by the execution of some of the commands above:

**rmcore** Removes any existing **gated** core dump file.  
**rmdump** Removes any existing **gated** state dump file.  
**rmparse** Removes the parse error file generated when a **checkconf** or **checknew** command is executed and syntax errors are encountered in the configuration file being checked.

## FILES

Many of default filenames listed below contain the string **%s**, which is replaced by the name with which **gated** is invoked. Normally this is **gated**, but if invoked as **gated-test**, **gated** will by default look for **/etc/gated-test.conf**. These paths may all be changed at compilation time.

|                             |                                          |
|-----------------------------|------------------------------------------|
| <b>/usr/sbin/gated</b>      | The <b>gated</b> binary.                 |
| <b>/etc/gated.conf</b>      | Current <b>gated</b> configuration file. |
| <b>/etc/gated.conf+</b>     | Newer configuration file.                |
| <b>/etc/gated.conf-</b>     | Older configuration file.                |
| <b>/etc/gated.conf--</b>    | Much older configuration file.           |
| <b>/var/run/gated.pid</b>   | Where <b>gated</b> stores its pid.       |
| <b>/var/tmp/gated_dump</b>  | <b>gated</b> 's state dump file.         |
| <b>/var/tmp/gated_parse</b> | Where config file parse errors go.       |
| <b>/var/tmp</b>             | Where <b>gated</b> drops its core file.  |

## AUTHOR

**gdc** was developed by Dennis Ferguson and Cornell University.

## SEE ALSO

**gated**(1M), **ospf\_monitor**(1M), **ripquery**(1M), **syslogd**(1M), **gated.conf**(4), *GateD Documentation*, *GateD Configuration Guide*.

## BUGS

Many commands only work when **gated** is installed in the system directory it was configured with.

There is not yet any way to tell **gdc** about systems which name their core dump other than **core** (**core.gated** is a less common possibility).

**NAME**

geocustoms - configure system language on multi-language systems

**SYNOPSIS**

**geocustoms** [-l *locale*]

**DESCRIPTION**

The **geocustoms** utility manages default selection and retention/removal of multiple languages installed on ignited systems. The geocustoms program is executed at first boot on ignited (Instant Ignition) systems with multiple languages available. On subsequent sessions, the command `/usr/sbin/geocustoms` starts **geocustoms**.

**Options:**

**-l *locale*** Sets the **LANG** variable (and all other appropriate dependencies, if applicable) to the value of *locale*. If the *locale* argument is not a valid option for that system, the User Interface (UI) will appear as if the option had not been used.

An additional locale value can be used in this context; **SET\_NULL\_LOCALE** can be the argument to the **-l** option, the result of which will be setting locale variables to **NULL** by default. A null locale will allow programs to execute without using localized message catalogs. This can increase system performance. All HP-UX messages appear in English if the locale is set to **NULL**.

**EXTERNAL INFLUENCES****Environment Variables**

**geocustoms** writes default values to system configuration files regarding the following environmental variables: **LANG**, **LC\_ALL**, **LC\_CTYPE**, **LC\_COLLATE**, **LC\_MONETARY**, **LC\_NUMERIC**, **LC\_TIME**, **LC\_MESSAGES**.

**International Code Set Support****Native Language Support (NLS):**

If the standard message catalogs exist, then they are in `/usr/lib/nls`. The **geocustoms** command will use the standard message catalogs, if they are on the system. If the standard message catalogs are not on the system, then the messages appear in English. (This is in accordance with standard NLS behavior). All European languages for CDE will be supported. For HP-UX 11.11, this includes English, French, German, Italian, Spanish and Swedish. All prompts and logging messages will be localized.

Locale (Language Variant) names are always localized in accordance with standard NLS behavior.

NLS is extended to allow multiple "fonts" on the initial screen at the same time through use of bitmapped images.

**RETURN VALUES**

- 0 Successful completion and/or clean exit from program.
- 1 Program was unable to complete all objectives.

**DIAGNOSTICS****Errors:**

**geocustoms** writes to **stderr**, and to `/var/adm/sw/lang.log`.

**Standard Output**

**geocustoms** does not write to **stdout**.

**Standard Error**

**geocustoms** only writes to **stderr** in case of command line error or request for syntax. Any UI error messages appear via an error window.

**Logging**

Both interactive and non-interactive sessions log summary events at:

`/var/adm/sw/lang.log`.

**EXAMPLES**

To set the default system language non-interactively to German:



```
/usr/sbin/geocustoms -l de_DE.iso88591
```

## DEPENDENCIES

ObAM 4.2

SD-UX 11.00 or greater

HP-UX 11.00 or greater

## Compatibility

This product is designed for compatibility with releases including and after HP-UX 11.00 with a Common Desktop Environment (CDE). No attempt has been made to support the Visual User Environment (VUE).

## Notes

If geocustoms is invoked by the user, it may be necessary to log out and log in again for language changes to take effect.

If language bundles have been marked for removal, that will occupy the `swagentd()` for some minutes at the next system boot.

## Limitations

geocustoms does *not* do the following:

- Manage languages at the codeset level.
- Provide a user interface for Asian languages.
- Manage keyboard selection.
- Create or remove locale definitions.
- Provide a special interface for restoring or adding languages to the system from separate media.

## AUTHOR

geocustoms was developed by HP.

## FILES:

geocustoms creates a text file `/var/adm/sw/lang.log`.

geocustoms creates, if necessary, and modifies `/etc/dt/config/Xconfig` and `/etc/rc.config.d/LANG`.

geocustoms will read NLS files, as discussed in Native Language Support above.

## SEE ALSO:

locale(1), swinstall(1M), swlist(1M), swremove(1M), setlocale(3C).

## STANDARDS CONFORMANCE

POSIX.2, UNIX95 (SPEC1170 and XPG4).

**NAME**

getext (vxfs) - get extent attributes

**SYNOPSIS**

`/usr/sbin/getext [-F vxfs] [-V] [-f] [-s] file...`

**DESCRIPTION**

**getext** displays extent attribute information associated with a set of files.

**Options**

- F vxfs** Specify the VxFS file system type.
- V** Echo the completed command line, but do not execute the command. The command line is generated by incorporating the user-specified options. This option allows the user to verify the command line.
- f** Do not print the filenames for which extent attributes are displayed.
- s** Do not print output for files that do not have fixed extent sizes or reservations.

**OUTPUT**

The following example shows a file with a block size of 1024 bytes, 36 blocks reserved, a fixed extent size of 3 blocks, and all extents aligned to 3 block boundaries:

```
file1:  Bsize 1024 Reserve 36 Extent Size 3 align noextend
```

The file size cannot be extended once the current reservation is exhausted. Reservations and fixed extent sizes are allocated in units of the file system block size.

**NOTES**

Only the **align** and **noextend** allocation flags (set through *setext*(1M) or the **VX\_SETEXT** ioctl) are persistent attributes of the file and therefore visible via **getext** or the **VX\_GETTEXT** ioctl. **trim** is also visible, although it is cleared and the reservation is reduced on the final close of the file.

**SEE ALSO**

*setext*(1M), *vxfsio*(7).

**NAME**

getmemwindow - extracts window ids of user processes from /etc/services.window

**SYNOPSIS**

getmemwindow *string*

**DESCRIPTION**

getmemwindow is the command used to extract window ids of user processes from the /etc/services.window file. User applications are encouraged to place a unique string defining an application and its associated window id in the /etc/services.window file and then extract that window id using the getmemwindow command.

This allows for changing the application window id in one central location, rather than changing scripts with hard-coded values, in the event of a memory window collision between two applications.

**EXAMPLES**

```
# Extract the window id for "HP" from the /etc/services.window file
# and start the program "HP_startup_script" with arguments arg1 and
# arg2.
#
WinId=$(getmemwindow HP)
setmemwindow -i $WinId HP_startup_script arg1 arg2
```

**AUTHOR**

getmemwindow was developed by HP.

**FILES**

/etc/services.window File containing applications' associated window id.

**SEE ALSO**

setmemwindow(1M), services.window(4), *11.0 Memory Windows White Paper*.

**NAME**

getprpw - display protected password database

**SYNOPSIS**

```
getprpw [-l | -n [domain]] [-r] [-m parm[,parm]] username
```

**DESCRIPTION**

**getprpw** displays the user's protected password database settings. This command is available only to the superuser in a trusted system. Normally it is only used via SAM, see *sam*(1M).

The database contains information for both local and NIS+ users. However, some NIS+ information is kept on the master. Since a user may be both local and NIS+, **getprpw** uses the *nsswitch.conf*(4) default if neither **-l** nor **-n** are specified.

**Options**

**getprpw** recognizes the following options...

- l** Specifies to get information from the local user.
- n** Can be specified with or without domain name; i.e., **-n [domain]**. If **-n [domain]** is specified, displays data for the NIS+ user. The *domain* name must be fully qualified, with a terminating period. If *domain* name is not specified, the local domain will be used.
- r** Displays the arguments supplied to **-m** in raw format
- m** Displays the database value for the argument passed.

An "invalid-opt" is printed if a list of options passed to **-m** contains an invalid option. The rest of the options will be processed. If **getprpw** is specified without **-m**, all parameters are displayed in the order given below.

Boolean values are returned as **YES**, **NO**, or **DFT** (for system default values in */tc/b/files/auth/system/default*).

Numeric values are specified as positive numbers, 0, or -1. A value of -1 indicates that the field has not been assigned a value in the database.

Units of time are returned in number of days (>=0), although the database keeps them in seconds. This and other minor differences between the command parameters and the database fields are consistent with *modprpw*(1M).

The following parameters for the user can be displayed using the **-m** option.

They are listed below in the order shown in *prot.h*. The database fields are fully explained in *prpwd*(4).

|                |                                                |
|----------------|------------------------------------------------|
| <b>uid</b>     | user uid                                       |
| <b>bootpw</b>  | boot authorization flag                        |
| <b>audid</b>   | audit id                                       |
| <b>audflg</b>  | audit flag                                     |
| <b>mintm</b>   | minimum time between password changes          |
| <b>maxpwn</b>  | maximum password length                        |
| <b>exptm</b>   | password expiration time                       |
| <b>lftm</b>    | password lifetime                              |
| <b>spwchg</b>  | last successful password change time           |
| <b>upwchg</b>  | last unsuccessful password change time         |
| <b>acctexp</b> | account expiration time                        |
| <b>llog</b>    | last login time interval                       |
| <b>expwarn</b> | password expiration warning time               |
| <b>usrpick</b> | whether user picks password, <b>YES/NO/DFT</b> |

|                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>sysnpw</b>   | whether system generates pronounceable passwords, <b>YES/NO/DFT</b>                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>rstrpw</b>   | whether password is restricted, i.e. checked for triviality, <b>YES/NO/DFT</b>                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>nullpw</b>   | NULL passwords are allowed, <b>YES/NO/DFT</b> . <b>Not recommended!</b>                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>syschpw</b>  | whether system generates passwords having characters only, <b>YES/NO/DFT</b>                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>sysltpw</b>  | whether system generates passwords having letters only, <b>YES/NO/DFT</b>                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>timeod</b>   | time of day allowed for login                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>slogint</b>  | time of last successful login                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>ulogint</b>  | time of last unsuccessful login                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>sloginy</b>  | tty of last successful login                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>culogin</b>  | consecutive number of unsuccessful logins so far                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>uloginy</b>  | tty of last unsuccessful login                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>umaxlntr</b> | maximum unsuccessful login tries                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>alock</b>    | administrator lock, <b>YES</b> if on, <b>NO</b> if off, <b>DFT</b> if not set.                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>lockout</b>  | returns the reason for a lockout in a "bit" valued string, where 0 = condition not present, 1 is present. The position, left to right represents: <ul style="list-style-type: none"> <li>1 past password lifetime</li> <li>2 past last login time (inactive account)</li> <li>3 past absolute account lifetime</li> <li>4 exceeded unsuccessful login attempts</li> <li>5 password required and a null password</li> <li>6 admin lock</li> <li>7 password is a *</li> </ul> |

**RETURN VALUE**

|   |                               |
|---|-------------------------------|
| 0 | success                       |
| 1 | user not privileged           |
| 2 | incorrect useage              |
| 3 | cannot find the password file |
| 4 | system is not trusted         |

**EXAMPLES**

Displays the database aging fields for user "someusr".

```
getprpw -m mintm,exptm,expwarn,lftm someusr
```

The command displays:

```
mintm=1, exptm=2, expwarn=-1, lftm=3
```

**WARNINGS**

This command is intended for SAM use only. It may change with each release and can not be guaranteed to be backward compatible.

Several database fields interact with others. The side effects of an individual change may not cause a problem till much later.

Special meanings may apply in the following cases:

- an absent field
- a field without a value
- a field with a zero value

**AUTHOR**

**getprpw** was developed by HP.

**FILES**

**/etc/passwd**                      System Password file

`/tcb/files/auth/*/*` Protected Password Database  
`/tcb/files/auth/system/default`  
System Defaults Database

**SEE ALSO**

`modprpw(1M)`, `prpwd(4)`, `nsswitch.conf(4)`.

## NAME

getty - set terminal type, modes, speed, and line discipline

## SYNOPSIS

```
/usr/sbin/getty [-h] [-t timeout] line [speed [type [linedesc]]]
/usr/sbin/getty -c file
```

## DESCRIPTION

*getty* is a program that is invoked by *init*(1M). It is the second process in the series, (*init-getty-login-shell*) that ultimately connects a user with the HP-UX system. Initially, if */etc/issue* exists, *getty* prints its contents to the user's terminal, followed by the login message field for the entry it is using from */etc/gettydefs*. *getty* reads the user's login name and invokes the *login*(1) command with the user's name as argument. While reading the name, *getty* attempts to adapt the system to the speed and type of terminal being used.

## Configuration Options and Arguments

*getty* recognizes the following arguments:

- |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |             |         |             |          |              |           |             |                        |             |             |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|---------|-------------|----------|--------------|-----------|-------------|------------------------|-------------|-------------|
| <i>line</i>       | Name of a tty line in <i>/dev</i> to which <i>getty</i> is to attach itself. <i>getty</i> uses this string as the name of a file in the <i>/dev</i> directory to open for reading and writing. By default <i>getty</i> forces a hangup on the line by setting the speed to zero before setting the speed to the default or specified speed. However, when <i>getty</i> is run on a direct port, <i>getty</i> does not force a hangup on the line since the driver ignores changes to zero speed on ports open in direct mode (see <i>modem</i> (7)).                                                                                                                                                                                                                                 |             |         |             |          |              |           |             |                        |             |             |
| <b>-h</b>         | Tells <i>getty</i> not to force a hangup on the line before setting the speed to the default or specified speed.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |             |         |             |          |              |           |             |                        |             |             |
| <b>-t timeout</b> | <i>getty</i> exits if the open on the line succeeds and no one types anything within <i>timeout</i> seconds.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |             |         |             |          |              |           |             |                        |             |             |
| <i>speed</i>      | A label to a speed and tty definition in the file <i>/etc/gettydefs</i> . This definition tells <i>getty</i> at what speed to initially run, what the login message should look like, what the initial tty settings are, and what speed to try next should the user indicate that the speed is inappropriate (by typing a <i>break</i> character). The default <i>speed</i> is 300 baud.                                                                                                                                                                                                                                                                                                                                                                                             |             |         |             |          |              |           |             |                        |             |             |
| <i>type</i>       | A character string describing to <i>getty</i> what type of terminal is connected to the line in question. <i>getty</i> understands the following types: <table border="0" style="margin-left: 40px;"> <tr> <td><b>none</b></td> <td>default</td> </tr> <tr> <td><b>vt61</b></td> <td>DEC vt61</td> </tr> <tr> <td><b>vt100</b></td> <td>DEC vt100</td> </tr> <tr> <td><b>hp45</b></td> <td>Hewlett-Packard HP2645</td> </tr> <tr> <td><b>c100</b></td> <td>Concept 100</td> </tr> </table> <p>The default terminal is <b>none</b>; i.e., any crt or normal terminal unknown to the system. Also, for terminal type to have any meaning, the virtual terminal handlers must be compiled into the operating system. They are available, but not compiled in the default condition.</p> | <b>none</b> | default | <b>vt61</b> | DEC vt61 | <b>vt100</b> | DEC vt100 | <b>hp45</b> | Hewlett-Packard HP2645 | <b>c100</b> | Concept 100 |
| <b>none</b>       | default                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |             |         |             |          |              |           |             |                        |             |             |
| <b>vt61</b>       | DEC vt61                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |             |         |             |          |              |           |             |                        |             |             |
| <b>vt100</b>      | DEC vt100                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |             |         |             |          |              |           |             |                        |             |             |
| <b>hp45</b>       | Hewlett-Packard HP2645                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |             |         |             |          |              |           |             |                        |             |             |
| <b>c100</b>       | Concept 100                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |             |         |             |          |              |           |             |                        |             |             |
| <i>linedesc</i>   | A character string describing which line discipline to use when communicating with the terminal. Hooks for line disciplines are available in the operating system, but there is only one presently available — the default line discipline, <b>LDISC0</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |             |         |             |          |              |           |             |                        |             |             |

When given no optional arguments, *getty* sets the *speed* of the interface to 300 baud, specifies that raw mode is to be used (awaken on every character), that echo is to be suppressed, either parity allowed, new-line characters will be converted to carriage return-line feed, and tab expansion performed on the standard output. It types the login message before reading the user's name a character at a time. If a null character (or framing error) is received, it is assumed to be the result of the user pushing the "break" key. This causes *getty* to attempt the next *speed* in the series. The series that *getty* tries is determined by what it finds in */etc/gettydefs*.

The user's name is terminated by a new-line or carriage-return character. The latter results in the system being set to treat carriage returns appropriately (see *ioctl*(2)).

The user's name is scanned to see if it contains any lowercase alphabetic characters; if not, and if the name is non-empty, the system is told to map any future uppercase characters into the corresponding lowercase

characters.

*getty* also understands the “standard” ESS2 protocols for erasing, killing and aborting a line, and terminating a line. If *getty* sees the ESS erase character, `_`, or kill character, `$`, or abort character, `&`, or the ESS line terminators, `/` or `!`, it arranges for this set of characters to be used for these functions.

Finally, *login* is called with the user’s name as an argument. Additional arguments can be typed after the login name. These are passed to *login*, which places them in the environment (see *login*(1)).

### Check Option

A check option is provided. When *getty* is invoked with the `-c` option and *file*, it scans *file* as if scanning `/etc/gettydefs` and prints the results on the standard output. If there are any unrecognized modes or improperly constructed entries, *getty* reports these. If the entries are correct, *getty* prints out the values of the various flags. See *ioctl*(2) for an interpretation of values. Note that some values are added to the flags automatically.

### DEPENDENCIES

#### HP 2334 MultiMux:

The modem control parameter *MRTS* must be present in the `/etc/gettydefs` file when using *getty* in conjunction with an HP 2334 or HP 2335 MultiMux to ensure that the RTS modem control signal is asserted correctly.

Example:

```
9600# B9600 HUPCL PARENB MRTS # B9600 SANE PARENB ISTRIP IXANY #login: #19200
```

MRTS is not intended for use with devices other than the HP 2334 or HP 2335 MultiMux.

### FILES

```
/etc/gettydefs
/etc/issue
```

### SEE ALSO

*ct*(1), *login*(1), *init*(1M), *ioctl*(2), *gettydefs*(4), *inittab*(4), *modem*(7), *termio*(7).

### BUGS

While *getty* does understand simple single character quoting conventions, it is not possible to quote the special control characters that *getty* uses to determine when the end of the line has been reached, which protocol is being used, and what the erase character is. Therefore it is not possible to log in by means of *getty* and type a `#`, `@`, `/`, `!`, `_`, backspace, `^U`, `^D`, or `&` as part of your login name or arguments. They will always be interpreted as having their special meaning as described above.



**NAME**

getx25 - get x25 line

**SYNOPSIS**

*/usr/sbin/getx25 line speed pad-type*

**DESCRIPTION**

**getx25** is functionally very similar to **getty** (see *getty(1M)*) but is used only for incoming lines that are connected to an X.25 PAD. It performs special functions such as setting up an initial PAD configuration. It also logs the number of the caller in */var/uucp/.Log/LOGX25*. The third parameter is the name of the PAD being used. HP 2334A is the only one supported at this time. A typical invocation would be:

*/usr/sbin/getx25 x25.1 2 HP2334A*

**AUTHOR**

**getx25** was developed by HP.

**SEE ALSO**

login(1), uucp(1), getty(1M).

  
g

**NAME**

groupadd - add a new group to the system

**SYNOPSIS**

```
groupadd [-g gid [-o] ] group
```

**DESCRIPTION**

The **groupadd** command creates a new group on the system by adding the appropriate entry to the **/etc/group** file. The **groupadd** command expects the *group* argument, which is the name of the new group. The name consists of a string of printable characters that may not include a colon (:) or newline (\n).

**Options**

The **groupadd** command may be used with the following options:

- g *gid* Specifies the group ID for the new group. *gid* must be a non-negative decimal integer less than MAXUID as defined in the **<param.h>** header file. By default the next available unique group ID in the valid range is allocated. Group IDs in the range 0-99 are reserved.
- o Allow the *gid* to be non-unique (i.e., a duplicate).

**NETWORKING FEATURES**

The **groupadd** command is aware of NIS user entries. Only local groups may be added with this command. Attempts to add an NIS group will result in an error. NIS groups must be administered from the NIS server. If **groupadd** is used on a system where NIS is installed, it may fail with the error

**group *x* is not unique**

(return value 9) if the group specified is not present in the local **/etc/group** file, but is an NIS group (see **group(4)**). NIS groups are also checked when verifying uniqueness of the new *gid*, which may result in the error

**GID # is not unique**

(return value 4).

**RETURN VALUE**

The **groupadd** command exits with one of the following values:

- 0 No error.
- 2 Invalid command syntax.
- 3 Invalid argument supplied to an option.
- 4 *gid* is not unique (when -o is not used).
- 9 *group* is not unique.
- 10 Cannot modify the **/etc/group** file.
- 11 **/etc/passwd** file or **/etc/ptmp** file busy. Another command may be modifying the **/etc/passwd** file.
- 12 Unable to open **/etc/ptmp** file or **/etc/passwd** file is non-existent.

**EXAMPLES**

Add the group **project1** to the **/etc/group** file.

```
groupadd project1
```

Add the group **project12** to the **/etc/group** file with the group ID 111 as long as no group currently exists with a group ID of 111.

```
groupadd -g 111 project12
```

**WARNINGS**

As many users may try to write the **/etc/passwd** file simultaneously, a passwd locking mechanism was devised. If this locking fails after subsequent retrying, **groupadd** terminates.

**FILES**

```
/etc/group
/etc/ptmp
```

**SEE ALSO**

users(1), groupdel(1M), groupmod(1M), logins(1M), useradd(1M), userdel(1M), usermod(1M), group(4).

**STANDARDS CONFORMANCE**

groupadd: SVID3

**NAME**

groupdel - delete a group from the system

**SYNOPSIS**

**groupdel** *group*

**DESCRIPTION**

The **groupdel** command deletes a group from the system by removing the appropriate entry from the */etc/group* file.

The **groupdel** command must be used with the *group* argument. *group* is the name of the group to be deleted, consisting of a string of printable characters.

**NETWORKING FEATURES**

This command is aware of NIS user entries. Only local groups may be deleted with **groupdel**. Attempts to delete an NIS group will result in an error. NIS groups must be administered from the NIS server. If **groupdel** is used on a system where NIS is installed, it may fail with the error

**group *x* does not exist**

(return value 6), if the group specified is an NIS group (see *group(4)*).

**RETURN VALUE**

**groupdel** exits with one of the following values:

- 0 No error.
- 2 Invalid command syntax.
- 3 Invalid argument supplied to an option.
- 6 *group* does not exist.
- 10 Cannot modify the */etc/group* file.
- 11 */etc/passwd* file or */etc/ptmp* file busy. Another command may be modifying the */etc/passwd* file.
- 12 Unable to open */etc/ptmp* or */etc/passwd* file is non-existent.

**EXAMPLES**

Delete the group **project1** from the */etc/group* file if it exists:

**groupdel project1**

**WARNINGS**

As many users may try to write the */etc/passwd* file simultaneously, a *passwd* locking mechanism was devised. If this locking fails after subsequent retrying, **groupdel** terminates.

**FILES**

*/etc/group*  
*/etc/ptmp*

**SEE ALSO**

*users(1)*, *groupadd(1M)*, *groupmod(1M)*, *logins(1M)*, *useradd(1M)*, *userdel(1M)*, *usermod(1M)*, *group(4)*.

**STANDARDS CONFORMANCE**

**groupdel**: SVID3

**NAME**

groupmod - modify a group on the system

**SYNOPSIS**

```
groupmod [-g gid [-o] ] [-n name] group
```

**DESCRIPTION**

The **groupmod** command modifies a group on the system by altering the appropriate entry in the `/etc/group` file.

The **groupmod** command must be used with the *group* argument, which is the name of the group to be modified.

**Options**

The **groupmod** command may be used with the following options:

- g *gid*** Change the value of the group ID to *gid*. *gid* must be a non-negative decimal integer less than MAXUID as defined in the `<param.h>` header file.
- o** Allow the *gid* to be non-unique (i.e., a duplicate).
- n *name*** Change the name of the group to *name*. *name* consists of a string of printable characters that may not include a colon (:) or newline (\n).

**NETWORKING FEATURES**

This command is aware of NIS user entries. Only local groups may be modified with **groupmod**. Attempts to modify an NIS group will result in an error. NIS groups must be administered from the NIS server. If **groupmod** is used on a system where NIS is installed, it may fail with the error

**group *x* does not exist**

(return value 6) if the group specified is an NIS group (see *group(4)*). However, NIS groups are checked when verifying uniqueness of the new *gid* or new group name, which may result in the above error, or the error

**GID # is not unique**

(return value 4).

**RETURN VALUES**

**groupmod** exits with one of the following values:

- 0 No error.
- 2 Invalid command syntax.
- 3 Invalid argument supplied to an option.
- 4 *gid* is not unique (when **-o** is not used).
- 6 *group* does not exist.
- 9 *group* is not unique.
- 10 Cannot modify the `/etc/group` file.
- 11 `/etc/passwd` file or `/etc/ptmp` file busy. Another command may be modifying the `/etc/passwd` file.
- 12 Unable to open `/etc/ptmp` file or the `/etc/passwd` file is non-existent.

**EXAMPLES**

Change the group ID of the group **project2** to 111 in the file `/etc/group` if the group **project2** exists. This is done even if the group ID 111 is already in use.

```
groupmod -g 111 -o project2
```

Change the name of **project2** to **project22** in the file `/etc/group` if the group **project22** does not already exist.

```
groupmod -n project22 project2
```

**WARNINGS**

As many users may try to write the `/etc/passwd` file simultaneously, a passwd locking mechanism was devised. If this locking fails after subsequent retrying, **groupmod** terminates.

**FILES**

/etc/group  
/etc/ptmp

**SEE ALSO**

users(1), groupadd(1M), groupdel(1M), logins(1M), useradd(1M), userdel(1M), usermod(1M), group(4).

**STANDARDS CONFORMANCE**

groupmod: SVID3

**NAME**

hosts\_to\_named - translate host table to name server file format

**SYNOPSIS**

**hosts\_to\_named** -d *domain* -n *network-number* [*options*]

**DESCRIPTION**

**hosts\_to\_named** translates the host table, */etc/hosts*, into files that are usable by the name server *named*(1M). The format of these files is defined in RFC1035. The files are created in the current directory. Once the host table is translated, the name server files can be maintained directly, or the translation can be repeated after each change to the host table.

If a line in the host table contains no domain names, all names on the line are assumed to be in the default domain. The first *domain* listed is the "default domain". If data is being created for more than 1 domain or if certain options are used, there must be domain names in the host table to determine which names belong in which domain.

The name server data is referred to as "resource records".

Options are:

-a *network-number*

Add the information about hosts in the local domain from network *network-number*. This is the same as the -n option except that no pointer (PTR) data is created. This is useful when there are multiple domains on a network and a different server is handling the address-to-name mapping for *network-number*.

-b *bootfile*

Name the boot file *bootfile*. The default is **named.boot** (if **named** is 4.x) or **named.conf** (if **named** is 8.x) in the current directory.

-c *subdomain*

Create alias (CNAME) records for hosts in *subdomain* of the default domain. When a subdomain is delegated, it is useful to create aliases for the old names in the default domain that point to the new names in the *subdomain*. After creating the alias (CNAME) records, ignore lines in the host table that contain names in the *subdomain*. This option can be used more than once on the command line. This option requires domain names in the host table. When the old names in this *domain* are no longer used, they can be ignored with the -e option. If the *subdomain* name does not have dots, the default domain is appended to *subdomain*.

-d *domain*

Create data for *domain*. This option can be used more than once on the command line if data is being created for more than 1 domain. The first *domain* listed is the "default domain". This option requires domain names in the host table for all hosts in domains except the default domain.

-e *subdomain*

Eliminate lines from the host table that contain names in the *subdomain* before translating. If the *subdomain* name does not have dots, the default domain is appended. This option may be used more than once on the command line. This option requires domain names in the host table.

-f *file*

Read command line options from *file*. The -f option is not allowed within a file.

-h *host*

Declare *host* to be the host in the start of authority (SOA) record that the name server data was created on. Also use *host* for the electronic mail address of the responsible user in the SOA record. The default is the host this command is run on.

-m *weight:mailhub*

For each canonical hostname from the host table, create mail exchanger (MX) records with the specified weight and mail hub. The weight is a positive integer. The mail hub is a hostname. If the mail hub name has no dots, the default domain is appended. This option can be used more than once on the command line.

-n *network-number[:mask]*

Create data for *network-number*. See below for description of *network-number*. If only one *domain* is listed with -d, all data for *network-number* is assumed to be in *domain*. The optional subnet mask *mask* can be used instead of supplying each *network-number* for a subnet using multiple -n options. *mask* must be in dot notation.

- o *refresh:retry:expire:min*  
Set the values in the start-of-authority (SOA) record to those specified. See below for description of the start-of-authority (SOA) record.
- p *domain*  
Create only pointer (PTR) data for hosts in *domain*. This is useful when there are multiple domains on a network and a different server is responsible for *domain*, but this server is responsible for the address-to-name mapping. This option can be used more than once on the command line. This option requires domain names in the host table.
- q  
Run quietly. No messages are printed.
- r  
Create name server data indicating that the name server is authoritative for `.` (the root of the domain tree). The file created is `db.root.` Use this only when your network is isolated from the Internet. If other root servers exist for the isolated network, they must be added manually.
- s *server*  
Create name server (NS) records that declare *server* is an authoritative name server for all of the domains created. If more than 1 server is authoritative, each needs to be declared. If the server name does not have any dots in it, the default domain is appended. The default server is the host this script is run on. This option can be used more than once on the command line.
- t  
Create text (TXT) records from the comments that appear with host data. The comments will all be in lower case because the host table is translated to lower case. If `[no smtp]` appears in a comment, it is omitted. The `[no smtp]` is used to control mail exchanger (MX) data.
- u *user*  
Declare *user* to be the electronic mail address of the person responsible for this domain. This is used in the start of authority (SOA) record. The format required in the name server data is *user.host* (host must be a domain name). If given as *user*, the host on which this script is run is appended. If given as *user@host*, the @ is replaced with a dot (.). The default user is `root`.
- w  
Create well known services (WKS) data declaring that the host provides the SMTP service. This is done only when mail exchanger (MX) data is also being created and only for hosts without `[no smtp]` in a comment.
- z *internet-address*  
Create a secondary boot file, `boot.sec.save` (if `named` is 4.x) or `conf.sec.save` (if `named` is 8.x), from the primary boot file listing *internet-address* as the server to load the data from. The boot file has the server back up the data on disk. The *internet-address* defaults to the value used with `-Z`. This option can be used more than once.
- A  
Do not create name server data for aliases in the host table.
- C *file*  
Create resource records from strings in the comment field of the host table. Each string in the comment field (except `[no smtp]`) is searched for in *file*. The format of *file* is a string, a colon, and a resource record. If the string in the comment field matches the string before the colon in *file*, a resource record is added consisting of the name of the host followed by everything after the colon from the matching line in *file*. For example, host information (HINFO) records can be created by adding `360:IN HINFO hp9000s360 hp-ux` to *file* and adding 360 to comments in the host table.
- D  
Do not create name server data for domain names in the host table.
- F  
By default, the serial number is incremented for a domain only if the data has changed (pointer (PTR) data only). This option forces the serial number to be incremented, even if the data has not changed.
- H *host-file*  
Use *host-file* instead of `/etc/hosts`.
- M  
Do not create mail exchanger (MX) records for hosts in the host table.
- N *mask*  
Apply the default subnet mask *mask* to each *network-number* specified with `-n` except for ones with their subnet masks already provided. *mask* must be in dot notation. This is the same as supplying each *network-number* for a subnet using multiple `-n` options.

h



- S server** This option is the same as the **-s** option, but it only applies to the last *domain* specified with **-d** or the last *network-number* specified with **-n**. This option is for when *server* is backing up some, but not all, of the domains.
- Z internet-address** Create a secondary boot file, **boot.sec** (if **named** is 4.x) or **conf.sec** (if **named** is 8.x), from the primary boot file listing *internet-address* as the server to load the data from. The boot file does not have the server back up the data on disk. The *internet-address* defaults to value used with **-z**. This option can be used more than once.
- 1** This option is obsolete.

**hosts\_to\_named** translates the host table to lower case to help eliminate duplicate data. Since the name server treats uppercase and lowercase as equivalent, names that differ only in case are considered the same.

Alias (CNAME) records are created for *subdomains* delegated with **-c**. Lines from the host table that contain names in *subdomains* from **-c** and **-e** are removed from the lowercase copy of the host table.

The host table is then used to create the name server data for each *network-number* declared on the command line. Do not include the trailing 0's in the network number. No distinction is made between class A, B, or C addresses nor is there any understanding of subnets unless a subnet mask is supplied. Example network numbers are: 10 (for all addresses of the form 10.\*.\*.\*), 10.1 (for addresses of the form 10.1.\*.\*), or 10.2.2 (for addresses of the form 10.2.2.\*).

Address (A) records are created for mapping hostnames to IP addresses. Alias (CNAME) records are created for aliases of hosts that are not multi-homed. The data are placed in a file named **db.DOMAIN** where *DOMAIN* is the first part of the domain from the command line. For the domain **div.inc.com**, the file is named **db.div**. All other name server data goes in this file except the pointer (PTR) records described below.

Pointer (PTR) records are created for mapping IP addresses to host names. PTR records are placed in a file named **db.NET** where *NET* is the network number from the command line. Network 10 data is placed in **db.10**. Network 10.1 data are placed in "db.10.1".

Mail exchanger (MX) records are created unless the **-M** option is used. The default MX record has a weight of 10 with the host itself as its mail exchanger. No default MX record is created for a host if [**no smtp**] is in the comment section of that line in the host table. MX records for each mail hub declared with the **-m** option are added for each host even if [**no smtp**] is in the comment section.

Well known services (WKS) records are created for each host that handles SMTP mail (does not have [**no smtp**]) if **-w** is used. The only service listed is SMTP.

Text (TXT) records are created for comments associated with hosts in the host table if **-t** is used. The comments do not include [**no smtp**].

For each domain, a start of authority (SOA) record is created. The SOA record requires 2 domain names: the host that the data is created on and the electronic mail address of the person responsible. The **-h** and **-u** options influence the names. In addition, the SOA record requires 5 values: a serial number, a refresh time, a retry time, an expire time, and a minimum ttl (time to live). The first time the data is created, the serial number is set to 1, the refresh time is set to 3 hours, the retry time is set to 1 hour, the expire time is set to 1 week, and the minimum ttl is set to 1 day. The **-o** option changes these values except for the serial number. Each subsequent time **hosts\_to\_named** is run, the serial number is incremented. If any of the other fields in the SOA record are modified, the changed values are retained.

If there are files named **spcl.DOMAIN** or **spcl.NET** in the current directory, **\$INCLUDE** directives are added to the corresponding **db.DOMAIN** or **db.NET** file for the **spcl** file. In this way, special data can be added to the data generated by **hosts\_to\_named**.

The first time **hosts\_to\_named** is run, it creates a default boot file for a primary name server. Each subsequent time **hosts\_to\_named** is run, the boot file is updated if necessary. New entries are made in the boot file for any additional networks or domains not already in the boot file. No entries are deleted from the boot file.

The boot file for a caching-only server, **boot.cacheonly** (if **bind** is 4.x) or **conf.cacheonly** (if **bind** is 8.x), is created if it does not exist. The boot files for secondary servers, (**boot.sec.save** or **conf.sec.save**) and (**boot.sec** or **conf.sec**), are created if the **-z** or **-Z** options are used. The boot files for secondary servers are created new each time from the primary server boot file so that they are

equivalent.

## EXAMPLES

Create name server data for networks 15.19.8 and 15.19.9 in **div.inc.com**.

```
hosts_to_named -d div.inc.com -n 15.19.8 -n 15.19.9
```

Create name server data for networks 15.19.8 and 15.19.9 in **div.inc.com**. Ignore aliases in the host table and include 2 mail hubs - **aaa.div.inc.com** and **bbb.mkt.inc.com**. Put all of the options in a file.

```
hosts_to_named -f option_file
```

**Option\_file** contains the following lines:

```
-d div.inc.com
-n 15.19.8 -n 15.19.9
-m 20:aaa
-m 30:bbb.mkt.inc.com
-A
```

Network 15.19.15 has hosts in the **xx.inc.com** domain and the **div.inc.com** domain. Create name server data for **xx.inc.com**. Create only pointer (PTR) data for hosts in **div.inc.com** on network 15.19.15 (this requires the hosts in **div.inc.com** to have the canonical name or an alias of the form **x.div.inc.com**).

```
hosts_to_named -d xx.inc.com -n 15.19.15 -p div.inc.com
```

Create name server data for network 15.19.8 in **div.inc.com**. Include **div.inc.com** data from network 15.19.15 but do not create pointer (PTR) data for 15.19.15 since that is being handled by the **xx.inc.com** server.

```
hosts_to_named -d div.inc.com -n 15.19.8 -a 15.19.15
```

## AUTHOR

**hosts\_to\_named** was developed by HP.

## FILES

|                          |                                           |
|--------------------------|-------------------------------------------|
| <b>/etc/hosts</b>        | The host table                            |
| <b>named.boot</b>        | Primary server boot file (4.x)            |
| <b>named.conf</b>        | Primary server boot file (8.x)            |
| <b>boot.cacheonly</b>    | Caching only server boot file (4.x)       |
| <b>conf.cacheonly</b>    | Caching only server boot file (8.x)       |
| <b>boot.sec.save</b>     | Secondary server boot file (4.x)          |
| <b>conf.sec.save</b>     | Secondary server boot file (8.x)          |
| <b>boot.sec</b>          | Secondary server boot file (4.x)          |
| <b>conf.sec</b>          | Secondary server boot file (8.x)          |
| <b>db.127.0.0</b>        | Pointer information for 127.0.0.1         |
| <b>db.cache</b>          | Stub cache file for root server addresses |
| <b>db.root</b>           | Data for servers for the root domain      |
| <b>db.DOMAIN</b>         | Address and other data for a domain       |
| <b>db.DOMAIN.in-addr</b> | Pointer data for all network-numbers      |
| <b>db.NET</b>            | Pointer data for a network-number         |

## SEE ALSO

**named(1M)**, **RFC1034**, **RFC1035**.

## NAME

hpux - HP-UX bootstrap

## SYNOPSIS

```
hpux [-F] [-lm] [-lq] [-a[C|R|S|D] devicefile] [-f number] [-i string] [boot] [devicefile]
hpux ll [devicefile] (same as hpux ls -aFln)
hpux ls [-aFln] [devicefile]
hpux set autofile devicefile string
hpux show autofile [devicefile]
hpux -v
hpux restore devicefile (Series 700 only; see DEPENDENCIES.)
```

## DESCRIPTION

**hpux** is the HP-UX specific secondary system loader (SSL) utility for bootstrap (see *isl*(1M) for the initial system loader). It supports the operations summarized below, as shown in the *SYNOPSIS* and detailed later in this *DESCRIPTION*.

|                      |                                                                                                                                                                    |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>boot</b>          | Loads an object file from an HP-UX file system or raw device and transfers control to the loaded image. (Note, the <b>boot</b> operation is position dependent).   |
| <b>ll</b>            | Lists the contents of HP-UX directories in a format similar to <b>ls -aFln</b> . (See <i>ls</i> (1); <b>ls</b> only works on a local disk with a HFS file system). |
| <b>ls</b>            | Lists the contents of HP-UX directories. (See <i>ls</i> (1); <b>ls</b> only works on a local disk with a HFS file system).                                         |
| <b>show autofile</b> | Displays the contents of the <b>autoexecute</b> file.                                                                                                              |
| <b>set autofile</b>  | Changes the contents of the <b>autoexecute</b> file to that specified by <i>string</i> .                                                                           |
| <b>-v</b>            | Displays the release and version numbers of the <b>hpux</b> utility.                                                                                               |
| <b>restore</b>       | Recovers the system from a properly formatted bootable tape. (Series 700 specific; see <i>DEPENDENCIES</i> .)                                                      |

**hpux** commands can be given interactively from the keyboard, or provided in an *isl* **autoexecute** file.

**hpux** is limited to operations on the interface initialized by *pdcc*(1M). In most cases, operations are limited to the boot device interface.

## Notation

**hpux** accepts numbers (numeric constants) in many of its options. Numbers follow the C language notation for decimal, octal, and hexadecimal constants. A leading 0 (zero) implies octal and a leading 0x or 0X implies hexadecimal. For example, 037, 0x1F, 0X1f, and 31 all represent the same number, decimal 31.

**hpux boot**, **ll**, **ls**, **set autofile**, **show autofile**, and **restore** operations accept *devicefile* specifications, which have the following format:

```
manager( w/x.y.z;n) filename
```

The *devicefiles* specification is comprised of a device name and a file name. The device name (*manager( w/x.y.z;n)*), consists of a generic name of an I/O system *manager* (device or interface driver) such as **disc**, a hardware path to the device, and minor number. The *manager* name can be omitted entirely if the default is used. *w/x.y.z* is the physical hardware path to the device, identifying bus converters, slot numbers, and hardware addresses. For Series 700 machines, there are a set of mnemonics that can be used instead of the hardware paths. The *n* is the minor number that controls *manager*-dependent functionality. The file name part, *filename*, is a standard HP-UX path name. Some **hpux** operations have defaults for particular components. A *devicefile* specification containing a device part only specifies a raw device. A *devicefile* specification containing a file name implies that the device contains an HP-UX file system, and that the *filename* resides in that file system.

A typical boot *devicefile* specification is

```
disc(2/4.0.0;0)/stand/vmunix
```

The *manager* is **disc**, the hardware path to the disk device is **2/4.0.0**, the minor number shown as **0** by default, and the **/stand/vmunix** is the *filename* for the boot device.

**hpux** now supports a consolidated list of managers: **disc**, **tape**, and **lan**. The manager **disc** manages all disks connected via SCSI, (formerly **disc3**), and all *autochanger* disk devices (formerly **disc30**). The

manager **lan** manages remote boot through the HP 28652A NIO based LAN interface (formerly **lan1**). Remote boot is currently supported on this card only and not on any CIO-based LAN card. The manager **tape** manages tape drives via SCSI (formerly **tape2**).

The hardware path in a *devicefile* specification is a string of numbers, each suffixed by slash, (/), followed by a string of numbers separated by dots (.), each number identifying a hardware component notated sequentially from the bus address to the device address. A hardware component suffixed by a slash indicates a bus converter and may not be necessary on your machine. For example, in *w/x.y.z* *w* is the address of the bus converter, *x* is the address of the MID-BUS module, *y* is the CIO slot number, and *z* is the HP27111 bus address.

The minor number, *n*, in a *devicefile* specification controls driver-dependent functionality. (See the manual, *Configuring HP-UX for Peripherals*, for minor-number bit assignments of specific drivers).

File names are standard HP-UX path names. No preceding slash (/) is necessary and specifying one will not cause problems.

### Defaults

Default values chosen by **hpux** to complete a command are obtained through a sequence of steps. First, any components of the command specified explicitly are used. If the command is not complete, **hpux** attempts to construct defaults from information maintained by **pdcc** (see *pdcc(1M)*). If sufficient information to complete the command is unavailable, the **autoexecute** file is searched. If the search fails, any remaining unresolved components of the command are satisfied by hard-coded defaults.

There is no hard-coded default choice for a *manager*; if none can be chosen, **hpux** reports an error.

When the hardware path to the boot device is not specified, **hpux** defaults to information maintained by **pdcc**. The hardware path element has no hard-coded default.

If the minor number element is not supplied, **hpux** takes its default from the **autoexecute** file. Failing that, the hard-coded default of 0 is used.

For the **boot** command, a *devicefile* specification without a file name indicates that the boot device does not contain an HP-UX file system. **hpux** interprets this as a NULL (instead of missing) file name and does not search for a default. If the entire *devicefile* specification is missing, **hpux** searches for a default; either the **autoexecute** file contents or the hard-coded default is chosen.

There are two possible hard-coded default *devicefile* specifications. One hard-coded default *devicefile* specification is **/vmunix**. The other hard-coded default *devicefile* specification is **/stand/vmunix**.

If you have a LVM system where the boot volume and the root volume are on different logical volumes, the kernel would be **/vmunix**. This is because the boot volume will be mounted under **/stand** when the system is up.

For all other configurations, the kernel would be **/stand/vmunix**.

The search order for the hard-coded defaults is **/stand/vmunix** and then **/vmunix**.

### boot Operation

The **boot** operation loads an object file from an HP-UX file system or raw device as specified by the optional *devicefile*. It then transfers control to the loaded image.

Any missing components in a specified *devicefile* are supplied with a default. For example, a *devicefile* of **vmunix.new** would actually yield:

```
disc(8.0.0;0)vmunix.new
```

and a *devicefile* of **(8/0/19/0.14.0)/stand/vmunix**, for booting from the disk at Ultra Wide SCSI address 14, would yield

```
disc(8/0/19/0.14.0;0)/stand/vmunix
```

Regardless of how incomplete the specified *devicefile* may be, **boot** announces the complete *devicefile* specification used to find the object file. Along with this information, **boot** gives the sizes of the **TEXT**, **DATA**, and **BSS** segments and the entry offset of the loaded image, before transferring control to it.

The **boot** operation accepts several options. Note that **boot** options *must* be specified positionally as shown in the syntax statement in the SYNOPSIS. Options for the **boot** operations are as follows:

- a[C|R|S|D] devicefile** Accept a new location (as specified by *devicefile*) and pass it to the loaded image. If that image is an HP-UX kernel, the kernel will erase its predefined I/O configuration, and configure in the specified *devicefile*. If the **C**, **R**, **S**, or **D** option is specified, the kernel configures the *devicefile* as the **console**, **root**, **swap**, or **dump** device, respectively. Note that **-a** can be repeated multiple times.
- f number** Use the number and pass it as the flags word to the loaded image.
- i string** Set the initial *run-level* for **init** (see *init(1M)*) when booting the system. The *run-level* specified will override any *run-level* specified in an *initdefault* entry in **/etc/inittab** (see *inittab(4)*).
- lm** Boot the system in LVM maintenance mode, configure only the root volume, and then initiate single user mode.
- lq** Boot the system with quorum override option. This option is used in a scenario where a disk is removed from the system or is otherwise unavailable, but the corresponding entry for the physical volume has not yet been removed from the volume group using **vgreduce**.
- F** Used with SwitchOver/UX software. However, SwitchOver/UX is not supported on HP-UX 10.30 or later systems. The **-F** option is used to ignore any locks on the boot disk. The **-F** option should be used only when it is known that the processor holding the lock is no longer running. (If this option is not specified and a disk is locked by another processor, the kernel will not boot from it, to avoid the corruption that would result if the other processor were still using the disk).

**boot** places some restrictions on object files it can load. It accepts only the HP-UX magic numbers **EXEC-MAGIC** (0407), **SHAREMAGIC** (0410), and **DEMANDMAGIC** (0413). See *magic(4)*. The object file must contain an Auxiliary Header of the **HPUX\_AUX\_ID** type and it must be the first Auxiliary Header (see *a.out(4)*).

## ll and ls Operations

The **ll** and **ls** operations list the contents of the HP-UX directory specified by the optional *devicefile*. The output is similar to that of **ls -aFl** command, except the date information is not printed.

The default *devicefile* is generated just as for **boot**, defaulting to the current directory.

## set autofile Operation

The **set autofile** operation overwrites the contents of the **autoexecute** file, *autofile*, with the string specified (see **autoexecute** in the *EXAMPLES* section).

## show autofile Operation

The **show autofile** operation displays the contents of the **autoexecute** file, *autofile* (see **autoexecute** in the *EXAMPLES* section).

## DIAGNOSTICS

If an error is encountered, **hpux** prints diagnostic messages to indicate the cause of the error. These messages fall into the General, Boot, Copy, Configuration, and System Call categories. System Call error messages are described in *errno(2)*. The remaining messages are listed below.

### General

**bad minor number in devicefile spec**

The minor number in the *devicefile* specification is not recognized.

**bad path in devicefile spec**

The hardware path in the *devicefile* specification is not recognized.

**command too complex for parsing**

The command line contains too many arguments.

**no path in devicefile spec**

The *devicefile* specification requires (but does not contain) a hardware path component.

**panic** (in **hpuxboot**): (**display**== *number*, **flags**== *number*) *string*  
 A severe internal **hpux** error has occurred. Report to your nearest HP Field Representative.

**Boot****bad magic**

The specified object file does not have a recognizable magic number.

**bad number in flags spec**

The flags specification in the **-f** option is not recognized.

**Exec failed: Cannot find /stand/vmunix or /vmunix.**

Neither /stand/vmunix or /vmunix could be found.

**booting from raw character device**

In booting from a raw device, the *manager* specified only has a character interface, which might cause problems if the block size is incorrect.

**isl not present, please hit system RESET button to continue**

An unsuccessful boot operation has overlaid **isl** in memory. It is impossible to return control to **isl**.

**short read**

The specified object file is internally inconsistent; it is not long enough.

**would overlay**

Loading the specified object file would overlay **hpux**.

**Configuration****cannot add path, error *number***

An unknown error has occurred in adding the hardware path to the I/O tree. The internal error number is given. Contact your HP Field Representative.

**driver does not exist**

The manager specified is not configured into **hpux**.

**driver is not a logical device manager**

The *manager* named is not that of a logical device manager and cannot be used for direct I/O operations.

**error rewinding device**

An error was encountered attempting to rewind a device.

**error skipping file**

An error was encountered attempting to forward-space a tape device.

**negative skip count**

The skip count, if specified, must be greater than or equal to zero.

**no major number**

The specified *manager* has no entry in the block or character device switch tables.

**path incompatible with another path**

Multiple incompatible hardware paths have been specified.

**path long**

The hardware path specified contains too many components for the specified *manager*.

**path short**

The hardware path specified contains too few components for the specified *manager*.

**table full**

Too many devices have been specified to **hpux**.

h

**EXAMPLES**

As a preface to the examples which follow, here is a brief overview of HP-UX system boot-up sequences.

**Automatic Boot**

Automatic boot processes on various HP-UX systems follow similar general sequences. When power is applied to the HP-UX system processor, or the system **Reset** button is pressed, processor-dependent code (firmware) is executed to verify hardware and general system integrity (see *pdcc(1M)*). After checking the hardware, *pdcc* gives the user the option to override the **autoboot** sequence by pressing the **Esc** key. At that point, a message resembling the following usually appears on the console.

```
(c) Copyright. Hewlett-Packard Company. 1994.
All rights reserved.
```

```
PDC ROM rev. 130.0
32 MB of memory configured and tested.
```

```
Selecting a system to boot.
```

```
To stop selection process, press and hold the ESCAPE key...
```

If no keyboard activity is detected, *pdcc* commences the **autoboot** sequence by loading *isl* (see *isl(1M)*) and transferring control to it. Since an **autoboot** sequence is occurring, *isl* finds and executes the **autoexecute** file which, on an HP-UX system, requests that **hpux** be run with appropriate arguments. Messages similar to the following are displayed by *isl* on the console:

```
Booting from: scsi.6 HP 2213A
Hard booted.
ISL Revision A.00.09 March 27, 1990
ISL booting hpux boot disk(;0)/stand/vmunix
```

**hpux**, the secondary system loader, then announces the operation it is performing, in this case **boot**, the *devicefile* from which the load image comes, and the **TEXT** size, **DATA** size, **BSS** size, and start address of the load image, as shown below, before control is passed to the image.

```
Booting disk(scsi.6;0)/stand/vmunix
966616+397312+409688 start 0x6c50
```

The loaded image then displays numerous configuration and status messages.

**Interactive Boot**

To use **hpux** interactively, *isl* must be brought up in interactive mode by pressing the **Esc** key during the interval allowed by *pdcc*. *pdcc* then searches for and displays all bootable devices and presents a set of boot options. If the appropriate option is chosen, *pdcc* loads *isl* and *isl* interactively prompts for commands. Information similar to the following is displayed:

```
Selection process stopped.
```

```
Searching for Potential Boot Devices.
```

```
To terminate search, press and hold the ESCAPE key.
```

| Device Selection | Device Path           | Device Type    |
|------------------|-----------------------|----------------|
| P0               | scsi.6.0              | QUANTUM PD210S |
| P1               | scsi.1.0              | HP 2213A       |
| p2               | lan.ffffff-ffffff.f.f | hpfooobar      |

```
b) Boot from specified device
s) Search for bootable devices
a) Enter Boot Administration mode
x) Exit and continue boot sequence
```

```
Select from menu: b p0 isl
```

```
Trying scsi.6.0
```

```
Boot path initialized.
```

```
Attempting to load IPL.
```

```
Hard booted.
```

```
ISL Revision A.00.2G Mar 27, 1994
```

```
ISL>
```

Although all of the operations and options of **hpux** can be used from **isl** interactively, they can also be executed from an **autoexecute** file. In the examples below, user input is the remainder of the line after each **ISL>** prompt shown. The remainder of each example is text displayed by the system. Before going over specific examples of the various options and operations of **hpux**, here is an outline of the steps taken in the automatic boot process. Although the hardware configuration and boot paths shown are for a single Series 800 machine, the user interfaces are consistent across all models. When the system **Reset** button is depressed, **pdcc** executes self-test, and assuming the hardware tests pass, **pdcc** announces itself, sends a BELL character to the controlling terminal, and gives the user 10 seconds to override the **autoboot** sequence by entering any character. Text resembling the following is displayed on the console:

```
Processor Dependent Code (PDC) revision 1.2
Duplex Console IO Dependent Code (IODC) revision 3

Console path          = 56.0.0.0.0.0.0    (dec)
                      38.0.0.0.0.0.0    (hex)

Primary boot path     = 44.3.0.0.0.0.0    (dec)
                      2c.00000003.0.0.0.0.0 (hex)

Alternate boot path   = 52.0.0.0.0.0.0    (dec)
                      34.0.0.0.0.0.0    (hex)

32 MB of memory configured and tested.
Autosearch for boot path enabled
To override, press any key within 10 seconds.
```

If no keyboard character is pressed within 10 seconds, **pdcc** commences the **autoboot** sequence by loading **isl** and transferring control to it. Because an **autoboot** sequence is occurring, **isl** merely announces itself, finds and executes the **autoexecute** file which, on an HP-UX system, requests that **hpux** be run with appropriate arguments. The following is displayed on the console.

```
10 seconds expired.
Proceeding with autoboot.

Trying Primary Boot Path
-----
Booting...
Boot IO Dependent Code (IODC) revision 2
HARD Booted.
ISL Revision A.00.2G Mar 20, 1994
ISL booting hpux
```

**hpux** then announces the operation it is performing, in this case **boot**, the *devicefile* from which the load image comes, and the **TEXT** size, **DATA** size, **BSS** size, and start address of the load image. The following is displayed before control is passed to the image.

```
Boot
: disc3(44.3.0;0)/stand/vmunix
3288076 + 323584 + 405312 start 0x11f3e8
```

Finally, the loaded image displays numerous configuration and status messages, then proceeds to **init run-level 2** for multiuser mode of operation.

h



**isl** must be brought up in interactive mode to use the operations and options of **hpux**. To do this, simply enter a character during the 10 second interval allowed by **pdcc**. **pdcc** then asks if the primary boot path is acceptable. Answering yes (Y) is usually appropriate. **pdcc** then loads **isl** and **isl** interactively prompts for commands. The following lines show the boot prompt, the Y response, subsequent boot messages, and finally the Initial System Loader (ISL) prompt that are sent to the display terminal:

```

Boot from primary boot path (Y or N)?> y
Interact with IPL (Y or N)?> y

Booting...
Boot IO Dependent Code (IODC) revision 2
HARD Booted.
ISL Revision A.00.2G Mar 20, 1994
ISL>

```

Although all of the operations and options of **hpux** can be used from **isl** interactively, they can also be executed from an **autoexecute** file. In the examples below, all user input follows the **ISL>** prompt on the same line. Subsequent text is resultant messages from the ISL.

### Default Boot

Entering **hpux** initiates the default boot sequence. The boot path read from **pdcc** is 8.0.0, the manager associated with the device at that path is **disc**, the minor number, in this case derived from the **autoexecute** file, is 4 specifying section 4 of the disk, and the object file name is **/stand/vmunix**.

```

ISL> hpux

Boot
: disc3(44.3.0;0)/stand/vmunix
3288076 + 323584 + 405312 start 0x11f3e8

```

### Booting Another Kernel

In this example, **hpux** initiates a boot operation where the name of the object file is **vmunix.new**.

```

ISL> hpux vmunix.new

Boot
: disc3(44.3.0;0)/stand/vmunix.new
3288076 + 323584 + 405312 start 0x11f3e8

```

### Booting From Another Section

In this example (shown for backward compatibility), a kernel is booted from another section of the root disk. For example, suppose kernel development takes place under **/mnt/azure/root.port** which happens to reside in its own section, section 3 of the root disk. By specifying a minor number of 3 in the above example, the object file **sys.azure/S800/vmunix** is loaded from **/mnt/azure/root.port**.

```

ISL> hpux (;3)sys.azure/S800/vmunix

Boot
: disc(8.0.0;0x3)sys.azure/S800/vmunix
966616+397312+409688 start 0x6c50

```

### Booting From Another Disk

Only the hardware path and file name are specified in this example. All other values are boot defaults. The object file comes from the file system on another disk.

```

ISL> hpux (52.5.0.0)/stand/vmunix

Boot
: disc(52.5.0.0)/stand/vmunix
966616+397312+409688 start 0x6c50

```

### Booting From LAN

This example shows how to boot a cluster client from the LAN. Though this example specifies a *devicefile*, you can also use default boot, as shown in a previous example. For a boot operation other than default boot, the file name must be specified and can be no longer than 11 characters. Booting to **isl** from a local disk then requesting an image to be loaded from the LAN is *not* supported.

```
ISL> hpux lan(32)/stand/vmunix
Boot
: lan(32;0x0)/stand/vmunix
966616+397312+409688 start 0x6c50
```

### Booting To Single User Mode

In this example, the **-i** option is used to make the system come up in *run-level s*, for single user mode of operation.

```
ISL> hpux -is
Boot
: disc(8.0.0;0x0)/stand/vmunix
966616+397312+409688 start 0x6c50
(Kernel Startup Messages Omitted)
INIT: Overriding default level with level 's'
INIT: SINGLE USER MODE
WARNING: YOU ARE SUPERUSER !!
#
```

### Booting With A Modified I/O Configuration

Here, a disc driver is configured in at CIO slot 5, SCSI address 5 as a dump device. Regardless of what was present in the kernel's original I/O configuration, the driver **disc** is now configured at that hardware path. Similarly, **asio0** is configured in at CIO slot 63 which is to be the console. The only other devices configured are the console and root device, which **boot** derived from **pdcc**.

```
ISL> hpux -aC asio0(8/0/63) -aD disc(8/16/5.5)
Boot
: disk(8/0/19/0.14.0.0.0.0.0;0)/stand/vmunix
: Adding console (8.0.63;0)...
: Adding dump (8.16.5.5;0)...
6463488 + 1101824 + 939616 start 0x39168
(Additional Kernel Startup Messages Omitted)
```

### Displaying The Autoexecute File

In this example, **show autofile** is used to print the contents of the **autoexecute** file residing in the boot LIF, on the device from which **hpux** was booted. Optionally, a *devicefile* can be specified in order to read the **autoexecute** file from the boot LIF of another boot device.

```
ISL> hpux show autofile
Show autofile
: AUTO file contains (hpux)
```

### Changing The Autoexecute File

This example shows how to change the contents of the **autoexecute** file. Once done, the system can be reset, and the new command will be used during any unattended boot.

```
ISL> hpux set autofile "hpux /stand/vmunix.std"
Set autofile
: disk(2/0/1.3.0.0.0.0.0;0)
: AUTO file now contains "(hpux /stand/vmunix.std)"
```

### Listing Directory Contents

The contents of the directory (**/stand**) on the root disk are listed. The format shows the file protections, number of links, user id, group id, and size in bytes for each file in the directory. There are three available kernels to boot: **vmunix**, **vmunix.test**, and **vmunix.prev**. Listing the files over the LAN is not supported.

```
ISL> hpux ll /stand
Ls
: disk(2/0/1.3.0.0.0.0.0;0)/stand
dr-xr-xr-x    3  2      2      1024 ./
```

|            |    |   |   |         |              |
|------------|----|---|---|---------|--------------|
| drwxr-xr-x | 17 | 0 | 0 | 1024    | ../          |
| -rw-r--r-- | 1  | 0 | 3 | 191     | bootconf     |
| drwxr-xr-x | 2  | 0 | 0 | 1024    | build/       |
| -rw-r--r-- | 1  | 0 | 0 | 632     | ioconfig     |
| -rw-r--r-- | 1  | 0 | 3 | 82      | kernrel      |
| -r--r--r-- | 1  | 0 | 3 | 426     | system       |
| -rw-r--r-- | 1  | 0 | 3 | 437     | system.prev  |
| -rwxr-xr-x | 1  | 0 | 3 | 7771408 | vmunix*      |
| -rwxr-xr-x | 1  | 0 | 3 | 7771408 | vmunix.prev* |

### Getting The Version

The **-v** option is used to get the version numbers of **hpux**.

```
ISL> hpux -v
Release: 10.00
Release Version:
@(#) X10.20.B HP-UX() #1: Dec  4 1995 16:55:08
```

### DEPENDENCIES

#### Series 700 Only

The **restore** operation is provided as a recovery mechanism in the event that a disk becomes totally corrupted. It copies data from a properly formatted bootable tape to disk. When this tape contains a backup image of the disk, the entire disk is restored. To create a properly formatted tape (DDS ONLY), the following commands should be executed:

```
dd if=/usr/lib/uxbootlf of=/dev/rmt/0mn bs=2k
dd if=/dev/rdisk/lss of=/dev/rmt/0m bs=64k
```

The first **dd** puts a boot area on the tape, making it a bootable image (see **dd(1)**). Once the boot image is on tape, the tape is *not* rewound. The next **dd** appends an image of the disk to the tape. The entire process takes about one hour for a 660 MB HP 2213 disk. To avoid later problems with **fsck** after the disk is restored, bring the system to single user mode and type **sync** a few times before doing the second **dd** (see **fsck(1M)**). Once created, the tape can be used to completely restore the disk:

1. Insert the tape into the tape drive.
2. Instruct the machine to boot to ISL from the tape. This is usually done by specifying **scsi.3** as the boot path.
3. Enter the following in response to the ISL prompt:

```
ISL> hpux restore disk(scsi.1;0)
```

This restores the disk image from the tape to the actual disk at **scsi.1**. *Any existing data on the disk will be lost.* This command destroys the contents of the device specified by *devicefile*. The restoration process takes about one hour for a 660 MB drive.

**NOTE:** There is a 2 GB limit on the amount of data that can be restored. The tape and disk must be on the boot device interface.

Also, this command may be replaced in the future by superior installation and recovery mechanisms. At that time, this command will be removed.

### SEE ALSO

boot(1M), fsck(1M), init(1M), isl(1M), pdc(1M), errno(2), a.out(4), inittab(4), magic(4).

**NAME**

i4admin - administer LicensePower/iFOR licensing

**SYNOPSIS**

```
i4admin [-Standard-X-Arguments]
i4admin -a [-n server-name] [-f filename] [-v "'vendor-name' [vendor-id vendor-password]"
-p "'product-name' 'product-version' license-password ['license-annotation']"]
i4admin -d [-n server-name] -v vendor-name -p product-name -t timestamp
i4admin -l s|v|p [-i] [-n "server-name..."] [-v "'vendor-name' ..."] [-p "'product-name' ..."]
[-u "user-name..."]
i4admin -s [-n "server-name..."] [-v "'vendor-name' ..."] [-p "'product-name' ..."]
[-u "user-name..."]
i4admin -r 1|2|3|4|5 [-e 1|234567] [-b start-date] [-g end-date] [-n "server-name..."]
[-v "'vendor-name' ..."] [-p "'product-name' ..."] [-u "user-name..."]
i4admin -x before-date -n "server-name..."
i4admin -h
```

**DESCRIPTION**

The LicensePower/iFOR Administration tool, **i4admin**, completely manages the LicensePower/iFOR licensing system. The tool can perform the following tasks:

- Perform basic license administration (e.g., adding and deleting licenses).
- Construct a single logical view of the license system from which current summary license usage and current detailed license usage reports can be generated.
- Generate detailed license event and license usage reports from logged server data.

The **i4admin** tool has a Graphical User Interface (GUI) and a Command Line Interface (CLI). If **i4admin** is invoked with non-X arguments, the CLI version is started, otherwise the GUI version is started.

A printable on-line administration guide is also available. (See the *FILES* section below.)

**CLI Actions**

The CLI is invoked with one of the following actions, and one or more action modifiers.

*XToolkit Options*

This command supports a subset of the standard X Toolkit options to control the appearance of the GUI. See the *X(1)* manual page for a definition of these options.

- a Add a product license to a specified license server. There are two ways to add a license to a license server.

If the license information has been provided in the form of a license certificate (a flat file describing the license), the license certificate can be added by specifying the *server-name* and the license certificate *filename*. If the server name is omitted, the license is added to the license server running on the local machine.

If the license information has not been provided in a license certificate, the parameters must be entered individually. All three vendor parameters are not always required. If the vendor for the product is already installed on the server, only the *vendor-name* must be specified, otherwise the *vendor-name*, *vendor-id* and *vendor-password* must be specified.

- d Delete a product license. To delete a compound password, or a use-once license, the license must have expired. If the server name is omitted, the license is deleted from the license server running on the local machine. The license *timestamp* must be specified to differentiate between licenses for the same product (same Vendor ID, Product ID, and Product version), which are installed on the same server. The license *timestamp* can be found using the list product details command:

```
i4admin -lp -i -p product-name
```

- l List installed license information. The command is qualified by the list type flag, *s|v|p*, to list servers, vendors, or products respectively.

## (TO BE OBSOLETE)

The vendor list can be limited to specific servers by entering one or more *server-names*. If more than one *server-name* is entered, the list must be enclosed in double quotes.

By default the product list contains a summary of product information. Detailed product information can be queried by specifying the *-i* parameter. The product list can be filtered by server, vendor, and user. If more than one *vendor-name* is entered, the list of *vendor-names* must be enclosed in double quotes. Any *vendor-name* which contains white space must also be enclosed in single quotes.

Specify one or more *user-names* to limit the product list to products currently in use by the those users.

- s Generate a status report containing detailed current license usage. For each product, the report includes the number of licenses in use, the user of the product and when license was acquired. By default the status report is generated based on all active license servers in the cell. The scope of the report can be limited by specifying *server-names*, *vendor-names*, *product-names*, or *user-names*.
- r Generates reports which are based on license events logged by the license server. The command will generate one of five reports specified by the report-type flag (1|2|3|4|5).
  - 1 Reports server log events. This command is further qualified by the event-flag which is described below.
  - 2 For each product lists the number of requests for licenses, the number of licenses granted, and the percent of rejected requests.
  - 3 Lists the same information as 2 but breaks out a separate entry for each user.
  - 4 For each product, lists the maximum concurrent nodes, maximum concurrent users, and average time in use.
  - 5 For each product, lists the number of times each user invoked the product and the average time the product was in use.
- x *before-date*  
Delete all log entries on the servers specified by *server-names* which are timestamped on or before *before-date*
- h Display a synopsis of command-line options

**CLI Action Modifiers**

- b *start-date*  
Specify the start date for generating log reports. By default the start date is Jan. 1 1970.
- e *event-type*  
Specify an event filter for the standard event report (-r1). By default all events are listed.
  - 1 All events (default)
  - 2 License related events (license request, license release, etc.)
  - 3 Vendor messages
  - 4 License database modifications (license added, license deleted, etc.)
  - 5 Error events (license request failed, vendor not found, etc.)
  - 6 Server start/stop
  - 7 Fatal error events (server out of memory, server file IO error, etc.)

Error events 2-7 can be combined, e.g., -e357 to list vendor messages, error events, and fatal error events.
- f *filename*  
Specifies filename for adding a license certificate.
- g *end-date*  
Specify the end date for generating log reports. By default the end date is current day.
- i Include license details (start date, end-date, multi-use rules, timestamp, etc.) when listing products.
- n "*server-name...*"  
Specify a server when performing administrative actions (adding a license, deleting a license, cleaning the log file), or limit the scope of a listing, status report or event report to a particular server, or

## (TO BE OBSOLETE)

servers. If more than one *server-name* is specified to limit the scope of a listing or report, the entire argument must be enclosed in double quotes.

-p "'*product-name*' '*product-version*' *license-password* ['*license-annotation*']"

Specify a product when adding a license (-a) which is not defined in a license certificate. The entire argument must be enclosed in double quotes. If the *product-name*, *product-version*, or *license-annotation* contains white space the argument must be enclosed in single quotes.

-p "'*product-name*'..."

Specify a product, or products to limit the scope of a product listing (-lp), a status report (-s), or a event report (-r). If multiple *product-names* are specified, the entire argument must be enclosed in double quotes. If any *product-name* contains white space it must be enclosed in single quotes to differentiate the argument from multiple single-word product names.

-u "*user-name*..."

Limit the scope of a status report, or event report to a specific user, or users. If more than one user is specified, the entire argument must be enclosed in double quotes.

-v "'*vendor-name*' [*vendor-id* *vendor-password*]"

Specify a vendor when adding a product license manually. If another product for this vendor has been installed on an active license server in this cell, only the *vendor-name* must be specified. If a product for this vendor has not been previously installed on an active server in this cell, the *vendor-id* and the *vendor-password* must also be specified.

## GUI Description

The i4admin GUI provides an intuitive dialog based interface to manage all aspects of the LicensePower/iFOR licensing system. The main window is divided into four functional areas:

- The menu bar contains pull-down menus which provide the interface to all administrative commands.
- The toolbar provides direct access to frequently used commands.
- All reports are displayed in the scrolling display area.
- When performing a task, the tool displays its progress in the status line at the bottom of the main window.

The GUI tool can perform the following tasks which will be described in detail in succeeding sections.

- Basic license administration which includes adding and deleting licenses.
- Extensive report generation based on current license usage and logged license events.

## GUI Administrative Tasks

The Administrative tasks are adding licenses, deleting licenses, and cleaning up stale licenses. There are two ways to add a license. If the license information has been provided in the form of a license certificate (a flat file describing the license), follow the first procedure. If the license information has been provided in any other form, follow the second procedure.

### Adding a license from a license certificate

1. Open the **Add** pull-down menu and select the **License...** menu item.
2. Select the server to add the license to from the **Server** drop-down listbox.
3. Select the **Read certificate...** button.
4. Enter the name of the license certificate in the **Selection** entry field. The **Filter** entry field and the **Filter** button can be used to limit the selection to a specific file or range of files.
5. Select **OK** to accept the file selection and close the dialog. Verify that the Vendor name, Product name, and Product version appear correctly on the **Add License** panel.
6. Select **OK** to add the license to the selected server and close the **Add license** dialog.

### Adding a license manually

1. Open the **Add** pull-down menu and select the **License...** menu item.
2. Select the server to add the license to from the **Server** drop-down listbox.

## (TO BE OBSOLETE)

3. Select the **Enter manually...** button.
4. Select the product's vendor from the drop down list of vendors which are displayed. If the product's vendor is not displayed, select the **New vendor** button to specify the vendor information.
5. Enter the Product name, Product version, License password, and optional License annotation (if provided) in the fields.
6. Select **OK** to accept the information and close the dialog. Verify that the Vendor name, Product name, and Product version appear correctly on the **Add license** dialog.

**Deleting a license**

1. Change to the **Product details** view. To change views select the desired view from the **View** pull-down menu.
2. Select a license to delete. Note that selected items which can be acted on are distinguished from plain text by the highlight color of the selection.
3. Select **Delete license** button from the **Selected** pulldown menu. The tool will ask for confirmation before deleting the license. Note that compound passwords, and use-once licenses cannot be deleted before their expiration date.

**Cleaning up stale licenses**

When a client application acquires a license from the license server, it also periodically checks back with the server to tell the server the application is still running. The interval between checks is referred to as the check-in period. The server does not automatically release licenses for applications which have missed their check-in period. However, if a client application attempts to acquire a license and none are available, the server will check all the outstanding licenses to make sure the respective clients have checked in. If a client has missed its check-in period, that client's license will be granted. The clean stale license command forces the server to iterate through the outstanding licenses, releasing the licenses which have not been checked.

To clean up stale licenses for a product or products:

1. Select one or more products from the **Product summary** view or the **Product status** view. Multiple entries can be selected by holding the Shift or Control key down while selecting.
2. Open the **Selected** menu and choose the **Clean stale licenses** menu item.

**GUI Usage and Installed License Reporting**

This set of reports are generated based on installed license details, and current usage information. The reports are generated based on a snapshot of the license system at a particular instant in time. Since the license system may be constantly changing, the information contained in these reports is only as current as the last snapshot.

These reports contain information which is summed across the license system. The **i4admin** tool constructs a single logical view of the license system from which these reports are generated. This logical view is referred to as a snapshot of the license system. There are three reports based on the snapshot. The reports are accessed via the **View** pulldown menu.

- The product summary is a terse view of a product's installed licenses and current license usage. From this view the administrator can quickly identify problem areas, i.e., a product has 10 licenses installed, and 10 are in use.
- The product details view reports detailed installed product information, including the number of license installed, the start and expiration date of the licenses, and the server that the license is installed on. >From this view, the administrator can select delete a license.
- The product status view generates a detailed current usage report which includes; the number of licenses installed, the number of licenses currently checked out, who is using the license from what node, and how long the user has had the license.

By default these reports are based on all the installed products and licenses on all the servers contained in the current snapshot. The scope of any of these reports can be limited by applying one or more View Filters. The View filter allows the report to be scoped by server, vendor, product, or user. To change the View filter:

1. Select **Filter...** from the **View** pulldown menu.

## (TO BE OBSOLETE)

2. From the **View filter** dialog select the type of filter to apply.
3. Select **OK** to close the individual filter selection dialog. Select **OK** to close the **View filter** dialog. The view will be immediately updated based on the new view when the **View filter** dialog is closed.

It is important to remember that these reports are only as current as the last snapshot. The snapshot can be updated manually or automatically.

To update the snapshot manually, select **Refresh now** from the **Snapshot** pulldown menu. The snapshot will be immediately updated.

To update the snapshot automatically, open the **Automatic refresh** dialog from the **Snapshot** pulldown menu. Select the **Automatic refresh** radio button, and enter a refresh interval in minutes.

### GUI License Event Reporting

These reports are generated by querying information directly from a server or servers. Since the amount of logged event information may be extensive it is impractical to create a local snapshot of all the log information to generate reports from.

The reports can be filtered using the same View Filter as previously discussed. A log report can be scoped by server, vendor, product, or user. By default, the View filter dialogs allow the administrator to select from the servers, vendors, products, and users which are contained in the current snapshot. If the desired filter item is not contained in the current snapshot, the administrator can manually specify the name in an entry field on the filter dialog.

There are five log reports which are summarized below.

- License event log reports which reports logged server events without deriving additional information. There are seven categories of events which can be included in this reports.
  1. All events
  2. (default) License related events (license request, license release, etc.)
  3. Vendor messages
  4. License database modifications.
  5. Error events (license request failed, vendor not found, etc.)
  6. Server start/stop
  7. Fatal error events (server out of memory, server file IO error, etc.)

Note that error events 2-7 can be combined.

- License requests by product. For each product lists the number of requests for licenses, the number of licenses granted, and the percent of rejected requests.
- License requests by user. Lists the same information and the previous reports, but breaks out a separate entry for each user.
- License use by product. For each product lists the maximum concurrent nodes, maximum concurrent users, and average time in use.
- License use by user. For each product, lists the number of times each user invoked the product and the average time the product was in use.

### WARNINGS

#### LSSERV Software Obsolescence

HP intends to remove LicensePower/iFOR from the core HP-UX product in a future release. The licensing product can be obtained directly from Isogon Corporation, the owner of the product.

For further information about LSSERV support, see the Isogon Corporation website at <http://www.isogon.com>.

Competitive information is also available at these websites:

FLEX/Im via <http://www.globetrotter.com>

SentinelLM via <http://www.rainbow.com>



**FILES**

`/opt/ifor/ls/conf/i4rpt.fmt` Report templates  
`/opt/ifor/ls/res/*.bmp` Icon bitmaps  
`/opt/ifor/ls/res/i4admin.pdl` Panel definitions  
`/opt/ifor/ls/doc/i4admin.pdf` LicensePower/iFOR Administrator's Guide (PDF format)

**AUTHOR**

**i4admin** was developed by HP and Gradient Technologies, Inc. It is currently a product owned by Isogon Corporation.

**SEE ALSO**

**i4lmd(1M)**, **i4start(1M)**, **i4stop(1M)**, **i4target(1M)**, **i4tv(1M)**.

*LicensePower/iFOR Administrator's Guide* available in `/opt/ifor/ls/doc/` in PDF format.

Isogon Corporation information on LicensePower/iFOR at:

<http://www.isogon.com/products/ptlpoweri/ptlpoweri.htm>.

HP-UX documentation at <http://docs.hp.com>.

## (TO BE OBSOLETE)

## NAME

i4lmd - starts the license server on a local node

## SYNOPSIS

```
i4lmd [-s[ecure]] [-l[ogname]] [-v[erbose]] [-z[debugging]] [-n[o] event_types]
      [-c[oldstart]]
```

## DESCRIPTION

The **i4lmd** command starts a license server on the local node. There is no graphic interface for this command, the shell script **i4config** is used to configure the license server. License servers should not be run manually.

A printable on-line administration guide is also available. (See the *FILES* section below.)

**NOTE:** Please refer to the release notes and **i4config** for information on how to automate the start-up of **i4lmd** on your specific platform.

## Options

- s** Secure mode. A LicensePower/iFOR license server running in secure mode will only permit modifications to its database from tools run locally (on the same node). Tools running on remote node are not permitted to modify the database.
- l log\_name** Redirects license server log entries to a file and location other than the default (`/opt/ifor/ls/conf/logdb*`). The alternate log file specification (*filename*) must be fully qualified starting from the root directory (`/`).
- v verbose** The verbose flag should only be used by administrators the event of a server failure. This command allows the administrator to review license calls and activity from the client programs. The **-v** option is used in conjunction with **-z**.
- z debugging** The debugging flag allows the administrator to review all rpc communication between the clients and the server. The **-z** option is used in conjunction with **-v**.
- no** Turns off logging of the events specified in *event\_list*. Any combination of events is valid, but items in the list of events must not be separated by spaces or other characters. Following are the event types that you may specify:
  - l** License-grant and license-release events.
  - c** License checkin events. (Licensed products usually check in with the license server at regular intervals while a user is using the product).
  - w** Waiting events: these include wait events (a user was waiting for a license), waitgrant events (a user was waiting for and then was granted a license), and waitremove events (a user was waiting for a license and then asked to be removed from the queues before a license was granted).
  - v** Vendor events: a vendor was added, renamed or deleted.
  - p** Product events: a product was added, renamed, or deleted.
  - e** Error events.
  - t** License timeout events. (When a licensed product fails to check in with the license server, it may stop running after it "times out." The vendor of the product sets the timeout interval, which is how long a product may run after it has lost contact with the license server).
  - m** Message events.
  - s** License server start/stop events.
- c** This option will delete all transactions records from the database and subsequently that cache during server startup.

## EXAMPLES

Start a license server; do not log checkin, vendor, product, timeout, or message events:

```
i4lmd -no cvptm
```

Start a license server, deleting all transactions from the database:

## (TO BE OBSOLETE)

**i4lmd -c**

Start a license server, overriding the default log file:

**i4lmd -l /logs/license\_server\_log**

**WARNINGS****LSSERV Software Obsolescence**

HP intends to remove LicensePower/iFOR from the core HP-UX product in a future release. The licensing product can be obtained directly from Isogon Corporation, the owner of the product.

For further information about LSSERV support, see the Isogon Corporation website at <http://www.isogon.com>.

Competitive information is also available at these websites:

FLEX/lm via <http://www.globetrotter.com>

SentinelLM via <http://www.rainbow.com>

**FILES**

**/opt/ifor/ls/bin/i4lmd**

**/opt/ifor/ls/bin/i4config**

**/opt/ifor/ls/doc/i4admin.pdf** LicensePower/iFOR Administrator's Guide (PDF format)

**AUTHOR**

**i4lmd** was developed by HP and Gradient Technologies, Inc. It is currently a product owned by Isogon Corporation.

**SEE ALSO**

**i4admin(1M)**, **i4start(1M)**, **i4stop(1M)**, **i4target(1M)**, **i4tv(1M)**.

*LicensePower/iFOR Administrator's Guide* available in **/opt/ifor/ls/doc/** in PDF format.

Isogon Corporation information on LicensePower/iFOR at:  
<http://www.isogon.com/products/ptlpoweri/ptlpoweri.htm>

HP-UX documentation at <http://docs.hp.com>.

**NAME**

i4start - LicensePower/iFOR server start tool

**SYNOPSIS**

**i4start**

**DESCRIPTION**

The **i4start** tool can be used to manually re-start a LicensePower/iFOR license server that has been stopped (for instance, with the **i4stop** tool). It will also start location brokers, if they are needed on the system. The settings of the tool are activated after the first invocation of **i4config**.

A printable on-line administration guide is also available. (See the *FILES* section below.)

**EXAMPLES**

**i4start**

**WARNINGS****LSSERV Software Obsolescence**

HP intends to remove LicensePower/iFOR from the core HP-UX product in a future release. The licensing product can be obtained directly from Isogon Corporation, the owner of the product.

For further information about LSSERV support, see the Isogon Corporation website at <http://www.isogon.com>.

Competitive information is also available at these websites:

FLEX/lm via <http://www.globetrotter.com>

SentinelLM via <http://www.rainbow.com>

**FILES**

**/opt/ifor/ls/bin/i4start**

**/opt/ifor/ls/bin/i4config**

**/opt/ifor/ls/doc/i4admin.pdf** LicensePower/iFOR Administrator's Guide (PDF format)

**AUTHOR**

**i4start** was developed by HP and Gradient Technologies, Inc. It is currently a product owned by Isogon Corporation.

**SEE ALSO**

**i4admin(1M)**, **i4lmd(1M)**, **i4stop(1M)**, **i4target(1M)**, **i4tv(1M)**.

*LicensePower/iFOR Administrator's Guide* available in **/opt/ifor/ls/doc/** in PDF format.

Isogon Corporation information on LicensePower/iFOR at:  
<http://www.isogon.com/products/ptlpoweri/ptlpoweri.htm>

HP-UX documentation at <http://docs.hp.com>.

**NAME**

i4stop - LicensePower/iFOR server stop tool

**SYNOPSIS**

**i4stop**

**DESCRIPTION**

The **i4stop** tool can be used to manually stop a LicensePower/iFOR license server (and location brokers) if they are running on the system. Use this tool **on** the system that contains the active LicensePower/iFOR license server that you want to stop. The tool is located in **/opt/ifor/ls/bin**.

A printable on-line administration guide is also available. (See the *FILES* section below.)

**EXAMPLES**

**i4stop**

**WARNINGS****LSSERV Software Obsolescence**

HP intends to remove LicensePower/iFOR from the core HP-UX product in a future release. The licensing product can be obtained directly from Isogon Corporation, the owner of the product.

For further information about LSSERV support, see the Isogon Corporation website at <http://www.isogon.com>.

Competitive information is also available at these websites:

FLEX/lm via <http://www.globetrotter.com>

SentinelLM via <http://www.rainbow.com>

**FILES**

**/opt/ifor/ls/bin/i4stop**

**/opt/ifor/ls/doc/i4admin.pdf** LicensePower/iFOR Administrator's Guide (PDF format)

**AUTHOR**

**i4stop** was developed by HP and Gradient Technologies, Inc. It is currently a product owned by Isogon Corporation.

**SEE ALSO**

**i4admin(1M)**, **i4lmd(1M)**, **i4start(1M)**, **i4target(1M)**, **i4tv(1M)**.

*LicensePower/iFOR Administrator's Guide* available in **/opt/ifor/ls/doc/** in PDF format.

Isogon Corporation information on LicensePower/iFOR at:  
<http://www.isogon.com/products/ptlpoweri/ptlpoweri.htm>

HP-UX documentation at <http://docs.hp.com>.

**NAME**

i4target - returns the local LicensePower/iFOR target id

**SYNOPSIS**

i4target

i4target [-c] [-C] [-h] [-H] [-o] [-O] [-q] [-Q] [-v] [-V]

**DESCRIPTION**

i4target is used to find the target ID that can be used by LicensePower/iFOR for locking licenses to a particular system.

To create LicensePower/iFOR licenses for an application, an application supplier will need the target ID of the machine where the LicensePower/iFOR licenses will be installed. The target ID tool (i4target) should be run on the machine where you want to identify a LicensePower/iFOR target ID. For server-based licensing, this will be the machine that is executing the license server (i4lmd) where you plan to install this application supplier's licenses. For nodelocked licensing, this will be the system where the application will be executing.

The algorithm that is used to identify a LicensePower/iFOR target ID may vary depending on operating system platform.

For example: On an HP-UX machine licenses managed by the i4lmd (concurrent and use once licenses), the LicensePower/iFOR target ID is derived from the link level address of the LAN card accessed by the device file /dev/i4target on the machine that is running the i4lmd. If /dev/i4target does not exist and the super-user is executing i4target, i4target will create /dev/i4target. On an HP 9000 Series 700 or 800, the device file will be for the lan0 LAN card. This is the same method used by the i4lmd for determining the LicensePower/iFOR ID of the machine on which it is executing.

On HP-UX, for LicensePower/iFOR nodelocked licenses, the LicensePower/iFOR ID is derived from:

- The LAN card accessed by /dev/i4target, or
- The built in SPU ID number, or
- An HIL ID Module.

A printable on-line administration guide is also available. (See the *FILES* section below.)

**Options**

- c -C Change the permanent target ID value.
- h -H Help. Display a list of options.
- o -O Display operating system name.
- q -Q Display target ID in quiet mode (without headers).
- v -V Display a verbose list of the LicensePower/iFOR target IDs from each possible source. The list consist of the link level address of the installed LAN cards. A super-user can then use the address to change to an alternate LAN card. This lets you change the IO slot where a LAN card is installed without losing the use of LicensePower/iFOR licenses locked to that LAN card.

**RETURN VALUE**

i4target always returns 0.

**DIAGNOSTICS**

Messages displayed during execution are self-explanatory.

**EXAMPLES**

To find the current local LicensePower/iFOR target ID(s):

i4target

Examples for each of the options are shown below:

i4target -c or i4target -C

Current Permanent Target ID: 3e53d0

## (TO BE OBSOLETE)

1. Target ID value: 3e53d0  
LAN card at logical unit 0

There is only one choice for the new Permanent Target ID.  
Enter '1' to select it; enter any other character to abort: 1

New Permanent Target ID: 3e53d0

NOTE: i4lmd must be restarted for the new  
Permanent Target ID to take effect.

i4target -h or i4target -H

Usage:

i4target [options]

options are:

- [vV] : verbose mode; detailed output
- [qQ] : quiet mode; no headers in output
- [cC] : change Permanent Target ID;
- [hH] : displays this message
- [oO] : displays os name

i4target -o or i4target -O

HP-UX

i4target -q or i4target -Q

3e53d0

i4target -v or i4target -V

Permanent Target ID: 3e53d0

SPU Target ID: 70328251

The Permanent Target ID is derived from a permanent hardware source on the system from which the i4target program is executed. This target ID may be used for all license types.

The SPU ID is derived from a hardware identification number on the SPU. It is used as the Permanent Target ID when no higher-priority sources for Permanent Target ID (i.e., LAN cards) are present.

**WARNINGS****LSSERV Software Obsolescence**

HP intends to remove LicensePower/iFOR from the core HP-UX product in a future release. The licensing product can be obtained directly from Isogon Corporation, the owner of the product.

For further information about LSSERV support, see the Isogon Corporation website at <http://www.isogon.com>.

Competitive information is also available at these websites:

FLEX/lm via <http://www.globetrotter.com>

SentinellM via <http://www.rainbow.com>

**FILES**

/opt/ifor/lm/bin/i4target

/opt/ifor/lm/doc/i4admin.pdf LicensePower/iFOR Administrator's Guide (PDF format)

**AUTHOR**

i4target was developed by HP and Gradient Technologies, Inc. It is currently a product owned by Isogon Corporation.

**SEE ALSO**

i4admin(1M), i4lmd(1M), i4start(1M), i4stop(1M), i4tv(1M).

**(TO BE OBSOLETE)**

*LicensePower/iFOR Administrator's Guide* available in `/opt/ifor/ls/doc/` in PDF format.

Isogon Corporation information on LicensePower/iFOR at:  
<http://www.isogon.com/products/ptlpoweri/ptlpoweri.htm>.

Complete HP-UX documentation at <http://docs.hp.com>.



## (TO BE OBSOLETE)

## NAME

i4tv - verify that LicensePower/iFOR License Servers are working

## SYNOPSIS

**i4tv** [**-n** *hostname* | **-z** | **-v**] [**-h** | **-usage** | **-version**]

## DESCRIPTION

The **i4tv** tool can be used after the license servers have been started to verify that they are running properly. The **i4tv** program resides in the `/opt/ibm/lis/bin` directory. A message describing a completed license transaction and a list of all license servers will be displayed. Once a license server has been configured using **i4config**, the **i4tv** tool is used to quickly verify the status of the license server **i4lmd**.

## Options

- n** *hostname* The **-n** option is used to check that the specified machine is running a license server. It returns 0 if the hostname is running **i4lmd** and it returns 1 if the hostname is not running **i4lmd**.
- z** The **-z** option turns on RPC tracing messages, which can be used to diagnose problems.
- v** Displays progress messages during the license request operation.
- h** Displays command usage information (same as **-usage**).
- usage** Displays command usage information (same as **-h**).
- version** Displays command version information.

If you can run **i4tv** successfully but are still having a problem with a licensed product, the problem is probably with the licenses, or possibly with the product itself: in this case, talk to the vendor of the licensed software product.

If you can not run **i4tv** successfully or it takes more than 10 seconds to retrieve a license, verify that **glbd** and **i4lmd** are running. Use the utility **lb\_admin** to clean the database. Answer YES to all database entries that do not respond. If you receive one of the error messages listed below, use the explanation of the error to fix the problem. Then try running **i4tv** again.

If you can not run **i4tv** successfully and receive an error that's not listed below, it means there is a problem with the software on which LicensePower/iFOR ARK is layered (for example, TCP), or a hardware problem.

A printable on-line administration guide is also available. (See the *FILES* section below.)

## ERROR MESSAGES

- netls\_no\_svrs\_found** No license servers are running or someone has deleted the LicensePower/iFOR Test Vendor from the license servers.
- netls\_license\_not\_found** Someone has deleted the Test Vendor licenses that each server automatically installs the first time that it starts. This prohibits anyone from using the test and verification tool (**i4tv**).
- netls\_not\_authorized** Someone has edited the user file to restrict the use of **i4tv**.
- netls\_bad\_timestamp** System clocks have not been synchronized to within 12 hours.

## EXAMPLES

Run the **i4tv** test and verification tool:

```
i4tv
```

```
i4TV Version 4.0 -- LicensePower/iFOR Test and Verification Tool
A product of Isogon Corporation
Completed license transaction on node 3541b8 running LicensePower/iFOR 4.0
Active LicensePower/iFOR Servers:
  hp_snake.gradient.com (HP-UX) running LicensePower/iFOR Version 3.0.0
```

Check for the presence of the license server hp1030:

**(TO BE OBSOLETE)**

`i4tv -n hp1030`

A product of Isogon Corporation  
hp1030 running

**WARNINGS****LSSERV Software Obsolescence**

HP intends to remove LicensePower/iFOR from the core HP-UX product in a future release. The licensing product can be obtained directly from Isogon Corporation, the owner of the product.

For further information about LSSERV support, see the Isogon Corporation website at <http://www.isogon.com>.

Competitive information is also available at these websites:

FLEX/Im via <http://www.globetrotter.com>

SentinellM via <http://www.rainbow.com>

**FILES**

`/opt/ifor/ls/bin/i4tv`

`/opt/ifor/ls/doc/i4admin.pdf` LicensePower/iFOR Administrator's Guide (PDF format)

**AUTHOR**

`i4tv` was developed by HP and Gradient Technologies, Inc. It is currently a product owned by Isogon Corporation.

**SEE ALSO**

`i4admin(1M)`, `i4lmd(1M)`, `i4start(1M)`, `i4stop(1M)`, `i4target(1M)`.

*LicensePower/iFOR Administrator's Guide* available in `/opt/ifor/ls/doc/` in PDF format.

Isogon Corporation information on LicensePower/iFOR at:  
<http://www.isogon.com/products/ptlpoweri/ptlpoweri.htm>.

Complete HP-UX documentation at <http://docs.hp.com>.

**NAME**

identd - TCP/IP IDENT protocol server

**SYNOPSIS**

```
/usr/sbin/identd [-i] [-w|-b] [-tseconds] [-uuid] [-ggid] [-pport] [-aaddress] [-charset]
[-n] [-o] [-e] [-l] [-V] [-m] [-N] [-d] [kernelfile [kmemfile]]
```

**DESCRIPTION**

**identd** is a server which implements the TCP/IP proposed standard IDENT user identification protocol as specified in the RFC 1413 document.

**identd** operates by looking up specific TCP/IP connections and returning the user name of the process owning the connection.

**Arguments**

- i** The **-i** flag, which is the default mode, should be used when starting the daemon from **inetd** with the "nowait" option in the **/etc/inetd.conf** file. Use of this mode will make **inetd** start one **identd** daemon for each connection request.
- w** The **-w** flag should be used when starting the daemon from **inetd** with the "wait" option in the **/etc/inetd.conf** file. This is the preferred mode of operation since that will start a copy of **identd** at the first connection request and then **identd** will handle subsequent requests without having to do the nlist lookup in the kernel file for every request as in the **-i** mode above. The **identd** daemon will run either forever, until a timeout, as specified by the **-t** flag, occurs.
- b** The **-b** flag can be used to make the daemon run in standalone mode without the assistance from **inetd**. This mode is the least preferred mode, and not supported by HP, since a bug or any other fatal condition in the server will make it terminate and it will then have to be restarted manually. Other than that it has the same advantage as the **-w** mode in that it parses the nlist only once.
- tseconds** The **-tseconds** option is used to specify the timeout limit. This is the number of seconds a server started with the **-w** flag will wait for new connections before terminating. The server is automatically restarted by **inetd** whenever a new connection is requested if it has terminated. A suitable value for this is 120 (2 minutes), if used. It defaults to no timeout (ie, will wait forever, or until a fatal condition occurs in the server).
- uuid** The **-uuid** option is used to specify a user id number which the **ident** server should switch to after binding itself to the TCP/IP port if using the **-b** mode of operation.
- ggid** The **-ggid** option is used to specify a group id number which the **ident** server should switch to after binding itself to the TCP/IP port if using the **-b** mode of operation.
- pport** The **-pport** option is used to specify an alternative port number to bind to if using the **-b** mode of operation. It can be specified by name or by number. Defaults to the IDENT port (113).
- aaddress** The **-aaddress** option is used to specify the local address to bind the socket to if using the **-b** mode of operation. Can only be specified by IP address and not by domain name. Defaults to the INADDR\_ANY address which normally means all local addresses.
- V** The **-V** flag makes **identd** display the version number and the exit.
- l** The **-l** flag tells **identd** to use the System logging daemon **syslogd** for logging purposes.
- o** The **-o** flag tells **identd** to not reveal the operating system type it is run on and to instead always return "OTHER".
- e** The **-e** flag tells **identd** to always return "UNKNOWN-ERROR" instead of the "NO-USER" or "INVALID-PORT" errors.
- charset** The **-charset** flags tells **identd** to add the optional (according to the IDENT protocol) character set designator to the reply generated. <charset> should be a valid character set as described in the MIME RFC in upper case characters.

- n** The **-n** flag tells **identd** to always return user numbers instead of user names if you wish to keep the user names a secret.
- N** The **-N** flag makes **identd** check for a file **.noident** in each homedirectory for a user which the daemon is about to return the user name for. If that file exists then the daemon will give the error **HIDDEN-USER** instead of the normal **USERID** response.
- m** The **-m** flag makes **identd** use a mode of operation that will allow multiple requests to be processed per session. Each request is specified one per line and the responses will be returned one per line. The connection will not be closed until the connecting part closes it's end of the line. *Please note that this mode violates the protocol specification as it currently stands.*
- d** The **-d** flag enables some debugging code that normally should *NOT* be enabled since that breaks the protocol and may reveal information that should not be available to outsiders.

*kernelfile* *kernelfile* defaults to the normally running kernel file.

*kmemfile* *kmemfile* defaults to the memory space of the normally running kernel.

## INSTALLATION

**identd** is invoked either by the internet server (see *inetd(1M)*) for requests to connect to the IDENT port as indicated by the **/etc/services** file (see *services(4)*) when using the **-w** or **-i** modes of operation or started manually by using the **-b** mode of operation.

## EXAMPLES

Since the server is located in **/usr/sbin/identd** one can put either:

```
ident stream tcp wait bin /usr/sbin/identd identd -w -t120
```

or:

```
ident stream tcp nowait bin /usr/sbin/identd identd -i
```

into the **/etc/inetd.conf** file.

To start it using the unsupported **-b** mode of operation one can put a line like this into the **/sbin/init.d/sendmail** file under the 'start' section:

```
/usr/sbin/identd -b -u2 -g2
```

This will cause **identd** to be started as daemon whenever **sendmail** is running. It will run in the background as user 2, group 2 (user 'bin', group 'bin').

## SEE ALSO

*inetd.conf(4)*.

**NAME**

ifconfig - configure network interface parameters

**SYNOPSIS**

**ifconfig** *interface address\_family* [*address* [*dest\_address*]] [*parameters*]

**ifconfig** *interface* [*address\_family*]

**DESCRIPTION**

The first form of the **ifconfig** command assigns an address to a network interface and/or configures network interface parameters. **ifconfig** must be used at boot time to define the network address of each interface present on a machine. It can also be used at other times to redefine an interface's address or other operating parameters.

The second form of the command, without *address\_family*, displays the current configuration for *interface*. If *address\_family* is also specified, **ifconfig** reports only the details specific to that address family.

Only a user with appropriate privileges can modify the configuration of a network interface. All users can run the second form of the command.

**Arguments**

**ifconfig** recognizes the following arguments:

|                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>address</i>             | Either a host name present in the host name database (see <i>hosts(4)</i> ), or a DARPA Internet address expressed in Internet standard dot notation (see <i>inet(3N)</i> ).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <i>address_family</i>      | Name of protocol on which naming scheme is based. An interface can receive transmissions in differing protocols, each of which may require separate naming schemes. Therefore, it is necessary to specify the <i>address_family</i> , which may affect interpretation of the remaining parameters on the command line. The only address family currently supported is <b>inet</b> (DARPA-Internet family).                                                                                                                                                                                                                                                                                                                                |
| <i>dest_address</i>        | Address of destination system. Consists of either a host name present in the host name database (see <i>hosts(4)</i> ), or a DARPA Internet address expressed in Internet standard dot notation (see <i>inet(3N)</i> ).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <i>interface</i>           | A string of the form <i>name unit</i> , such as <b>lan0</b> . (See the Interface Naming subsection given below.)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <i>parameters</i>          | One or more of the following operating parameters:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>up</b>                  | Mark an interface "up". Enables interface after an <b>ifconfig down</b> . Occurs automatically when setting the address on an interface. Setting this flag has no effect if the hardware is "down".                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>down</b>                | Mark an interface "down". When an interface is marked "down", the system will not attempt to transmit messages through that interface.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>broadcast</b>           | (Inet only) Specify the address that represents broadcasts to the network. The default broadcast address is the address with a host part of all 1's.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>metric</b> <i>n</i>     | Set the routing metric of the interface to <i>n</i> . The default is 0. The routing metric is used by the routing protocol (see <i>gated(1M)</i> ). Higher metrics have the effect of making a route less favorable; metrics are counted as additional hops to the destination network or host.                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>netmask</b> <i>mask</i> | (Inet only) Specify how much of the address to reserve for subdividing networks into sub-networks or aggregating networks into supernets. <i>mask</i> can be specified as a single hexadecimal number with a leading <b>0x</b> , with a dot-notation Internet address, or with a pseudo-network name listed in the network table (see <i>networks(4)</i> ). For subdividing networks into sub-networks, <i>mask</i> must include the network part of the local address, and the subnet part which is taken from the host field of the address. <i>mask</i> must contain 1's in the bit positions in the 32-bit address that are to be used for the network and subnet parts, and 0's in the host part. The 1's in the <i>mask</i> must be |

contiguous starting from the leftmost bit position in the 32-bit field. *mask* must contain at least the standard network portion, and the subnet field must be contiguous with the network portion. The subnet field must contain at least 1 bit. For aggregating networks into supernets, *mask* must only include a portion of the network part. *mask* must contain contiguous 1's in the bit positions starting from the leftmost bit of the 32-bit field.

|                |                                                                                                                                                                                                                                                                                                         |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>arp</b>     | Enable the user of the Address Resolution Protocol in mapping between network level addresses and link level addresses (default). If an interface already had the Address Resolution Protocol disabled, the user must "unplumb" the interface before it can be enabled for Address Resolution Protocol. |
| <b>-arp</b>    | Disable the use of the Address Resolution Protocol. If an interface already had the Address Resolution Protocol enabled, the user must "unplumb" the interface before it can be disabled for Address Resolution Protocol.                                                                               |
| <b>plumb</b>   | Setup the Streams plumbing needed for TCP/IP for a primary interface name. (See the Interface Naming subsection given below.). By default, the <b>plumb</b> operation is done automatically when an IP address is specified for an interface.                                                           |
| <b>unplumb</b> | Tear down the Streams plumbing for a primary interface name. (See the Interface Naming subsection given below.) Secondary interface does not require "plumbing" and it can be removed by assigning an IP address of 0.0.0.0.                                                                            |

### Interface Naming

The *interface* name associated with a network card is composed of the *name* of the interface (e.g. **lan** or **snap**), the *ppa number* which identifies the card instance for this interface, and an optional *IP index number* which allows the configuration of multiple IP addresses for an interface. For LAN cards, the *interface* name **lan** will be used to designate Ethernet encapsulation and **snap** for IEEE 802.3 encapsulation. The **lanscan** command can be used to display the *interface* name and *ppa number* of each interface that is associated with a network card (see **lanscan(1M)**).

Multiple IP addresses assigned to the same *interface* may be in different subnets. An example of an interface name without an *IP index number* is **lan0**. An example of an interface name with a *IP index number* is **lan0:1**. Note: specifying **lan0:0** is equivalent to **lan0**.

### Loopback Interface

The loopback interface (**lo0**) is automatically configured when the system boots with the TCP/IP software. The default IP address and netmask of the loopback interface are 127.0.0.1 and 255.0.0.0, respectively. The user is not permitted to change the address of the primary loopback interface (**lo0:0**). It is permissible to assign other IP addresses to lo0 with non-zero *IP index numbers* (lo0:1, lo0:2, etc). This allows a system to have a "system IP" address that is available as long as one interface remains usable.

### Supernets

A supernet is a collection of smaller networks. Supernetting is a technique of using the netmask to aggregate a collection of smaller networks into a supernet.

This technique is particularly useful when the limit of 254 hosts per class C network is too restrictive. In those situations a netmask containing only a portion of the network part may be applied to the hosts in these networks to form a supernet. This supernet netmask should be applied to those interfaces that connect to the supernet using the *ifconfig* command. For example, a host can configure its interface to connect to a class C supernet, 192.6, by configuring an IP address of 192.6.1.1 and a netmask of 255.255.0.0 to its interface.

### DIAGNOSTICS

Messages indicate if the specified interface does not exist, the requested address is unknown, or the user is not privileged and tried to alter an interface's configuration.

**AUTHOR**

**ifconfig** was developed by HP and the University of California, Berkeley.

**SEE ALSO**

netstat(1), lanscan(1M), hosts(4), routing(7).

**NAME**

inetd - Internet services daemon

**SYNOPSIS**

```
/usr/sbin/inetd [-c]
/usr/sbin/inetd [-k]
/usr/sbin/inetd [-l]
```

**DESCRIPTION**

The **inetd** daemon is the Internet superserver, which invokes Internet server processes as needed. It must be running before other hosts can connect to the local host through **ftp**, **rcp**, **remsh**, **rlogin**, and **telnet**. The **inetd** daemon also supports services based on the Remote Procedure Call (RPC) protocol (NFS), such as **rwalld** and **rusersd**. If RPC servers are started by **inetd**, the **portmap** server (see *portmap(1M)*) must be started before **inetd**.

The **inetd** daemon is designed to invoke all the Internet servers as needed, thus reducing load on the system. It is normally started at system boot time. Only one **inetd** can run at any given time.

The **inetd** daemon starts servers for both stream and datagram type services. For stream services, **inetd** listens for connection requests on Internet stream sockets. When a connection is requested for one of its sockets, **inetd** decides which service the socket will support, forks a process, invokes an appropriate server for the connection, and passes the connected socket to the server as **stdin** and **stdout**. Then **inetd** returns to listening for connection requests.

For datagram services, **inetd** waits for activity on Internet datagram sockets. When an incoming datagram is detected, **inetd** forks a process, invokes an appropriate server, and passes the socket to the server as **stdin** and **stdout**. Then **inetd** waits, ignoring activity on that datagram socket, until the server exits.

The **inetd** daemon is normally started by the `/sbin/init.d/inetd` script, which is invoked during the boot-time initialization. Otherwise, **inetd** can be started only by the superuser.

The Internet daemon and the servers it starts inherit the **LANG** and **TZ** environment variables and the **umask** of the process that started **inetd**. If **inetd** is started by the superuser, it inherits the superuser's **umask**, and passes that **umask** to the servers it starts.

When invoked, **inetd** reads `/etc/inetd.conf` and configures itself to support whatever services are included in that file (see *inetd.conf(4)*). The **inetd** daemon also performs a security check if the file `/var/adm/inetd.sec` exists (see *inetd.sec(4)*). If the Internet daemon refuses a connection for security reasons, the connection is shut down. Most RPC-based services, if their first connection is refused, attempt to connect four more times at 5-second intervals before timing out. In such cases, **inetd** refuses the connection from the same service invocation five times. This is visible in the system log if **inetd** connection logging and **syslogd** logging for the **daemon** facility are both enabled (see *syslogd(1M)*).

The **inetd** daemon provides several "trivial" services internally by use of routines within itself. The services are **echo**, **discard**, **chargen** (character generator), **daytime** (human readable time), and **time** (machine readable time in the form of the number of seconds since midnight, January 1, 1900). The **inetd** daemon provides both TCP- and UDP-based servers for each of these services. See *inetd.conf(4)* for instructions on configuring internal servers.

**Options**

**inetd** recognizes the following options. These options can be used only by a superuser.

- c Reconfigure the Internet daemon; in other words, force the current **inetd** to reread `/etc/inetd.conf`. This option sends the signal **SIGHUP** to the Internet daemon that is currently running. Any configuration errors that occur during the reconfiguration are logged to the **syslogd** daemon facility.
- k Kill the current **inetd**. This option sends the signal **SIGTERM** to the Internet daemon that is currently running, causing it to exit gracefully. This option is the preferred method of killing **inetd**.
- l By default, **inetd** starts with connection logging disabled. If no **inetd** is running, the **-l** option causes the **inetd** to start with connection logging enabled. Otherwise the **-l** option causes **inetd** to send the signal **SIGQUIT** to the **inetd** that is already running, which causes it to toggle the state of connection logging.



When connection logging is enabled, the Internet daemon logs attempted connections to services. It also logs connection attempts which fail the security check. This information can be useful when trying to determine if someone is repeatedly trying to access your system from a particular remote system (in other words, trying to break into your system). Successful connection attempts are logged to the **syslogd** daemon facility at the **info** log level. Connection attempts failing the security check are logged at the **notice** log level. **inetd** also logs whether the connection logging has been enabled or disabled at the *info* log level.

## DIAGNOSTICS

The following diagnostics are returned by the Internet daemon before it disconnects from the terminal.

### An inetd is already running

An attempt was made to start an Internet daemon when one was already running. It is incorrect to call the Internet daemon a second time without the **-c**, **-k**, or **-l** option.

### There is no inetd running

An attempt was made to reconfigure an Internet daemon when none was running.

### Inetd not found

This message occurs if **inetd** is called with **-c** and another Internet daemon is running but cannot be reconfigured. This occurs if the original Internet daemon died without removing its semaphore.

*Next step:* Use the **inetd -k** command to remove the semaphore left by the previous Internet daemon; then restart the daemon.

The following diagnostics are logged to the **syslogd** daemon facility. Unless otherwise indicated, messages are logged at the **error** log level.

### /etc/inetd.conf: Unusable configuration file

The Internet daemon is unable to access the configuration file **/etc/inetd.conf**. The error message preceding this one specifies the reason for the failure.

### /etc/inetd.conf: line number: error

There is an error on the specified line in **/etc/inetd.conf**. The line in the configuration file is skipped. This error does not stop the Internet daemon from reading the rest of the file and configuring itself accordingly.

*Next step:* Fix the line with the error and reconfigure the Internet daemon by executing the **inetd -c** command.

### system\_call: message

*system\_call* failed. See the corresponding manual entry for a description of *system\_call*. The reason for the failure is explained in *message*.

### Cannot configure inetd

None of the services/servers listed in the configuration file could be set up properly, due to configuration file errors.

### Too many services (max n)

The number of active services listed in the configuration file exceeds the "hard" limit that can be supported by the system (see *setrlimit(2)*).

*Next step:* Reduce the number of services listed in the configuration file, then reconfigure the Internet daemon by running the command **inetd -c**.

### file: \ found before end of line line

*file* can be either **inetd.conf** or **inetd.sec**. If a backslash is not immediately followed by an end of line, it is ignored and the information up to the end of line is accepted. In this case, the next line of the file is not appended to the end of the current line. Unless all the information required is present on a single line, configuration file error messages are also output. This message is logged at the **warning** log level.

### service/protocol: Unknown service

The call to the library routine `getservbyname` (see `getservent(3N)`) failed. The service is not listed in `/etc/services`.

*Next step:* Include that service in `/etc/services` or eliminate the entry for the service in `/etc/inetd.conf`.

*service/protocol:* **Server failing (looping), service terminated.**

When `inetd` tries to start 40 servers within 60 seconds for a datagram service, other than `bootp`, `rpc`, or `tftp`, it assumes that the server is failing to handle the connection. To avoid entering a potentially infinite loop, `inetd` issues this message, discards the packet requesting the socket connection, and refuses further connections for this service. After 10 minutes, `inetd` tries to reinstate the service, and once again accepts connections for the service.

*service/protocol:* **socket:** *message*

*service/protocol:* **listen:** *message*

*service/protocol:* **getsockname:** *message*

Any one of the three errors above makes the service unusable. For another host to communicate with the server host through this service, the Internet daemon needs to be reconfigured after any of these error messages.

*service/protocol:* **bind:** *message*

If this error occurs, the service is temporarily unusable. After 10 minutes, `inetd` tries again to make the service usable by binding to the Internet socket for the service.

*service/protocol:* **Access denied to remote\_host (address)**

The remote host failed to pass the security test for the indicated service. This information can be useful when trying to determine if someone is repeatedly trying to access your system from a particular remote system (in other words, trying to break into your system). This message is logged at the **warning** log level.

*service/protocol:* **Connection from remote\_host (address)**

When connection logging is enabled, this message indicates a successful connection attempt to the specified service. This message is logged at the **notice** log level.

*service/protocol:* **Added service, server executable**

Keeps track of the services added when reconfiguring the Internet daemon. This message is logged at the **info** log level.

*service/protocol:* **New list**

Lists the new user IDs, servers or executables used for the service when reconfiguring the Internet daemon. This message is logged at the **info** log level.

*service/protocol:* **Deleted service**

Keeps track of the services deleted when reconfiguring the Internet daemon. This message is logged at the **info** log level.

### Security File (inetd.sec) Errors

The following errors, prefixed by `/var/adm/inetd.sec:`, are related to the security file `inetd.sec`:

**Field contains other characters in addition to \* for service**

For example, field 2 of the Internet address `10.5*.8.7` is incorrect.

**Missing low value in range for service**

For example, field 2 of the Internet address `10.-5.8.7` is incorrect.

**Missing high value in range for service**

For example, field 2 of the Internet address `10.5-.8.7` is incorrect.

**High value in range is lower than low value for service**

For example, field 2 of the Internet address `10.5-3.8.7` is incorrect.

**allow/deny field does not have a valid entry for service**

The entry in the allow/deny field is not one of the keywords **allow** or **deny**. No security for this service is implemented by **inetd** since the line in the security file is ignored. This message is logged at the **warning** log level.

#### RPC Related Errors for NFS Users

These errors are specific to RPC-based servers:

```
/etc/inetd.conf: line number: Missing program number
/etc/inetd.conf: line number: Missing version number
```

Error on the specified line of **/etc/inetd.conf**. The program or version number for an RPC service is missing. This error does not stop the Internet daemon from reading the rest of the file and configuring itself accordingly. However, the service corresponding to the error message will not be configured correctly.

*Next step:* Fix the line with the error, then reconfigure the Internet daemon by executing the **inetd -c** command.

```
/etc/inetd.conf: line number: Invalid program number
```

Error on the specified line of **/etc/inetd.conf**. The program number for an RPC service is not a number. This error does not stop the Internet daemon from reading the rest of the file and configuring itself accordingly. However, the service corresponding to the error message will not be correctly configured.

*Next step:* Fix the line with the error, then reconfigure the Internet daemon by executing the **inetd -c** command.

#### AUTHOR

**inetd** was developed by HP and the University of California, Berkeley.

NFS was developed by Sun Microsystems, Inc.

#### FILES

|                           |                                    |
|---------------------------|------------------------------------|
| <b>/etc/inetd.conf</b>    | List of Internet server processes. |
| <b>/var/adm/inetd.sec</b> | Optional security file.            |

#### SEE ALSO

umask(1), portmap(1M), syslogd(1M), getservent(3N), inetd.conf(4), inetd.sec(4), protocols(4), services(4), environ(5).

**NAME**

inetsvcs\_sec - enable/disable secure internet services

**SYNOPSIS**

**inetsvcs\_sec** [**enable** | **disable** | **status**]

**DESCRIPTION**

/usr/sbin/inetsvcs\_sec is used to enable or disable secure internet services (SIS) by updating inetsvcs.conf(4) with the appropriate entry. SIS provide network authentication when used in conjunction with HP DCE security services, the HP Praesidium/Security Server, or other software products that provide a Kerberos V5 Network Authentication Services environment.

**Options**

**inetsvcs\_sec** recognizes the following options:

- |                |                                                                                                                            |
|----------------|----------------------------------------------------------------------------------------------------------------------------|
| <b>enable</b>  | The secure internet services are enabled. The services now provide network authentication through Kerberos V5.             |
| <b>disable</b> | The secure internet services are disabled. The services now follow the traditional behavior of prompting for passwords.    |
| <b>status</b>  | This option displays the current authentication mechanism used ( i.e., whether Kerberos authentication is enabled or not). |

**SEE ALSO**

sis(5), inetsvcs.conf(4).

**NAME**

infocmp - compare or print out terminfo descriptions

**SYNOPSIS**

```
infocmp [-d] [-c] [-n] [-I] [-L] [-C] [-r] [-u] [-s d|i|l|c] [-v] [-V] [-l]
[-w width] [-A directory] [-B directory] [termname...]
```

**DESCRIPTION**

**infocmp** can be used to compare a binary **terminfo** entry with other terminfo entries, rewrite a **terminfo** description to take advantage of the **use=** terminfo field, or print out a **terminfo** description from the binary file (**term**) in a variety of formats. In all cases, the boolean fields will be printed first, followed by the numeric fields, followed by the string fields.

**Default Options**

If no options are specified and zero or one *termnames* are specified, the **-I** option will be assumed. If more than one *termname* is specified, the **-d** option will be assumed.

**Comparison Options [-d] [-c] [-n]**

**infocmp** compares the **terminfo** description of the first terminal *termname* with each of the descriptions given by the entries for the other terminal's *termnames*. If a capability is defined for only one of the terminals, the value returned will depend on the type of the capability: **F** for boolean variables, **-1** for integer variables, and **NULL** for string variables.

- d** produces a list of each capability that is different between two entries. This option is useful to show the difference between two entries, created by different people, for the same or similar terminals.
- c** produces a list of each capability that is common between two entries. Capabilities that are not set are ignored. This option can be used as a quick check to see if the **-u** option is worth using.
- n** produces a list of each capability that is in neither entry. If no *termnames* are given, the environment variable **TERM** will be used for both of the *termnames*. This can be used as a quick check to see if anything was left out of a description.

**Source Listing Options [-I] [-L] [-C] [-r]**

The **-I**, **-L**, and **-C** options will produce a source listing for each terminal named.

- I** use the **terminf** names
- L** use the long C variable name listed in **<term.h>**
- C** use the **termcap** names
- r** when using **-C**, put out all capabilities in **termcap** form

If no *termnames* are given, the environment variable **TERM** will be used for the terminal name.

The source produced by the **-C** option may be used directly as a **termcap** entry, but not all of the parameterized strings may be changed to the **termcap** format. **infocmp** will attempt to convert most of the parameterized information, but anything not converted will be plainly marked in the output and commented out. These should be edited by hand.

All padding information for strings will be collected together and placed at the beginning of the string where **termcap** expects it. Mandatory padding (padding information with a trailing **'**) will become optional.

All **termcap** variables no longer supported by **terminfo**, but which are derivable from other **terminfo** variables, will be output. Not all **terminfo** capabilities will be translated; only those variables which were part of **termcap** will normally be output. Specifying the **-r** option will take off this restriction, allowing all capabilities to be output in **termcap** form.

Note that because padding is collected to the beginning of the capability, not all capabilities are output. Mandatory padding is not supported. Because **termcap** strings are not as flexible, it is not always possible to convert a **terminfo** string capability into an equivalent **termcap** format. A subsequent conversion of the **termcap** file back into **terminfo** format will not necessarily reproduce the original **terminfo** source.

Some common **terminfo** parameter sequences, their **termcap** equivalents, and some terminal types which commonly have such sequences, are:

| <b>terminfo</b>           | <b>termcap</b> | Representative Terminals |
|---------------------------|----------------|--------------------------|
| %p1%c                     | %.             | adm                      |
| %p1%d                     | %d             | hp, ANSI standard, vt100 |
| %p1%'x'%;%c               | %+x            | concept                  |
| %i                        | %i             | ANSI standard, vt100     |
| %p1%?'x'%;%t%p1%'y'%;     | %>xy           | concept                  |
| %p2 is printed before %p1 | %r             | hp                       |

**Use= Option [-u]**

- u** produces a **terminfo** source description of the first terminal *termname* which is relative to the sum of the descriptions given by the entries for the other terminals *termnames*. It does this by analyzing the differences between the first *termname* and the other *termnames* and producing a description with **use=** fields for the other terminals. In this manner, it is possible to retrofit generic terminfo entries into a terminal's description. Or, if two similar terminals exist, but were coded at different times or by different people so that each description is a full description, using **infocmp** will show what can be done to change one description to be relative to the other.

A capability will get printed with an at-sign (@) if it no longer exists in the first *termname*, but one of the other *termname* entries contains a value for it. A capability's value gets printed if the value in the first *termname* is not found in any of the other *termname* entries, or if the first of the other *termname* entries that has this capability gives a different value for the capability than that in the first *termname*.

The order of the other *termname* entries is significant. Since the terminfo compiler **tic** does a left-to-right scan of the capabilities, specifying two **use=** entries that contain differing entries for the same capabilities will produce different results depending on the order that the entries are given in. **infocmp** will flag any such inconsistencies between the other *termname* entries as they are found.

Alternatively, specifying a capability after a **use=** entry that contains that capability will cause the second specification to be ignored. Using **infocmp** to recreate a description can be a useful check to make sure that everything was specified correctly in the original source description.

Another error that does not cause incorrect compiled files, but will slow down the compilation time, is specifying extra **use=** fields that are superfluous. **infocmp** will flag any other *termname* **use=** fields that were not needed.

**Other Options [-s d|i|l|c] [-v] [-V] [-1] [-w width]**

- s** sorts the fields within each type according to the argument below:

- d** leave fields in the order that they are stored in the **terminfo** database.
- i** sort by **terminfo** name.
- l** sort by the long C variable name.
- c** sort by the **termcap** name.

If the **-s** option is not given, the fields printed out will be sorted alphabetically by the **terminfo** name within each type, except in the case of the **-C** or the **-L** options, which cause the sorting to be done by the **termcap** name or the long C variable name, respectively.

- v** prints out tracing information on standard error as the program runs.
- V** prints out the version of the program in use on standard error and exit.
- 1** causes the fields to be printed out one to a line. Otherwise, the fields will be printed several to a line to a maximum width of 60 characters.
- w** changes the output to *width* characters.

**Changing Databases [-A directory] [-B directory]**

The location of the compiled **terminfo** database is taken from the environment variable **TERMINFO**. If the variable is not defined, or the terminal is not found in that location, the system **terminfo** database, usually in **/usr/lib/terminfo**, will be used. The options **-A** and **-B** may be used to override this location. The **-A** option will set **TERMINFO** for the first *termname* and the **-B** option will set **TERMINFO** for the other *termnames*. With this, it is possible to compare descriptions for a terminal with the same name located in two different databases. This is useful for comparing descriptions for the same terminal created by different people.

**FILES**

`/usr/lib/terminfo/?/*`      Compiled terminal description database.

**SEE ALSO**

`curses_intro(3X)`, `captoinfo(1M)`, `terminfo(4)`, `tic(1M)`.

**NAME**

init - process control initialization

**SYNOPSIS**

```
/sbin/init [0|1|2|3|4|5|6|S|s|Q|q|a|b|c]
```

**DESCRIPTION**

The **init** daemon and command is a general process spawner. Its primary role is to create processes from a script stored in the file `/etc/inittab` (see *inittab(4)*). This file usually has **init** spawn a **getty** on each line where users can log in. It also controls autonomous processes required by any particular system.

At boot time, **init** is started as a system daemon.

While the system is running, a user-spawned **init** directs the actions of the boot **init**. It accepts a one-character argument and signals the boot **init** with the `kill()` system call to perform the appropriate action.

The arguments have the following effect:

- 0-6 Place the system in one of the run levels 0 through 6.
- a|b|c Process the **inittab** entries that have the special "run level" **a**, **b**, or **c**, without changing the numeric run level.
- Q|q Re-examine the **inittab** entries without changing the run level.
- S|s Enter the single-user environment. When this level change occurs, the logical system console `/dev/syscon` is changed to the terminal from which the command was executed.

Boot **init** considers the system to be in a **run level** at any given time. A run level can be viewed as a software configuration of the system, where each configuration allows only a selected group of processes to exist. The processes spawned by boot **init** for each of these run levels are defined in the **inittab** file. Boot **init** can be in one of eight run levels, 0-6, and **S** or **s**. The run level is changed by having a privileged user run the **init** command. This user-spawned **init** sends appropriate signals to the boot **init**.

Boot **init** is invoked inside the HP-UX system as the last step in the boot procedure. Boot **init** first performs any required machine-dependent initialization, such as setting the system context. Next, boot **init** looks for the **inittab** file to see if there is an entry of the type **initdefault** (see *inittab(4)*). If an **initdefault** entry is found, boot **init** uses the run level specified in that entry as the initial run level to enter. If this entry is not in **inittab**, or **inittab** is not found, boot **init** requests that the user enter a run level from the logical system console, `/dev/syscon`. If **S** or **s** is entered, boot **init** goes into the **single-user** level. This is the only run level that does not require the existence of a properly formatted **inittab** file. If **inittab** does not exist, then by default the only legal run level that boot **init** can enter is the single-user level.

In the single-user level, the logical system console terminal `/dev/syscon` is opened for reading and writing, and the command `/usr/bin/su`, `/usr/bin/sh`, or `/sbin/sh` is invoked immediately. To exit from the single-user run level, one of two options can be selected:

- If the shell is terminated with an end-of-file, boot **init** reprompts for a new run level.
- User **init** can signal boot **init** and force it to change the current system run level.

When attempting to boot the system, some processes spawned by boot **init** may send display messages to the system console (depending on the contents of **inittab**). If messages are expected but do not appear during booting, it may be caused by the logical system console (`/dev/syscon`) being linked to a device that is not the physical system console (`/dev/systty`). If this occurs, you can force boot **init** to relink `/dev/syscon` to `/dev/systty` by pressing the DEL (delete) key (ASCII 127) on the physical system console.

When boot **init** prompts for the new run level, you can only enter one of the digits 0 through 6 or the letter **S** or **s**. If you enter **S**, boot **init** operates as previously described in single-user mode with the additional result that `/dev/syscon` is linked to the user's terminal line, thus making it the logical system console. A message is generated on the physical system console, `/dev/systty`, identifying the new logical system console.

When boot **init** comes up initially, and whenever it switches out of single-user state to normal run states, it sets the states (see *ioctl(2)*) of the logical system console, `/dev/syscon`, to those modes saved in the file `/etc/ioctl.syscon`. This file is written by boot **init** whenever single-user mode is entered. If



this file does not exist when boot **init** wants to read it, a warning is printed and default settings are assumed.

If 0 through 6 is entered, boot **init** enters the corresponding run level. Any other input is rejected and a new prompt is issued. If this is the first time boot **init** has entered a run level other than single-user, boot **init** first scans **inittab** for special entries of the type **boot** and **bootwait**. These entries are performed — provided that the run level entered matches that of the entry — before any normal processing of **inittab** takes place. In this way, any special initialization of the operating system, such as mounting file systems, can take place before users are allowed onto the system. The **inittab** file is scanned to find all entries that are to be processed for that run level.

Run levels in HP-UX are defined as follows:

- 0        Shut down HP-UX.
- S | s    Use for system administration (also known as "single-user state"). When booting into run level **S** at powerup, the only access to the system is through a shell spawned at the system console as the root user. The only processes running on the system will be kernel daemons started directly by the HP-UX kernel, daemon processes started from entries of type **sysinit** in **/etc/inittab**, the shell on the system console, and any processes started by the system administrator. Administration operations that require the system to be in a quiescent state (such as the **fsck(1M)** operation to repair a file system) should be run in this state. Transitioning into run level **S** from a higher run level does not terminate other system activity and does not result in a "single-user state"; this operation should not be done.
- 1        Start a subset of essential system processes. This state can also be used to perform system administration tasks.
- 2        Start most system daemons and login processes. This state is often called the "multi-user state". Login processes either at local terminals or over the network are possible.
- 3        Export filesystems and start other system processes. In this state NFS filesystems are often exported, as may be required for an NFS server.
- 4        Activate graphical presentation managers and start other system processes.
- 5–6      These states are available for user-defined operations.

The default run level is usually run level 3 or 4, depending on the system configuration.

When **init** transitions into a new run level 0–6, the master sequencer script **rc** is invoked. **rc** in turn invokes each of the start or kill scripts for each installed subsystem for each intervening run level. When transitioning to a higher run level start scripts are invoked, and when transitioning to a lower run level kill scripts are invoked. See **rc(1M)**.

In a multiuser environment, the **inittab** file is usually set up so that boot **init** creates a process for each terminal on the system.

For terminal processes, ultimately the shell terminates because of an end-of-file either typed explicitly or generated as the result of hanging up. When boot **init** receives a child death signal telling it that a process it spawned has died, it records the fact and the reason it died in **/etc/utmp** and **/var/adm/wtmp**, if they exist (see **who(1)**). A history of the processes spawned is kept in **/var/adm/wtmp**, if it exists.

To spawn each process in the **inittab** file, boot **init** reads each entry and, for each entry that should be respawned, it forks a child process. After it has spawned all of the processes specified by the **inittab** file, boot **init** waits for one of its descendant processes to die, a powerfail signal, or until it is signaled by a user **init** to change the system's run level. When one of the above three conditions occurs, boot **init** re-examines the **inittab** file. New entries can be added to the **inittab** file at any time. However, boot **init** still waits for one of the above three conditions to occur. For an instantaneous response, use the **init Q** (or **init q**) command to wake up boot **init** to re-examine the **inittab** file without changing the run level.

If boot **init** receives a powerfail signal (**SIGPWR**) and is not in single-user mode, it scans **inittab** for special **powerfail** entries. These entries are invoked (if the run levels permit) before any other processing takes place by boot **init**. In this way, boot **init** can perform various cleanup and recording functions whenever the operating system experiences a power failure. Note, however, that although boot **init** receives **SIGPWR** immediately after a power failure, boot **init** cannot handle the signal until it resumes execution. Since execution order is based on scheduling priority, any eligible process with a higher priority executes before boot **init** can scan **inittab** and perform the specified functions.

When boot **init** is requested to change run levels via a user **init**, it sends the warning signal **SIGTERM** to all processes that are undefined in the target run level. Boot **init** waits 20 seconds before forcibly terminating these processes with the kill signal **SIGKILL**. Note that boot **init** assumes that all these processes (and their descendants) remain in the same process group that boot **init** originally created for them. If any process changes its process group affiliation with either **setpgrp()** or **setpgrp2()** (see **setsid(2)** and **setpgid(2)**), it will not receive these signals. (Common examples of such processes are the shells **cs**h and **ks**h (see **cs**h(1) and **ks**h(1).) Such processes need to be terminated separately.

A user **init** can be invoked only by users with appropriate privileges.

## DIAGNOSTICS

If boot **init** finds that it is continuously respawning an entry from **inittab** more than 10 times in 2 minutes, it will assume that there is an error in the command string, generate an error message on the system console, and refuse to respawn this entry until either 5 minutes have elapsed or it receives a signal from a user **init**. This prevents boot **init** from using up system resources if there is a typographical error in the **inittab** file or a program is removed that is referenced in **inittab**.

## WARNINGS

Boot **init** assumes that processes and descendants of processes spawned by boot **init** remain in the same process group that boot **init** originally created for them. When changing init states, special care should be taken with processes that change their process group affiliation, such as **cs**h and **ks**h.

One particular scenario that often causes confusing behavior can occur when a child **cs**h or **ks**h is started by a login shell. When boot **init** is asked to change to a run level that would cause the original login shell to be killed, the shell's descendant **cs**h or **ks**h process does not receive a hangup signal since it has changed its process group affiliation and is no longer affiliated with the process group of the original shell. Boot **init** cannot kill this **cs**h or **ks**h process (or any of its children).

If a **getty** process is later started on the same tty as this previous shell, the result may be two processes (the **getty** and the job control shell) competing for input on the tty.

To avoid problems such as this, always be sure to manually kill any job control shells that should not be running after changing init states. Also, always be sure that user **init** is invoked from the lowest level (login) shell when changing to an init state that may cause your login shell to be killed.

## FILES

```
/dev/syscon
/dev/systty
/etc/default/security
/etc/inittab
/etc/ioctl.syscon
/etc/utmp
/var/adm/wtmp
```

## SEE ALSO

**cs**h(1), **ks**h(1), **login**(1), **sh**(1), **who**(1), **getty**(1M), **rc**(1M), **ioctl**(2), **kill**(2), **setpgid**(2), **setsid**(2), **inittab**(4), **security**(4), **utmp**(4).

## STANDARDS CONFORMANCE

**init**: SVID2, SVID3

**NAME**

insf - install special (device) files

**SYNOPSIS**

**/sbin/insf**

**/sbin/insf** [-C *class* | -d *driver*] [-D *directory*] [-e] [-H *hw-path*] [-I *instance*]  
[-n *npty*] [-q | -v] [-s *nstrpty*] [-p *first-optical-disk:last-optical-disk*]

**DESCRIPTION**

The **insf** command installs special files in the devices directory, normally **/dev**. If required, **insf** creates any subdirectories that are defined for the resulting special file.

If no options are specified, special files are created for all new devices in the system. New devices are those devices for which no special files have been previously created. A subset of the new devices can be selected with the **-C**, **-d**, and **-H** options.

With the **-e** option, **insf** reinstalls the special files for pseudo-drivers and existing devices. This is useful for restoring special files when one or more have been removed.

Normally, **insf** displays a message as the special files are installed for each driver. The **-q** (quiet) option suppresses the installation message. The **-v** (verbose) option displays the installation message and the name of each special file as it is created.

**Options**

**insf** recognizes the following options.

- C *class*** Match devices that belong to a given device class, *class*. Device classes can be listed with the **lsdev** command (see **lsdev(1M)**). They are defined in the files in the directory **/usr/conf/master.d**. The special class **pseudo** includes all pseudo-drivers. This option cannot be used with **-d**.
- d *driver*** Match devices that are controlled by the specified device driver, *driver*. Device drivers can be listed with the **lsdev** command (see **lsdev(1M)**). They are defined in the files in the directory **/usr/conf/master.d**. This option cannot be used with **-C**.
- D *directory*** Override the default device installation directory **/dev** and install the special files in *directory* instead. *directory* must exist; otherwise, **insf** displays an error message and exits. See WARNINGS.
- e** Reinstall the special files for pseudo-drivers and existing devices. This is useful for restoring special files if one or more have been removed.
- H *hw-path*** Match devices at a given hardware path, *hw-path*. Hardware paths can be listed with the **ioscan** command (see **ioscan(1M)**). A hardware path specifies the addresses of the hardware components leading to a device. It consists of a string of numbers separated by periods (.), such as **52** (a card), **52.3** (a target address), and **52.3.0** (a device). If a hardware component is a bus converter, the following period, if any, is replaced by a slash (/) as in **2**, **2/3**, and **2/3.0**.  
  
If the specified path contains fewer numbers than are necessary to reach a device, special files are made for all devices at addresses that extend the given path. If the specified path is **56**, then special files are made for the devices at addresses **56.0**, **56.1**, **56.2**, etc.
- I *instance*** Match a device with the specified *instance* number. Instances can be listed with the **-f** option of the **ioscan** command (see **ioscan(1M)**).  
  
This option is effective only if the **-e** option is specified or if an appropriate device class or driver is specified with a **-C** or **-d** option.
- n *npty*** Install *npty* special files for each specified **ptym** and **ptys** driver. The **pty** driver specifies both the **ptym** and **ptys** drivers. *npty* is a decimal number.  
  
This option is effective only if the **-e** option is specified or if an appropriate device class or driver is specified with a **-C** or **-d** option.  
  
If this option is omitted, *npty* defaults to 60 for the **ptym** and **ptys** drivers.

**-p** *first-optical-disk*: *last-optical-disk*

Install the special files for those optical disks located in slots in the range *first-optical-disk* to *last-optical-disk*. The two variables can have values from the set **1a**, **1b**, ..., **32a**, **32b**. This option only applies to the **autox0** and **schgr** drivers. If it is omitted, the 64 special files for both sides of 32 optical disks (**1a** through **32b**) will be installed.

**-q** Quiet option. Normally, **insf** displays a message as each driver is processed. This option suppresses the driver message, but not error messages. See the **-v** option.

**-s** *nstrpty* Install *nstrpty* slave-side stream special files for the **pts** driver. *nstrpty* is a decimal number. This option only applies to the **pts** special file installation.

This option is effective only if the **-e** option is specified or if an appropriate device class or driver is specified with a **-C** or **-d** option.

If this option is omitted, *nstrpty* defaults to 60.

**-v** Verbose option. In addition to the normal processing message, display the name of each special file as it is created. See the **-q** option.

### Naming Conventions

Many special files are named using the **c***card***t***target***d***device* naming convention. These variables have the following meaning wherever they are used.

*card* The unique interface card identification number from **ioscan** (see **ioscan(1M)**). It is represented as a decimal number with a typical range of 0 to 255.

*target* The device target number, for example the address on a HP-FL or SCSI bus. It is represented as a decimal number with a typical range of 0 to 15.

*device* A address unit within a device, for example, the unit in a HP-FL device or the LUN in a SCSI device. It is represented as a decimal number with a typical range of 0 to 15.

### Special Files

This subsection shows which special files are created and the permissions for each device driver.

The special file names are relative to the installation directory, normally **/dev**. This directory may be overridden with the **-D** option.

**insf** sets the file permissions and the owner and group IDs. They are shown here in a format similar to that of the **ll** command:

```
special-file           permissions owner group
```

For example:

```
tty                   rw-rw-rw- bin bin
```

### Device Driver Special Files and Description

**arp** The following special file is installed:

```
arp                   rw-rw-rw- root sys
```

#### asio0

For the built-in serial port, the following special files are installed for each card instance:

```
ttycardp0            rw--w--w- bin bin
Direct connect
```

#### asio0

For the SAS console ports, the following special files are installed for each card instance:

```
ttycardp0            rw--w--w- bin bin
local console port (direct connect)

ttycardp1            rw--w--w- bin bin
remote session port (direct connect)

sassy                rw----- root sys
internal console port (direct connect)
```

```

ttycard+lp0    rw--w--w- bin bin
               UPS port (direct connect)

ttycard+lp1    rw--w--w- bin bin
               local session port (direct connect)

```

**asyncdsk**

The following special file is installed:

```
asyncdsk      rw-rw-rw- bin bin
```

**audio**

The following special files are installed. Note the underscore (\_) before *card* in each special file name.

For *card* 0, the device files are linked to files without the trailing *\_0* in their names.

```

audio_card    rw-rw-rw- bin bin
               Default audio device

audioCtl_card rw-rw-rw- bin bin
               Audio control device

audioBA_card  rw-rw-rw- bin bin
               All outputs, A-law format

audioBL_card  rw-rw-rw- bin bin
               All outputs, 16-bit linear format

audioBU_card  rw-rw-rw- bin bin
               All outputs, Mu-law format

audioEA_card  rw-rw-rw- bin bin
               External output, A-law format

audioEL_card  rw-rw-rw- bin bin
               External output, 16-bit linear format

audioEU_card  rw-rw-rw- bin bin
               External output, Mu-law format

audioIA_card  rw-rw-rw- bin bin
               Internal speaker output, A-law format

audioIL_card  rw-rw-rw- bin bin
               Internal speaker output, 16-bit linear format

audioIU_card  rw-rw-rw- bin bin
               Internal speaker output, Mu-law format

audioLA_card  rw-rw-rw- bin bin
               Line output, A-law format

audioLL_card  rw-rw-rw- bin bin
               Line output, 16 bit linear format

audioLU_card  rw-rw-rw- bin bin
               Line output, Mu-law format

audioNA_card  rw-rw-rw- bin bin
               No output, A-law format

audioNL_card  rw-rw-rw- bin bin
               No output, 16 bit linear format

audioNU_card  rw-rw-rw- bin bin
               No output, Mu-law format

```

**autox0 schgr**

Special file names for **autox0** and **schgr** use the format:

```
ccardttargetd device_surface
```

*surface*: 1a through 32b, unless modified by the **-p** option. Note the underscore (\_) between *device* and *surface*.

For each autochanger device, the following special files are installed:

```
ac/c cardt targetd device_surface  rw-r----- bin sys
Block entry
```

```
rac/c cardt targetd device_surface  rw-r----- bin sys
Character entry
```

```
rac/c cardt targetd device          rw----- bin sys
Character entry
```

#### beep

The following special file is installed:

```
beep          rw-rw-rw- bin bin
```

#### CentIf

For each card instance, the following special file is installed.

```
ccardt targetd device_lp          rw-rw-rw- lp bin
Handshake mode 2, character entry
```

#### consp1

For each card instance, the following special files are installed:

```
ttycardp0      rw--w--w- bin bin
Direct connect
```

#### cn

The following special files are installed:

```
syscon          rw--w--w- bin bin
systty          rw--w--w- bin bin
console         rw--w--w- root sys
ttyconf         rw----- root sys
```

#### devconfig

The following special file is installed:

```
config          rw-r----- root sys
```

#### diag0

The following special file is installed:

```
diag/diag0      rw----- bin bin
```

#### diag1

The following special file is installed:

```
diag/diag1      rw----- bin bin
```

#### diag2

The following special files are installed:

```
diag2           rw----- bin bin
diag/diag2      rw----- bin bin
```

#### disc3 sdisk

For each disk device, the following special files are installed:

```
dsk/c cardt targetd device        rw-r----- bin sys
Block entry
```

```
rdsk/c cardt targetd device        rw-r----- bin sys
Character entry
```

For **disc3** instances, the following additional special files are installed:

```
floppy/c cardt targetd device      rw-r----- bin sys
Block entry
```

```
rfloppy/c cardt targetd device      rw-r----- bin sys
```

## Character entry

**dlpi**

The following special files are installed:

```
dlpi          rw-rw-rw-  root sys
dlpi0         rw-rw-rw-  root sys
dlpi1         rw-rw-rw-  root sys
dlpi2         rw-rw-rw-  root sys
dlpi3         rw-rw-rw-  root sys
dlpi4         rw-rw-rw-  root sys
```

**dmem**

The following special file is installed:

```
dmem          rw-----  bin bin
```

**echo**

The following special file is installed:

```
echo          rw-rw-rw-  root sys
```

**eisa\_mux0 pci\_mux0**

For each instance of an EISA mux or PCI mux card, the following "Direct Connect" special files are created. The term "card" below refers to the instance number of the mux card.

```
ttycardport_module port
                    rw--w--w-  bin bin
                    letter: a to p, port module name
                    port: 1 to 16, direct connect

muxcard           rw-----  bin bin
diag/muxcard      rw-----  bin bin
diag/muxcard_1    rw-----  bin bin
diag/muxcard_2    rw-----  bin bin
```

**fddi**

The following special file is installed:

```
lancard        rw-rw-rw-  bin bin
```

**framebuf**

For each graphics device, the following special files are installed.

```
crtdevice_number rw-rw-rw-  bin bin
ocrtdevice_number
                    rw-rw-rw-  bin bin
```

*device\_number* is 0 indexed and is assigned in the order in which the devices appear in *ioscan(1M)* output.

If the console device is a graphics device, the files **crt** and **ocrt** are created as the console device. If the console is not a graphics device, **crt** and **ocrt** are identical to **crt0** and **ocrt0**.

**hil** For each device, the following special files are installed. Note the underscore (**\_**) before *card* in each special file name.

For *card* 0, the device files are linked to files named **hiladdr** for the link addresses 1 to 7; **hilkbd** for the cooked keyboard device; and **rhil** for the **hil** controller device.

```
hil_card.addr    rw-rw-rw-  bin bin
                  addr: link addresses 1 to 7

hilkbd_card      rw-rw-rw-  bin bin
rhil_card        rw-rw-rw-  bin bin
```

**inet\_clts**

The following special file is installed:

```
inet_clts      rw-rw-rw-  root sys
```

**inet\_cots**

The following special file is installed:

```
inet_cots      rw-rw-rw-  root sys
```

**ip** The following special file is installed:

```
ip             rw-rw-rw-  root sys
```

**kepd**

The following special file is installed:

```
kepd           rw-r--r--  root other
```

**klog**

The following special file is installed:

```
klog           rw-----  bin bin
```

**lan0 lan1 lan2 lan3**

For each card instance, the following special files are installed:

```
lancard        rw-rw-rw-  bin bin
```

```
ethercard       rw-rw-rw-  bin bin
```

```
diag/lancard   rw-----  bin bin
```

**lantty0**

For each card instance, the following special files are installed:

```
lanttycard      rw-rw-rw-  bin bin
Normal access
```

```
diag/lanttycard rw-rw-rw-  bin bin
Exclusive access
```

**lpr2 lpr3**

For each card instance, the following special files are installed:

```
ccardttargetddevice_lp  rw-----  lp bin
```

```
diag/ccardttargetddevice_lp  rw-----  bin bin
```

**mm** The following special files are installed:

```
mem            rw-r-----  bin sys
Minor 0
```

```
kmem           rw-r-----  bin sys
Minor 1
```

```
null          rw-rw-rw-  bin bin
Minor 2
```

**mux0**

For each instance of a 6-channel card, the following special files are installed:

```
ttycardpport     rw--w--w-  bin bin
port: 0 to 5, direct connect
```

```
muxcard         rw-----  bin bin
```

```
diag/muxcard    rw-----  bin bin
```

For each instance of a 16-channel card, the following special files are installed:

```
ttycardpport     rw--w--w-  bin bin
port: 0 to 15, direct connect
```

```
muxcard         rw-----  bin bin
```



```

diag/mux card    rw----- bin bin
mux2
For each instance of an 16-channel card, the following special files are installed:
ttycardport      rw--w--w- bin bin
                  port: 0 to 15, direct connect
mux card         rw----- bin bin
diag/mux card    rw----- bin bin
For each card instance of an 8-channel card, the following special files are
installed:
ttycardport      rw--w--w- bin bin
                  port: 0 to 7, direct connect
mux card         rw----- bin bin
diag/mux card    rw----- bin bin
For each card instance of an 3-channel card, the following special files are
installed:
ttycardport      rw--w--w- bin bin
                  port: 0, 1, and 7, direct connect
mux card         rw----- bin bin
diag/mux card    rw----- bin bin
mux4
For each card instance, the following special files are installed:
ttycardport      rw--w--w- bin bin
                  port: 0 and 1, direct connect
netqa
The following special file is installed:
netqa            rw-rw-rw- root sys
nuls
The following special file is installed:
nuls             rw-rw-rw- root sys
pci_mux0
The following "Direct Connect" special files are created. The term "card" below refers to the instance
number of the mux card.
ttycardport_module port
                  rw--w--w- bin bin
                  port_module: a to p, port module name
                  port: 1 to 16, port number
mux card         rw----- bin bin
diag/mux card    rw----- bin bin
diag/mux card_1  rw----- bin bin
diag/mux card_2  rw----- bin bin
pflop sflop
For each card instance, the following special files are installed:
floppy/c cardt targetd device  rw-r----- bin sys
                                Block entry
rfloppy/c cardt targetd device  rw-r----- bin sys
                                Character entry
ps2 The following special files are installed:

```

```

ps2kbd          rw-rw-rw-  bin bin
                 Autosearch for first ps2 keyboard

ps2mouse        rw-rw-rw-  bin bin
                 Autosearch for first ps2 mouse

ps2_0           rw-rw-rw-  bin bin
                 ps2 port 0

ps2_1           rw-rw-rw-  bin bin
                 ps2 port 1

```

**ptm** The following special file is installed:

```
ptmx            rw-rw-rw-  root sys
```

**pts** The following special files are installed:

```
pts/number      rw-rw-rw-  root sys
                 number: 0 to 59
```

**pty** Specifying this driver tells **insf** to install the special files for both the master and slave pty drivers, **ptym** and **ptys**. The command **insf -d pty** is equivalent to the two commands **insf -d ptym** and **insf -d ptys**.

**ptym**

The following special files are installed:

```

ptym/clone      rw-r--r--  root other

ptym/pty index number
                 index: p to z, a to c, e to o; number: 0 to f (hexadecimal)

                 The first 48 special files ptym/pty* are linked to pty*.

ptym/pty index number
                 rw-rw-rw-  bin bin
                 index: p to z, a to c, e to o; number: 00 to 99

ptym/pty index number
                 rw-rw-rw-  bin bin
                 index: p to z, a to c, e to o; number: 000 to 999

```

**ptys**

The following special files are installed:

```

pty/tty index number
                 rw-rw-rw-  bin bin
                 index: p to z, a to c, e to o; number: 0 to f (hexadecimal)

                 The first 48 special files pty/tty* are linked to tty*.

pty/tty index number
                 rw-rw-rw-  bin bin
                 index: p to z, a to c, e to o; number: 00 to 99

pty/tty index number
                 rw-rw-rw-  bin bin
                 index: p to z, a to c, e to o; number: 000 to 999

```

**rawip**

The following special file is installed:

```
rawip          rw-rw-rw-  root sys
```

**root**

The following special files are installed:

```

root          rw-r-----  bin sys
rroot        rw-r-----  bin sys

```

**sad** The following special file is installed:

```
sad           rw-rw-rw-  root sys
```

**sastty**

For each card instance, the following special files are installed:

```
ttycardport    rw--w--w-  bin bin
```

*port:* 0 to 1, direct connect

**schgr**  
See **autox0**.

**sdisk**  
See **disc3**.

**sflop**  
See **pflop**.

#### **stape tape2**

For each driver instance, different special files are installed depending on the number of characters allowed in the target directory. There are two lists below, one for long file name directories and one for short file name directories (14 characters maximum). Short file names are used for files installed on an NFS file system.

Note that the first four special files in each list for tape driver instances 0-9 are also linked to **rmt / instance~~m~~**, **rmt / instance~~m~~b**, **rmt / instance~~m~~n**, and **rmt / instance~~m~~nb**, respectively.

For installation in a long file name directory:

|                                                                            |                                                                                                |
|----------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|
| <b>rmt / c card<del>t</del> target<del>d</del> device<del>BEST</del></b>   | <b>rw-rw-rw- bin bin</b><br>AT&T-style, best available density, character entry                |
| <b>rmt / c card<del>t</del> target<del>d</del> device<del>BEST</del>b</b>  | <b>rw-rw-rw- bin bin</b><br>Berkeley-style, best available density, character entry            |
| <b>rmt / c card<del>t</del> target<del>d</del> device<del>BEST</del>n</b>  | <b>rw-rw-rw- bin bin</b><br>AT&T-style, no rewind, best available density, character entry     |
| <b>rmt / c card<del>t</del> target<del>d</del> device<del>BEST</del>nb</b> | <b>rw-rw-rw- bin bin</b><br>Berkeley-style, no rewind, best available density, character entry |

For installation in a short file name directory:

|                                                                          |                                                                                                |
|--------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|
| <b>rmt / c card<del>t</del> target<del>d</del> device<del>f</del>0</b>   | <b>rw-rw-rw- bin bin</b><br>AT&T-style, best available density, character entry                |
| <b>rmt / c card<del>t</del> target<del>d</del> device<del>f</del>0b</b>  | <b>rw-rw-rw- bin bin</b><br>Berkeley-style, best available density, character entry            |
| <b>rmt / c card<del>t</del> target<del>d</del> device<del>f</del>0n</b>  | <b>rw-rw-rw- bin bin</b><br>AT&T-style, no rewind, best available density, character entry     |
| <b>rmt / c card<del>t</del> target<del>d</del> device<del>f</del>0nb</b> | <b>rw-rw-rw- bin bin</b><br>Berkeley-style, no rewind, best available density, character entry |

For both long and short file name directories, the following additional files are created.

|                                                                  |                                                                                     |
|------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| <b>rmt / driver_name_config</b>                                  | <b>rw-r--r-- bin bin</b><br>Tape configuration, character entry                     |
| <b>diag / rmt / c card<del>t</del> target<del>d</del> device</b> | <b>rw----- bin bin</b><br>For <b>tape2</b> only, diagnostic access, character entry |

#### **stcpmap**

The following special file is installed:

|                |                           |
|----------------|---------------------------|
| <b>stcpmap</b> | <b>rw-rw-rw- root sys</b> |
|----------------|---------------------------|

#### **strlog**

The following special file is installed:

|               |                           |
|---------------|---------------------------|
| <b>strlog</b> | <b>rw-rw-rw- root sys</b> |
|---------------|---------------------------|

**sy** The following special file is installed:

|            |                          |
|------------|--------------------------|
| <b>tty</b> | <b>rw-rw-rw- bin bin</b> |
|------------|--------------------------|

#### **tape2**

See **stape**.

**tcp** The following special file is installed:

```

tcp                rw-rw-rw- root sys
telm
    The following special file is installed:
telnetm            rw-rw-rw- root sys
tels
    The following special files are installed:
pts/tnumber        rw-rw-rw- root sys
                    number: 0 to 59
tlclts
    The following special file is installed:
tlclts            rw-rw-rw- root sys
tlcots
    The following special file is installed:
tlcots            rw-rw-rw- root sys
tlcotsod
    The following special file is installed:
tlcotsod          rw-rw-rw- root sys
token2
    The following special file is installed:
lancard            rw-rw-rw- bin bin
udp
    The following special file is installed:
udp                rw-rw-rw- root sys
unix_clts
    The following special file is installed:
unix_clts          rw-rw-rw- root sys
unix_cots
    The following special file is installed:
unix_cots          rw-rw-rw- root sys

```

## RETURN VALUE

**insf** exits with one of the following values:

- 0 Successful completion, including warning diagnostics.
- 1 Failure.

## DIAGNOSTICS

Most diagnostic messages from **insf** are self-explanatory. Listed below are some messages deserving further clarification.

### Warnings

Device driver *name* is not in the kernel

Device class *name* is not in the kernel

The indicated device driver or device class is not present in the kernel. A device driver and/or device class can be added to the kernel using *config*(1M).

No instance number available for device class *name*

All of the instance numbers available for the device class are already assigned. Use the **rmsf** command to remove any unneeded devices from the system (see *rmsf*(1M)).

Don't know how to handle driver *name* - no special files created for *path*

**insf** does not know how to create special files for the specified device driver. Use **mknod** to create special files for the device (see *mknod*(1M)).

**EXAMPLES**

Install special files for all new devices belonging to the **tty** device class:

```
insf -C tty
```

Install special files to the new device added at hardware path **2/4.0.0**:

```
insf -H 2/4.0.0
```

**WARNINGS**

**insf** should only be run in single-user mode. It can change the mode, owner, or group of an existing special file, or unlink and recreate one; special files that are currently open may be left in an indeterminate state.

Many commands and subsystems assume their device files are in **/dev**, therefore the use of the **-D** option is discouraged.

**AUTHOR**

**insf** was developed by HP.

**FILES**

|                      |                                   |
|----------------------|-----------------------------------|
| <b>/dev/config</b>   | I/O system special file           |
| <b>/etc/ioconfig</b> | I/O system configuration database |

**SEE ALSO**

**config(1M)**, **ioscan(1M)**, **lsdev(1M)**, **lssf(1M)**, **mknod(1M)**, **mksf(1M)**, **rmsf(1M)**.

**NAME**

install-sd - utility used to install the Software Distributor (SD) product

**SYNOPSIS**

```
install-sd -s source_depot_location
```

**DESCRIPTION**

The **install-sd** command installs the SD product onto a workstation or server. You may need to install SD if the SD currently on your system is unusable, or if you want to install a newer version of SD. This command also installs any SD patches that exist in the source depot.

**Prerequisites**

The **install-sd** command and an accompanying **swagent.Z** file require at least 2 MB of free space in the **/var/tmp** directory. If there is not enough space in the temporary directory **install-sd** will fail.

**Options**

The **install-sd** command supports the following option:

**-s** *source\_depot\_location*

Specifies the source media location, where *source\_depot\_location* is an absolute path. Possible media locations are:

- A local directory.
- A CD-ROM mount point that has an SD media CD-ROM loaded.
- A remote machine (or *host*) and depot combination.

If *source\_depot\_location* is a remote machine and depot combination, specify the machine name, a colon, then the absolute path to the remote depot. The syntax is:

```
[machine_name][:][/depot_path]
```

For example:

```
swtest:/var/spool/sw
```

**Replacing an Unusable Version of SD**

If the SD on the target system is unusable, you must first load the **install-sd** utility and an accompanying **swagent.Z** file onto your system into **/var/tmp**, then use **install-sd** to re-install SD.

The **install-sd** utility ships in the **catalog/SW-DIST/pfiles** directory. Use **cp** or **rcp** (depending on whether the new software is on CD or a remote system in a software depot) to load **install-sd** onto your system.

For example, to load **install-sd** from a local CD-ROM mounted at **/SD\_CDRom** into **/var/tmp**:

- 1) Copy **install-sd** onto your system from the CD-ROM:

```
cp /SD_CDRom/catalog/SW-DIST/pfiles/install-sd /var/tmp
```

- 2) Copy the **swagent.Z** file from the CD-ROM:

```
cp /SD_CDRom/catalog/SW-DIST/pfiles/swagent.Z /var/tmp
```

- 3) Make **install-sd** executable:

```
chmod +x /var/tmp/install-sd
```

- 4) Execute **install-sd**:

```
/var/tmp/install-sd -s /SD_CDRom
```

The SW-DIST product will then install itself onto your system from the CD-ROM.

**Getting a Newer Version of SD**

If you want to install a newer version of SD on your system and **/usr/sbin/install-sd** is not yet on your system, use this procedure:

- 1) Type on the command line:

```
/usr/sbin/swinstall -r -s source_depot_location \  
SW-DIST.SD-UPDATE > \@/var/adm/sw/install-sd.root 2>/dev/null
```

2) Install the newer SD:

```
/usr/sbin/install-sd -s source_depot_location
```

where *source\_depot\_location* is the absolute path to the depot or media containing the newer version of SD.

## RETURN VALUES

The `install-sd` command returns:

- 0 Successful completion
- 1 Error during execution

## DIAGNOSTICS

### Standard Output

An `install-sd` session writes messages for significant events. These include:

- Begin session and end session messages.
- Major task messages.

### Standard Error

An `install-sd` session writes messages for all WARNING and ERROR conditions to stderr.

### Logging

An `install-sd` session logs detailed events to the `install-sd` log file located at:

```
/var/adm/sw/install-sd.log
```

## EXAMPLES

To load `install-sd` from a local CD-ROM mounted at `/SD_CDROM` into `/var/tmp`:

```
cp /SD_CDROM/catalog/SW-DIST/pfiles/install-sd /var/tmp
cp /SD_CDROM/catalog/SW-DIST/pfiles/swagent.Z /var/tmp
chmod +x /var/tmp/install-sd
/var/tmp/install-sd -s /SD_CDROM
```

To install a newer version of SD onto your system from the `/var/spool/sw` depot on `swtest` (when `/usr/sbin/install-sd` is not yet on your system):

```
/usr/sbin/swinstall -r -s swtest:/var/spool/sw \
    SW-DIST.SD-UPDATE > \@/var/adm/sw/install-sd.root 2>/dev/null
/usr/sbin/install-sd -s swtest:/var/spool/sw
```

## AUTHOR

`install-sd` was developed by the Hewlett-Packard Company.

## FILES

```
/usr/sbin/install-sd
    Location of the install-sd command.

/var/adm/sw/install-sd.log
    Location of the install-sd log file.
```

## SEE ALSO

`sd(4)`, `sd(5)`, `update-ux(1M)`,

*Software Distributor Administration Guide*, available at <http://docs.hp.com>.

**NAME**

install - install commands

**SYNOPSIS**

```
/usr/sbin/install [-c dira] [-f dirb] [-i] [-n dirc] [-o] [-g group] [-s] [-u user]
file [dirx ...]
```

**DESCRIPTION**

**install** is a command most commonly used in “makefiles” (see *make(1)*) to install a *file* (updated target file) in a specific place within a file system. Each *file* is installed by copying it into the appropriate directory, thereby retaining the mode and owner of the original command. The program prints messages telling the user exactly what files it is replacing or creating and where they are going.

**install** is useful for installing new commands, or new versions of existing commands, in the standard directories (i.e. **/usr/bin**, **/usr/sbin**, etc.).

If no options or directories (*dirx*...) are given, **install** searches a set of default directories (**/usr/bin**, **/usr/sbin**, **/sbin**, and **/usr/sbin**, in that order) for a file with the same name as *file*. When the first occurrence is found, **install** issues a message saying that it is overwriting that file with *file* (the new version), and proceeds to do so. If the file is not found, the program states this and exits without further action.

If one or more directories (*dirx*...) are specified after *file*, those directories are searched before the directories specified in the default list.

**Options**

Options are interpreted as follows:

- c** *dira*      Installs a new command (*file*) in the directory specified by *dira*, only if it is not found. If it is found, **install** issues a message saying that the file already exists, and exits without overwriting it. Can be used alone or with the **-s** option.
- f** *dirb*      Forces *file* to be installed in given directory, whether or not one already exists. If the file being installed does not already exist, the mode and owner of the new file will be set to **755** and **bin**, respectively. If the file already exists, the mode and owner will be that of the already existing file. Can be used alone or with the **-o** or **-s** options.
- i**            Ignores default directory list, searching only through the given directories (*dirx*...). Can be used alone or with any other options other than **-c** and **-f**.
- n** *dirc*      If *file* is not found in any of the searched directories, it is put in the directory specified in *dirc*. The mode and owner of the new file will be set to **755** and **bin**, respectively. Can be used alone or with any other options other than **-c** and **-f**.
- o**            If *file* is found, this option saves the “found” file by copying it to **OLDfile** in the directory in which it was found. This option is useful when installing a normally busy text file such as **/usr/bin/sh** or **/usr/sbin/getty**, where the existing file cannot be removed. Can be used alone or with any other options other than **-c**.
- g** *group*    Causes *file* to be owned by group *group*. This option is available only to users who have appropriate privileges. Can be used alone or with any other option.
- u** *user*      Causes *file* to be owned by user *user*. This option is available only to users who have appropriate privileges. Can be used alone or with any other option.
- s**            Suppresses printing of messages other than error messages. Can be used alone or with any other options.

When no directories are specified (*dirx*...), or when *file* cannot be placed in one of the directories specified, **install** checks for the existence of the file **/etc/syslist**. If **/etc/syslist** exists, it is used to determine the final destination of *file*. If **/etc/syslist** does not exist, the default directory list is further scanned to determine where *file* is to be located.

The file **/etc/syslist** contains a list of absolute pathnames, one per line. The pathname is the “official” destination (for example **/usr/bin/echo**) of the file as it appears on a file system. The file **/etc/syslist** serves as a master list for system command destinations. If there is no entry for *file* in the file **/etc/syslist** the default directory list is further scanned to determine where *file* is to be located.



**Cross Generation**

The environment variable `ROOT` is used to locate the locations file (in the form `$ROOT/etc/syslist`). This is necessary in cases where cross generation is being done on a production system. Furthermore, each pathname in `$ROOT/etc/syslist` is appended to `$ROOT` (for example, `$ROOT/usr/bin/echo`), and used as the destination for *file*. Also, the default directories are also appended to `$ROOT` so that the default directories are actually `$ROOT/usr/bin`, `$ROOT/usr/sbin`, `$ROOT/sbin`, and `$ROOT/usr/lib`.

The file `/etc/syslist` (`$ROOT/etc/syslist`) does not exist on a distribution tape; it is created and used by local sites.

**WARNINGS**

`install` cannot create alias links for a command (for example, `vi(1)` is an alias link for `ex(1)`).

**SEE ALSO**

`make(1)`, `cpset(1M)`.

**NAME**

ioinit - test and maintain consistency between the kernel I/O data structures and /etc/ioconfig

**SYNOPSIS**

```
/sbin/ioinit -i [-r]
/sbin/ioinit -c
/sbin/ioinit -f infile [-r]
```

**DESCRIPTION**

The **ioinit** command is invoked by the **init** process when the system is booted, based on the **ioin** entry in **/etc/inittab**:

```
ioin::sysinit:/sbin/ioinitrc > /dev/console 2>&1
```

where **ioinitrc** is a script to invoke **ioinit** with the **-i** and **-r** options. Given the **-i** option, **ioinit** checks consistency between the kernel I/O data structures (initialized with **/stand/ioconfig**, which is accessible for NFS-diskless support when the system boots up) and information read from **/etc/ioconfig**. If these are consistent, **ioinit** invokes **insf** to install special files for all new devices. If the kernel is inconsistent with **/etc/ioconfig**, **ioinit** updates **/stand/ioconfig** from **/etc/ioconfig**, and, if the **-r** option is given, reboots the system.

If **/etc/ioconfig** is corrupted or missing when the system reboots, **ioinitrc** brings the system up in single-user mode. The user should then restore **/etc/ioconfig** from backup or invoke the **ioinit** with the **-c** option to recreate **/etc/ioconfig** from the kernel.

If the **-f** option is given, **ioinit** reassigns instance numbers to existing devices within a given class based on **infile**. Reassignment takes effect when the system reboots. If **ioinit** finds no errors associated with the reassignment, and the **-r** option is given, the system is rebooted. (See the WARNINGS section.)

If the **-c** option is given, **ioinit** recreates **/etc/ioconfig** from the existing kernel I/O data structures.

**Options**

**ioinit** recognizes the following options:

- i** Invoke **insf** to install special files for new devices after checking consistency between the kernel and **/etc/ioconfig**.
- f infile** Use the file **infile** to reassign instance numbers to devices within a specified class. **infile** may have multiple entries, each to appear on a separate line, each field in the entry separated by 1 or more blanks. Entries should conform to the following format:

| h/w_path | class_name | instance_# |
|----------|------------|------------|
|----------|------------|------------|

**ioinit** preprocesses the contents of **infile**, looking for invalid entries, and prints out explanatory messages. An entry is considered to be invalid if the specified hardware path or class name does not already exist in the system, or if the specified instance number already exists for the given class.

- r** Reboot the system when it is required to correct the inconsistent state between the kernel and **/etc/ioconfig**, as used with the **-i** option. When used with the **-f** option, if there are no errors associated with the instance reassignment, **-r** reboots the system.
- c** Recreate **/etc/ioconfig**, if the file is corrupted or missing and cannot be restored from backup. If **-c** is invoked, any previous binding of hardware path to device class and instance number is lost.

**RETURN VALUE**

- 0 No errors occurred, although warnings might be issued.
- 1 **ioinit** encountered an error.

**DIAGNOSTICS**

Most of the diagnostic messages from **ioinit** are self-explanatory. Listed below are some messages deserving further clarification. Errors cause **ioinit** to halt immediately.

**Errors**

**/etc/ioconfig is missing.**

**/etc/ioconfig is corrupted.**

Either restore **/etc/ioconfig** from backup and then reboot, or recreate **/etc/ioconfig** using **ioinit -c**.

**Permission to access /etc/ioconfig is denied.**

Change permissions to **/etc/ioconfig** to allow access by **ioinit**.

**exec of insf failed.**

**ioinit** completed successfully, but **insf** failed.

**Instance number is already in kernel.**

Instance number already exists for a given class. Use **rmsf** to remove the existing instance number, then retry.

**Hardware path is not in the kernel.**

The given hardware path is not in the kernel. Use **ioscan -k** to get the correct hardware path, then retry.

**Device class name is not in the kernel.**

The given class name is not in the kernel. Use **ioscan -k** to get the correct class name, then retry.

**EXAMPLES**

To reassign an instance number to a device and class (specified in *infile*) and reboot the system:

```
/sbin/ioinit -f infile -r
```

where **infile** contains the following:

```
56.52          scsi          2
```

**56.52** is the *h/w\_path*, **scsi** is the *class\_name*, and **2** is the *instance\_#*.

**WARNINGS**

Running **rmsf** or **insf** overwrites the effect of reassignment by **ioinit** before the system is rebooted.

**AUTHOR**

**ioinit** was developed by HP.

**FILES**

**/stand/ioconfig**

**/etc/ioconfig**

**SEE ALSO**

**init(1M)**, **insf(1M)**, **ioscan(1M)**, **rmsf(1M)**, **inittab(4)**, **ioconfig(4)**.

**NAME**

ioscan - scan I/O system

**SYNOPSIS**

```
/usr/sbin/ioscan [-k|-u] [-d driver|-C class] [-I instance] [-H hw_path] [-f|-n] [-F|-n]
[devfile]
```

```
/usr/sbin/ioscan -M driver -H hw_path [-I instance]
```

**DESCRIPTION**

**ioscan** scans system hardware, usable I/O system devices, or kernel I/O system data structures as appropriate, and lists the results. For each hardware module on the system, **ioscan** displays by default the hardware path to the hardware module, the class of the hardware module, and a brief description.

By default, **ioscan** scans the system and lists all reportable hardware found. The types of hardware reported include processors, memory, interface cards and I/O devices. Scanning the hardware may cause drivers to be unbound and others bound in their place in order to match actual system hardware. Entities that cannot be scanned are not listed.

In the second form shown, **ioscan** forces the specified software driver into the kernel I/O system at the given hardware path and forces software driver to be bound. This can be used to make the system recognize a device that cannot be recognized automatically; for example, because it has not yet been connected to the system, does not support autoconfiguration, or because diagnostics need to be run on a faulty device.

Generally, **ioscan** requires superuser privileges. A non root user may use the **-k** option, only to display the kernel hardware tree. Driver binding and actual hardware scanning is restricted to root.

**Options**

**ioscan** recognizes the following options:

- C class** Restrict the output listing to those devices belonging to the specified *class*. Cannot be used with **-d**.
- d driver** Restrict the output listing to those devices controlled by the specified *driver*. Cannot be used with **-C**.
- f** Generate a full listing, displaying the module's class, instance number, hardware path, driver, software state, hardware type, and a brief description.
- F** Produce a compact listing of fields (described below), separated by colons. This option overrides the **-f** option.
- H hw\_path** Restrict the scan and output listing to those devices connected at the specified hardware path. The hardware path must be a bus path. Scanning below the bus level will not probe the hardware and may produce incorrect results. For example, specifying the path at the target level will always change the state of the device attached to it as NO\_HW. When used with **-M**, this option specifies the full hardware path at which to bind the software modules.
- I instance** Restrict the scan and output listing to the specified instance, when used with either **-d** or **-C**. When used with **-M**, specifies the desired instance number for binding.
- k** Scan kernel I/O system data structures instead of the actual hardware and list the results. No binding or unbinding of drivers is performed. The **-d**, **-C**, **-I**, and **-H** options can be used to restrict listings. Cannot be used with **-u**. This option does not require superuser privileges.
- M driver** Specifies the software driver to bind at the hardware path given by the **-H** option. Must be used with the **-H** option.
- n** List device file names in the output. Only special files in the **/dev** directory and its subdirectories are listed.
- u** Scan and list usable I/O system devices instead of the actual hardware. Usable I/O devices are those having a driver in the kernel and an assigned instance number. The **-d**, **-C**, **-I**, and **-H** options can be used to restrict listings. The **-u** option cannot be used with **-k**.

The **-d** and **-C** options can be used to obtain listings of subsets of the I/O system, although the entire system is still scanned. Specifying **-d** or **-C** along with **-I**, or specifying **-H** or a *devfile* causes **ioscan** to restrict both the scan and the listing to the hardware subset indicated.

### Fields

The **-F** option can be used to generate a compact listing of fields separated by colons (:), useful for producing custom listings with **awk**. Fields include the module's bus type, *cdio*, *is\_block*, *is\_char*, *is\_pseudo*, block major number, character major number, minor number, class, driver, hardware path, identify bytes, instance number, module path, module name, software state, hardware type, a brief description, and card instance. If a field does not exist, consecutive colons hold the field's position. Fields are defined as follows:

|                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                |                                                                    |                  |                                   |                |                                                       |                  |                                                      |                  |                                                                     |               |                                |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|--------------------------------------------------------------------|------------------|-----------------------------------|----------------|-------------------------------------------------------|------------------|------------------------------------------------------|------------------|---------------------------------------------------------------------|---------------|--------------------------------|
| <i>class</i>           | A device category, defined in the files located in the directory <b>/usr/conf/master.d</b> and consistent with the listings output by <b>lsdev</b> (see <i>lsdev</i> (1M)). Examples are <b>disk</b> , <b>printer</b> , and <b>tape</b> .                                                                                                                                                                                                                                                                                                                                                                                |                |                                                                    |                  |                                   |                |                                                       |                  |                                                      |                  |                                                                     |               |                                |
| <i>instance</i>        | The instance number associated with the device or card. It is a unique number assigned to a card or device within a class. If no driver is available for the hardware component or an error occurs binding the driver, the kernel will not assign an instance number and a (-1), is listed.                                                                                                                                                                                                                                                                                                                              |                |                                                                    |                  |                                   |                |                                                       |                  |                                                      |                  |                                                                     |               |                                |
| <i>hw path</i>         | A numerical string of hardware components, notated sequentially from the bus address to the device address. Typically, the initial number is appended by slash (/), to represent a bus converter (if required by your machine), and subsequent numbers are separated by periods (.). Each number represents the location of a hardware component on the path to the device.                                                                                                                                                                                                                                              |                |                                                                    |                  |                                   |                |                                                       |                  |                                                      |                  |                                                                     |               |                                |
| <i>driver</i>          | The name of the driver that controls the hardware component. If no driver is available to control the hardware component, a question mark (?) is displayed in the output.                                                                                                                                                                                                                                                                                                                                                                                                                                                |                |                                                                    |                  |                                   |                |                                                       |                  |                                                      |                  |                                                                     |               |                                |
| <i>software state</i>  | The result of software binding. <table> <tr><td><b>CLAIMED</b></td><td>software bound successfully</td></tr> <tr><td><b>UNCLAIMED</b></td><td>no associated software found</td></tr> <tr><td><b>DIFF_HW</b></td><td>software found does not match the associated software</td></tr> <tr><td><b>NO_HW</b></td><td>the hardware at this address is no longer responding</td></tr> <tr><td><b>ERROR</b></td><td>the hardware at this address is responding but is in an error state</td></tr> <tr><td><b>SCAN</b></td><td>node locked, try again later</td></tr> </table>                                                   | <b>CLAIMED</b> | software bound successfully                                        | <b>UNCLAIMED</b> | no associated software found      | <b>DIFF_HW</b> | software found does not match the associated software | <b>NO_HW</b>     | the hardware at this address is no longer responding | <b>ERROR</b>     | the hardware at this address is responding but is in an error state | <b>SCAN</b>   | node locked, try again later   |
| <b>CLAIMED</b>         | software bound successfully                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                |                                                                    |                  |                                   |                |                                                       |                  |                                                      |                  |                                                                     |               |                                |
| <b>UNCLAIMED</b>       | no associated software found                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                |                                                                    |                  |                                   |                |                                                       |                  |                                                      |                  |                                                                     |               |                                |
| <b>DIFF_HW</b>         | software found does not match the associated software                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                |                                                                    |                  |                                   |                |                                                       |                  |                                                      |                  |                                                                     |               |                                |
| <b>NO_HW</b>           | the hardware at this address is no longer responding                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                |                                                                    |                  |                                   |                |                                                       |                  |                                                      |                  |                                                                     |               |                                |
| <b>ERROR</b>           | the hardware at this address is responding but is in an error state                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                |                                                                    |                  |                                   |                |                                                       |                  |                                                      |                  |                                                                     |               |                                |
| <b>SCAN</b>            | node locked, try again later                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                |                                                                    |                  |                                   |                |                                                       |                  |                                                      |                  |                                                                     |               |                                |
| <i>hardware type</i>   | Entity identifier for the hardware component. It is one of the following strings: <table> <tr><td><b>UNKNOWN</b></td><td>There is no hardware associated or the type of hardware is unknown</td></tr> <tr><td><b>PROCESSOR</b></td><td>Hardware component is a processor</td></tr> <tr><td><b>MEMORY</b></td><td>Hardware component is memory</td></tr> <tr><td><b>BUS_NEXUS</b></td><td>Hardware component is bus converter or bus adapter</td></tr> <tr><td><b>INTERFACE</b></td><td>Hardware component is an interface card</td></tr> <tr><td><b>DEVICE</b></td><td>Hardware component is a device</td></tr> </table> | <b>UNKNOWN</b> | There is no hardware associated or the type of hardware is unknown | <b>PROCESSOR</b> | Hardware component is a processor | <b>MEMORY</b>  | Hardware component is memory                          | <b>BUS_NEXUS</b> | Hardware component is bus converter or bus adapter   | <b>INTERFACE</b> | Hardware component is an interface card                             | <b>DEVICE</b> | Hardware component is a device |
| <b>UNKNOWN</b>         | There is no hardware associated or the type of hardware is unknown                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                |                                                                    |                  |                                   |                |                                                       |                  |                                                      |                  |                                                                     |               |                                |
| <b>PROCESSOR</b>       | Hardware component is a processor                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                |                                                                    |                  |                                   |                |                                                       |                  |                                                      |                  |                                                                     |               |                                |
| <b>MEMORY</b>          | Hardware component is memory                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                |                                                                    |                  |                                   |                |                                                       |                  |                                                      |                  |                                                                     |               |                                |
| <b>BUS_NEXUS</b>       | Hardware component is bus converter or bus adapter                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                |                                                                    |                  |                                   |                |                                                       |                  |                                                      |                  |                                                                     |               |                                |
| <b>INTERFACE</b>       | Hardware component is an interface card                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                |                                                                    |                  |                                   |                |                                                       |                  |                                                      |                  |                                                                     |               |                                |
| <b>DEVICE</b>          | Hardware component is a device                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                |                                                                    |                  |                                   |                |                                                       |                  |                                                      |                  |                                                                     |               |                                |
| <i>bus type</i>        | Bus type associated with the node.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                |                                                                    |                  |                                   |                |                                                       |                  |                                                      |                  |                                                                     |               |                                |
| <i>cdio</i>            | The name associated with the Context-Dependent I/O module.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                |                                                                    |                  |                                   |                |                                                       |                  |                                                      |                  |                                                                     |               |                                |
| <i>is_block</i>        | A boolean value indicating whether a device block major number exists. A <b>T</b> or <b>F</b> is generated in this field.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                |                                                                    |                  |                                   |                |                                                       |                  |                                                      |                  |                                                                     |               |                                |
| <i>is_char</i>         | A boolean value indicating whether a device character major number exists. A <b>T</b> or <b>F</b> is generated in this field.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                |                                                                    |                  |                                   |                |                                                       |                  |                                                      |                  |                                                                     |               |                                |
| <i>is_pseudo</i>       | A boolean value indicating a pseudo driver. A <b>T</b> or <b>F</b> is generated in this field.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                |                                                                    |                  |                                   |                |                                                       |                  |                                                      |                  |                                                                     |               |                                |
| <i>block major</i>     | The device block major number. A -1 indicates that a device block major number does not exist.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                |                                                                    |                  |                                   |                |                                                       |                  |                                                      |                  |                                                                     |               |                                |
| <i>character major</i> | The device character major number. A -1 indicates that a device character major                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                |                                                                    |                  |                                   |                |                                                       |                  |                                                      |                  |                                                                     |               |                                |

|                       |                                                                 |
|-----------------------|-----------------------------------------------------------------|
|                       | number does not exist.                                          |
| <i>minor</i>          | The device minor number.                                        |
| <i>identify bytes</i> | The identify bytes returned from a module or device.            |
| <i>module path</i>    | The software components separated by periods (.).               |
| <i>module name</i>    | The module name of the software component controlling the node. |
| <i>description</i>    | A description of the device.                                    |
| <i>card instance</i>  | The instance number of the hardware interface card.             |

**RETURN VALUE**

**ioscan** returns **0** upon normal completion and **1** if an error occurred.

**EXAMPLES**

Scan the system hardware and list all the devices belonging to the disk device class.

```
ioscan -C disk
```

Forcibly bind driver **tape2** at the hardware path **8.4.1**.

```
ioscan -M tape2 -H 8.4.1
```

**AUTHOR**

**ioscan** was developed by HP.

**FILES**

```
/dev/config  
/dev/*
```

**SEE ALSO**

**config(1M)**, **lsdev(1M)**, **ioconfig(4)**.

**NAME**

isl - initial system loader

**DESCRIPTION**

*isl* implements the operating system independent portion of the bootstrap process. It is loaded and executed after self-test and initialization have completed successfully.

The processor contains special purpose memory for maintaining critical configuration related parameters (e.g. Primary Boot, Alternate Boot, and Console Paths). Two forms of memory are supported: Stable Storage and Non-Volatile Memory (NVM).

Typically, when control is transferred to *isl*, an *autoboot* sequence takes place. An *autoboot* sequence allows a complete bootstrap operation to occur with no intervention from an operator. *isl* executes commands from the *autoexecute* file in a script-like fashion. *autoboot* is enabled by a flag in Stable Storage.

*autosearch* is a mechanism that automatically locates the boot and console devices. For further information, see *pdcc(1M)*.

During an *autoboot* sequence, *isl* displays its revision and the name of any utility it executes. However, if *autoboot* is disabled, after *isl* displays its revision, it then prompts for input from the console device. Acceptable input is any *isl* command name or the name of any utility available on the system. If a non-fatal error occurs or the executed utility returns, *isl* again prompts for input.

**Commands**

There are several commands available in *isl*. The following is a list with a short description. Parameters may be entered on the command line following the command name. They must be separated by spaces. *isl* prompts for any necessary parameters that are not entered on the command line.

|                   |                                                                                |
|-------------------|--------------------------------------------------------------------------------|
| <b>?</b>          |                                                                                |
| <b>help</b>       | Help - List commands and available utilities                                   |
| <b>listf</b>      |                                                                                |
| <b>ls</b>         | List available utilities                                                       |
| <b>autoboot</b>   | Enable or disable the <i>autoboot</i> sequence                                 |
|                   | Parameter - on or off                                                          |
| <b>autosearch</b> | Enable or disable the <i>autosearch</i> sequence                               |
|                   | Parameter - on or off                                                          |
| <b>primpath</b>   | Modify the Primary Boot Path                                                   |
|                   | Parameter - Primary Boot Path in decimal                                       |
| <b>altpath</b>    | Modify the Alternate Boot Path                                                 |
|                   | Parameter - Alternate Boot Path in decimal                                     |
| <b>conspath</b>   | Modify the Console Path                                                        |
|                   | Parameter - Console Path in decimal                                            |
| <b>lsautofl</b>   |                                                                                |
| <b>listautofl</b> | List contents of the <i>autoexecute</i> file                                   |
| <b>display</b>    | Display the Primary Boot, Alternate Boot, and Console Paths                    |
| <b>readnvm</b>    | Display the contents of one word of NVM in hexadecimal                         |
|                   | Parameter - NVM address in decimal or standard hexadecimal notation            |
| <b>readss</b>     | Display the contents of one word of Stable Storage in hexadecimal              |
|                   | Parameter - Stable Storage address in decimal or standard hexadecimal notation |

**DIAGNOSTICS**

*isl* displays diagnostic information through error messages written on the console and display codes on the LED display.

For the display codes, **CE0x** are informative only. **CE1x** and **CE2x** indicate errors, some of which are fatal and cause the system to halt. Other errors merely cause *isl* to display a message.

## (Series 800 Only)

Non-fatal errors during an *autoboot* sequence cause the *autoboot* sequence to be aborted and *isl* to prompt for input. After non-fatal errors during an interactive *isl* session, *isl* merely prompts for input.

Fatal errors cause the system to halt. The problem must be corrected and the system **RESET** to recover.

- CE00** *isl* is executing.
- CE01** *isl* is *autobooting* from the *autoexecute* file.
- CE02** Cannot find an *autoexecute* file. *autoboot* aborted.
- CE03** No console found, *isl* can only *autoboot*.
- CE05** Directory of utilities is too big, *isl* reads only 2K bytes.
- CE06** *autoexecute* file is inconsistent. *autoboot* aborted.
- CE07** Utility file header inconsistent: SOM values invalid.
- CE08** *autoexecute* file input string exceeds 2048 characters. *autoboot* aborted.
- CE09** *isl* command or utility name exceeds 10 characters.
- CE0F** *isl* has transferred control to the utility.
- CE10** Internal inconsistency: Volume label - **FATAL**.
- CE11** Internal inconsistency: Directory - **FATAL**.
- CE12** Error reading *autoexecute* file.
- CE13** Error reading from console - **FATAL**.
- CE14** Error writing to console - **FATAL**.
- CE15** Not an *isl* command or utility.
- CE16** Utility file header inconsistent: Invalid System ID.
- CE17** Error reading utility file header.
- CE18** Utility file header inconsistent: Bad magic number.
- CE19** Utility would overlay *isl* in memory.
- CE1A** Utility requires more memory than is configured.
- CE1B** Error reading utility into memory.
- CE1C** Incorrect checksum: Reading utility into memory.
- CE1D** Console needed - **FATAL**.
- CE1E** Internal inconsistency: Boot device class - **FATAL**.
- CE21** Destination memory address of utility is invalid.
- CE22** Utility file header inconsistent: *pdccache* entry.
- CE23** Internal inconsistency: *iode\_entry\_init* - **FATAL**.
- CE24** Internal inconsistency: *iode\_entry\_init* - console - **FATAL**.
- CE25** Internal inconsistency: *iode\_entry\_init* - boot device - **FATAL**.
- CE26** Utility file header inconsistent: Bad aux\_id.
- CE27** Bad utility file type.

**SEE ALSO**

boot(1M), pdc(1M).



**NAME**

itemap - load an ITE (Internal Terminal Emulator) keyboard mapping

**SYNOPSIS**

**itemap** [*options*]

**DESCRIPTION**

The **itemap** command loads a keyboard mapping into the ITE (the graphics console driver), or displays ITE keyboard mappings. **itemap** is run by **/etc/bcheckrc** automatically. It is not usually explicitly invoked by the user.

**Options**

**-d** *name*

**-d** *keyboard\_ID*

Dump a keymap to standard output in hexadecimal notation.

**-h** Load the specified keymap into the kernel mapping table used for **HP-HIL** keyboards.

**-i** Interactively prompt for a **PS2 DIN** keyboard mapping. **itemap** scans the keymap database file for all mapping names beginning with a **PS2\_DIN** prefix. Each of these names is displayed, and one must be selected.

**-k** *database\_file\_name*

The name of the keymap database file to be used for input. The default is **/etc/X11/XHPKeymaps**.

**-L** Load the appropriate keymap. **itemap** scans the hardware for a keyboard, determines the language of that keyboard, and loads the keymap corresponding to that keyboard.

Because **itemap** cannot determine the language of **PS2 DIN** keyboards, use the **-i** option when using **-L** with **PS2 DIN** keyboards.

**-l** *name*

**-l** *keyboard\_ID*

Load a specified keyboard map. Once loaded, ITE uses the specified mapping.

When loading a keyboard mapping with the **-l** option, **itemap** matches the suffix of the name of the specified keyboard mapping with those found in **/etc/X11/XHPKeymaps** to determine the keyboard language. This information is used by the ITE to perform ISO 7-to-8 bit conversion. Keymap names added by users, via

**/usr/contrib/bin/X11/keymap\_ed**

should use the same suffixes as those already used in **/etc/X11/XHPKeymaps**. For example, a French keyboard mapping can be named **New\_French**, for consistency with existing **ITF\_French** and **PS2\_French** mappings. A mapping called **New\_Stuff** would not match any suffix patterns found by **itemap**, and would result in incorrect ISO 7-to-8 bit conversion.

**-p** Load the specified keymap into the kernel mapping table used for **PS2 DIN** keyboards.

**-v** Perform actions verbosely.

**-w** *file\_name* If a keymap for a **PS2 DIN** keyboard is loaded, write its name to *file\_name*.

**EXAMPLES**

To automatically install the correct mapping for an **HP-HIL** keyboard:

**itemap -L**

To explicitly load the **ITF\_French** mapping for an **HP-HIL** keyboard:

**itemap -h -l ITF\_French**

To explicitly load the **PS2\_DIN\_French** mapping for a **PS2 DIN** keyboard:

**itemap -p -l PS2\_DIN\_French**

To interactively choose a PS2 DIN keyboard mapping:

```
itemap -Li
```

To generate a list of the available keyboard mappings:

```
/usr/contrib/bin/X11/keymap_ed -l
```

**FILES**

```
/usr/contrib/bin/X11/keymap_ed
```

Keymap database editor

```
/etc/X11/XHPKeymaps
```

System keymap database

```
/etc/kbdlang
```

Contains mapping name configured for PS2 DIN keyboards

**SEE ALSO**

ps2(7), termio(7), keymap\_ed(1X111).

**NAME**

keyenvoy - talk to keyserver

**SYNOPSIS**

**keyenvoy**

**Remarks**

The Network Information Service (NIS) was formerly known as Yellow Pages (yp). Although the name has changed, the functionality of the service remains the same.

**DESCRIPTION**

**keyenvoy** is a setuid root process that is used by some RPC programs to intermediate between a user process and the keyerv process, *keyserv*(1M), which will not talk to anything but a root process.

This program cannot be run interactively.

**AUTHOR**

**keyenvoy** was developed by Sun Microsystems, Inc.

**SEE ALSO**

keyserv(1M).

**NAME**

keyserv - server for storing private encryption keys

**SYNOPSIS**

**keyserv** [ **-d** ] [ **-D** ] [ **-n** ]

**DESCRIPTION**

**keyserv** is a daemon that is used for storing the private encryption keys of each user logged into the system. These encryption keys are used for accessing secure network services such as NIS+.

Normally, root's key is read from the file **/etc/.rootkey** when the daemon is started. This is useful during power-fail reboots when no one is around to type a password.

**Options**

- d** Disable the use of default keys for **nobody**.
- D** Run in debugging mode and log all requests to **keyserv**.
- n** Root's secret key is not read from **/etc/.rootkey**. Instead, **keyserv** prompts the user for the password to decrypt root's key stored in the **publickey** database and then stores the decrypted key in **/etc/.rootkey** for future use. This option is useful if the **/etc/.rootkey** file ever gets out of date or corrupted.

**FILES**

**/etc/.rootkey**

**AUTHOR**

**keyserv** was developed by Sun Microsystems, Inc.

**SEE ALSO**

keylogin(1), keylogout(1), publickey(4).

k

**NAME**

killall - kill all active processes

**SYNOPSIS**

**/usr/sbin/killall** [*signal*]

**DESCRIPTION**

**killall** is a procedure used by **/usr/sbin/shutdown** to kill all active processes not directly related to the shutdown procedure.

**killall** is chiefly used to terminate all processes with open files so that the mounted file systems are no longer busy and can be unmounted. **killall** sends the specified *signal* to all user processes in the system, with the following exceptions:

the **init** process;

all processes (including background processes) associated with the terminal from which **killall** was invoked;

any **ps -ef** process, if owned by **root**;

any **sed -e** process, if owned by **root**;

any **shutdown** process;

any **killall** process;

any **/sbin/rc** process.

**killall** obtains its process information from **ps**, and therefore may not be able to perfectly identify which processes to signal (see *ps(1)*).

If no *signal* is specified, a default of **9** (kill) is used.

**killall** is invoked automatically by **shutdown**. The use of **shutdown** is recommended over using **killall** by itself (see *shutdown(1M)*).

**FILES**

**/usr/sbin/shutdown**

**SEE ALSO**

fuser(1M), kill(1), ps(1), shutdown(1M), signal(5).

**STANDARDS CONFORMANCE**

**killall**: SVID2, SVID3

k

**NAME**

killsm - kill the sendmail daemon

**SYNOPSIS**

**/usr/sbin/killsm**

**DESCRIPTION**

**killsm** reads the **/etc/mail/sendmail.pid** file to find the pid number of the currently running sendmail daemon, and then kills that daemon. The **"/sbin/init.d/sendmail stop"** command does the same thing.

HP recommends that system administrators use **"/sbin/init.d/sendmail start"** and **"/sbin/init.d/sendmail stop"** to start and stop sendmail; these startup scripts are used when the system is booting to start sendmail. Advanced system administrators can put **/usr/sbin** into their search path and just reference **"sendmail -bd -q30m"** to start sendmail, and **killsm** to stop it.

The previous **sendmail -bk** option of former releases is no longer supported.

**SEE ALSO**

sendmail(1M).

k

**NAME**

kl - control kernel logging

**SYNOPSIS**

```
/usr/sbin/kl -e [-q qsize] [-s fsize] [-w (on|off)] [-l {d|e|w|i} {subsys_name... / all}]
/usr/sbin/kl -d
/usr/sbin/kl -l {d|e|w|i} {subsys_name... / all}
/usr/sbin/kl -w (on [-s fsize] | off)
/usr/sbin/kl -s fsize
/usr/sbin/kl -i
/usr/sbin/kl -p filename [-w on [-s fsize]]
/usr/sbin/kl -q qsize
```

**DESCRIPTION**

The **kl** command controls the operation of the Kernel Logging facility. Kernel Logging is a high-availability feature that gives system administrators the ability to collect the information necessary to diagnose problems with the HP-UX kernel while the system is running. **kl** is used to specify the levels of events to be logged and the kernel subsystems that will write messages to memory or disk. **kl** also provides for managing the contents of the logfile in memory and on disk.

At startup, Kernel Logging determines its default configuration by reading the file `/etc/nettlgen.conf`. See `nettlgen.conf(4)` for an explanation of the file format. The **kl** command permits only temporary changes to the default Kernel Logging configuration without having to stop and restart Kernel Logging facility. Note that any values you specify on the **kl** command line do not modify the contents of the `/etc/nettlgen.conf` file. To make permanent changes to the values in the `/etc/nettlgen.conf` file, run `nettlconf(1M)`.

Only users with appropriate privileges (root) can invoke the **kl** command to control the Kernel Logging facility.

**Options**

**kl** recognizes the following options, which can be used only in the combinations indicated in the *SYNOPSIS* section. All options and keywords are case-sensitive.

**-e** Enable the Kernel Logging facility and start up default logging as defined in the file `/etc/nettlgen.conf`.

If the **-l** option is used in conjunction with the **-e** option, the **-l** option must be specified as the last option on the command line.

**-d** Disable the Kernel Logging facility. Once this option is issued, Kernel Logging stops accepting logging calls from the kernel subsystems.

**-w (on | off)** Turn on/off write-to-disk logging. If write-to-disk logging is enabled, log messages residing on the log queue in memory are written to disk and removed from the queue.

The name of the log file on disk is formed by adding the suffix `.KLOG0` to the log file name specified in the `/etc/nettlgen.conf` file. If the log file (including suffix) already exists, one of the following events takes place:

- If the existing file contains messages logged during the system run when a panic occurred, the file will be preserved (in a new location). See the *USAGE* section for more information on how this situation is handled.
- Otherwise, the existing file and whatever information it contains will be lost; that is, the contents of the file are overwritten with new kernel logging data.

When write-to-disk logging is turned off, messages are not written out to disk, but continue to be collected in memory (circular buffer).

**-l { d | e | w | i } { subsys\_name ... | all }**

Modify the level of log messages to be captured for the specified subsystem(s).

The *subsys\_name* argument is a subsystem name specified in the file `/etc/nettlgen.conf`. The list of available subsystem names can be obtained using

the command **nettlconf -KL -status**. The keyword **all** changes the logging level for all subsystems specified in the file **/etc/nettngen.conf**.

The following table describes the classes of messages that can be logged

| Message Class | Description                                                                                                                                                                       |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Disaster      | Signals an event or condition which affected the the operation of an entire subsystem, or the entire kernel, causing several programs to fail or the entire machine to shut down. |
| Error         | Signals an event or condition which did not affect the overall operation of an entire subsystem, or the entire kernel, but may have caused an application program to fail.        |
| Warning       | Indicates abnormal events, possibly caused by problems in an individual subsystem.                                                                                                |
| Informative   | Describes routine operations and current system values.                                                                                                                           |

The following table identifies the classes of messages that are captured at each log level

| Log Level | Classes of Messages                      |
|-----------|------------------------------------------|
| <b>d</b>  | Disaster                                 |
| <b>e</b>  | Disaster and Error                       |
| <b>w</b>  | Disaster, Error and Warning              |
| <b>i</b>  | Disaster, Error, Warning and Informative |

Note that, although the log level is specified as a single keyword, messages are logged according to the following rule: if level *x* is specified, then all messages whose severity is greater than or equal to the severity of class *x* will be logged. The order of severity is as follows: Disaster (the most severe), Error, Warning, Informative (the least severe).

**-s *fsize*** Set the size of the two log files used to store logged messages when write-to-disk is enabled. The maximum *fsize* is 1 gigabyte; the minimum *fsize* is 8 kilobytes.

When write-to-disk is started, the default log file size is taken from the file **/etc/nettngen.conf**. The **-s** option allows you to modify the size of the log file without stopping write-to-disk operations.

Note that, when setting the file size, you can specify a numeric value followed by the character **'M'** or **'K'**, which indicates that *fsize* is being defined in units of megabytes or kilobytes, respectively. This suffix character is case-insensitive.

**-p *filename*** Take a picture (snapshot) of the current contents in memory of the kernel logging buffer. This option causes all messages residing in the log queue to be dumped to *filename* and removed from the queue.

Note that taking a picture (snapshot) and write-to-disk are mutually exclusive operations, because there is no reason to take a picture of Kernel Logging messages if write-to-disk component already writes them to disk.

If the **-w on** option is specified in conjunction with the **-p** option, write-to-disk logging will be started immediately after the picture is taken.

**-q *qsize*** Set the size of the Kernel Logging queue. The maximum *qsize* is 10000 messages; the minimum *qsize* is 100 messages. Note that *qsize* indicates the number of messages the queue can hold in memory, not the amount of memory the messages themselves would occupy. At startup, this value is taken from the file **/etc/nettngen.conf**.

**-i** Report information about the status of the Kernel Logging facility.

Information returned by the **-i** option includes:

- kernel logging facility is on or off.
- write-to-disk logging is on or off.
- picture (snapshot) tool is on or off.
- current size of the Kernel Logging queue.
- number of messages currently held in the queue.



- name of the log file used by write-to-disk.
- maximum file size of the log file used by write-to-disk, together with the minimum file size available for the current session of write-to-disk.
- number of messages not written to file (could be due to the lack of memory or small size of the circular buffer).
- list of all subsystems currently specified in the `/etc/nettlgen.conf` file and the associated message class(es) logged for each subsystem.

## USAGE

### When to Use Kernel Logging

Mission critical systems should have KL always enabled. Failing to enable Kernel Logging causes diagnostic information about any suspicious events that might occur on the system to be lost. The recommended classes to capture are Disaster, Error and Warning. Use `kl -l w all` command to do so.

To minimize Kernel Logging's impact on a running system, use the `kl -l e all` command to set all kernel subsystems to capture error-level log messages only.

### Log File Management

The write-to-disk facility uses two files to hold logging information. The base name of the log files is specified in `/etc/nettlgen.conf`. The default base log file name is `/var/adm/kl`; see *nettlgen.conf*(4). The most current data is always in the file with suffix `.KLOG0`. If the size of the `.KLOG0` file reaches a user-defined maximum, Kernel Logging renames the `.KLOG0` file to `.KLOG1`, overwriting the previous contents of the `.KLOG1` file, then continues writing messages to the `.KLOG0` file. (Specify maximum log file size in `/etc/nettlgen.conf` or use the `-s` option.)

The Kernel Logging facility has a protection feature for saving old log files. When write-to-disk starts and encounters old log files that contain messages collected during a prior run of the system when a panic occurred (thus, the log files may contain important information about the panic), then log files are not overwritten. Instead, write-to-disk first attempts to move the old log files to the default crash directory (typically `/var/adm/crash/crash.ID`, where *ID* is a numeric counter). If the move fails, then write-to-disk tries to rename the log files with the prefix `OLD`, allowing the files to remain in the same directory. For example, if write-to-disk finds an old log file named `/var/adm/kl.KLOG0`, it first tries to move the file into the default crash directory. If this move is not successful, write-to-disk then tries to move the file to `/var/adm/OLDkl.KLOG0`.

If both attempts to save old log files fail, the write-to-disk component is not started. To preserve the messages in the old log files, do one of the following:

- If possible, eliminate the conditions that caused both attempts to move the old log files to fail.
- Manually move old log files.
- Using the `nettlconf` command, modify the file `/etc/nettlgen.conf` to specify a different log file name for write-to-disk logging.

Once the problem is corrected, start write-to-disk again. If successful, the Kernel Logging facility will notify you about the move of the old log files.

## RETURN VALUE

`kl` exits with one of the following values:

- 0 Operation was successful
- 1 `kl` command aborted due to error(s)

## EXAMPLES

1. Enable the default Kernel Logging facility.  
`kl -e`
2. Display the information about the Kernel Logging facility.  
`kl -i`
3. Change level to log disaster, error and warning messages for all subsystems.  
`kl -l w all`

4. Request a picture (snapshot) of the collected log messages.

```
kl -p kl_snap.kl
```

5. Turn on default write-to-disk logging.

```
kl -w on
```

6. Change the maximum size of the current log file to 512KB.

```
kl -s 512K
```

7. Turn off write-to-disk logging.

```
kl -w off
```

8. Disable the Kernel Logging facility.

```
kl -d
```

9. Enable on the Kernel Logging facility with the following parameters: queue size is 1000, write-to-disk component is on, maximum file size is 4 megabytes, and log level for the KL\_PM subsystem is Warning.

```
kl -e -q 1000 -w on -s 4M -l w KL_PM
```

Note that `-l w KL_PM` changes the initial level of the KL\_PM subsystem only. This means that the initial levels of the rest of the subsystems specified in the `/etc/nettlgen.conf` file remain unchanged.

10. Take a picture (snapshot) and start write-to-disk logging with a maximum file size of 128K.

```
kl -p -w on 128K
```

**k****AUTHOR**

kl was developed by HP in partnership with NEC.

**FILES**

|                                 |                                                                   |
|---------------------------------|-------------------------------------------------------------------|
| <code>/dev/kernlog</code>       | Kernel log pseudo-device file.                                    |
| <code>/etc/nettlgen.conf</code> | NetTL and KL subsystem configuration file.                        |
| <code>/var/adm/kl.KLOG0</code>  | Default log files as specified in <code>/etc/nettlgen.conf</code> |
| <code>/var/adm/kl.KLOG1</code>  |                                                                   |

**SEE ALSO**

netfmt(1M), nettl(1M), nettlconf(1M), nettlgen.conf(4).

**NAME**

kadmin - kernel module administration

**SYNOPSIS**

```
/usr/sbin/kadmin -d directory_name |-D
/usr/sbin/kadmin -k
/usr/sbin/kadmin -L module_name ... | pathname ...
/usr/sbin/kadmin -q module_id ...
/usr/sbin/kadmin -Q module_name ...
/usr/sbin/kadmin -s | -S
/usr/sbin/kadmin -u module_id ...
/usr/sbin/kadmin -U module_name ...
```

**DESCRIPTION**

**kadmin** is the administrative command for static and loadable kernel modules. It performs the following functions:

- loads a kernel module into a running system
- unloads a kernel module from a running system
- displays the status of kernel module(s) currently loaded or registered
- modifies the search path for kernel modules

The loadable modules feature enables adding a module to a running system without rebooting the system or rebuilding the kernel. When the module is no longer needed, this feature also allows the module to be dynamically removed, thereby freeing system resources for other use.

Loadable modules are maintained in individual object files in the same manner as statically configured modules. Unlike static modules, loadable modules:

- are not linked to the kernel until they are needed
- must be configured into the system and registered with the running kernel using the **config** command, before they can be loaded
- must be configured in loadable form (requires writing additional module initialization or *wrapper* code)
- can be loaded and unloaded by using the **kadmin** command
- can be loaded by the kernel itself (called an auto load)

Auto-load occurs when the kernel detects a particular loadable module is required to accomplish some task, but is not currently loaded. The kernel automatically loads the module.

**Options**

The **kadmin** options have the following meanings:

**-d** *pathname*

Prepend the *pathname* to the current loadable modules search path, where *pathname* specifies directories that should be searched:

for all subsequent demand loads initiated by a **kadmin** command with the option **-L** and a named *module\_name*,

for all subsequent loads performed by the kernel's auto-load mechanism (see note below),

prior to searching any directories already prepended to the search path by a prior **kadmin** command with the **-d** option, and

prior to searching the default search path **/stand/dlkm/mod.d** or **/stand/dlkm.current.vmunix/mod.d**.

*pathname* must specify an absolute pathname or a list of absolute pathnames delimited by colons. The directories identified by *pathname* do not have to exist on the system at the time the request to modify the search path using **kadmin** is made. If these directories do not exist at the time a load takes place, the load operation ignores them.

All modifications to the search path made using this option take effect immediately and affect all subsequent loads (demand and auto-load) and all users on the system.

- D      Reset the kernel modules search path to its default value. The default value can be one of two search paths depending upon the running kernel. When the running kernel is */stand/vmunix*, the default value is */stand/dlkm/mod.d*. When the running kernel is */stand/current.vmunix*, the default value is */stand/dlkm.current.vmunix/mod.d*. The reset takes effect immediately and affects all subsequent loads (demand and auto-load) and all users on the system.
- k      Print a list of all statically configured modules.
- L *module\_name*  
         Load the named module(s), using the current value of the search path to locate the module's object file on disk.  
         This option searches for a matching file in all directories specified in the search path. The default search *pathname* can be one of two values. The *pathname* is */stand/dlkm.current.vmunix/mod.d* when the running kernel is */stand/current.vmunix* or *pathname* is */stand/dlkm/mod.d* when the running kernel is */stand/vmunix*.  
         The load operation performs all tasks associated with link editing the module to the kernel and making the module accessible to the system. If the module depends on other kernel modules (as defined in */usr/conf/master.d*), and these modules are not currently loaded, **kmadmin** will automatically load the dependent modules during the load operation.  
         When loading completes, an integer *module\_id* prints on the standard output to identify the module(s) that was loaded.
- L *pathname*  
         Same as -L *module\_name*, except the absolute pathname, *pathname*, is used to locate the kernel module's object file.
- U *module\_name*  
         Unload the named module(s) *module\_name*.  
         The unload operation performs all tasks associated with disconnecting the module from the kernel and releasing any memory acquired by the module. When unloading completes, a message is displayed to standard output notify the user that the module(s) that has been unloaded.  
         If the module(s) to be unloaded are currently in use, are dependents of a loadable module that is currently loaded, or are currently being loaded or unloaded, the unload request will fail.
- u *module\_id*  
         Same as -U *module\_name*, except that module(s) to be unloaded is identified by the integer value *module\_id*. If *module\_id* is 0 (zero), **kmadmin** attempts to unload all loaded modules.
- q *module\_id*  
         Print the status of loaded or registered module(s) identified by the integer value *module\_id*. Information returned by this option includes:
  - module name
  - module identifier (*module\_id*)
  - the module's *pathname*
  - module status
  - module size
  - the module's virtual load address
  - the memory size of BSS
  - the base address of BSS
  - the module's reference count
  - the module's dependent count
  - the module's unload delay value
  - the module's descriptive name
  - the type of module

Depending on the type of module, information on the module's character major number, block major number and flags may also be printed.

**-Q** *module\_name*

Same as **-q** *module\_id*, except the module(s) for which status information is to be reported is specified by *module\_name* rather than *module\_id*.

**-s** Print an abbreviated status for all modules currently registered or loaded. This option returns a listing of module name, module id, status and type.

Example:

| Name   | ID | Status   | Type |
|--------|----|----------|------|
| hello  | 1  | UNLOADED | Misc |
| misato | 2  | UNLOADED | WSIO |
| stape  | 3  | UNLOADED | WSIO |

**-S** Print the full status for all modules currently loaded. This option returns status information of the form returned by the **-q** options.

## DIAGNOSTICS

**kmadmin** fails in the following cases:

**kmadmin: Incorrect usage**

Command line input contained one or more syntax errors. See the **SYNOPSIS** section for the correct usage.

**kmadmin: module\_id: Invalid argument**

Unable to load the module corresponding to *module\_id* because the module does not exist.

**kmadmin : Device busy**

Unable to load a module because the module is currently in-use.

**kmadmin : Non-numeric ID string: string**

Unable to unload or obtain status for a module because the *module\_id* string specified a non-numeric value.

**kmadmin: modstat: Invalid argument**

Unable to obtain status for module, *module\_id*, because the module does not exist.

**kmadmin: Module: module\_name, not found**

Unable to obtain status for *module* because the module is currently not registered.

## FILES

**/stand/dlkm/mod.d/\*** Default search path for kernel modules when **/stand/vmunix** is the running kernel.

**/stand/dlkm.current.vmunix/mod.d/\*** Default search path for kernel modules when **/stand/current.vmunix** is the running kernel.

## SEE ALSO

config(1M), kmmodreg(1M), kmtune(1M), modload(2), modpath(2), modstat(2), moduload(2), loadmods(4).

**NAME**

kminstall - add, delete, update a kernel module

**SYNOPSIS**

```
/usr/sbin/kminstall [-a|-d|-u] [-s] module_name
```

**DESCRIPTION**

**kminstall** will add (**-a**), delete (**-d**) or update (**-u**) a module's component files on the system.

When called with the **-a** or **-u** option, **kminstall** expects to find the module's component files in the current directory, and installs or updates copies of the files under subdirectories of the **/usr/conf** and **/stand** directories.

**Options**

The options for **kminstall** are:

- a** Add the component files for the named module, *module\_name*, to the appropriate system-specific directories.

A module's component files consist of the following:

|                  |                 |
|------------------|-----------------|
| <b>mod.o</b>     | <i>required</i> |
| <b>master</b>    | <i>required</i> |
| <b>system</b>    | <i>required</i> |
| <b>space.h</b>   | <i>optional</i> |
| <b>Modstub.o</b> | <i>optional</i> |

**kminstall -a** expects minimally a readable **mod.o**, **master**, and **system** file in the current directory. It creates the required system-specific target directories if they do not exist. If *module\_name* already exists on the system, **kminstall -a** prints a message and fails.

- d** Remove the component files for the named module, *module\_name*, from the system-specific directories.

**kminstall -d** deletes the files that have been previously installed via **kminstall -a** or **kminstall -u**. If the module has been configured for use with the current kernel, and **kminstall** can bring the module to an unregistered state, **kminstall -d** will also remove the loadable image of the module from disk. If the module cannot be unregistered, **kminstall -d** prints a warning message and the loadable image remains.

If *module\_name* is configured as a loadable module and its entry is in the **/etc/loadmods** file (see *loadmods(4)*), then **kminstall** prints a warning message and removes the module entry from **/etc/loadmods**.

- s** Silence all warning messages.
- u** Update the component files for the named module, *module\_name*, in the system-specific directories.

**kminstall -u** expects minimally the same required component files in the current directory as the **-a** option. If *module\_name* already exists on the system, **kminstall** updates the module. When updating an existing module, any administrator-configurable attributes (see *kmsystem(1M)* and *kmtune(1M)*) of the existing module will be preserved and applied to the updated module.

If *module\_name* does not exist on the system, then **kminstall -u** prints a warning and proceeds to add the module to the system.

**kminstall** creates the required system-specific target directories if they do not exist.

**RETURN VALUE**

An exit value of zero indicates success. If an error occurs, **kminstall** exits with a non-zero value and reports an error message. Error messages are self-explanatory.

**FILES**

**/usr/conf/master.d/\*** Default input master kernel configuration tables  
**/stand/dlkm\*/mod.d/\*** Configured loadable images associated with a kernel

**SEE ALSO**

config(1M), kmsystem(1M), kmtune(1M), loadmods(4), master(4).

**NAME**

kmmodreg - register or unregister loadable kernel modules with the running kernel

**SYNOPSIS**

```
/usr/sbin/kmmodreg [[-M module_name]...] [-r mod_register_root] [-c mod_reg_root]
/usr/sbin/kmmodreg [[-U module_name]...] [-r mod_register_root] [-c mod_reg_root]
```

**DESCRIPTION**

**kmmodreg** registers all of the loadable kernel modules listed in the **mod\_register** file located under either **/stand/dlkm.current\_vmunix/** when the running kernel is **current\_vmunix**, or **/stand/dlkm** when the running kernel is **/stand/vmunix**. All loadable kernel modules need to be registered by **kmmodreg** before they can be automatically-loaded by the running kernel (i.e., upon module access by an application or user process), or demand-loaded by an administrator issuing the **kmadmin** command.

The **mod\_register** file is generated whenever **config** is run to create a new kernel and contains the registration information for any (and all) configured loadable modules. When **config -M** is run to configure a loadable kernel module, the entries for the module are appended to the **mod\_register**. The **mod\_register** file is not expected to be edited manually. An individual module's registration information is also created by **config** and stored in the **mod\_reg** file located under **/stand/dlkm/mod\_bld.d** directory.

**Options**

**kmmodreg** takes the following options:

- r *mod\_register\_root***  
Use to specify a directory other than **/stand/dlkm.current\_vmunix** or **/stand/dlkm/** as the location for the **mod\_register** file that is used to register modules.
- c *mod\_reg\_root***  
Use the individual module registration information under the *mod\_reg\_root* directory instead of **/stand/dlkm/mod\_bld.d**.
- M *module\_name* [*module\_name*]**  
Register the specified loadable kernel module, and append an entry (or entries) for the module(s) to the **mod\_register** file. This will effect registration of the specified module(s) at every system reboot.
- U *module\_name***  
Unregister the specified loadable kernel module, and remove an entry (or entries) for the module from the **mod\_register** file, so it will not be registered every time the system is rebooted.

**NOTES**

The **kmmodreg** command is executed automatically at every system reboot. **kmupdate** also calls **kmmodreg**, with the **-M** option, when a loadable kernel module configuration is requested. **kmmodreg** can also be invoked as a user-level command to register all of the loadable kernel modules.

**WARNINGS**

The **mod\_register** file format may change or be eliminated in the future.

**FILES**

|                                                         |                                            |
|---------------------------------------------------------|--------------------------------------------|
| <b>/stand/dlkm</b>                                      | Default <i>mod_register_root</i> directory |
| <b>/stand/dlkm.current_vmunix/mod_register</b>          | Default <i>mod_register</i> file           |
| <b>/stand/dlkm/mod_bld.d/<i>module_name</i>/mod_reg</b> | Module registration information            |

Each **mod\_register** file entry provides registration information about a single module. The information is contained in a single-line entry. All fields are positional and are separated by colons. The subfields are separated by commas. The entry is of the form:

*module-name:module-type:type-specific-data*

where:

- *module-name* identifies the module to which the entry belongs
- *module-type* contains an integer representing the module type



- *type-specific-data* includes additional information that depends on the type of the module

**RETURN VALUE**

An exit value of zero indicates successful completion of the command. If errors occur, **kmmodreg** reports error messages for each error and exits with the return value 1. If the error is a failure to register a module, an error message is reported, but the command continues processing the remaining modules listed in the **mod\_register** file. If no modules are processed, **kmmodreg** returns a value of 2.

**SEE ALSO**

config(1M), kmadmin(1M), kmupdate(1M).

**NAME**

kmsystem - set, query configuration and loadable flags for a module

**SYNOPSIS**

```
/usr/sbin/kmsystem [-S system_file]
/usr/sbin/kmsystem [-c {Y|y|N|n}] [-l {Y|y|N|n}] [-q] [-S system_file] module_name
```

**DESCRIPTION**

Without any option or with the **-S** option only, **kmsystem** prints the information on the **\$LOADABLE** and **\$CONFIGURATION** flags of all modules. The **-q** option may be used to print information about the specified module only. The **\$CONFIGURATION** flag for *module\_name* is set using the **-c** option, and the **\$LOADABLE** flag is set with the **-l** flag. When *module\_name* is specified on the command line, one or more of the **-c**, **-l**, or **-q** flags must also be specified.

**Options**

**-c** *value* Set the configuration status of *module\_name* to *value*. *value* must be **Y** or **y** to configure the module, or **N** or **n** to not configure it.

If the system file for the module (*/stand/system.d/module\_name*) exists but does not contain the **\$CONFIGURE** flag, then an error message is printed. Otherwise, the flag is set to *value*.

If the system file for the module does not exist, then the standard system file (see **-S** option) is searched. *module\_name* is added or removed from that system file according to *value*.

**-l** *value* Set the **\$LOADABLE** flag in the system file of *module\_name* to *value*. *value* must be **Y** or **y** to make the module loadable, or **N** or **n** to specify that it should be statically linked. If the system file for the module does not exist, **kmsystem** exits with an error. If the system file exists, but the **\$LOADABLE** flag is not present in the file, then the module is a static module, and **kmsystem** exits with an error.

**-q** Print the loadable and configuration flag information for *module\_name*. If the loadable information does not apply, then a **-** is printed.

**-S** *system\_file*

Specify the HP-UX system description file name. Users should specify the complete path to the file name; otherwise, **kmsystem** will search the current directory for the specified file. The default HP-UX system description file if the **-S** option is not specified is */stand/system*. This option is for backward compatibility.

**EXAMPLES**

To display the configuration and loadable status of the stape module:

```
/usr/sbin/kmsystem -q stape
```

To specify that the stape module should be statically linked:

```
/usr/sbin/kmsystem -l N stape
```

**NOTES**

System administrators are encouraged to use **kmsystem** and **kmtune** instead of editing system description files manually. File format of system description files are subject to change, and **kmsystem** provides compatibility in the event of a format change.

**RETURN VALUE**

Upon successful completion, **kmsystem** returns with one a 0; otherwise it returns with a 1.

**DIAGNOSTICS**

Output for queries is sent to stdout. Error messages are sent to stderr. Messages from **kmsystem** are self explanatory.

**FILES**

|                             |                                                           |
|-----------------------------|-----------------------------------------------------------|
| <i>/usr/conf/master.d/*</i> | Master configuration tables for kernel and kernel modules |
| <i>/stand/system</i>        | Default HP-UX system description file                     |

`/stand/system.d/*`

Kernel module system description files

**SEE ALSO**

kmtune(1M), master(4).

  
k

NAME

kmtune - query, set, or reset system parameters

SYNOPSIS

`/usr/sbin/kmtune [-l] [-d] [[-q name]...] [-S system_file]`  
`/usr/sbin/kmtune [-u] [[-s name{=|+}value]...] [[-r name]...] [-S system_file]`

DESCRIPTION

**kmtune** is used to query, set, or reset system parameters. **kmtune** displays the value of all system parameters when used without any options or with the **-S** or **-l** option. **kmtune** reads the master files and the system description files of the kernel and kernel modules. Information on the running kernel is gathered with the **tuneinfo()** system call.

Changes to kernel parameters generally do not take effect until the kernel has been rebuilt and the system has been rebooted. Some kernel parameters can be changed in the running kernel. **kmtune** with the **-u** option will alter these parameters and avoid the need to rebuild and reboot.

Options

The following options are recognized by **kmtune**:

- l** Print a detailed report. The **-l** option cannot be used with the **-r**, **-s**, or **-u** options.
- d** Print a brief difference report. Only parameters where the planned and current values are different are listed. If the planned value is an formula, it is NOT considered different from the current value. This option is useful to insure that only desired changes will be made on a subsequent invocation of **kmtune -u**. The **-d** option cannot be used with the **-r**, **-s**, or **-u** options.
- q name** Query the value of the specified system parameter.
- r name** Reset the value of a system parameter to the default.
- s name{=|+}value** Set the value to a system parameter. If the separator is an equal sign (=), the parameter is set to the value specified. If the separator is a plus sign (+), the parameter is incremented by the value specified. Negative values cannot be used with plus sign (+). The **name{=|+}value** format must not include spaces or tabs.
- S system\_file** Specify the HP-UX system description file name. If not specified, **/stand/system** is used as the default.
- u** Update the currently running kernel. This modifies the **-s** and **-r** options, to update the running kernel as well as the system files. The values specified must be numeric. The whole command will fail if any non dynamic parameters are being set. The **-s** and **-r** options are processed in the order they are listed. When incrementing a value with the **-s** option the kernel and the system files are set to the value in the system file plus the increment value.  
  
When used alone, or with **-S**, it updates all the dynamic parameters to match the numeric values stored in the system files. A warning is reported if any static parameters have differences. The order the parameters are listed in the system files is not important. The **-d** option can be used to look at the differences to get a preview of what a lone **-u** will do.  
  
Use of **-u** requires super user privileges.

If the **-q** query option is specified, **kmtune** displays the following format:

Brief report without **-l** option

| Parameter | Current Dyn Planned | Module Version |
|-----------|---------------------|----------------|
| =====     | =====               | =====          |
| name      | value Y/- value     | module version |

Detailed report with **-l** option

|            |                                   |
|------------|-----------------------------------|
| Parameter: | name                              |
| Current:   | current value from running kernel |
| Planned:   | planned value from system file    |

|                 |                                       |
|-----------------|---------------------------------------|
| <b>Default:</b> | <i>default value from master file</i> |
| <b>Minimum:</b> | <i>minimum</i>                        |
| <b>Module:</b>  | <i>module</i>                         |
| <b>Version:</b> | <i>version</i>                        |
| <b>Dynamic:</b> | <i>Yes/No</i>                         |

If the **-l** option is specified without the **-q** query option, a detailed report on all the parameters is displayed. The information between the parameters is separated by blank lines.

If the parameter has no minimum value specified in a master file, *minimum* will be displayed as '-'. If the parameter is not supplied by kernel modules, *module* will be displayed as '-'. In the brief listing the module and version fields are blank unless the parameter comes from a dynamically loadable kernel module.

The *Current* field is an integer read from the running kernel. Some parameters are not stored in the running kernel and are displayed as '-'. The *Value* string is a copy of the default value or what is stored in a system file.

The *Dynamic* field, in the **-l** output, is either Yes or No, and indicates whether the parameter can be changed using **-u**. This avoids rebuilding and rebooting. In the brief listing Y is used to indicate dynamic parameters.

If the **-s** set option is specified with an equal (=) separator and the minimum value of the parameter is described in a master file, the value range is checked. If the minimum value or the specified value is a formula, the check is not made. More restrictive range checking may be done by the kernel whenever **-u** is used.

If the **-s** set option with a plus (+) separator is specified and the original value is non numeric, an error is reported.

## NOTES

System administrators are encouraged to use **kmsystem** and **kmtune** instead of editing description files manually. File format of description files are subject to change, and **kmtune** is intended to provide compatibility in case of format change.

The **-u** option is new functionality that allows changes to values in the kernel that have always been constant before. Programs using **pstat()**, **getrlimit()**, or **sysconf()** may have been written with the previously correct assumption that the returned variables do not vary while the system is running. The use of **kmtune** to modify the running kernel may cause such programs to produce erroneous results or even abort.

When lowering system limits, processes that exceed the new limit will be "grandfathered". Such processes may not be able to reallocate a resource that had just been released and may fail in ways that were very unlikely before. After lowering **maxfiles**, a process may be in violation of the new limit. If it closes a file and then reopens it, the open may fail. See *setttune(2)*.

## RETURN VALUE

Upon completion, **kmtune** returns with one of the following exit values:

- 0 Successful.
- 1 Requested parameter is not found, the value is out of range, or the type of value is formula.
- 2 Syntax error.
- 3 Changes in static parameters were found when trying to dynamically update the running kernel.
- >3 Environmental error.

Results of query requests are sent to stdout. Error and warning messages are sent to stderr.

## EXAMPLES

```
# kmtune -q shmseg
Parameter          Current Dyn Planned      Module Version
=====
shmseg              120   Y   120
# kmtune -s shmseg=155
# kmtune -l -q shmseg
```

```
Parameter:      shmseg
Current:        120
Planned:        155
Default:        120
Minimum:        -
Module:         -
Version:        -
Dynamic:        Yes
# kmtune -u shmseg
shmseg has been set to 155 (0x9b).
# kmtune -r shmseg
# kmtune -d
Parameter      Current Dyn Planned  Module Version
=====
shmseg          155   Y   120
```

**FILES**

- /usr/conf/master.d/\* Master configuration tables for kernel and kernel modules
- /stand/system Default HP-UX system description file
- /stand/system.d/\* Kernel module system description files

**SEE ALSO**

kmsystem(1M), settune(2), tuneinfo(2), master(4).

k

**NAME**

kmupdate - update default kernel file and files associated with the kernel, or update specified kernel modules

**SYNOPSIS**

```
/usr/sbin/kmupdate [kernel_file]
```

```
/usr/sbin/kmupdate -M module_name [[-M module_name]...] [-i | -a]
```

**DESCRIPTION**

This command can be invoked to either update the kernel and the kernel modules associated with the kernel (i.e., */stand/dlkm*, which is the kernel function set directory), or to update only the specified kernel modules.

**Updating the Kernel and the Associated Kernel Function Set Directory**

The first form of **kmupdate** is used to initiate the move of the specified *kernel\_file* to the default kernel located at */stand/vmunix* during the next system shutdown or startup. The directory associated with the specified *kernel\_file*, the kernel function set directory, is also moved to */stand/dlkm* at the next shutdown or startup. If *kernel\_file* is not specified, */stand/build/vmunix\_test* is used as the *kernel\_file* to use for the update.

**kmupdate** is useful in cases where the kernel is built either by **config** without its **-u** option, or by **mk\_kernel** with its **-o** option (which specifies a kernel other than the default). In these cases the administrator should use **kmupdate** to update the kernel file and its associated kernel function set directory for the next shutdown or startup.

**NOTE:** Overwriting or replacing the kernel file and associated kernel function set directory using commands like **cp** or **mv** should be avoided.

**Options for Updating Specified Loadable Kernel Modules**

The second form of **kmupdate** supports the following options.

**-M *module\_name***

Update specified *module\_name* module. Without **-a** or **-i**, **kmupdate** will attempt to update *module\_name* immediately. If *module\_name* cannot be updated immediately, the module will be updated asynchronously, as described below.

**-i** When specified, **kmupdate** will only attempt an immediate update.

**-a** When specified, **kmupdate** will update asynchronously without attempting an immediate update.

**Immediate Update of Specified Kernel Modules**

**kmupdate** may be used for immediately updating the loadable image of a newly created kernel module, without a reboot. If the *module\_name* is loaded, **kmupdate** tries to unload it and, if the **-i** option is specified and the module cannot be unloaded, **kmupdate** exits with an error. If the kernel module was either not loaded or successfully unloaded, **kmupdate** checks if it is registered, and if so, unregisters the module. If the kernel module cannot be unregistered, **kmupdate** exits with an error if **-i** is specified; otherwise the module will be updated asynchronously. If the unregistration succeeds, **kmupdate** overlays the existing loadable image of the module with the newly generated image. It then registers the module with the latest registry information and performs module type specific initialization, if required. If the module was loaded originally, **kmupdate** reloads the module before exiting.

**Asynchronous Update of Specified Kernel Modules**

If the **-a** option is specified, the module will be updated asynchronously without first attempting an immediate update. An asynchronous update occurs at shutdown. When the system shuts down, the module's loadable image is updated. The module is registered when the system is restarted.

**RETURN VALUE**

**kmupdate** returns 0 upon normal completion, and 1 if an error occurred.

**DIAGNOSTICS**

Messages that notify an update is successful are sent to stdout. Error messages are sent to stderr.

**FILES**

`/stand/vmunix`

Default kernel file

`/stand/dlkm`

Default kernel function set directory

**SEE ALSO**

`mk_kernel(1M)`, `config(1M)`.

k



**NAME**

krsd - kernel registry services daemon

**SYNOPSIS**

**krsd -l**

**krsd [-i] [-d *seconds*]**

**DESCRIPTION**

**krsd** saves data, flagged as persistent in the core kernel KRS tree, to files on disk. These files are read when the system is booted, in order to restore the aforementioned persistent data.

When running in daemon mode, **krsd** sleeps, waking periodically to check if any persistent data has changed since the last save. If persistent data has changed, the new state of the data is saved to disk. Otherwise, **krsd** sleeps for another period of time and then repeats the process.

When running in *one time* (non-daemon) mode, **krsd** unconditionally saves persistent data to disk and then exits.

Normally, **krsd** is executed in daemon mode by *init*(1M). The */etc/inittab* entry for **krsd** will cause **krsd** to be respawned automatically if it is terminated.

**krsd** uses the **syslog** message logging facility to log all **krsd** activity.

Persistent KRS data is maintained in files based on the **class** of the data in question. Currently, the following **classes** are recognized:

**system specific**

Data that applies to the system in general, regardless of what kernel is booted. This data is saved in the file: **system.krs**.

**kernel specific**

Data specific to a given bootable kernel. This data is saved in the file: *kernel\_name.krs*. Where *kernel\_name* is the full path name of the kernel in question, with slashes replaced by underscores. NOTE: This naming scheme is subject to change in the future.

These files are maintained in the following directories:

**/stand/krs**

The primary KRS directory. This is the directory from which the files are loaded when the system is booted. It contains the most recent, consistent copy of persistent KRS data.

**/stand/krs\_lkg**

This directory contains the *last known good* copies of the KRS data. The files in **/stand/krs** are copied to this directory when the system successfully reaches the **sysinit** init state. They are loaded, during boot, if the files in **/stand/krs** are missing or corrupt.

This scheme ensures that the system will be able to boot to **sysinit**, in the event the primary KRS files are found to be corrupt. Once the **sysinit** state is reached, more recent versions of the KRS data can be restored to **/stand/krs** and the system rebooted.

**/stand/krs\_tmp**

This directory contains temporary copies of the KRS data files. Data is first saved to files in this directory. Once the save is complete, the files are linked to **/stand/krs** and unlinked from **/stand/krs\_tmp**.

**Options**

**krsd** recognizes the following options:

- l** *One time* mode, unconditionally save persistent data to disk, then exit.
- i** Run from **inittab**, parent does not spawn a child nor exit. Required if **respawn** from **inittab** is to work properly.
- d *seconds*** Set the delay time to *seconds* seconds. This is the time interval between attempted saves of persistent data. The default interval is 300 seconds.

**AUTHOR**

**krsd** was developed by Hewlett-Packard Company.

**FILES**

|                               |                            |
|-------------------------------|----------------------------|
| <code>/stand/krs/*</code>     | Primary KRS files          |
| <code>/stand/krs_lkg/*</code> | Last known good KRS files. |
| <code>/stand/krs_tmp/*</code> | Temporary KRS files.       |
| <code>/dev/devkrs</code>      | Pseudo driver.             |

**SEE ALSO**

`krs_flush(1M)`, `krs(5)`.

**NAME**

krs\_flush - flush kernel registry services data to disk

**SYNOPSIS**

**krs\_flush**

**DESCRIPTION**

**krs\_flush** causes persistent, in core, kernel KRS data to be flushed to files on disk.

This command signals the KRS daemon, *krsd*(1M), causing it to save the KRS data to disk. The data will only be saved if it has changed from the time of the last save.

**AUTHOR**

**krs\_flush** was developed by Hewlett-Packard Company.

**FILES**

|                         |                           |
|-------------------------|---------------------------|
| <b>/stand/krs/*</b>     | Primary KRS files         |
| <b>/stand/krs_lkg/*</b> | Last known good KRS files |
| <b>/stand/krs_tmp/*</b> | Temporary KRS files       |
| <b>/dev/devkrs</b>      | Pseudo driver             |

**SEE ALSO**

krsd(1M), krs(5).

**NAME**

lanadmin - local area network administration program

**SYNOPSIS**

```
/usr/sbin/lanadmin [-e] [-t]
/usr/sbin/lanadmin [-a] [-A station_addr] [-b] [-B on|off ] [-m] [-M mtu_size] [-R] [-s]
[-S speed] [-x options] [-X options] PPA
```

**DESCRIPTION**

The **lanadmin** program administers and tests the Local Area Network (LAN). For each interface card, it allows you to:

- Display and change the station address.
- Display and change the 802.5 Source Routing options (RIF).
- Display and change the maximum transmission unit (MTU).
- Display and change the speed setting.
- Clear the network statistics registers to zero.
- Display the interface statistics.
- Reset the interface card, thus executing its self-test.

For operations other than display, you must have superuser privileges.

**lanadmin** reads commands from standard input, writes prompts and error messages to standard error, and writes status information to standard output. When the program is run from a terminal, the interrupt key (usually ^C) interrupts a currently executing command; the eof key (usually ^D) terminates the program.

**lanadmin** operates in two modes: Menu Mode (see the first SYNOPSIS line) and Immediate Mode (see the second SYNOPSIS line). If at least one **-aAbBmMRsS** option is supplied, **lanadmin** executes in Immediate Mode. Otherwise, it executes in Menu Mode.

**NOTE:** **lanadmin** replaces the now obsolete **landiag** command beginning at 10.0.

**Options and Arguments**

**lanadmin** recognizes the following Immediate Mode options and arguments. At least one **-aAbBmMRsS** option and the *PPA* argument must be supplied.

|                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>PPA</i>             | The Physical Point of Attachment (PPA) number of the LAN interface. This argument is ignored if none of the <b>-aAbBmMRsS</b> options are used (Menu Mode). Any options specified after <i>PPA</i> are ignored. Appropriate values can be displayed with the <b>lanscan</b> command (see <i>lanscan(1M)</i> ).                                                                                                                                                                                                                                                                                                                     |
| <b>-a</b>              | Display the current station address of the interface corresponding to <i>PPA</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>-A station_addr</b> | Set the new station address of the interface corresponding to <i>PPA</i> . The <i>station_addr</i> must be entered in hex format with a '0x' prefix. You must have superuser privileges. When <i>station_addr</i> is <b>DEFAULT</b> , the factory default physical address will be restored.<br><br>WARNING: To ensure the interface and the system work correctly, the interface <b>MUST</b> be brought down before setting the new station address. After the new station address is set, the interface should be brought up in order to be functional. See <i>ifconfig(1M)</i> for bringing down and bringing up the interface. |
| <b>-b</b>              | Display the current 802.5 source routing option for the interface corresponding to <i>PPA</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>-B on off</b>       | Turn the 802.5 source routing option "on" or "off" for the interface corresponding to <i>PPA</i> . The default value for HP devices is "on". You must have superuser privileges.                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>-m</b>              | Display the current MTU size of the interface corresponding to <i>PPA</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>-M mtu_size</b>     | Set the new MTU size of the interface corresponding to <i>PPA</i> . The <i>mtu_size</i> value must be within the link specific range. You must have superuser privileges.                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>-R</b>              | Reset the MTU size of the interface corresponding to <i>PPA</i> to the default for that link type. You must have superuser privileges.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

- s** Display the current link speed setting of the interface corresponding to *PPA*.
- s *speed*** Set the new link speed setting of the interface corresponding to *PPA*. You must have superuser privileges.
- x *options*** Get and display driver specific *options* of the interface corresponding to *PPA*.
- x *options*** Set driver specific *options* of the interface corresponding to *PPA*. You must have superuser privileges.

**lanadmin** recognizes the following Menu Mode options. They are ignored if they are given with an Immediate Mode option.

- e** Echo the input commands on the output device.
- t** Suppress the display of the command menu before each command prompt. This is equivalent to the Test Selection Mode **terse** command. The default is **verbose**.

### Immediate Mode

In Immediate Mode, you can display the station address, source routing option, MTU size, and link speed of LAN interface *PPA*. For certain interfaces, if you have superuser privileges you can also modify the station address, source routing option, MTU size, and link speed. See "Options and Arguments" above.

### Menu Mode

In Menu Mode, you can select an interface card, display statistics for the selected card, reset the card, and clear the statistics registers.

Menu Mode accepts either complete command words or unique abbreviations, and no distinction is made between uppercase and lowercase letters in commands. Multiple commands can be entered on one line if they are separated by spaces, tabs, or commas.

### Test Selection Mode Menu

This menu is entered when Menu Mode is first selected. The available Test Selection Mode commands are:

- lan** Select the LAN Interface Test Mode menu.
- menu** Display the Test Selection Mode command menu.
- quit** Terminate the **lanadmin** program.
- terse** Suppress the display of command menus.
- verbose** Restore the display of command menus.

### LAN Interface Test Mode Menu

The following commands are available:

- clear** Clear the LAN interface network statistics registers to zero. You must have superuser privileges.
- display** Display the RFC 1213 MIB II statistics. Depending on the link, the type-specific MIB statistics may also be displayed. For instance, for Ethernet links, the RFC 1398 Ethernet-like statistics are displayed.
- end** Return **lanadmin** to Test Selection Mode.
- menu** Display the LAN Interface Test Mode command menu.
- ppa** Prompt for a *PPA* that corresponds to a LAN interface card. It defaults to the first LAN interface encountered in an internal list. Appropriate values can be displayed with the **lanscan** command (see **lanscan(1M)**).
- quit** Terminate the **lanadmin** program.
- reset** Reset the local LAN interface card, causing it to execute its self-test. Local access to the network is interrupted during execution of **reset**. You must have superuser privileges.
- specific** Display and execute commands from a driver **specific** menu.

**WARNINGS**

Changes made to an interface's station address or mtu interactively with the **lanadmin** command will not be preserved between system reboots. A user must modify the initialization configuration files for this feature, either manually editing configuration files or through the **SAM** interface.

**AUTHOR**

**lanadmin** was developed by HP.

**SEE ALSO**

netstat(1), lanscan(1M), linkloop(1M), ping(1M), lan(7).

IETF Requests for Comments: RFC 1213, RFC 1398.

**NAME**

lanscan - display LAN device configuration and status

**SYNOPSIS**

**lanscan** [-aimnpqv] [*system* [*core*]]

**DESCRIPTION**

**lanscan** displays the following information about each LAN device that has software support on the system:

- Hardware Path.
- Active Station Address (also known as Physical Address).
- Card Instance Number
- Hardware State.
- Network Interface "NamePPA". The Network Interface "Name" and the "PPA" (Physical Point of Attachment) number are concatenated together. A single hardware device may have multiple "NamePPA" identifiers, which indicates multiple encapsulation methods may be supported on the device. For Ethernet/IEEE 802.3 links, the "Name" **lan** is used to designate Ethernet encapsulation, and **snap** for IEEE 802.3 encapsulation. For other links (FDDI, Token Ring), only the **lan** encapsulation designation is used.
- Network Management ID.
- MAC Type.
- HP DLPI Supported. Indicates whether or not the lan device driver will work with HP's Common Data Link Provider Interface.
- DLPI Major Number.
- Extended Station Address for those interfaces which require more than 48 bits. This is displayed only when the **-v** option is selected.
- Encapsulation Methods that the Network Interface supports. This is displayed only when the **-v** option is selected.

The arguments *system* and *core* allow substitution for the default values **/stand/vmunix** and **/dev/kmem**.

**Options**

**lanscan** recognizes the following command-line options:

- a** Display station addresses only. No headings.
- i** Display interface names only. No headings.
- m** Display MAC types only. No headings.
- n** Display Network Managements IDs only. No headings.
- p** Display PPA numbers only. No headings.
- q** Same as **-p**, except link aggregate PPA's will be followed by a list of LAN interface PPA's that are configured in the corresponding link aggregate. No headings.
- v** Verbose output. Two lines per interface. Includes displaying of extended station address and supported encapsulation methods.

**WARNINGS**

**lanscan** does not display information about LAN devices that do not have software support such as LAN interface cards that fail to bind properly at boot-up time.

**AUTHOR**

**lanscan** was developed by HP.

**SEE ALSO**

ifconfig(1M), ioscan(1M), lanadmin(1M), linkloop(1M), lan(7).

**NAME**

libcadmin - libc administration command

**SYNOPSIS**

/usr/sbin/libcadmin

**DESCRIPTION**

The **libcadmin** command is used to perform administrative functions for **libc**. Currently the only function that the command performs is to replace the shared 32-bit PA-RISC2.0 library in **/usr/lib/libc.2** with the shared 32-bit PA-RISC1.1 version of that library.

There are no required arguments or options used with this command. The **/usr/lib/libc.a** archive library and the **libc.sl** symbolic link are unaffected by the **libcadmin** command.

During the system installation process, the shared 32-bit PA-RISC1.1 library is copied into **/usr/lib/pa11\_32**. This library is the same as the PA-RISC2.0 library except that it was built with different compiler options. The PA-RISC2.0 library provides much higher performance than the PA-RISC1.1 library. However, if the system administrator desires to have the PA-RISC1.1 library installed instead, **libcadmin** allows the system administrator to accomplish this task. When **libcadmin** is executed, the PA-RISC2.0 library is stored in **/usr/lib/pa20\_32** and the PA-RISC1.1 library is copied into **/usr/lib/libc.2**.

It is recommended that the system be re-booted after executing the **libcadmin** command so that applications which are linked shared against **libc** will use the shared 32-bit PA-RISC1.1 version of that library.

Once the command is executed, the only way to put the PA-RISC2.0 library back in place is to re-install the operating system.

The **libcadmin** command can only be executed by a user with superuser privileges.

**ERRORS**

Multiple executions of the command will result in an error message indicating the PA-RISC1.1 library is already installed.

Executing this command on a PA-RISC1.1 machine will result in an error message indicating the PA-RISC1.1 library is already installed.

**RETURN VALUES**

Upon successful completion, **libcadmin** returns zero (0). Otherwise a one (1) is returned.

**AUTHOR**

**libcadmin** was developed by HP.



**NAME**

`link`, `unlink` - execute `link()` and `unlink()` system calls without error checking

**SYNOPSIS**

`/usr/sbin/link file1 file2`

`/usr/sbin/unlink file`

**DESCRIPTION**

The `link` and `unlink` commands perform their respective system calls (`link()` or `unlink()`) on their arguments, abandoning most error checking.

These commands can be executed only by users who have appropriate privileges.

**EXTERNAL INFLUENCES****Environment Variables**

`LC_MESSAGES` determines the language in which messages are displayed.

If `LC_MESSAGES` is not specified in the environment or is set to the empty string, the value of `LANG` is used as a default for each unspecified or empty variable. If `LANG` is not specified or is set to the empty string, a default of "C" (see *lang(5)*) is used instead of `LANG`.

If any internationalization variable contains an invalid setting, `link` behaves as if all internationalization variables are set to "C". See *environ(5)*.

**International Code Set Support**

Single- and multi-byte character code sets are supported.

**RETURN VALUE**

`link` and `unlink` return the following values:

- 0    Operation successful.
- 1    Input syntax error.
- 2    The `link()` or `unlink()` call failed.

**WARNINGS**

If a directory that contains files other than `.` and `..` is unlinked, the files become orphans, unless they are also linked by some other directory.

Not all file systems permit linking to directories.

**SEE ALSO**

`ln(1)`, `rm(1)`, `link(2)`, `unlink(2)`.

**STANDARDS CONFORMANCE**

`link`: SVID2, SVID3

`unlink`: SVID2, SVID3

**NAME**

linkloop - verify LAN connectivity with link-level loopback

**SYNOPSIS**

**linkloop** [-i *PPA*] [-n *count*] [-r *rif*] [-s *size*] [-t *timeout*] [-v] *linkaddr* ...

**DESCRIPTION**

The **linkloop** command uses IEEE 802.2 link-level test frames to check connectivity within a local area network (LAN).

*linkaddr* is the hardware station address of a remote node. Several addresses can be specified at one time.

**linkloop** tests the connectivity of the local node and the remote node specified by each hardware station address. The hardware station address of a remote node can be found by executing **lanscan** on the remote node. This hardware station address is usually represented as a hexadecimal string prefixed with **0x**. It can also be represented as a octal string prefixed with **0** or as a decimal string. The hardware station address must not be a multicast or broadcast address.

**Options**

**linkloop** recognizes the following options:

- i *PPA* Specify the *PPA* to use. If this option is omitted, **linkloop** uses the first *PPA* it encounters in an internal data structure.
- n *count* Set the number of frames to transmit. If *count* is 0, **linkloop** transfers frames indefinitely until an interrupt signal (defined by the user shell) is received. The default value for *count* is 1.
- r *rif* Specify the particular bridge route over which token ring packets should be delivered. *rif* is the *routing information field* used for token-ring networks. Its value is given as an even number of hexadecimal bytes separated by colons, up to a maximum of 16 bytes.
- s *size* Set the size in bytes of the data message to send. The maximum data size is dependent on the type of LAN link being used. The default value is the maximum data byte count that can be used for the particular link.
- t *timeout* Set the amount of time in seconds to wait for a reply from the remote node before aborting. If *timeout* is 0, **linkloop** waits indefinitely for a reply. The default value is 2 seconds.
- v Set the verbose option. In addition to the regular summary of test results, this option displays more extensive error information. If there are header or length errors, appropriate messages are displayed. All verbose output is preceded by the number of replies accepted before an error occurred.

**Connectivity Test Results**

**linkloop** aborts upon receipt of an interrupt signal. If aborted, the current results are printed.

**linkloop** prints the result of the link-level connectivity test. If the test fails, it prints a summary of the test and indicates the type of error. The possible messages are:

**address has bad format**

An incorrect hardware station address was entered on the command line.

**address is not individual**

The station address entered on the command line is either a multicast or broadcast address.

**frames sent**

Total number of frames sent.

**frames received correctly**

Total number of frames received without errors.

**frames with length error**

Received frame length does not match transmitted frame length. If the verbose option is set, the length received is printed.

**frames with data error**

Received frame does not match transmitted frame.

**frames with header error**

Number of frames received containing unexpected frame header information. Either the source address does not match the remote address, the destination address does not match the local address, or the control field is not the TEST **frame control field**. These frames are ignored. **linkloop** continues to try to receive the reply frame until the **read** operation times out.

**reads that timed out**

Count of how many **read** operations timed out before the reply was received.

**DIAGNOSTICS****illegal count parameter**

The *count* specified in the **-n** option is a negative integer, or the number specified is too large for the local computer.

**illegal timeout parameter**

The *timeout* specified in the **-t** option is a negative integer, or the value specified multiplied by 1000 is too large for the local computer.

**illegal size parameter**

The *size* specified in the **-s** option is not in the range from 0 to the maximum link data size. Remember that the maximum link data size can vary in value for different LAN connection types. The current MTU can be obtained with the **linkloop** command.

**No valid interface associated with PPA**

The *PPA* specified in the **-i** option is not a valid PPA.

**Unable to open device file /dev/dlpi**

Device file **/dev/dlpi** does not exist.

**invalid rif parameter**

The *rif* value in the **-r** option is invalid.

**rif parameter too long**

The number of bytes in *rif* in the **-r** option exceeded 16, which is the maximum allowed.

**rif parameter length must be even**

The number of bytes in *rif* in the **-r** option is odd. The number of bytes must be even.

**AUTHOR**

**linkloop** was developed by HP.

**SEE ALSO**

lanadmin(1M), lanscan(1M), lan(7).

**NAME**

localedef - generate a locale environment

**SYNOPSIS**

```
localedef [-cenvw] [-C compiler_options] [-L loader_options]
          [-m method_file] [-f charmap_file] [-i locale_definition] locale_name
```

**DESCRIPTION**

**localedef** sets up the language environment for the named locale. **localedef** reads a **locale definition** file (see *localedef(4)* for a detailed description) from standard input (default) or from *locale\_definition* file, creates a locale file with the same name as specified for the *locale\_name* parameter, and optionally installs this locale in the appropriate directory. Installation of public locales (those accessible to all users) requires appropriate privileges. Creation of locales (both private and public) requires access to the ANSI C compiler.

**Options**

**localedef** recognizes the following options:

- c Create permanent output even if warning messages have been generated.
- e Generate 64-bit locale in addition to the 32-bit locale. This is the default on a 64-bit operating system and is included to allow cross platform development.
- n (noinstall) Create the locale file in the current directory.
- v (verbose) Generate as many diagnostic messages as possible.
- w Generate additional warning messages for duplicate definitions and ellipses use in the LC\_COLLATE category.
- f *charmap\_file*  
If **locale definition** file contains symbolic names (of the form <name>) use *charmap\_file*. See *charmap(4)* for a description of the format of a *charmap\_file*.
- i *locale\_definition*  
Use *locale\_definition* file as input, instead of standard input (default).
- m *method\_file*  
Use the specified *method\_file* to overwrite use of default methods in processing the **locale definition**.
- C *compiler\_options*  
Specify additional compiler options to be applied in compiling the locale. See *cc\_bundled(1)* for a complete list of options. Use with care on a 64-bit operating system since the additional default option includes +DA2.0W.
- L *loader\_options*  
Specify additional loader options to be applied in linking the locale. See *ld(1)* for a complete list of options.
- locale\_name* This argument is required, and identifies the name of the language following the naming convention of the **LANG** environment variable (see *environ(5)*):  
*language[\_territory][.codeset]*

The following is a brief description of the components that make up a locale. For a complete description of the form and syntax of a **locale definition** file, see *localedef(4)*. For a complete description of the form and effects of a charmap file, see *charmap(4)*.

Six categories of data in the **locale\_name** file are recognized by *setlocale(3C)*, and make up a language definition:

- LC\_COLLATE** Information in this category affects behavior of regular-expressions and NLS string-collation functions.
- LC\_CTYPE** Information in this category affects behavior of character classification and conversion functions.
- LC\_MONETARY** Information in this category affects behavior of functions that handle monetary values.

|                    |                                                                                                                                 |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------|
| <b>LC_NUMERIC</b>  | Information in this category affects handling of the radix character in formatted-input/output and string-conversion functions. |
| <b>LC_TIME</b>     | Information in this category affects behavior of time-conversion functions.                                                     |
| <b>LC_MESSAGES</b> | This category contains information affecting interpretation of yes/no responses.                                                |

A **locale definition** file also consists of six categories. The beginning of each category is identified by a **category tag** having the form **LC\_category** where *category* is one of the following: **CTYPE**, **COLLATE**, **MONETARY**, **NUMERIC**, **TIME**, or **MESSAGES**. The end of each category is identified by a tag consisting of the word **END** followed by a space and the category identifier; for example, **END LC\_COLLATE**. Categories can appear in any order in the **locale definition** file. At least one category specifications is required. If a category is not specified, **setlocale()** sets up the default “C” locale for that category (see **setlocale(3C)** and **lang(5)**).

Each category is composed of one or more statements. Each statement begins with a keyword followed by one or more expressions. An expression is a set of well-formed metacharacters, strings, and constants. **localedef** also recognizes comments and separators.

More than one definition specified for each category constitutes a hard error (causes **localedef** to exit without generating a locale). Any category can be specified by the keyword **copy** followed by the name of a valid locale. This causes the information for the category to be identical to that in the named locale. Note that the **copy** keyword, if used for a category, must be the first and only keyword following the category tag.

A methods file is used to create locales for user-specific character encoding schemes.

### Operating System Requirements

For cross platform development and development on a 64-bit operating system several requirements must be observed. Both the 32-bit and 64-bit method libraries must exist. In the case of the 64-bit shared library it must be in the directory **pa20\_64** under the location where the 32-bit library is located. When the **-e** option is specified, or when executing on a 64-bit operating system, the resulting locale is placed in the directory **pa20\_64** under the current working directory unless the **install** option has been specified.

### NOTE

Locale built for one system cannot be used on other systems.

### EXTERNAL INFLUENCES

#### Environment Variables

**LANG** determines the locale to use when neither **LC\_ALL** or the other category variables specify a locale.

**LC\_ALL** determines locale to be used. It overrides any values specified by **LANG** or any other **LC\_\*** variables.

**LC\_COLLATE** and **LC\_CTYPE** have no effect on the processing of **localedef**, which behaves as if these two variables were set to the C locale.

**LC\_MESSAGES** determines the language in which messages are displayed.

### International Code Set Support

Single- and multi-byte character code sets are supported.

### RETURN VALUE

**localedef** returns the following values:

- 0 No errors occurred and the locale was successfully created.
- 1 Warnings occurred and the locale was successfully created.
- 2 The locale specification exceeded implementation limits or the coded character set used is not supported.
- >3 Warnings or errors occurred, and no output was generated.

### AUTHOR

**localedef** was developed by OSF and HP.

### FILES

**/usr/lib/nls/config**  
**/usr/lib/nls/loc/src**

```
/usr/lib/nls/loc/charmaps  
/usr/lib/nls/loc/methods  
/usr/lib/nls/loc/pa20_64/methods  
/usr/lib/nls/loc/locales/language[_territory][.codeset]
```

**SEE ALSO**

locale(1), localedef(4), charmap(4), setlocale(3C), environ(5).

**STANDARDS CONFORMANCE**

localedef: XPG4, POSIX.2

**NAME**

lockd - network lock daemon

**SYNOPSIS**

`/usr/sbin/rpc.lockd [-l log_file] [-t timeout] [-g graceperiod]`

**DESCRIPTION**

**lockd** is an RPC server that processes NFS file locking requests from the local kernel or from another remote lock daemon. **lockd** forwards lock requests for remote data to the server site's lock daemon through the RPC/XDR package (see *rpc(3C)*). **lockd** then requests the status monitor daemon, **statd** for monitor service (see *statd(1M)*). The reply to the lock request is not sent to the kernel until the status daemon and the server site's lock daemon have replied.

If either the status monitor or server site's lock daemon is unavailable, the reply to a lock request for remote data is delayed until all daemons become available.

When a server recovers, it waits for a grace period for all NFS client-site **lockd**s to submit reclaim requests. Client-site *lockd*s are notified by the **statd** of the server recovery, and promptly resubmit previously granted lock requests. If a **lockd** fails to secure a previously granted lock at the server site, the **lockd** sends a **SIGLOST** to the process holding that lock.

**Options**

**lockd** recognizes the following options and command-line arguments:

- |                              |                                                                                                                                                                                                                                                                                                                                                                        |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-l</b> <i>log_file</i>    | <b>lockd</b> Log any errors to the named log file <i>log_file</i> . Errors are not logged if the <b>-l</b> option is not specified.<br><br>Information logged to the file includes date and time of the error, host name, process ID and name of the function generating the error, and the error message.                                                             |
| <b>-t</b> <i>timeout</i>     | <b>lockd</b> uses <i>timeout</i> (seconds) as the interval instead of the default value (10 seconds) to retransmit a lock request to the remote server. Note that changing this value also changes the value for grace period duration.                                                                                                                                |
| <b>-g</b> <i>graceperiod</i> | <b>lockd</b> uses $[1 + (\textit{graceperiod} / \textit{timeout})] \times \textit{timeout}$ (seconds) as the grace period duration instead of the default value ( $5 \times \textit{timeout}$ seconds). If both <b>-t</b> and <b>-g</b> are specified, the <b>-t</b> should appear first since the grace period duration is dependent on the value of <i>timeout</i> . |

**AUTHOR**

**lockd** was developed by Sun Microsystems, Inc., and HP.

**SEE ALSO**

*fcntl(2)*, *lockf(2)*, *signal(2)*, *statd(1M)*.

**NAME**

logins - display system and user login data

**SYNOPSIS**

**logins** [-admpstux] [-g *groups*] [-l *logins*]

**DESCRIPTION**

**logins** displays data concerning system and user logins. The format and content of the output is controlled by command options and may include: system or user login, user ID number, `/etc/passwd` comment field value (e.g., user name, etc...), primary group name, primary group ID, supplementary group names, supplementary group IDs, home directory, login shell, user security level, user audit events, and password aging parameters. The default data is: login, user ID, primary group name, primary group ID, and `/etc/passwd` comment field value. Output is sort by user ID, with user logins following system logins. The default output consists of login, user ID, primary group, primary group ID and comment field formatted into columns.

The following options are available to this command:

- a Displays two account expiration fields. The fields show how long the account can be unused (in days) before it becomes inactive and the date the account will expire.
- d Display logins with duplicate UIDs.
- m Show multiple group membership data.
- o Display with alternate format of one line of colon separated fields.
- p Display logins with no passwords
- s Display all system logins
- t Sort output by login rather than UID.
- u Display all user logins.
- x Display extended information about selected users. This extended information includes home directory, login shell and password aging data, each on its own line. Password information consists of password status (PS for valid password, LK for locked and NP for no password) and, if a password is present, date of last change, required number of days between changes, and number of days allowed between changes. In the case of non-trusted systems, the date of last change will be the latest Thursday since the change.
- g *groups*  
Display all users belonging to *groups*, sorted by login. A comma separated list specifies multiple groups.
- l *logins*  
Display the requested *logins*. A comma separated list specifies multiple logins.

Multiple options may be used. Any login matching any of the criteria will be displayed. A login will be displayed only once, even if it meets multiple criteria.

**EXAMPLES**

**logins** List all logins in default format.

**logins -p -d** List all logins that have no password or have a duplicate UID in default format.

**logins -s -o** List all system logins in the alternate format.

**FILES**

`/etc/passwd` HP-UX password file.

`/etc/group` HP-UX group file.

**SEE ALSO**

listusers(1), passwd(1), group(4), passwd(4).

**STANDARDS COMPLIANCE**

logins: SVID3



**NAME**

logprint - allows you to format and print the contents of the ARMServer log files

**SYNOPSIS**

```
logprint [-d log_directory_name] [-s start_time] [-e stop_time] [-t record_type...]
        [-a array_serial_number]
```

**DESCRIPTION**

logprint allows you to select and print the contents of the various log files maintained by ARMServer.

**Options**

logprint supports the following options:

- none** Print all log records, except performance records, from the default ARMServer log directory.
- a *array\_serial\_number*** Limit the records printed to those associated with the disk array identified by *array\_serial\_number*. The default is to print the records for all disk arrays.
- d *log\_directory\_name*** Specify the directory that contains the ARMServer log files. If not specified, the default is `/var/opt/hparray/log`.
- e *stop\_time*** Specify the latest log record to print. Any log records with a time later than *stop\_time* will not be printed. The default is the time of the latest log record.  
The format for entering time is *mmddhhmm[yy]*:  
  - mm* Month (01-12)
  - dd* Day (01-31)
  - hh* Hour (00-23)
  - mm* Minute (00-59)
  - yy* Year (optional)
- s *start\_time*** Specify the earliest log record to print. Any log records with a time earlier than *start\_time* will not be printed. The default is the time of the oldest log record. See above for the format for entering time.
- t *record\_type*** Specify the type(s) of log records to print. If not specified, the default log records include usage, disk, controller, and change.  
Valid log types include the following:
  - usage** System usage log
  - disk** Disk error log
  - ctrlr** Controller error log
  - change** System change log
  - perf** Performance log

**RETURN VALUE**

logprint returns the following values:

- 0 Successful completion.
- 1 An error in execution (I/O, subsystem, security, etc.) occurred.
- 2 An error in command syntax occurred: for example, an unknown command-line option was passed.

**EXAMPLES**

Print all log records that occurred after 0800 on 16 April:

```
logprint -s 04160800
```

Print only the log records for disk array serial number 00786b5c0000. Further limit the log records to usage log and disk log entries that occurred before 0800 on April 16:

```
logprint -e 04140800 -t usage -t disk -a 00786b5c0000
```

**LOG OUTPUT DATA**

logprint outputs the following data fields:

**Controller timestamp**

A number indicating the elapsed time since the disk array was commissioned. The number is incremented each millisecond the disk array is operating. This number does not relate in any way to system time and is primarily intended to show the temporal relationship between the log entries on a disk array. If there are multiple disk arrays on the system, each will maintain its own unique timestamp.

**Event code**

A brief description of the event.

**Event count**

Indicates the number of times the event occurred.

**Component ID**

Indicates which disk array controller was involved (6 = controller Y, 7 = controller X).

**FRU ID**

The code for the hardware assembly involved. The FRU codes map to the disk array hardware assemblies as follows:

- 0 Disk in slot A1
- 1 Disk in slot B1
- 2 Disk in slot A2
- 3 Disk in slot B2
- 4 Disk in slot A3
- 5 Disk in slot B3
- 6 Disk in slot A4
- 7 Disk in slot B4
- 8 Disk in slot A5
- 9 Disk in slot B5
- 10 Disk in slot A6
- 11 Disk in slot B6
- 129 Reporting controller, or No FRU
- 130 First installable SIMM on X controller
- 131 Second installable SIMM on X controller
- 132 First installable SIMM on Y controller
- 133 Second installable SIMM on Y controller
- 134 Other (non-reporting) controller
- 135 Battery
- 136 Fan 1
- 137 Fan 2
- 138 Fan 3
- 139 Power Supply 1
- 140 Power Supply 2
- 141 Controller X
- 142 Controller Y
- 192 Internal SCSI Bus 0
- 193 Internal SCSI Bus 1
- 194 Internal SCSI Bus 2
- 195 Internal SCSI Bus 3

**SECURITY CONFIGURATION**

This command is modified for all security configurations.

**Security Behavior/Restrictions**

Use of this command is restricted to authorized users only.

**Command Authorizations**

This command requires the **sysadmin** authorization to successfully execute.

**Privileges**

The command has been modified to support least privilege. The potential privileges possessed by the command and their uses include:

**allowdacread** This privilege is raised to provide discretionary read access to the devices.

**allowmacread** This privilege is raised to provide mandatory read access to the devices.  
**filesysops** This privilege is raised to allow the *mknod(2)* system call to succeed.

**AUTHOR**

**logprint** was developed by HP.

**SEE ALSO**

ARMServer(1M), arraycfg(1M), arraydsp(1M), arrayfmt(1M), arraylog(1M), arraymgr(1M), arrayrbld(1M), download(1M), drivetest(1M), dteststat(1M).

**NAME**

lpadmin - configure the LP spooling system

**SYNOPSIS**

```
/usr/sbin/lpadmin -pprinter [options]
/usr/sbin/lpadmin -xdest
/usr/sbin/lpadmin -d[dest]
```

**DESCRIPTION**

**lpadmin** configures LP spooling systems to describe printers, classes and devices. It is used to add and remove destinations, change membership in classes, change devices for printers, change printer interface programs, and to change the system default destination. **lpadmin** cannot be used when the LP scheduler, *lpsched*(1M), is running, except where noted below.

Exactly one of the **-p**, **-x** or **-d** options must be present for every legal invocation of *lpadmin*.

- pprinter** Names a *printer* to which all of the *options* below refer. If *printer* does not exist, it will be created.
- xdest** Removes destination *dest* from the LP system. If *dest* is a printer and is the only member of a class, the class is deleted, too. No other *options* are allowed with **-x**.
- d[dest]** Makes existing destination *dest* the new system default destination. If *dest* is not supplied, there is no system default destination. This option can be used when *lpsched*(1M) is running. No other *options* are allowed with **-d**.

The following *options* are only useful with **-p** and can appear in any order. For ease of discussion, the printer is referred to below as printer *P*.

- cclass** Inserts printer *P* into the specified *class*. *class* is created if it does not already exist.
- epprinter** Copies an existing *printer's* interface program to be the new interface program for printer *P*.
- gpriority** Sets the default priority for printer *P* associated with *lp*(1). If omitted, the default priority is set to 0.
- h** Indicates that the device associated with printer *P* is hardwired. This *option* is assumed when creating a new printer unless the **-l** option is specified.
- iinterface** Establishes a new interface program for printer *P*. *interface* is the pathname of the new program.
- l** Indicates that the device associated with printer *P* is a login terminal. The LP scheduler (see *lpsched*(1M)) disables all login terminals automatically each time it is started. Before re-enabling printer *P*, its current *device* should be established using *lpadmin*.
- mmodel** Selects a model interface program for printer *P*. *model* is one of the model interface names supplied with the LP software (see Models below).
- rclass** Removes printer *P* from the specified *class*. If printer *P* is the last member of the *class*, the *class* is removed.
- vdevice** Associates a new *device* with printer *P*. *device* is the pathname of a file that is writable by the LP administrator *lp*. Note that there is nothing to stop an administrator from associating the same *device* with more than one *printer*. If only the **-p** and **-v** options are supplied, **lpadmin** can be used while the scheduler is running.

The following *options* are only useful with **-p** and can appear in any order. They are provided with systems that provide remote spooling.

- ob3** Uses three-digit request numbers associated with the printer directory. This is for contact with BSD systems. The default is to not use three-digit request numbers.
- ociremcancel** Specifies that the local command *remcancel* is used to cancel requests to remote printers. To ensure that the correct command is used, specify the full path name.

- ocmremcancel** Specifies that the local model *remcancel* is used to cancel requests to remote printers.
- ormmachine** The name of the remote machine is *machine*.
- orpprinter** The name of the printer to use on the remote machine is *printer*.
- orc** Restricts users to canceling only their own requests. Default is to not restrict the cancel command.
- osi remstatus** Specifies that the command *remstatus* is used to obtain the status of requests to remote printers. To ensure that the correct command is used, specify the full path name.
- osm remstatus** Specifies that the model *remstatus* is used to obtain the status of requests to remote printers.

### Restrictions

When creating a new printer, the **-v** option and one of the **-e**, **-i**, or **-m** options must be specified. Only one of the **-e**, **-i** or **-m** options can be specified. The **-h** and **-l** key letters are mutually exclusive. Printer and class names must not exceed 14 characters and must consist entirely of the characters **A-Z**, **a-z**, **0-9** and **\_** (underscore).

### Models

Model interface programs are supplied with the LP software. They are shell procedures, C programs, or other executable programs that interface between *lpsched*(1M) and devices. All printer models reside in directory **/usr/lib/lp/model** and can be used without modification with **lpadmin -m**. All cancel models reside in directory **/usr/lib/lp/cmodel** and can be used without modification with **lpadmin -ocm**. All status models reside in directory **/usr/lib/lp/smodel** and can be used without modification with **lpadmin -osm**. Models should have 644 permission if owned by **lp** and **bin**, or 664 permission if owned by **bin** and **bin**. Model file names must not exceed 14 characters. Alternatively, LP administrators can modify copies of models then use **lpadmin -m** to associate them with printers.

The LP model interface program does the actual printing on the device that is currently associated with the printer. The LP spooler sets standard input to **/dev/null** and standard output and standard error output to the device specified in the **-v** option of *lpadmin*. The interface program is then invoked for printer *P* from the directory **/etc/lp** as follows:

```
interface/ P id user title copies options file . . .
```

where arguments are as follows:

- id* request id returned by *lp*(1).
- user* login name of the user who made the request.
- title* optional title specified with the **-t** option of *lp*(1).
- copies* number of copies to be printed.
- options* blank-separated list of class-dependent or printer-dependent options specified with the **-o** option of *lp*(1). Options from a BSD system have the character sequence **BSD** attached to the beginning of the option (for example, **BSDI**).
- file* full pathname of the file to be printed.

Given the command line arguments and the output directed to the device, interface programs can format their output in any way they choose.

When printing is completed, it is the responsibility of the interface program to exit with a code indicative of the success of the print job. Only return values of **0** indicating that the job completed successfully, or values of positive **1** through **127** indicating that some error was encountered that does not affect future print jobs should be used. Negative values and positive values greater than **127** are reserved for system use and should not be used by interface programs. *lpsched*(1M) notifies users by mail when there is an error in printing the request. If problems are detected that are likely to affect future print jobs, the interface program should disable the printer so that other pending print requests are not lost.

The cancel and status model interface programs perform the actual communication with the remote system to cancel requests or get the status of requests. See *rcancel*(1M) and *rlpstat*(1M) for command line arguments.

**EXTERNAL INFLUENCES****Environment Variables**

LANG determines the language in which messages are displayed.

If LANG is not specified or is set to the empty string, a default of "C" (see *lang(5)*) is used instead of LANG.

If any internationalization variable contains an invalid setting, **lpadmin** behaves as if all internationalization variables are set to "C" (see *environ(5)*).

**EXAMPLES**

Assuming an existing Hewlett-Packard HP 2934A line printer named **lp1**, it will use the **hp2934a** model interface through **/dev/lp** after the command:

```
/usr/sbin/lpadmin -plp1 -mhp2934a -v/dev/lp
```

Assuming a printer **lp** on a remote system **system2**, the command:

```
/usr/sbin/lpadmin -plp3 -v/dev/null -mrmodel -ocmrcmodel -osmrmodel  
-ob3 -ormsystem2 -orlp -v/dev/null
```

causes the spool system to use the local line printer **lp3** and the model **rmodel**. The spool system also uses the model **rcmodel** to cancel remote requests and **rsmodel** to get status from **system2**. In addition, the three-digit sequence numbers, the remote system name **system2** and the remote printer **lp** are used.

**WARNINGS**

When installing remote printers, use the option **-ocmrcmodel** instead of **-oci/usr/sbin/rcancel** to specify the method used to cancel remote requests. The option **-osmrmodel** should be used instead of **-osi/usr/sbin/rlpstat** to specify the method used for displaying remote status.

*classes* must not include *remote* printers. HP-UX systems do not have the ability to distribute print jobs in this way. Printing to a class of printers on a remote system (**systemB** for example) must be accomplished by creating the class on the remote system, then identifying that class by using a command resembling the following (though you might have to change some of the specific values shown in the example):

```
lpadmin -plocal_name -ormsystemB -orpsystemB_class_name -v /dev/null  
-mrmodel -ocmrcmodel -osmrmodel
```

**FILES**

```
/var/spool/lp/*  
/var/adm/lp/*  
/etc/lp/*  
/usr/lib/lp/*
```

**SEE ALSO**

**enable(1)**, **lp(1)**, **lpstat(1)**, **nroff(1)**, **accept(1M)**, **lpana(1M)**, **lpsched(1M)**, **rcancel(1M)**, **rlp(1M)**, **rlpdaemon(1M)**, **rlpstat(1M)**.

**NAME**

**lpana** - print LP spooler performance analysis information

**SYNOPSIS**

**lpana** [-**d** *dest*]

**DESCRIPTION**

**lpana** prints LP spooler performance information, which system administrators can use to optimize the configuration of the entire spooler system.

**Options**

**lpana** recognizes one option:

**-d** *dest* Choose *dest* as the printer or the class of printers. If *dest* is a printer, the performance analysis information is printed on that specific printer. If *dest* is a class of printers, the performance analysis information is printed on the printers that are members of the class. By default, **lpana** prints the performance analysis information for all printers and/or classes.

**lpana** examines `/var/adm/lp/lpana.log` for the following items:

|                        |                                                                        |
|------------------------|------------------------------------------------------------------------|
| <b>Wait AV</b>         | Average waiting time from when job is spooled until start of printing. |
| <b>Wait SD</b>         | Standard Deviation for waiting time.                                   |
| <b>Print AV</b>        | Average printing time from start to end of job.                        |
| <b>Print SD</b>        | Standard Deviation for printing time.                                  |
| <b>Bytes AV</b>        | Average of number of bytes printed per request.                        |
| <b>Bytes SD</b>        | Standard Deviation for number of bytes.                                |
| <b>Sum KB</b>          | Sum of bytes printed for all requests (in kilobytes).                  |
| <b>Num of Requests</b> | Total number of requests since logging started.                        |

**EXTERNAL INFLUENCES****Environment Variables**

**LANG** determines the language in which messages are displayed.

**WARNINGS**

**lpana** performs its operation on the local system only.

**AUTHOR**

**lpana** was developed by HP.

**FILES**

`/var/adm/lp/lpana.log`

**SEE ALSO**

`lp(1)`, `lpstat(1)`, `lpadmin(1M)`, `lpsched(1M)`.

**NAME**

**lpsched**, **lpshut**, **lpmove**, **lpfence** - start/stop the LP request scheduler, move requests, and define the minimum priority for printing

**SYNOPSIS**

```
/usr/sbin/lpsched [-v] [-a]
/usr/sbin/lpshut
/usr/sbin/lpmove requests dest
/usr/sbin/lpmove dest1 dest2
/usr/sbin/lpfence printer fence
```

**DESCRIPTION**

**lpsched** Schedules requests taken by *lp(1)* for printing on line printers. *lpsched(1M)* is typically invoked in */sbin/rc*. This creates a process which runs in the background until **lpshut** is executed. The activity of the process is recorded in */var/adm/lp/log*.

**lpsched** recognizes the following options:

- v Write a verbose record of the **lpsched** process on */var/adm/lp/log*.
- a Write *lpana(1M)* logging data on */var/adm/lp/lpana.log*.

**lpshut** Shuts down the line printer scheduler. All printers that are printing at the time **lpshut** is invoked stop printing. Requests that were printing at the time a printer was shut down are reprinted in their entirety after **lpsched** is started again. All LP commands perform their functions even when **lpsched** is not running.

**lpmove** Moves requests that were queued by *lp(1)* between LP destinations. This command can be used only when **lpsched** is not running.

The first form of the command moves the named *requests* to the LP destination, *dest*. *requests* are request ids as returned by *lp(1)*. The second form moves all requests for destination *dest1* to destination *dest2*. As a side effect, *lp(1)* rejects requests for *dest1*.

Note that **lpmove** never checks the acceptance status (see *accept(1M)*) for the new destination when moving requests.

**lpfence** Defines the minimum required *priority* for the spooled file to be printed. *fence* must be in between 0 (lowest fence) and 7 (highest fence). Each *printer* has its own *fence*, which is initialized to 0 when it is configured by the *lpadmin(1M)* command. **lpfence** is used only when **lpsched** is not running.

**EXTERNAL INFLUENCES****Environment Variables**

**LC\_TIME** determines the format and contents of date and time strings.

**LANG** determines the language in which messages are displayed.

If **LC\_TIME** is not specified in the environment or is set to the empty string, the value of **LANG** is used as a default for each unspecified or empty variable. If **LANG** is not specified or is set to the empty string, a default of "C" (see *lang(5)*) is used instead of **LANG**. If any internationalization variable contains an invalid setting, *lpsched*, *lpmove*, and **lpshut** behave as if all internationalization variables are set to "C". See *environ(5)*.

**FILES**

```
/var/spool/lp/*
/var/adm/lp/*
/etc/lp/*
/usr/lib/lp/*
```

**WARNINGS**

Moving requests associated with remote printers can cause unpredictable results.

*lpsched*, *lpshut*, *lpmove*, and **lpfence** perform their operation on the local system only.

**SEE ALSO**

*accept(1M)*, *cancel(1)*, *enable(1)*, *lp(1)*, *lpadmin(1M)*, *lpana(1M)*, *lpstat(1)*, *rcancel(1M)*, *rlp(1M)*, *rlpdaemon(1M)*, *rlpstat(1M)*.



**NAME**

lsdev - list device drivers in the system

**SYNOPSIS**

```
/usr/sbin/lsdev [-h] [-d driver | -C class] [-b block_major] [-c char_major] [-e major]
[major ...]
```

**DESCRIPTION**

The **lsdev** command lists, one pair per line, the major device numbers and driver names of device drivers configured into the system and available for invocation via special files. A **-1** in either the block or character column means that a major number does not exist for that type.

If no arguments are specified, **lsdev** lists all drivers configured into the system.

If the **-h** option is specified, **lsdev** will not print a heading. This option may be useful when the output of **lsdev** will be used by another program.

The **-d**, **-C**, **-b**, **-c**, and **-e** options are used to select specific device drivers for output. If more than one option is specified, all drivers that match the criteria specified by those options will be listed. These search options are divided into two types: name search keys (the **-d** and **-C** options) and major number search keys (the **-b**, **-c**, and **-e** options). If both types of options are present, only entries that match both types are printed. The same type of option may appear more than once on the command line with each occurrence providing an ORing effect of that search type. The **-d** and **-C** options may not be specified at the same time.

The ability to process *major* arguments is provided for compatibility and functions like the **-e** option.

**Options**

|                       |                                                                                                 |
|-----------------------|-------------------------------------------------------------------------------------------------|
| <b>-C class</b>       | List device drivers that match <i>class</i> .                                                   |
| <b>-d driver</b>      | List device drivers with the name <i>driver</i> .                                               |
| <b>-b block_major</b> | List device drivers with a block major number of <i>block_major</i> .                           |
| <b>-c char_major</b>  | List device drivers with a character major number of <i>char_major</i> .                        |
| <b>-e major</b>       | List device drivers with either a character major number or block major equal to <i>major</i> . |

**DIAGNOSTICS****Invalid combination of options**

The **-d** and **-C** options may not be specified at the same time.

**Invalid major number**

A major number is malformed or out of range.

**EXAMPLES**

To output entries for all drivers in the **pseudo** class:

```
lsdev -C pseudo
```

To output entries that are in the class **disk** that have either a block or character major number of 0:

```
lsdev -C disk -e 0
```

To get the character major number of **my\_driver** into a shell environment variable:

```
C_MAJOR=$(lsdev -h -d my_driver | awk '{print $1}')
```

**WARNINGS**

Some device drivers available from the system may be intended for use by other drivers. Attempting to use them directly from a special file may produce unexpected results.

A driver may be listed even when the hardware requiring the driver is not present. Attempts to access a driver without the corresponding hardware will fail.

**lsdev** only lists drivers that are configured into the currently executing kernel. For a complete list of available drivers, please run **sam** (see *sam*(1M)).

**DEPENDENCIES**

Since **lsdev** relies on the device driver information provided in a *driver\_install* routine, **lsdev** may not list drivers installed by other means.

**AUTHOR**

**lsdev** was developed by HP.

**SEE ALSO**

**sam(1M)**.

Section 7 entries related to specific device drivers.

*Managing Systems and Workgroups* manual.

**NAME**

**lssf** - list a special file

**SYNOPSIS**

**/sbin/lssf** *special\_file* ...

**DESCRIPTION**

**lssf** lists information about a special file. For each *special\_file* name, **lssf** determines the major number of the special file and whether it is block or character (using *stat(2)*). It then scans the system for the device that is associated with the special file. When the device is found, the minor number of the special file is decoded. A mnemonic description of the minor number is printed on standard output along with the hardware path (i.e., address) of the device. Mnemonics used to describe the fields are closely related to the options used with **mksf** (see *mksf(1M)*).

**DIAGNOSTICS**

Most diagnostic messages from **lssf** are self explanatory. Listed below are some messages deserving further clarification. Warnings allow **lssf** to continue.

**Warnings****No such device in the system**

There is no information about the device in the kernel. The special file is not usable. Use **rmsf** to remove the special file (see *rmsf(1M)*).

**Character major <major> is not in the kernel****Block major <major> is not in the kernel**

The major number associated with the special file is not in the kernel. Use **config** to add the appropriate driver to the kernel (see *config(1M)*).

**Device driver <name> is not in the kernel****Device class <name> is not in the kernel**

The indicated device driver or device class is not present in the kernel. An **open()** of a special file pointing to an unusable device fails. To make the device usable, the appropriate device driver and/or device class must be added to the **config** input file and a new kernel generated (see *config(1M)*). If the device is no longer needed, **rmsf** should be used to remove the special files and update **/etc/ioconfig**.

**<special\_file> is not a special file**

The file is not associated with an I/O device.

**EXAMPLES**

Suppose a special file is created with the command **mksf -d tape2 -H 8.6.1 -b 1600 -a rmt/c2t6d0m**. The command **lssf rmt/c2t6d0m** then produces:

```
tape2 instance 2 bpi 1600 att address 8.6.1 rmt/c2t6d0m
```

**AUTHOR**

**lssf** was developed by HP.

**FILES**

**/dev/config** I/O system special file

**/etc/ioconfig** I/O system configuration database

**SEE ALSO**

*config(1M)*, *insf(1M)*, *mksf(1M)*, *rmsf(1M)*.

**NAME**

lvchange - change LVM logical volume characteristics

**SYNOPSIS**

```
/usr/sbin/lvchange [-a availability] [-A autobackup] [-c mirror_consistency] [-C contiguous]
                  [-d schedule] [-D distributed] [-M mirror_write_cache] [-p permission] [-r relocate] [-s strict]
                  [-t IO_timeout] lv_path
```

**Remarks**

Mirrored disk operations require the installation of the optional HP MirrorDisk/UX software, which is not included in the standard HP-UX operating system.

**lvchange** cannot be performed if the volume group is activated in shared mode.

**DESCRIPTION**

The **lvchange** command changes certain characteristics of a logical volume. Other characteristics can be changed with the **lvextend** and **lvreduce** commands (see *lvextend(1M)* and *lvreduce(1M)*).

The command-line options specify the type and extent of change. Each current characteristic for a logical volume remains in effect until explicitly changed by the corresponding option. All options take effect immediately, except **-s**, which takes effect only when new extents are allocated by the **lvextend** command.

If a logical volume is striped, its scheduling policy is always parallel and its allocation policy is always strict and noncontiguous; these attributes cannot be changed with **lvchange**.

The **lvchange** command can also be used to change the timeout value for a logical volume. This can be useful to control how long an IO request will be retried (for a transient error, like a device timeout), before giving up and declaring a pending IO to be failed. The default behavior is for the system to continue to retry an IO for a transient error until the IO can complete. Thus, the IO will not be returned to the caller until the IO can complete. By setting a non-zero IO timeout value, this will set the maximum length of time that the system will retry an IO. If the IO cannot complete before the length of time specified by the IO timeout, then the IO will be returned to the caller with an error. The actual duration of the IO request may exceed the logical volume's maximum IO timeout value when the underlying physical volume(s) have timeouts which either exceed the logical volume's timeout value or are not an integer multiple of the logical volume's timeout value (see *pvchange(1M)* for details on how to change the IO timeout value on a physical volume).

**Options and Arguments**

The **-c**, **-d**, **-M**, and **-s** options are meaningful only if the optional HP MirrorDisk/UX software has been installed on the system.

**lvchange** recognizes the following options and arguments:

|                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>lv_path</i>                | The block device path name of a logical volume.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>-a</b> <i>availability</i> | Set logical volume availability. <i>availability</i> can have one of the following values: <ul style="list-style-type: none"> <li><b>y</b> Make a logical volume available. An open of the logical volume will succeed.</li> <li><b>n</b> Make a logical volume temporarily unavailable. An open of the logical volume will fail. However, all current processes that have the logical volume open remain open.</li> </ul>                                                                                                       |
| <b>-A</b> <i>autobackup</i>   | Set automatic backup for this invocation of this command. <i>autobackup</i> can have one of the following values: <ul style="list-style-type: none"> <li><b>y</b> Automatically back up configuration changes made to the logical volume. This is the default.               <p>After this command executes, the <b>vgcfgbackup</b> command (see <i>vgcfgbackup(1M)</i>) is executed for the volume group to which the logical volume belongs.</p> </li> <li><b>n</b> Do not back up configuration changes this time.</li> </ul> |

- c** *mirror\_consistency* Set mirror consistency recovery. This option is effective only when **-M n** is specified or previously set. *mirror\_consistency* can have one of the following values:
- y** Set mirror consistency recovery on. LVM achieves mirror consistency during volume group activation by going through all logical extents and copying data from a nonstale copy to the other mirror copies.
  - n** Set mirror consistency recovery off. LVM does not perform mirror consistency recovery on this logical volume when the volume group is activated.
- C** *contiguous* Set the contiguous allocation policy. *contiguous* can have one of the following values:
- y** Set a contiguous allocation policy. Physical extents are allocated in ascending order without any gap between adjacent extents and all extents are contained in a single physical volume.
  - n** Do not set a contiguous allocation policy.
- A nonempty logical volume that has a noncontiguous allocation policy cannot be changed to a contiguous allocation policy unless it happens to meet all the requirements of the contiguous allocation policy. See *lvcreate*(1M) for more information about the contiguous allocation policy.
- d** *schedule* Set the scheduling policy when a logical extent with more than one mirror is written. (The scheduling policy of a striped logical volume is striped and cannot be changed.) *schedule* can have one of the following values:
- p** Establish a parallel scheduling policy.
  - s** Establish a sequential scheduling policy. Use this value with care, because it leads to performance loss in most cases.
- D** *distributed* Change the distributed allocation policy. *distributed* can have one of the following values:
- y** Turn on distributed allocation.
  - n** Turn off distributed allocation.
  - f** Force distributed allocation to be on.
- When the distributed allocation policy is turned on, only one free extent is allocated from the first available physical volume. The next free extent is allocated from the next available physical volume. Allocation of free extents proceeds in round-robin order on the list of available physical volumes.
- When the distributed allocation policy is turned off, all available free extents are allocated from each available physical volume before proceeding to the next available physical volume.
- The distributed allocation policy **REQUIRES** the PVG-strict allocation policy ( **-s g** ) to ensure that mirrors of distributed extents do not overlap (for maximum availability).
- The distributed allocation policy is incompatible with the striped scheduling policy ( **-i stripes** ) and the contiguous allocation policy ( **-C y** ).
- See *lvcreate*(1M) for more information on the distributed allocation policy.
- The **-D y** option will fail if the existing logical volume has any two consecutive logical extents on the same physical volume. To override this failure, use the **-D f** option.
- If a logical volume with the distributed allocation policy has at least two consecutive logical extents on the same physical volume, then *lvdisplay*(1M) will display the allocation as **partially-distributed** (vs. **distributed**).

See *lvdisplay(1M)* for display values.

**-M** *mirror\_write\_cache* Set the Mirror Write Cache flag. This option is allowed only when the logical volume is not opened. *mirror\_write\_cache* can have one of the following values:

**y** Set Mirror Write Cache on. Every write to a mirror copy is recorded in the Mirror Write Cache and written into the Mirror Consistency Record on the disk if a cache-miss occurs. This allows LVM to determine whether all mirror copies are identical, even across system crashes. When the volume group is activated, the Mirror Consistency Record is used to perform mirror consistency recovery.

**n** Set Mirror Write Cache off. Mirror write does not incur an additional write to the Mirror Consistency Record on the disk.

**-p** *permission* Set the access permission. *permission* can have one of the following values:

**w** Set the access permission to read-write.

**r** Set the access permission to read-only.

**-r** *relocate* Set the bad block relocation policy. *relocate* can have one of the following values:

**y** Allow bad block relocation. Upon a media failure (detection of a bad block of data on disk), LVM will mark the failed block in the Bad Block Directory, and attempt to relocate the block to a new location on disk. If relocation is successful then no error will be returned, and future I/O requests which contain the bad block will be directed to the new location. If relocation is unsuccessful, an I/O error will be returned, and subsequent I/O requests containing the bad block will again attempt relocation.

**n** Prevent bad block relocation. Upon a media failure, LVM will mark the failed block as bad in the Bad Block Directory, but will NOT attempt to relocate the bad block to a new location on disk. Future I/O requests which contain the bad block will return with an I/O error. No attempt will be made to access the bad block.

**N** Disable bad block relocation and the Bad Block Directory. Upon a media failure, LVM will NOT attempt to relocate the bad block. In addition it will NOT enter the block in the Bad Block Directory. LVM will have no record of the block being bad, and will attempt to access it on future I/O requests.

**-s** *strict* Set the strict allocation policy. Mirror copies of a logical extent can be allocated to share or not share the same physical volume or physical volume group. This option only makes sense when the physical volumes of the volume group that owns the specified logical volume reside on different physical disks. *strict* can have one of the following values:

**y** Set a strict allocation policy. Mirrors of a logical extent cannot share the same physical volume.

**g** Set a PVG-strict allocation policy. Mirrors of a logical extent cannot share the same physical volume group.

**n** Do not set a strict or a PVG-strict allocation policy. Mirrors of a logical extent can share the same physical volume.

When a logical volume is mirrored, the following changes are not allowed:

- From nonstrict to strict
- From nonstrict to PVG-strict
- From strict to PVG-strict

**-t** *IO\_timeout* Set the *IO\_timeout* for the logical volume to the number of seconds indicated. This value will be used to determine how long to wait for IO requests to complete before concluding that an IO request cannot be completed. An *IO\_timeout* value of zero (0) causes the system to use the default value of

"forever". NOTE: The actual duration of the request may exceed the specified *IO\_timeout* value when the underlying physical volume(s) have timeouts which either exceed this *IO\_timeout* value or are not integer multiples of this value.

## EXTERNAL INFLUENCES

### Environment Variables

**LANG** determines the language in which messages are displayed.

If **LANG** is not specified or is null, it defaults to "C" (see *lang(5)*).

If any internationalization variable contains an invalid setting, all internationalization variables default to "C" (see *environ(5)*).

## EXAMPLES

Change the permission of a logical volume to read-only:

```
lvchange -p r /dev/vg01/lvol3
```

Change the allocation policy of a logical volume to nonstrict:

```
lvchange -s n /dev/vg01/lvol7
```

Turn the mirror write cache off on a logical volume:

```
lvchange -M n /dev/vg01/lvol1
```

Change the IO timeout value of a logical volume to 1 minute (60 seconds):

```
lvchange -t 60 /dev/vg01/lvol1
```

## WARNINGS

For root, swap or dump logical volumes, the allocation policy is always contiguous. This attribute cannot be changed with **lvchange**.

## SEE ALSO

lvcreate(1M), lvdisplay(1M), lvextend(1M).

**NAME**

lvcreate - create logical volume in LVM volume group

**SYNOPSIS**

```
/usr/sbin/lvcreate [-A autobackup] [-c mirror_consistency] [-C contiguous] [-d schedule]
[-D distributed] [-i stripes -I stripe_size] [-l le_number | -L lv_size] [-m mirror_copies]
[-M mirror_write_cache] [-n lv_name] [-p permission] [-r relocate] [-s strict] vg_name
```

**Remarks**

Mirrored disk operations require the installation of the optional HP MirrorDisk/UX software, which is not included in the standard HP-UX operating system.

**lvcreate** cannot be performed if the volume group is activated in shared mode.

Logical volumes that were created using the striped option are not supported in shared mode.

**DESCRIPTION**

The **lvcreate** command creates a new logical volume within the volume group specified by *vg\_name*. Up to 255 logical volumes can be created in one volume group.

If you specify the **-n** *lv\_name* option, a new logical volume is created with that name. Otherwise, a system-generated name of the form **lv01N** is created, where *N* is the decimal equivalent of the two least significant bytes of the minor number of the new logical volume, in the range 1 to 255 (see *lvm(7)*). Two device files are created in *vg\_name*: a block device file named *lv\_name* or **lv01N**, and a character (raw) device file named **r***lv\_name* or **rlv01N**.

If you omit the **-l** and **-L** options, the logical volume is created with zero length. This permits you to choose its physical volume location when you allocate logical extents with the **lvextend** command (see *lvextend(1M)*). If you specify **-l** or **-L**, the location is determined automatically.

The default settings provide the most commonly used characteristics. Use the options to tailor the logical volume to the requirements of the system. Once a logical volume is created, some of its characteristics can be changed with the **lvchange**, **lvextend**, and **lvreduce** commands (see *lvchange(1M)*, *lvextend(1M)*, and *lvreduce(1M)*).

**Options and Arguments**

The **-c**, **-d**, **-m**, **-M**, and **-s** options are only meaningful if the optional HP MirrorDisk/UX software has been installed on the system.

**lvcreate** recognizes the following options and arguments:

- |                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>vg_name</i>                      | The path name of a volume group.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>-A</b> <i>autobackup</i>         | Set automatic backup for this invocation of this command. <i>autobackup</i> can have one of the following values: <ul style="list-style-type: none"> <li><b>y</b> Automatically back up configuration changes made to the logical volume. This is the default.</li> <li>After this command executes, the <b>vgcfgbackup</b> command (see <i>vgcfgbackup(1M)</i>) is executed for the volume group to which the logical volume belongs.</li> <li><b>n</b> Do not back up configuration changes this time.</li> </ul>                                                                                                                                                               |
| <b>-c</b> <i>mirror_consistency</i> | Set mirror consistency recovery. This option is effective only when <b>-M</b> <b>n</b> is specified. It is ignored for <b>-M</b> <b>y</b> . <i>mirror_consistency</i> can have one of the following values: <ul style="list-style-type: none"> <li><b>y</b> Set mirror consistency recovery on. This is the default.</li> <li>LVM achieves mirror consistency during volume group activation by going through all logical extents and copying data from a nonstale copy to the other mirror copies.</li> <li><b>n</b> Set mirror consistency recovery off. LVM does not perform mirror consistency recovery on this logical volume when the volume group is activated.</li> </ul> |



**-C** *contiguous*

Set the contiguous allocation policy. A contiguous logical volume has three characteristics:

- Physical extents are allocated in ascending order,
- No gap is allowed between physical extents within a mirror copy,
- Physical extents of any mirror copy all reside on a single physical volume.

Use the strict (**-s**) and contiguous (**-C**) options together to form various combined allocation policies on a logical volume. For example, **-s y -C y** defines a logical volume such that each mirror copy is contiguous, yet mirror copies of a logical extent cannot share the same physical volume.

*contiguous* can have one of the following values:

**y** Set a contiguous allocation policy.

**n** Do not set a contiguous allocation policy. This is the default.

**-d** *schedule*

Set the scheduling policy when a logical extent with more than one mirror is written. (The scheduling policy of a striped logical volume is striped and cannot be changed.) *schedule* can have one of the following values:

**p** Establish a parallel scheduling policy. This is the default.

**s** Establish a sequential scheduling policy. Use this value with care, because it leads to performance loss in most cases.

**-D** *distributed*

Set the distributed allocation policy. *distributed* can have one of the following values:

**y** Turn on distributed allocation.

**n** Turn off distributed allocation. This is the default.

When the distributed allocation policy is turned on, only one free extent is allocated from the first available physical volume. The next free extent is allocated from the next available physical volume. Allocation of free extents proceeds in round-robin order on the list of available physical volumes.

When the distributed allocation policy is turned off, all available free extents are allocated from each available physical volume before proceeding to the next available physical volume. This is the default.

The distributed allocation policy REQUIRES the PVG-strict allocation policy ( **-s g** ) to ensure that mirrors of distributed extents do not overlap (for maximum availability).

*lvcreate*(1M) will obtain the list of available physical volumes from **/etc/lvm/pvg**. See *vgextend*(1M) for more information on physical volume groups and **/etc/lvm/pvg**.

When a logical volume with distributed extents is mirrored, the resulting layout is commonly referred to as EXTENT-BASED MIRRORED STRIPES.

Note that EXTENT-BASED MIRRORED STRIPES can be created without the distributed allocation policy by adding one extent at a time to the desired physical volumes through *lvextend*(1M).

The distributed allocation policy is incompatible with the striped scheduling policy ( **-i stripes** ) and the contiguous allocation policy ( **-C y** ).

The *lvchange*(1M) command can be used to assign the distributed allocation policy to an existing logical volume.

See *lvdisplay*(1M) for display values.

See EXAMPLES.

**-i** *stripes*

Set the number of disks to stripe across. *stripes* must be in the range 2 to the number of disks in the current volume group. **-i** and **-I** must be specified together.

- I** *stripe\_size* Set the size in kilobytes of the stripe. *stripe\_size* should be a power of 2 in the range 4 to 32768. **-i** and **-I** must be specified together.
- l** *le\_number* Allocate space to the logical volume, specified in logical extents. *le\_number* is a decimal value in the range 1 to 65535 (the implementation limit). The default is described above.
- Either **-l** or **-L** can be specified, but not both.
- L** *lv\_size* Allocate space to the logical volume, specified in megabytes. *lv\_size* is a decimal value in the range 1 to 16777216 (the implementation limit). *lv\_size* is rounded up to the nearest multiple of the logical extent size, equivalent to the physical extent size defined for the volume group by the **vgcreate** command (see **vgcreate(1M)**). The default is described above.
- Either the **-l** or the **-L** option can be specified, but not both.
- m** *mirror\_copies* Set the number of mirror copies allocated for each logical extent. A mirror copy contains the same data as the original. *mirror\_copies* can have the value 1 or 2. The default value is 0 (no mirror copies).
- M** *mirror\_write\_cache* Set the Mirror Write Cache flag. *mirror\_write\_cache* can have one of the following values:
- y** Set Mirror Write Cache on. This is the default.  
Every write to a mirror copy is recorded in the Mirror Write Cache. The Mirror Consistency Record in the Volume Group Reserved Area on the disk is updated whenever there is a write to a logical track group that is not already recorded in the cache. This allows LVM to determine whether all the mirror copies are identical, even across system crashes. When the volume group is activated, the Mirror Consistency Record is used to perform mirror consistency recovery.
  - n** Set Mirror Write Cache to off. Mirror write does not incur an additional write to the Mirror Consistency Record.
- n** *lv\_name* Set the name of the new logical volume to *lv\_name*, where *lv\_name* is a simple file name, not a path name. The default is described above.
- p** *permission* Set the access permission. *permission* can have one of the following values:
- w** Set the access permission to read-write. This is the default.
  - r** Set the access permission to read-only.
- r** *relocate* Set the bad block relocation policy. *relocate* can have one of the following values:
- y** Allow bad block relocation. Upon a media failure (detection of a bad block of data on disk), LVM will mark the failed block in the Bad Block Directory, and attempt to relocate the block to a new location on disk. If relocation is successful then no error will be returned, and future I/O requests which contain the bad block will be directed to the new location. If relocation is unsuccessful, an I/O error will be returned, and subsequent I/O requests containing the bad block will again attempt relocation. This is the default.
  - n** Prevent bad block relocation. Upon a media failure, LVM will mark the failed block as bad in the Bad Block Directory, but will NOT attempt to relocate the bad block to a new location on disk. Future I/O requests which contain the bad block will return with an I/O error. No attempt will be made to access the bad block.
  - N** Disable bad block relocation and the Bad Block Directory. Upon a media failure, LVM will NOT attempt to relocate the bad block. In addition it will NOT enter the block in the Bad Block Directory. LVM will have no record of the block being bad, and will attempt to access it on future I/O requests.

**-s** *strict*

Set the strict allocation policy. Mirror copies of a logical extent can be allocated to share or not share the same physical volume or physical volume group. *strict* can have one of the following values:

- y** Set a strict allocation policy. Mirrors of a logical extent cannot share the same physical volume. This is the default.
- g** Set a PVG-strict allocation policy. Mirrors of a logical extent cannot share the same physical volume group. A PVG-strict allocation policy cannot be set on a logical volume in a volume group that does not have a physical volume group defined.
- n** Do not set a strict or PVG-strict allocation policy. Mirrors of a logical extent can share the same physical volume.

Striped logical volumes are only allocated using the *strict* or *PVG-strict* allocation policies. The number of extents for a striped logical volume is always a multiple of the number of disks the logical volume is striped across. A logical volume striped across **n** disks, is allocated in sets of **n** extents, and each extent of a given set is allocated on a different physical volumes in the volume group.

## EXTERNAL INFLUENCES

### Environment Variables

**LANG** determines the language in which messages are displayed.

If **LANG** is not specified or is null, it defaults to "C" (see *lang(5)*).

If any internationalization variable contains an invalid setting, all internationalization variables default to "C" (see *environ(5)*).

## EXAMPLES

Create a logical volume in volume group **/dev/vg02**:

```
lvcreate /dev/vg02
```

Create a logical volume in volume group **/dev/vg03** with nonstrict allocation policy:

```
lvcreate -s n /dev/vg03
```

Create a logical volume of size 100 MB in volume group **/dev/vg03**:

```
lvcreate -L 100 /dev/vg03
```

Create a logical volume of size 90 MB striped across 3 disks with a stripe size of 64 KB:

```
lvcreate -L 90 -i 3 -I 64 /dev/vg03
```

### Distributed Allocation Policy

This example shows how the **-D y** option can be used to create EXTENT-BASED MIRRORED STRIPES.

Assume that volume group **/dev/vgtest** has two physical volume groups: **pvg1**, **pvg2**.

Assume that each physical volume group has 2 physical volumes.

Assume that the first physical volume in each pvg has 3 extents free and the second physical volume in each pvg has 2 extents free.

The following command creates a logical volume in **vgtest** with EXTENT-BASED MIRRORED STRIPES:

```
lvcreate -D y -s g -m 1 -l 5 /dev/vgtest
```

The distributed allocation proceeds as follows:

- A free extent is allocated from the 1st pvol in **pvg1**.
- A free extent is allocated from the 2nd pvol in **pvg1**.
- A free extent is allocated from the 1st pvol in **pvg1**.
- A free extent is allocated from the 2nd pvol in **pvg1**.
- A free extent is allocated from the 1st pvol in **pvg1**.
- Mirrors for the five extents are then allocated from the free extents in **pvg2** in a similar manner.

## WARNINGS

The root, swap, and dump logical volumes (see *lvlnboot(1M)*) must be created with contiguous allocation policy.

**SEE ALSO**

lvchange(1M), lvdisplay(1M), lvextend(1M), lvreduce(1M), pvchange(1M).

## NAME

lvdisplay - display information about LVM logical volumes

## SYNOPSIS

`/usr/sbin/lvdisplay [-k] [-v] lv_path ...`

## Remarks

Mirrored disk information requires the installation of the optional HP MirrorDisk/UX software, which is not included in the standard HP-UX operating system.

## DESCRIPTION

The **lvdisplay** command displays the characteristics and status of each logical volume specified by *lv\_path*.

## Options and Arguments

**lvdisplay** recognizes the following options and arguments:

- lv\_path* The block device path name of a logical volume, for example, `/dev/vg00/lvol11`.
- v** For each logical volume, display the physical volume distribution, and the mapping of the logical extents onto the physical extents of the physical volumes.
- k** This option displays the same information as the **-v** option, except in the column where **PV Name** is displayed, the **pvkey** (Physical Volume Number in VG) will be displayed instead. Use this option with the **-v** option.

## Display Without -v Option

If you omit the **-v** option, **lvdisplay** displays the following information for each logical volume:

--- Logical volumes ---

|                             |                                                                                                                                   |                                                                                                                                                                            |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>LV Name</b>              | The block device path name of the logical volume.                                                                                 |                                                                                                                                                                            |
| <b>VG Name</b>              | The path name of the volume group.                                                                                                |                                                                                                                                                                            |
| <b>LV Permission</b>        | Access permission: <b>read-only</b> or <b>read/write</b> .                                                                        |                                                                                                                                                                            |
| <b>LV Status</b>            | State of the logical volume:                                                                                                      |                                                                                                                                                                            |
|                             | <b>available/stale</b>                                                                                                            | Available but contains physical extents that are not current.                                                                                                              |
|                             | <b>available/syncd</b>                                                                                                            | Available and synchronized.                                                                                                                                                |
|                             | <b>available</b>                                                                                                                  | Available but the stale or synchronized state cannot be confidently determined because both Mirror Write Cache and Mirror Consistency Recovery are turned off.             |
|                             | <b>unavailable</b>                                                                                                                | Not available for use.                                                                                                                                                     |
| <b>Mirror copies</b>        | Number of physical extents beyond the original allocated for each logical extent; i.e., the number of mirrors: 0, 1, or 2.        |                                                                                                                                                                            |
| <b>Consistency Recovery</b> | Mode of mirror consistency recovery which determines how LVM performs mirror consistency recovery during volume group activation: |                                                                                                                                                                            |
|                             | <b>MWC</b>                                                                                                                        | Recover mirror consistency by using the Mirror Write Cache and Mirror Consistency Record. Implies that Mirror Write Cache is on.                                           |
|                             | <b>NOMWC</b>                                                                                                                      | Recover mirror consistency by going through all logical extents and copying data from a non-stale copy to the other mirror copies. Implies that Mirror Write Cache is off. |
|                             | <b>NONE</b>                                                                                                                       | No mirror consistency recovery during volume group activation on this logical volume. Implies that Mirror Write Cache is off.                                              |
| <b>Schedule</b>             | Striped, sequential or parallel scheduling policy. Striped policy is by default parallel scheduling for mirrored I/O.             |                                                                                                                                                                            |

**LV Size (Mbytes)**

Size of the logical volume in megabytes (MB).

**Current LE**

Number of logical extents currently in the logical volume.

**Allocated PE**

Number of physical extents allocated to the logical volume.

**Stripes**

The number of stripes. If this field is 0, then the logical volume is not striped.

**Stripe Size (Kbytes)**

The size of each stripe in kilobytes (KB).

**Bad block**

Bad block relocation policy.

**Allocation**

Current allocation state, displayed as one of:

**non-strict****non-strict/contiguous****strict****strict/contiguous****PVG-strict****PVG-strict/contiguous****PVG-strict/distributed****PVG-strict/partially-distributed****contiguous** Physical extents are allocated in an ascending order without any gap between adjacent extents. All physical extents of a given mirror are contained in a single physical volume.**distributed**Distributed allocation is turned on and any two consecutive logical extents are guaranteed to be located on different physical volumes. See *lvcreate(1M)* for more information.**partially-distributed**Distributed allocation is turned on but any two consecutive logical extents are NOT guaranteed to be located on different physical volumes. See *lvchange(1M)* for more information.**non-strict**

Physical extents that belong to the same logical extent can be allocated on the same physical volume or physical volume group.

**PVG-strict**

Mirror copies for a logical extent are not allocated on the same physical volume group.

**strict**

Mirror copies for a logical extent are not allocated on the same physical volume.

**IO Timeout (Seconds)**

The IO timeout used by LVM for all IO to this logical volume. A value of default, indicates that the system will use the value of "forever". (Note: the actual duration of a request may exceed this timeout value when the underlying physical volume(s) have timeouts which either exceed this value or are not integer multiples thereof.)

**Display With -v Option**If you specify the **-v** option, **lvdisplay** also lists the distribution of each logical volume across the physical volumes of the volume group and the mapping of each logical extent of the logical volume on the physical extents of the physical volume.**--- Distribution of logical volume ---**The distribution of logical volume *lv\_path* across the physical volumes of the volume group, displayed in the following columns:**PV Name**

The block device path name of the physical volume where the logical extents are allocated.

**PVNUM**The Physical Volume Number in VG (if **-k** option is specified).**LE on PV**

Number of logical extents allocated on the physical volume.

**PE on PV**

Number of physical extents allocated on the physical volume.

**--- Logical extents ---**

The mapping of logical extents onto physical extents, displayed in the following columns:

|                 |                                                                                                                                        |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------|
| <b>LE</b>       | Logical extent number.                                                                                                                 |
| <b>PV1</b>      | The block device path name of the physical volume that corresponds to the location of the first physical extent of the logical extent. |
| <b>PE1</b>      | First physical extent number allocated to the logical extent.                                                                          |
| <b>Status 1</b> | Status of the first physical extent: <b>stale</b> or <b>current</b> .                                                                  |

The following columns are displayed for one or two mirror copies:

|                 |                                                                                                                                                      |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>PV2</b>      | The block device path name of the physical volume that corresponds to the location of the second physical extent (first copy) of the logical extent. |
| <b>PE2</b>      | Second physical extent number allocated to the logical extent.                                                                                       |
| <b>Status 2</b> | Status of the second physical extent: <b>stale</b> or <b>current</b> .                                                                               |

The following columns are displayed for two mirror copies:

|                 |                                                                                                                                                      |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>PV3</b>      | The block device path name of the physical volume that corresponds to the location of the third physical extent (second copy) of the logical extent. |
| <b>PE3</b>      | Third physical extent number allocated to the logical extent.                                                                                        |
| <b>Status 3</b> | Status of the third physical extent: <b>stale</b> or <b>current</b> .                                                                                |

## EXTERNAL INFLUENCES

### Environment Variables

**LANG** determines the language in which messages are displayed.

If **LANG** is not specified or is null, it defaults to "C" (see *lang(5)*).

If any internationalization variable contains an invalid setting, all internationalization variables default to "C" (see *environ(5)*).

## EXAMPLES

Display information about a logical volume:

```
lvdisplay /dev/vg01/lvol3
```

Display all the available information about a logical volume, including the characteristics, status and distribution map:

```
lvdisplay -v /dev/vg01/lvol3
```

Display all the available information about a logical volume, but display **pvkey** instead of **PV Name** in the status and distribution map.

```
lvdisplay -v -k /dev/vg01/lvol3
```

## SEE ALSO

lvchange(1M), lvcreate(1M), lvextend(1M), lvreduce(1M), pvdisplay(1M), vgdisplay(1M).

**NAME**

lvextend - increase space, increase mirrors for LVM logical volume

**SYNOPSIS**

```
/usr/sbin/lvextend [-A autobackup] {-l le_number | -L lv_size | -m mirror_copies} lv_path
[pv_path ... | pvg_name ...]
```

**Remarks**

Mirrored disk operations require the installation of the optional HP MirrorDisk/UX software, which is not included in the standard HP-UX operating system.

**lvextend** cannot be performed if the volume group is activated in shared mode.

Existing logical volumes that were created using the striped option are not supported in shared mode.

**DESCRIPTION**

The **lvextend** command can increase a logical volume's allocated extents, or increase its number of mirrored copies.

Other logical volume characteristics can be modified with the **lvchange** and **lvreduce** commands (see *lvchange(1M)* and *lvreduce(1M)*).

To limit the allocation to specific physical volumes, specify the physical volume names as *pv\_path* arguments or specify the physical volume group names as *pvg\_name* arguments. Otherwise, all of the physical volumes in a volume group are available for allocating new physical extents. LVM always ensures that physical extent allocation can satisfy the current allocation policy or policies. If a physical volume is not suitable for use with a certain allocation policy, it is not used during physical extent allocation, even it is specified in a *pv\_path* argument or indirectly in a *pvg\_name* argument.

LVM striped logical volumes are always allocated using a strict allocation policy. Consequently, striped logical volumes may only be extended by a number extents that is a multiple of disks the logical volume is striped across. For example, for a logical volume striped across 3 disks, the logical volume will be extended in increments of 3 extents, with each of the 3 extents allocated on a different disk in the volume group.

The *pvg\_name* argument is allowed only if one of the allocation policies of the logical volume is PVG-strict.

**Options and Arguments**

The **-m** option is only meaningful if the optional HP MirrorDisk/UX software has been installed on the system.

**lvextend** recognizes the following options and arguments:

|                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>lv_path</i>       | The block device path name of a logical volume.                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <i>pv_path</i>       | The block device path name of a physical volume.                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <i>pvg_name</i>      | The name of a physical volume group (see <i>lvmpvg(4)</i> ).                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>-A autobackup</b> | Set automatic backup for this invocation of this command. <i>autobackup</i> can have one of the following values: <ul style="list-style-type: none"> <li><b>y</b> Automatically back up configuration changes made to the logical volume. This is the default.</li> <li>After this command executes, the <b>vgcfgbackup</b> command (see <i>vgcfgbackup(1M)</i>) is executed for the volume group to which the logical volume belongs.</li> <li><b>n</b> Do not back up configuration changes this time.</li> </ul> |
| <b>-l le_number</b>  | Increase the space allocated to the logical volume, specified in logical extents. <i>le_number</i> is a decimal value greater than the current number of logical extents, in the range 1 to 65535 (the implementation limit).<br>One, and only one, <b>-l</b> , <b>-L</b> , or <b>-m</b> option must be supplied.                                                                                                                                                                                                   |
| <b>-L lv_size</b>    | Increase the space allocated to the logical volume, specified in megabytes. <i>lv_size</i> is a decimal value greater than the current logical volume size, in the range 1 to 16777216 (the implementation limit). <i>lv_size</i> is rounded up to the nearest multiple of the logical extent size, equivalent to the physical extent size defined for the volume group by the <b>vgcreate</b> command                                                                                                              |



(see *vgcreate*(1M)).

One, and only one, **-l**, **-L**, or **-m** option must be specified.

**-m** *mirror\_copies*

Set the number of mirror copies allocated for each logical extent. A mirror copy contains the same data as the original. *mirror\_copies* can have the value 1 or 2. It must be greater than the current value.

Data in the new copies is synchronized. The synchronization process can be time consuming, depending on hardware characteristics and the amount of data.

One, and only one, **-l**, **-L**, or **-m** option must be specified.

## EXTERNAL INFLUENCES

### Environment Variables

**LANG** determines the language in which messages are displayed.

If **LANG** is not specified or is null, it defaults to "C" (see *lang*(5)).

If any internationalization variable contains an invalid setting, all internationalization variables default to "C" (see *environ*(5)).

## EXAMPLES

Increase the number of the logical extents of a logical volume to 100:

```
lvextend -l 100 /dev/vg01/lvol3
```

Increase the logical volume size to 400 MB:

```
lvextend -L 400 /dev/vg01/lvol4
```

Allocate two mirrors (that is, two copies of the original) for each logical extent of a logical volume:

```
lvextend -m 2 /dev/vg01/lvol5
```

Mirror a logical volume onto a particular physical volume.

```
lvextend -m 1 /dev/vg00/lvol3 /dev/dsk/c0t3d0
```

Increase the size of a file system existing on a logical volume.

First, increase the size of the logical volume.

```
lvextend -L 400 /dev/vg06/lvol3
```

Unmount the file system.

```
umount /dev/vg06/lvol3
```

Extend the file system to occupy the entire (larger) logical volume.

```
extendfs /dev/vg06/r1vol3
```

Remount the file system.

```
mount /dev/vg06/lvol3 /mnt
```

## SEE ALSO

*lvchange*(1M), *lvcreate*(1M), *lvdisplay*(1M), *lvreduce*(1M), *pvchange*(1M), *pvdisplay*(1M).

**NAME**

**lvinboot** - prepare LVM logical volume to be root, boot, primary swap, or dump volume

**SYNOPSIS**

```
/usr/sbin/lvinboot [[-A autobackup] { -b boot_lv | -d dump_lv | -r root_lv |
-R | -s swap_lv }] [-v] [vg_name]

/usr/sbin/lvinboot [-c]
```

**Remarks**

**lvinboot** cannot be performed if the volume group is activated in shared mode.

**DESCRIPTION**

The **lvinboot** command updates all physical volumes in the volume group so that the logical volume becomes the root, boot, primary swap, or a dump volume when the system is next booted on the volume group. If a nonexistent logical volume is specified, this command fails. If a different logical volume is already linked to the root or primary swap, the command fails.

This command should be run in recovery mode (**-R**) whenever the configuration of the root volume group is affected by one of the following commands: **lvextend**, **lvmerge**, **lvreduce**, **lvsplit**, **pvmove**, **lvremove**, **vgextend**, or **vgreduce** (see **lvextend(1M)**, **lvmerge(1M)**, **lvreduce(1M)**, **lvsplit(1M)**, **pvmove(1M)**, **lvremove(1M)**, **vgextend(1M)**, and **vgreduce(1M)**). Starting with HP-UX Release 10.0, this is done automatically.

**Options and Arguments**

**lvinboot** recognizes the following options and arguments:

- |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>vg_name</i>              | The path name of a volume group.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>-A</b> <i>autobackup</i> | Set automatic backup for this invocation of this command. <i>autobackup</i> can have one of the following values: <ul style="list-style-type: none"> <li><b>y</b> Automatically back up configuration changes made to the logical volume. This is the default.</li> <li>After this command executes, the <b>vgcfgbackup</b> command (see <b>vgcfgbackup(1M)</b>) is executed for the volume group to which the logical volume belongs.</li> <li><b>n</b> Do not back up configuration changes this time.</li> </ul>                                                                                                                                                                                      |
| <b>-b</b> <i>boot_lv</i>    | Define <i>boot_lv</i> to be the boot volume the next time the system is booted on the volume group. <i>boot_lv</i> must be the first logical volume on the physical volume. <i>boot_lv</i> must be contiguous, and must not allow bad block relocation.<br><i>boot_lv</i> is used to locate the boot file system during the boot process. The boot file system has the kernel which is read by the boot loader <b>hpux(1M)</b> .                                                                                                                                                                                                                                                                         |
| <b>-d</b> <i>dump_lv</i>    | Define <i>dump_lv</i> to be one of the dump volumes the next time the system is booted on the volume group. <i>dump_lv</i> must be a contiguous logical volume and cannot have Bad Block Relocation enabled.<br><br>The command updates the Boot Data Reserved Area of each bootable physical volume in the volume group (see <b>pvcreate(1M)</b> ).<br><br>The combined size of all the dump volumes should be at least 2048 bytes larger than the total memory of the system. The additional 2 KB is used to safeguard against a dump to the bottom of the disk.<br><br>Multiple dump devices can be configured, but each <i>dump_lv</i> must be entered with a separate <b>lvinboot</b> command line. |
| <b>-r</b> <i>root_lv</i>    | Define <i>root_lv</i> to be the root volume the next time the system is booted on this volume group. <i>root_lv</i> must be a contiguous logical volume and cannot have bad block relocation enabled.<br><br>If <i>root_lv</i> is the first logical volume on the physical volume, then it is configured as the combined root-boot volume. Otherwise, <i>root_lv</i> is configured as the separate root volume in which case a separate boot volume needs to be configured using the <b>lvinboot -b</b> option.                                                                                                                                                                                          |

Either the separate root or the separate boot volume can be configured first.

The command updates the Boot Data Reserved Area of each bootable physical volume (see *pvcreate*(1M)) to enable the volume group to be used to locate the root file system. *root\_lv* is also used as the root volume during a maintenance-mode boot (see *hpux*(1M)).

The physical volumes containing *root\_lv* must have been created using the **pvcreate -B** option (see *pvcreate*(1M)), indicating that that physical volume is to be used as a bootable physical volume. Also, the **mkboot** command (see *mkboot*(1M)) must have been run on the physical volume to create the LIF area at the top of the physical volume (see *lif*(4)).

**-R** Recover any missing links to all of the logical volumes specified in the Boot Data Reserved Area and update the Boot Data Reserved Area of each bootable physical volume in the volume group (see *pvcreate*(1M)).

**-s swap\_lv** Define *swap\_lv* to be the primary swap volume the next time the system is booted on the volume group. *swap\_lv* must be a contiguous logical volume, and a root logical volume must have been previously defined with this command.

The command updates the Boot Data Reserved Area of each bootable physical volume in the volume group (see *pvcreate*(1M)). Any existing swap area previously defined must be removed via *lvrmboot*(1M).

**-c** During normal boots (vs. maintenance-mode boots, see *hpux*(1M)), this command is automatically executed by **/sbin/ioinitrc** (see *inittab*(4)).

Since this command is performed during boot, it does not need to be performed manually unless **/stand/rootconf** is missing in a separate root/boot configuration (or alternatively, performing a normal reboot will recreate this file).

This command updates the **/stand/rootconf** file with the location of the root volume in the currently booted volume group.

The **/stand/rootconf** file is used during maintenance-mode boots to locate the root volume for volume groups with separate boot and root volumes.

During maintenance-mode boots, since the root volume group is not activated, **lvlnboot -c** does not update **/stand/rootconf**. For separate root/boot configurations, maintenance-mode boot will fail if **/stand/rootconf** does not already exist with the correct location of the root volume. See *WARNINGS*.

When a new volume group with separate boot and root volumes is created, the first boot must be a normal boot (versus. a maintenance-mode boot), so that **/stand/rootconf** gets created.

This option does not allow updating **/stand/rootconf** for any volume group other than the one that is booted.

**-v** Print verbose messages. With no other arguments present, print information on root, boot, swap, and dump logical volumes. If a combined root-boot volume is configured, no information for the boot volume is displayed.

## EXTERNAL INFLUENCES

### Environment Variables

**LANG** determines the language in which messages are displayed.

If **LANG** is not specified or is null, it defaults to "C" (see *lang*(5)).

If any internationalization variable contains an invalid setting, all internationalization variables default to "C" (see *environ*(5)).

## EXAMPLES

The following examples show configuration of a combined root-boot volume.

Create a root volume group, **vglvmroot**, containing root, swap, and dump logical volumes. Assume that an appropriate directory called **/dev/vglvmroot** and a corresponding **group** file already exist (see *lv*(7)).

First, initialize the disk, say `/dev/dsk/c0t0d0`, so that it can be used as an LVM boot disk.

```
pvccreate -B /dev/rdisk/c0t0d0
```

Place the LIF information on the disk using the `mkboot` command.

```
mkboot /dev/rdisk/c0t0d0
```

Create the volume group `vglvmroot`.

```
vgcreate /dev/vglvmroot /dev/dsk/c0t0d0
```

Create a logical volume that is suitable for use as the root volume. This logical volume has to be the first in the volume group and should be a contiguous volume with bad block relocation turned off.

```
lvcreate -n root -L 120 -C y -r n /dev/vglvmroot
```

Create a logical volume that will be used as primary swap. This volume should be contiguous.

```
lvcreate -n swap -L 64 -C y /dev/vglvmroot
```

Create a logical volume that will be used as the dump volume. This volume should be contiguous.

```
lvcreate -n dump -L 64 -C y /dev/vglvmroot
```

Specify that the logical volume, `root`, will be used as the root volume.

```
lvlnboot -r /dev/vglvmroot/root
```

Specify that the logical volume, `swap`, will be used as the primary swap.

```
lvlnboot -s /dev/vglvmroot/swap
```

Specify that the logical volume, `dump`, will be used as the dump volume.

```
lvlnboot -d /dev/vglvmroot/dump
```

Display the results of the previous operations.

```
lvlnboot -v /dev/vglvmroot
```

The following examples show configuration of separate root and boot volumes.

Create a root volume group, `vglvmroot`, containing `root`, `boot`, `swap`, and `dump` logical volumes. Assume that an appropriate directory called `/dev/vglvmroot` and a corresponding `group` file already exist (see `lvmm(7)`).

First, initialize the disk, say `/dev/dsk/c0t0d0`, so that it can be used as an LVM boot disk.

```
pvccreate -B /dev/rdisk/c0t0d0
```

Place the LIF information on the disk using the `mkboot` command.

```
mkboot /dev/rdisk/c0t0d0
```

Create the volume group `vglvmroot`.

```
vgcreate /dev/vglvmroot /dev/dsk/c0t0d0
```

Create a logical volume that is suitable for use as the boot volume. This logical volume has to be the first in the volume group and should be a contiguous volume with bad block relocation turned off.

```
lvcreate -n boot -L 24 -C y -r n /dev/vglvmroot
```

Create a logical volume that is suitable for use as the root volume. This logical volume should be a contiguous volume with bad block relocation turned off.

```
lvcreate -n root -L 64 -C y -r n /dev/vglvmroot
```

Create a logical volume that will be used as primary swap. This volume should be contiguous.

```
lvcreate -n swap -L 64 -C y /dev/vglvmroot
```

Create a logical volume that will be used as the dump volume. This volume should be contiguous.

```
lvcreate -n dump -L 64 -C y /dev/vglvmroot
```

Specify that the logical volume, `root`, will be used as the root volume.

```
lvlnboot -r /dev/vglvmroot/root
```

Specify that the logical volume, **boot**, will be used as the boot volume.

```
lvlnboot -b /dev/vg1vmroot/boot
```

Specify that the logical volume, **swap**, will be used as the primary swap.

```
lvlnboot -s /dev/vg1vmroot/swap
```

Specify that the logical volume, **dump**, will be used as the dump volume.

```
lvlnboot -d /dev/vg1vmroot/dump
```

Display the results of the previous operations.

```
lvlnboot -v /dev/vg1vmroot
```

The following example shows configuration of multiple dump volumes.

Specify that logical volumes `/dev/vg00/swap1`, `/dev/vg00/dump2`, and `/dev/vg00/dump3` should be used as the dump logical volumes and that `/dev/vg00/swap1` should also be used as primary swap. Assume that the volume group and the logical volumes have been created and the logical volumes are contiguous.

```
lvlnboot -s /dev/vg00/swap1
lvlnboot -d /dev/vg00/swap1
lvlnboot -d /dev/vg00/dump2
lvlnboot -d /dev/vg00/dump3
```

## WARNINGS

### Dump Volume Warnings

At the HP-UX 11.00 release and forward, the `lvlnboot` command will support any size dump device depending upon the IODC firmware addressability of the system. If the configured dump logical volume is out of the range of what the firmware can address, the `lvlnboot` command will return an error message such as "Unable to configure dump logical volume. Dump logical volume size beyond the IODC max address."

### Separate Root/Boot Warnings

Whenever `mkboot(1M)` is used to restore the LIF area of a damaged root physical volume, the `-b boot_lv` option of `lvlnboot` must be performed afterwards to record the boot volume information inside the new LIF (see `lif(4)`). Subsequent `lvlnboot` commands such as `lvlnboot -R` are dependent on the `boot_lv` information inside the LIF.

If the `-v` option does not locate the boot volume `boot_lv`, and the `-r root_lv` has not yet been performed, then performing the `-r root_lv` option will enable the boot volume to be located. The `lvlnboot` command derives the location of boot volume from the location of the root volume.

### Separate Root/Boot Maintenance-Mode Warnings

When creating additional root volumes with separate root/boot, a normal boot must be performed on each new root volume so that `/stand/rootconf`, which is required for maintenance-mode boots (see `hpux(1M)`), gets created for each new root volume.

Mirrored `root_lv` volumes should start at the same offset on each physical volume so that the location stored in `/stand/rootconf` works for maintenance-mode boots off of any mirror.

### Cross-Cabinet Volume Group Warnings

On multi-cabinet V-Class systems, it is possible to create a volume group using physical volumes that are physically attached to different cabinets. However, due to limitations in the HP-UX boot sequence, such a cross-cabinet volume group cannot contain root, boot, swap, or dump logical volumes. Attempts to specify logical volumes in a cross-cabinet volume group for use as root, boot, swap, or dump using `lvlnboot` will fail.

## FILES

|                              |                                                                                                                                                                                            |
|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>/stand/rootconf</code> | Contains the location of the root volume. Used during maintenance-mode boots (see <code>hpux(1M)</code> ) to locate the root volume for volume groups with separate boot and root volumes. |
|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

**SEE ALSO**

lvcreate(1M), lvrmbboot(1M), mkboot(1M), pvcreate(1M), vgcreate(1M), inittab(4), lif(4), lvm(7).

**NAME**

lvmerge - merge two LVM logical volumes into one logical volume

**SYNOPSIS**

`/usr/sbin/lvmerge [-A autobackup] dest_lv_path src_lv_path`

**Remarks**

This command requires the installation of the optional HP MirrorDisk/UX software, which is not included in the standard HP-UX operating system.

**lvmerge** cannot be performed if the volume group is activated in shared mode.

**DESCRIPTION**

The **lvmerge** command merges two logical volumes of the same size. The number of mirrored copies of the *dest\_lv\_path* is increased by the number of copies in the *src\_lv\_path*.

Data previously contained in the *dest\_lv\_path* is resynchronized using the data in the *src\_lv\_path*. All new data on the *dest\_lv\_path* is destroyed.

Whenever a mirrored logical volume is split into two logical volumes, a bit map is stored that keeps track of all writes to either logical volume in the split pair. When the two logical volumes are subsequently merged using **lvmerge**, the bit map is used to decide which areas of the logical volumes need to be resynchronized. This bit map continues to exist until the merge is completed, or one of the logical volumes is extended or reduced, or the system is rebooted.

If there is no bit map available, the entire logical volume is resynchronized.

The normal usage for this command is to merge previously mirrored logical volumes that have been split using the **lvsplit** command (see *lvsplit*(1M)). However, the two logical volumes are not required to have been the result of a previous **lvsplit** operation.

**Options and Arguments**

**lvmerge** recognizes the following options and arguments:

- |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>dest_lv_path</i>         | The block device path name of a logical volume.                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <i>src_lv_path</i>          | The block device path name of a logical volume.                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>-A</b> <i>autobackup</i> | Set automatic backup for this invocation of this command. <i>autobackup</i> can have one of the following values: <ul style="list-style-type: none"> <li><b>y</b> Automatically back up configuration changes made to the logical volume. This is the default.</li> <li>After this command executes, the <b>vgcfgbackup</b> command (see <i>vgcfgbackup</i>(1M)) is executed for the volume group to which the logical volume belongs.</li> <li><b>n</b> Do not back up configuration changes this time.</li> </ul> |

**EXTERNAL INFLUENCES****Environment Variables**

**LANG** determines the language in which messages are displayed.

If **LANG** is not specified or is null, it defaults to "C" (see *lang*(5)).

If any internationalization variable contains an invalid setting, all internationalization variables default to "C" (see *environ*(5)).

**EXAMPLES**

Merge `/dev/vg00/lvol1b` with `/dev/vg00/lvol1`: Data in `/dev/vg00/lvol1b` will be overridden by `/dev/vg00/lvol1`.

```
lvmerge /dev/vg00/lvol1b /dev/vg00/lvol1
```

**WARNINGS**

If no bit map is found, all data on *dest\_lv\_path* is lost after the merge.

**lvmerge** does not check to guarantee that the allocation policy of *src\_lv\_path* is preserved after the merge.

**SEE ALSO**

lvcreate(1M), lvextend(1M), lvsplit(1M).



**NAME**

lvmigrate - prepare root file system for migration from partitions to LVM logical volumes

**SYNOPSIS**

```
/usr/sbin/lvmigrate [-d disk_special_file] [-e file_system ...] [-f] [-i file_system ...] [-n]
[-v]
```

**DESCRIPTION**

The **lvmigrate** command records the configuration information of the current system in the LIF volume of the boot section for use with a subsequent cold-install process. If there is no LIF volume on the disk, **lvmigrate** creates it using *lifinit*(1), then records the information in a LIF file named **CUSTOM**. A copy of the LIF file is saved as **/tmp/LVMMIGRATE.CFG**. The information is also written to file **/tmp/LVMMIGRATE** for reviewing. The install process looks for the LIF file **CUSTOM**, and if it exists, uses the information found as the configuration defaults for the root volume group and the root file systems. After the install process has completed, a copy of the **CUSTOM** final configuration can be found on the newly created system in the file **/usr/lib/sw/hpux.install/config.local**.

All file system entries in the **/etc/mnttab** and **/etc/fstab** files are read. **lvmigrate** also searches for unmounted file systems and possible character data sections in unused disk areas. The file systems appropriate for the root volume group are marked for migration. The default file systems are: **/**, **/home**, **/opt**, **/tmp**, **/usr**, **/var**, and any file system with a mount path beginning with: **/home/**, **/opt/**, **/tmp/**, **/usr/**, **/var/**.

**lvmigrate** displays the following information on the standard output: disks and file system names that are marked for migration, disk areas and file systems to be backed up by the user, and instructions for reinstallation.

After executing **lvmigrate**, the user *must* back up the file systems and any raw device section having useful data to tape. The system is then reinstalled on logical volumes using the configuration information recorded by **lvmigrate**.

**Options**

**lvmigrate** recognizes the following options:

- d *disk\_special\_file*** Use the specified root disk for reinstallation. Without this option, the current root disk (where root file system **/** is currently located) is assumed and the configuration is recorded in the boot section.
- e *file\_system* ...** Exclude each specified default file system from the root volume group. Note that the **/** file system cannot be excluded.
- f** Force the recording of configuration information. Information is recorded in a LIF file named **CUSTOM** in the boot section. Without this option, if there is a file system or LVM record in the boot section, no write is done and a warning message is displayed.
- i *file\_system* ...** Include each specified file system in the root volume group, along with the default file systems.
- n** Perform a "no write" operation for preview purposes. Migration information is displayed on the terminal screen, but is not recorded in the boot section of the disk. The **CUSTOM** LIF file is not written, but the files **/tmp/LVMMIGRATE** and **/tmp/LVMMIGRATE.CFG** are still created.
- v** Display all disks, file systems, and possible raw sections present in the system.

**EXAMPLES**

Prepare a system for migration to root logical volumes. Create a file in the LIF area that the cold-install can use to read default configuration information. Specify verbose mode. Create files **/tmp/LVMMIGRATE** and **/tmp/LVMMIGRATE.CFG**:

```
lvmigrate -v
```

Display a detailed list of the disks, file systems, and possible raw data sections present in the current system.

```
lvmmigrate -v -n
```

Include file system **/mnt** in the root volume group for migration and exclude file system **/usr/source**. Write configuration information in the boot section of disk **/dev/dsk/clt0d0**:

```
lvmmigrate -d /dev/dsk/clt0d0 -i /mnt -e /usr/source
```

**WARNINGS**

Use of the **-f** option results in overwriting the contents of the boot section. Before using the **-f** option be sure to back up all data on the boot section of the disk specified with the **-d** option.

If there is no LIF volume, **lvmmigrate** uses **lifinit** to create it (see *lifinit(1)*). If file **CUSTOM** already exists in the LIF volume, **lvmmigrate** rewrites it.

*Caution:* All data on disks being used for reinstallation must be backed up to a *separate device* because the install process overwrites data on all disks used in the new root volume group.

**SEE ALSO**

*lifinit(1)*.

**NAME**

lvreduce - decrease space allocation or the number of mirror copies of logical volumes

**SYNOPSIS**

```
/usr/sbin/lvreduce [-A autobackup] [-f] -l le_number lv_path
/usr/sbin/lvreduce [-A autobackup] [-f] -L lv_size lv_path
/usr/sbin/lvreduce [-A autobackup] -m mirror_copies lv_path [pv_path ...]
/usr/sbin/lvreduce [-A autobackup] -k -m mirror_copies lv_path
```

**Remarks**

Mirrored disk operations require the installation of the optional HP MirrorDisk/UX software, which is not included in the standard HP-UX operating system.

**lvreduce** cannot be performed if the volume group is activated in shared mode.

**DESCRIPTION**

The **lvreduce** command reduces the number of logical extents allocated to a logical volume specified by *lv\_path*. The excess physical extents in the logical volume and any mirror copies are deallocated.

Alternatively, it reduces the number of mirror copies in the logical volume. The physical extents that comprise the deleted mirror copy or copies are deallocated. If *pv\_path ...* is specified, the mirror or mirrors to be removed will be deallocated from those specific physical volumes.

**lvreduce** asks for confirmation before deallocating logical extents if the **-f** option is omitted.

**Options and Arguments**

The **-m** option and *pv\_path* argument are only meaningful if the optional HP MirrorDisk/UX software has been installed on the system.

**lvreduce** recognizes the following options and arguments:

|                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>lv_path</i>       | The block device path name of a logical volume.                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <i>pv_path</i>       | The block device path name of a physical volume.                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>-A autobackup</b> | Set automatic backup for invocation of this command. <i>autobackup</i> can have one of the following values:                                                                                                                                                                                                                                                                                                                                                      |
| <b>y</b>             | Automatically back up configuration changes made to the logical volume. This is the default.                                                                                                                                                                                                                                                                                                                                                                      |
|                      | After this command executes, the <b>vgcfgbackup</b> command (see <b>vgcfgbackup(1M)</b> ) is executed for the volume group to which the logical volume belongs.                                                                                                                                                                                                                                                                                                   |
| <b>n</b>             | Do not back up configuration changes.                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>-f</b>            | Force reduction of the number of logical extents without first requesting confirmation.                                                                                                                                                                                                                                                                                                                                                                           |
|                      | This option can be dangerous when there is a file system on the <i>lv_path</i> that is larger than the size that the logical volume is being reduced to. If the file system is unmounted, the <b>-f</b> option forces the reduction of the logical volume without reducing the file system. The file system becomes corrupt and is not mountable. If the file system is mounted, <b>lvreduce</b> fails, preventing a mounted file system from becoming corrupted. |
| <b>-l le_number</b>  | Decrease the space allocated to the logical volume, specified in logical extents. <i>le_number</i> is a decimal value smaller than the current number of logical extents, in the range 1 to 65535 (the implementation limit).                                                                                                                                                                                                                                     |
|                      | One, and only one, <b>-l</b> , <b>-L</b> , or <b>-m</b> option must be supplied.                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>-L lv_size</b>    | Decrease the space allocated to the logical volume, specified in megabytes. <i>lv_size</i> is a decimal value smaller than the current logical volume size, in the range 1 to 16777216 (the implementation limit). <i>lv_size</i> is rounded up to the nearest multiple of the logical extent size, equivalent to the physical extent size defined for the volume group by the <b>vgcreate</b> command (see <b>vgcreate(1M)</b> ).                                |

- One, and only one, **-l**, **-L**, or **-m** option must be specified.
- m** *mirror\_copies* Reduce the number of mirror copies allocated for each logical extent. A mirror copy contains the same data as the original. *mirror\_copies* can have the value 0 or 1. It must be smaller than the current value.
- If optional *pv\_path* arguments are specified, the mirror copies are deallocated from the specified physical volumes.
- One, and only one, **-l**, **-L**, or **-m** option must be specified.
- k** This option should be used only in the special instance when you want to reduce a mirrored logical volume on a physical volume that is missing or has failed.
- Use this option with the **-m** option.

## EXTERNAL INFLUENCES

### Environment Variables

**LANG** determines the language in which messages are displayed.

If **LANG** is not specified or is null, it defaults to "C" (see *lang(5)*).

If any internationalization variable contains an invalid setting, all internationalization variables default to "C" (see *environ(5)*).

## EXAMPLES

Decrease the number of the logical extents of a logical volume to one hundred:

```
lvreduce -l 100 /dev/vg01/lvol3
```

Reduce to one mirror (that is, an original and one copy) for each logical extent of a logical volume:

```
lvreduce -m 1 /dev/vg01/lvol5
```

Remove mirror copies of logical extents of a logical volume from the physical volume **/dev/dsk/c1t0d0**:

```
lvreduce -m 0 /dev/vg01/lvol4 /dev/dsk/c1t0d0
```

Remove a logical volume from a one-way mirrored set on the specified *pv\_path*.

```
lvreduce -m 0 -k /dev/vg01/lvol1 /dev/dsk/c1t0d0
```

## WARNINGS

LVM does not store any information about which physical extents within a logical volume contain useful data; therefore, reducing the space allocated to a logical volume without doing a prior backup of the data could lead to the loss of useful data. The **lvreduce** command on a logical volume containing a file system of greater length than the size being reduced to will cause data corruption.

To reduce a logical volume being used for swap, that swap area must not be currently in use.

## SEE ALSO

**lvcreate(1M)**, **lvdisplay(1M)**, **lvextend(1M)**, **pvchange(1M)**, **pvdisk(1M)**.

**NAME**

lvremove - remove one or more logical volumes from LVM volume group

**SYNOPSIS**

```
/usr/sbin/lvremove [-A autobackup] [-f] lv_path ...
```

**Remarks**

**lvremove** cannot be performed if the volume group is activated in shared mode.

**DESCRIPTION**

The **lvremove** command removes each logical volume specified by *lv\_path* ....

Logical volumes must be closed before they can be removed. For example, if the logical volume contains a file system, unmount the file system before removing it.

**Options and Arguments**

**lvremove** recognizes the following options and arguments:

- |                             |                                                                                                                                                                 |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>lv_path</i>              | The block device path name of a logical volume.                                                                                                                 |
| <b>-A</b> <i>autobackup</i> | Set automatic backup for this invocation of this command. <i>autobackup</i> can have one of the following values:                                               |
| <b>y</b>                    | Automatically back up configuration changes made to the logical volume. This is the default.                                                                    |
|                             | After this command executes, the <b>vgcfgbackup</b> command (see <i>vgcfgbackup</i> (1M)) is executed for the volume group to which the logical volume belongs. |
| <b>n</b>                    | Do not back up configuration changes this time.                                                                                                                 |
| <b>-f</b>                   | Specify that no user confirmation is required.                                                                                                                  |

**EXTERNAL INFLUENCES****Environment Variables**

**LANG** determines the language in which messages are displayed.

If **LANG** is not specified or is null, it defaults to "C" (see *lang*(5)).

If any internationalization variable contains an invalid setting, all internationalization variables default to "C" (see *environ*(5)).

**EXAMPLES**

Remove a logical volume without requiring user confirmation:

```
lvremove -f /dev/vg01/lvol5
```

**WARNINGS**

This command destroys all data in the specified logical volumes.

**SEE ALSO**

lvchange(1M), umount(1M).

**NAME**

lvrmbboot - remove LVM logical volume link to root, primary swap, or dump volume

**SYNOPSIS**

```
/usr/sbin/lvrmbboot [-A autobackup] [-d dump_lv] [-r] [-s] [-v] vg_name
```

**Remarks**

**lvrmbboot** cannot be performed if the volume group is activated in shared mode.

**DESCRIPTION**

The **lvrmbboot** command updates all physical volumes contained in the volume group *vg\_name* such that the logical volume is removed as a root, primary swap, or dump volume when the system is next booted on the volume group.

**Options and Arguments**

**lvrmbboot** recognizes the following options and arguments:

|                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |          |                                                                                              |  |                                                                                                                                                                 |          |                                                 |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|----------------------------------------------------------------------------------------------|--|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|-------------------------------------------------|
| <i>vg_name</i>              | The path name of the volume group.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |          |                                                                                              |  |                                                                                                                                                                 |          |                                                 |
| <b>-A</b> <i>autobackup</i> | Set automatic backup for this invocation of this command. <i>autobackup</i> can have one of the following values: <table> <tr> <td><b>y</b></td><td>Automatically back up configuration changes made to the logical volume. This is the default.</td></tr> <tr> <td></td><td>After this command executes, the <b>vgcfgbackup</b> command (see <b>vgcfgbackup(1M)</b>) is executed for the volume group to which the logical volume belongs.</td></tr> <tr> <td><b>n</b></td><td>Do not back up configuration changes this time.</td></tr> </table> | <b>y</b> | Automatically back up configuration changes made to the logical volume. This is the default. |  | After this command executes, the <b>vgcfgbackup</b> command (see <b>vgcfgbackup(1M)</b> ) is executed for the volume group to which the logical volume belongs. | <b>n</b> | Do not back up configuration changes this time. |
| <b>y</b>                    | Automatically back up configuration changes made to the logical volume. This is the default.                                                                                                                                                                                                                                                                                                                                                                                                                                                       |          |                                                                                              |  |                                                                                                                                                                 |          |                                                 |
|                             | After this command executes, the <b>vgcfgbackup</b> command (see <b>vgcfgbackup(1M)</b> ) is executed for the volume group to which the logical volume belongs.                                                                                                                                                                                                                                                                                                                                                                                    |          |                                                                                              |  |                                                                                                                                                                 |          |                                                 |
| <b>n</b>                    | Do not back up configuration changes this time.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |          |                                                                                              |  |                                                                                                                                                                 |          |                                                 |
| <b>-d</b> <i>dump_lv</i>    | Remove the definition of <i>dump_lv</i> as one of the dump volumes. Update the Boot Data Reserved Area.                                                                                                                                                                                                                                                                                                                                                                                                                                            |          |                                                                                              |  |                                                                                                                                                                 |          |                                                 |
| <b>-r</b>                   | Remove the definitions of all of the root, primary swap, and all dump volumes from the given volume group. Update the Boot Data Reserved Area.                                                                                                                                                                                                                                                                                                                                                                                                     |          |                                                                                              |  |                                                                                                                                                                 |          |                                                 |
| <b>-s</b>                   | Remove the definition of the primary swap volume from the given volume group. Update the Boot Data Reserved Area.                                                                                                                                                                                                                                                                                                                                                                                                                                  |          |                                                                                              |  |                                                                                                                                                                 |          |                                                 |
| <b>-v</b>                   | Print verbose messages.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |          |                                                                                              |  |                                                                                                                                                                 |          |                                                 |

**EXTERNAL INFLUENCES****Environment Variables**

**LANG** determines the language in which messages are displayed.

If **LANG** is not specified or is null, it defaults to "C" (see *lang(5)*).

If any internationalization variable contains an invalid setting, all internationalization variables default to "C" (see *environ(5)*).

**EXAMPLES**

Specify that the logical volume **/dev/vg00/lvol3** should be removed as one of the dump logical volumes:

```
lvrmbboot -v -d lvol3 /dev/vg00
```

Specify that volume group **/dev/vg00** should no longer be a root volume group. Primary swap and dump are also removed.

```
lvrmbboot -r /dev/vg00
```

**SEE ALSO**

lvlnboot(1M).

## (Requires Optional HP MirrorDisk/UX Software)

## NAME

lvsplit - split mirrored LVM logical volume into two logical volumes

## SYNOPSIS

```
/usr/sbin/lvsplit [-A autobackup] [-s suffix] [-g PhysicalVolumeGroup] lv_path ...
```

## Remarks

This command requires the installation of the optional HP MirrorDisk/UX software (not included in the standard HP-UX operating system) before it can be used.

**lvsplit** cannot be performed if the volume group is activated in shared mode.

## DESCRIPTION

The **lvsplit** command splits a single- or double-mirrored logical volume, *lv\_path*, into two logical volumes. A second logical volume is created containing one copy of the data. The original logical volume is appropriately reset as unmirrored or single-mirrored.

If the **-s** option is specified, the new logical volume name has the form *lv\_pathsuffix*. If **-s** is not specified, *suffix* defaults to **b**, as in *lv\_pathb*.

If more than one *lv\_path* is specified on the command line, **lvsplit** ensures that all logical volumes are brought offline together in one system call, ensuring predictable results among the logical volumes. Up to 127 logical volumes can be specified on the command line. All logical volumes must belong to the same volume group, and there must be enough unused logical volumes remaining in the volume group to hold the newly split logical volumes. A volume group can contain up to 255 logical volumes.

If *PhysicalVolumeGroup* is specified, the offline logical volumes are created using the mirror copies on the physical volumes contained in the specified physical volume group.

Whenever a mirrored logical volume is split into two logical volumes, a bit map is stored that keeps track of all writes to either logical volume in the split pair. When the two logical volumes are subsequently merged using **lvmerge**, the bit map is used to decide which areas of the logical volumes need to be resynchronized (see **lvmerge**(1M)). This bit map remains in existence until the merge is completed, until one of the logical volumes is extended, reduced, or split again, or until the system is rebooted.

The new logical volume must be checked with the **fsck** command before it is mounted (see **fsck**(1M)). **lvsplit** flushes the file system to a consistent state except for pipes and unlinked but open files.

To rejoin two split copies of a logical volume, use the **lvmerge** command (see **lvmerge**(1M)).

## Options and Arguments

**lvsplit** recognizes the following options and arguments:

|                                      |                                                                                                                                                                                                                     |
|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>lv_path</i>                       | The block device path name of a logical volume. Up to 127 logical volumes in the same volume group can be specified at one time.                                                                                    |
| <b>-A</b> <i>autobackup</i>          | Set automatic backup for invocation of this command. <i>autobackup</i> can have one of the following values:                                                                                                        |
| <b>y</b>                             | Automatically back up configuration changes made to the logical volume. This is the default.                                                                                                                        |
|                                      | After this command executes, the <b>vgcfgbackup</b> command (see <b>vgcfgbackup</b> (1M)) is executed for the volume group to which the logical volume belongs.                                                     |
| <b>n</b>                             | Do not back up configuration changes this time.                                                                                                                                                                     |
| <b>-g</b> <i>PhysicalVolumeGroup</i> | The offline logical volumes will be created using the mirror copies on the physical volumes in the specified <i>PhysicalVolumeGroup</i> .                                                                           |
| <b>-s</b> <i>suffix</i>              | Specify the suffix to use to identify the new logical volume. The new logical volume name has the form <i>lv_pathsuffix</i> . If <b>-s</b> is omitted, <i>suffix</i> defaults to <b>b</b> , as in <i>lv_pathb</i> . |

**EXTERNAL INFLUENCES****Environment Variables**

**LANG** determines the language in which messages are displayed.

If **LANG** is not specified or is null, it defaults to "C" (see *lang(5)*).

If any internationalization variable contains an invalid setting, all internationalization variables default to "C" (see *environ(5)*).

**EXAMPLES**

Split the mirrored logical volume `/dev/vg00/lvol1` into two copies. Call the new logical volume `/dev/vg00/lvol1backup`:

```
lvsplit -s backup /dev/vg00/lvol1
```

Split the mirrored logical volume `/dev/vg00/lvol1` into two copies. The offline logical volume will be created using the mirror copy on the physical volumes contain in the physical volume group `pvgr1`.

```
lvsplit -g pvgr1 /dev/vg00/lvol1
```

Split an online logical volume which is currently mounted on `/usr` so that a backup can take place:

```
lvsplit /dev/vg00/lvol1
fsck /dev/vg00/lvol1b
mount /dev/vg00/lvol1b /usr.backup
```

Perform a backup operation, then:

```
umount /usr.backup
lvmerge /dev/vg00/lvol1b /dev/vg00/lvol1
```

Split two logical volumes at the same time:

```
lvsplit /dev/vg01/database1 /dev/vg01/database2
```

Perform operation on split logical volumes, then rejoin them:

```
lvmerge /dev/vg01/database1b /dev/vg01/database1
lvmerge /dev/vg01/database2b /dev/vg01/database1
```

**WARNINGS**

After a two-way mirrored logical volume has been split once, it cannot be split again without merging the logical volumes using the `lvmerge` command (see *lvmerge(1M)*).

**SEE ALSO**

`lvcreate(1M)`, `lvextend(1M)`, `lvmerge(1M)`.



**NAME**

lvsync - synchronize stale mirrors in LVM logical volumes

**SYNOPSIS**

`/usr/sbin/lvsync lv_path ...`

**Remarks**

This command requires the installation of the optional HP MirrorDisk/UX software (not included in the standard HP-UX operating system) before it can be used.

**DESCRIPTION**

The **lvsync** command synchronizes the physical extents of each logical volume specified by *lv\_path*. Synchronization occurs only on physical extents that are stale mirrors of the original logical extent. The synchronization process can be time consuming, depending on the hardware characteristics and the amount of data.

**Arguments**

**lvsync** recognizes the following argument:

*lv\_path*                      The block device path name of a mirrored logical volume.

**EXTERNAL INFLUENCES****Environment Variables**

**LANG** determines the language in which messages are displayed.

If **LANG** is not specified or is null, it defaults to "C" (see *lang(5)*).

If any internationalization variable contains an invalid setting, all internationalization variables default to "C" (see *environ(5)*).

**EXAMPLES**

Synchronize the mirrors on a logical volume:

```
lvsync /dev/vg01/lvol5
```

**SEE ALSO**

lvdisplay(1M), vgsync(1M).

**NAME**

makedbm - make a Network Information System database

**SYNOPSIS**

```
/usr/sbin/makedbm [-b] [-l] [-s] [-i nis_input_file] [-o nis_output_name] [-d
nis_domain_name] [-m nis_master_name] infile outfile
/usr/sbin/makedbm -u database_name
```

**Remarks**

The Network Information Service (NIS) was formerly known as Yellow Pages (yp). Although the name has changed, the functionality of the service remains the same.

**DESCRIPTION**

**makedbm** generates databases (maps) for the Network Information System (NIS) from *infile*. A database created by **makedbm** consists of two files: *outfile.pag* and *outfile.dir*. A **makedbm** database contains records called **dbm records** composed of key-value pairs.

Each line of *infile* is converted to a single dbm record; all characters up to the first tab or space form the key, and the remainder of the line is the value. If a value read from *infile* ends with \, the value for that record is continued onto the next line. The NIS clients must interpret the # character (which means that **makedbm** does not treat the # as if it precedes a comment). If *infile* is a hyphen (-), **makedbm** reads standard input.

**makedbm** always generates a special dbm record with the key **YP\_LAST\_MODIFIED**, whose value is the time of last modification of *infile* (or the current time, if *infile* is -). This value is also known as the order number of a map, and **yppoll** prints it for a specified NIS map (see **yppoll(1M)**).

Another special dbm record created by **makedbm** has the key **YP\_MASTER\_NAME**. Its value is usually the host name retrieved by **gethostname()**; however, the **-m** option can be used to specify a different value (see **gethostname(2)**).

If the **-b** option is used, another special dbm record with the **YP\_INTERDOMAIN** key is created. When this key exists in the NIS *host.by\** maps and the NIS host name resolution fails, the ypserv process will query the Internet domain name server, **named(1M)**, to provide the host name resolution. Before using the **-b** option, it is recommended that the name services switch, *switch(4)*, be set to allow NIS host name resolution first. (Note that, since the ypserv process only checks *hosts.byname* and *hosts.byaddr* for the existence of the **YP\_INTERDOMAIN** key, using the **-b** option on any other NIS map will have no effect. Also, the **-b** option should be used on both the *hosts.byname* and *hosts.byaddr* maps, not one exclusively.)

If the **-s** option is used, another special dbm record created is the **YP\_SECURE** key. If this key exists in an NIS map, ypserv will only allow privileged processes (applications that can create reserved ports) to access the data within the map.

**Options**

**makedbm** recognizes the following options and command-line arguments.

- b** Create a special dbm record with the key **YP\_INTERDOMAIN**. This key, which is in the *hosts.byname* and *hosts.byaddr* maps, allows the ypserv process to query the Internet domain name server, (see **named(1M)**).
- l** Convert the keys of the given map to lowercase. This command option allows host name matches to work independent of character-case distinctions.
- s** Accept connections from secure NIS networks only.
- i** Create a special dbm record with the key **YP\_INPUT\_FILE** and the value *nis\_input\_file*. If the **-s** option is used, another special dbm record created is the **YP\_SECURE** key. If this key exists in an NIS map, ypserv will only allow privileged processes to access the data within the map. (i.e. applications that can create reserved ports.)
- o** Create a special dbm record with the key **YP\_OUTPUT\_NAME** and the value *nis\_output\_name*.
- d** Create a special dbm record with the key **YP\_DOMAIN\_NAME** and the value *nis\_domain\_name*.
- m** Replace the value of the special dbm record whose key is **YP\_MASTER\_NAME** with *nis\_master\_name*.

- u Undo the *database\_name* (i.e., write the contents of *database\_name* to the standard output), one dbm record per line. A single space separates each key from its value.

**EXAMPLES**

Shell scripts can be written to convert ASCII files such as `/etc/netgroup` to the key-value form used by **makedbm**. For example,

```
#!/usr/bin/sh
/usr/bin/awk 'BEGIN { FS = ":" } { print $1, $0 }' \
    /etc/netgroup | \
makedbm - netgroup
```

converts the file `/etc/netgroup` to a form that is read by **makedbm** to make the NIS map **netgroup**. The keys in the database are *netgroup*(4) names, and the values are the remainders of the lines in the `/etc/netgroup` file.

**AUTHOR**

**makedbm** was developed by Sun Microsystems, Inc.

**SEE ALSO**

`domainname(1)`, `ypinit(1M)`, `ypmake(1M)`, `yppoll(1M)`, `gethostname(2)`, `netgroup(4)`, `ypfiles(4)`.

**NAME**

**makemap** - creates database maps for sendmail

**SYNOPSIS**

**makemap** [-N] [-n] [-d] [-f] [-o] [-r] [-s] [-v] *maptype mapname*

**DESCRIPTION**

**makemap** creates the database maps used by the keyed map lookups in *sendmail*(1M). It reads input from the standard input and outputs them to the indicated *mapname*.

**makemap** handles up to three different database formats, selected using the *maptype* parameter. They may be

**dbm**        DBM format maps. (.pag,.dir)  
**btree**      B-Tree format maps. (.db)  
**hash**       Hash format maps. (.db)

In all cases, **makemap** reads lines from the standard input consisting of two words separated by white space. The first is the database key, the second is the value. The value may contain *%n* strings to indicated parameter substitution. Literal parentheses should be doubled (%%). Blank lines and lines beginning with pound sign (#) are ignored.

**Flags**

- N**        Include the null byte that terminates strings in the map. This must match the **-N** flag in the **sendmail.cf** **K** line.
- n**        Create NIS compatible alias database.
- d**        Allow duplicate keys in the map. This is only allowed on B-Tree format maps. If two identical keys are read, they will both be inserted into the map.
- f**        Normally all upper case letters in the key are folded to lower case. This flag disables that behaviour. This is intended to mesh with the **-f** flag in the **K** line in **sendmail.cf**. The value is never case folded.
- o**        Append to an old file. This allows you to augment an existing file.
- r**        Allow replacement of existing keys. Normally **makemap** complains if you repeat a key, and does not do the insert.
- s**        Ignore safety checks on maps being created. This includes checking for hard or symbolic links in world writable directories.
- v**        Verbosely print what it is doing.

**SEE ALSO**

*sendmail*(1M).

**HISTORY**

The **makemap** command appeared in 4.4BSD.

**NAME**

map-mbone - Multicast Router Connection Mapper

**SYNOPSIS**

```
/usr/sbin/map-mbone [-d debuglevel] [-f] [-g] [-n] [-r retries] [-t timeout] [ multicast-router ]
```

**DESCRIPTION**

**map-mbone** requests the multicast router connection information from the *multicast-router*, and prints the information to the standard out. **map-mbone** sends out the *ASK\_NEIGHBORS* igmp message to the multicast-router. When the multicast-router receives the request, it sends back its configuration information. *multicast-router* can be either an ip address or a system name.

If the *multicast-router* is not specified, *flood* mode is on by default and the igmp request message is sent to all the multicast router on the local network. With *flood* mode on, when **map-mbone** finds new neighbor routers from the replies, it will send the same igmp request to the new neighbor routers. This activity continues until no new neighbor routers are reported in the replies.

The command line options are:

- d debuglevel** Sets the level for printing out the debug message. The default is 0, which prints only error and warning messages. Debug level three prints most the messages.
- r retries** Sets the retry times to poll the routing daemon for information. The default is 1.
- t timeout** It specifies the timeout value in seconds for waiting the reply. The default value is 2 seconds.
- f** Sets the *flood* mode on. It is the default value when no *multicast-router* is given on the command line input.
- g** Generates output in GGraphEd format.
- n** Disable DNS lookup for the multicast router names.

The output contains the interface configuration information of the requested router(s). The format for each interface output is:

```
interface_addr -> neighbor_addr (neighbor_name) [metrics/thresh/flags]
```

If there are multiple neighbor routers on one interface, they will all be reported. The *neighbor\_name* will not be printed if the **-n** option is specified on the command line.

The possible values for **flags** are:

- tunnel** Neighbors are reached via tunnel.
- srcrt** The tunnel uses IP source routing.
- down** The interface is down.
- disabled** The interface is administratively disabled for multicast routing.
- querier** The local router is the querier of the subnet.

The format of the GGraphEd output is:

```
interface_addr_in_integer { $ NP low_byte_addr high_byte_addr } node_name  
[ neighbor_addr_in_integer metrics/threshold/flags ]
```

If there is no neighbor router on an interface, then a **\*** will be put next to the *node\_name*. If there are multiple neighbor routers on one interface, all of them will be reported. The possible values for **flags** are:

- E** The neighbor is reached via tunnel.
- P** The neighbor is on the same network/subnet.
- D** The interface is down.

Please see *mrouted*(1M) for **metrics** and **thresh**.

**EXAMPLES**

Querying **camden.cup.hp.com** for the multicast router connection information.

```
map-mbone hpntclt.cup.hp.com
```

```
127.0.0.1 (localhost) [version 3.3]:
193.2.1.39 -> 0.0.0.0 (all-zeros-broadcast) [1/1/disabled]
15.13.106.144 -> 15.255.176.33 (matmos.hpl.hp.com) [10/1/tunnel]
15.13.106.144 -> 15.17.20.7 (hpspddc.vid.hp.com) [10/1/tunnel/down]
```

Querying **hpntcbs.cup.hp.com** for multicast router connectivity with **-g** option:

```
map-mbone -g hpntcbs.cup.hp.com
```

```
GRAPH "Multicast Router Connectivity: Wed Feb 1 17:34:59 1995"=UNDIRECTED
252537488 {$ NP 1440 1060 $} "hpntclt.cup.hp.com*"
;
252538974 {$ NP 940 1120 $} "hpntcbs.cup.hp.com"
252537488 "10/1E"
252539807 "1/1P"
;
252539807 {$ NP 1590 1150 $} "hpntclh.cup.hp.com*"
;
```

#### **Note**

**map-mbone** must be run as root.

#### **AUTHOR**

**map-mbone** was developed by Pavel Curtis.

#### **SEE ALSO**

**mrouted** (1M), **mrinfo**(1M).

**m**

**NAME**

mc - media changer manipulation utility

**SYNOPSIS**

```
mc [-p device] [-a num] [-q] [-c <src_element_type><dest_element_type>]
mc [-p device] [-b 0|1] [-l 0|1] [-e element_type]
mc [-p device] [-n drive_number] [-r element_type]
mc [-p device] -s <element_type><num> -d <element_type><num>
mc [-h | -?]
```

**DESCRIPTION**

The **mc** utility provides users with a command-line interface to send media manipulation commands to an autoloader or media changer device. It takes "element types" as arguments to most of the options. The valid element types (*element\_types*) are:

- D** Specifies a Data Transfer (DT) element.
- I** Specifies an Import/Export (IE) element.
- M** Specifies a Medium Transport (MT) element.
- S** Specifies a Storage (ST) element.

An example of a Data Transfer element is the embedded tape drive(s) of the autoloader. An example of an Import/Export element is the slot(s) by which an item of the media may be inserted or removed from the autoloader. An example of a Medium Transport element is the robotic picker assembly of the autoloader. An example of a Storage element is the media slot(s) of the autoloader.

Please see examples below for usage.

**Options**

**mc** recognizes the following options and arguments:

- a num** Prints the SCSI bus address of the drive slot specified by *num*.
- b 0 / 1** Issue a Reserve (1) or Release (0) SCSI command to the device.
- c <src\_element\_type><dest\_element\_type>**  
Determines whether a move from source to destination is valid. Uses device capabilities mode page and will return TRUE or FALSE. There should be no spaces in the source and destination element type values. For example, **-c DS** specifies a Data Transfer element as the source and a Storage element as the destination.
- e element\_type**  
Prints out the number of elements of element type. See element types above. Multiple types can be specified. For example, **-e IDSM** specifies all the valid element types.
- h | -?** Prints out usage description.
- l 0 / 1** Allow (0) or prevent (1) media removal.
- n num** Prints the serial number of the drive in the slot specified by *num* (if information is available).
- p device** Specifies the pass-through device file to the library device. If the **LIBRARY** environment variable is set then the **-p** option may be omitted.
- q** Prints out Vendor ID, Product ID and Product Rev standard inquiry information.
- r element\_type**  
Prints out the status (FULL/EMPTY/NONE) of element slots of element type(s). See element types above. If barcode information is supported by the device, this will be appended to the status. Multiple types can be specified. For example, **-r IDSM** specifies all the valid element types.
- s <element\_type><num>**  
Specifies the element type and slot number (<num>) for the move medium source. There should be no space between the element type and the slot number. For example, **-sS1** specifies a Storage element in slot number 1. This option cannot be specified more than

twice per invocation.

**-d** *<element\_type><num>*

Specifies the element type and slot number for the move medium destination. There should be no space between the element type and the slot number. For example, **-dD3** specifies a Data Transfer element in slot number 3. This option cannot be specified more than twice per invocation.

## RETURN VALUE

**mc** returns 0 upon successful completion and -1 otherwise.

## DIAGNOSTICS

**ERROR: 0x5 Illegal Request: 0x3b0d Medium Destination element full**

The above error message could be a result of the command **mc -s S2 -d D1** that was used to move media to an embedded drive that is already full.

**ERROR: /dev/scsi/3: No such file or directory**

If the default SCSI pass-through device file does not exist and no other device file is specified, then the above error message will be printed.

## EXAMPLES

Using a DDS-2 autoloader with a six-cartridge magazine as an example:

To see the status of the autoloader's Data Transfer and Storage element types:

```
mc -r DS
```

The following shows an example of output from the above command. The output indicates that there is an item of media in slot 2 (ST\_slot\_2), an item of media in the embedded drive (DT\_slot\_1), and all the other slots are empty.

```
DT_slot_1 FULL
ST_slot_1 EMPTY
ST_slot_2 FULL
ST_slot_3 EMPTY
ST_slot_4 EMPTY
ST_slot_5 EMPTY
ST_slot_6 EMPTY
```

To move media from an embedded drive to slot 5 and then move media from slot 2 to an embedded drive:

```
mc -s D1 -d S5 -s S2 -d D1
```

To check if a move from a Data Transfer element to a Storage element is possible:

```
mc -c DS
```

The following shows an example of output from the above command. The output indicates that moves from Data Transfer element types to Storage element types are valid.

```
DT->ST: TRUE
```

## WARNINGS

Note for all HP DDS autoloaders: After the **mc** command has been used for the first time, the autoloader will enter into random mode. Once in random mode, all front panel button features are disabled except for the Eject Button. To go back to stacker mode, the magazine must be ejected and then reinserted.

Note for some autoloaders or libraries: The media must be unloaded using the **mt offline** command prior to using the **mc** command.

## DEPENDENCIES

The **mc** command supports all HP-supported devices that comply with the SCSI-2 Medium Changer command specification.



A SCSI pass-through driver must be configured and the device file created before this command can be used to manipulate the autoloader.

For devices connected to HPPB SCSI interface cards, the **spt** pass-through driver must be configured. See *scsi\_pt(7)*.

For devices connected to all other SCSI interface cards (HSC, PCI, or EISA), the **sctl** pass-through driver must be configured. See *scsi\_ctl(7)*.

**AUTHOR**

**mc** was developed by Hewlett-Packard.

**FILES**

`/dev/scsi/3`      Default pass-through device file.

**SEE ALSO**

*mt(1)*, *scsi(7)*, *scsi\_ctl(7)*, *scsi\_pt(7)*.

**NAME**

mkboot, rmboot - install, update or remove boot programs from disk

**SYNOPSIS**

```
/usr/sbin/mkboot [-b boot_file_path] [-c [-u] | -f | -h | -u] [-i included_lif_file]
[-p preserved_lif_file] [-l | -H | -W] [-v] device

/usr/sbin/mkboot [-a auto_file_string] [-v] device

/usr/sbin/rmboot device
```

**DESCRIPTION**

**mkboot** is used to install or update boot programs on the specified device file.

The position on *device* at which boot programs are installed depends on the disk layout of the device. **mkboot** examines *device* to discover the current layout and uses this as the default. If the disk is uninitialized, the default is LVM layout. The default can be overridden by the **-l**, **-H**, or **-W** options.

Boot programs are stored in the boot area in Logical Interchange Format (LIF), which is similar to a file system. For a device to be bootable, the LIF volume on that device must contain at least the **ISL** (the initial system loader) and **HPUX** (the HP-UX bootstrap utility) LIF files. If, in addition, the device is an LVM physical volume, the **LABEL** file must be present (see *lvlnboot(1M)*).

**Options**

**mkboot** recognizes the following options:

- a *auto\_file\_string*** If the **-a** option is specified, **mkboot** creates an autoexecute file **AUTO** on *device*, if none exists. **mkboot** deposits *auto\_file\_string* in that file. If this string contains spaces, it must be quoted so that it is a single parameter.
- b *boot\_file\_path*** If this option is given, boot programs in the pathname specified by *boot\_file\_path* are installed on the given device.
- c** If this option is specified, **mkboot** checks if the available space on *device* is sufficient for the boot programs. If the **-i** option is also specified, **mkboot** checks if each *included\_lif\_file* is present in the boot programs. If the **-p** option is specified, it checks if each *preserved\_lif\_file* is present on the *device*. If all these checks succeed, **mkboot** exits with a status code of 0. If any of these checks fail, **mkboot** exits with a status code of 1. If the verbose option is also selected, a message is also displayed on the standard output.
- f** This option forces the information contained in the boot programs to be placed on the specified *device* without regard to the current swapping status. Its intended use is to allow the boot area to grow without having to boot the system twice (see **-h** option).  
  
This option should only be used when the system is in the single user state.  
  
This could be a dangerous operation because swap space that is already allocated and possibly in use will be overwritten by the new boot program information. A message is also displayed to the standard output stating that the operator should immediately reboot the system to avoid system corruption and to reflect new information on the running system.  
  
A safer method for reappportioning space is to use the **-h** option.
- h** This option is valid only if *device* has the Whole Disk layout.  
  
Specifying this option shrinks the available space allocated to swap in the LIF header by the amount required to allow the installation of the new boot programs specified by *boot\_file\_path*.  
  
After the LIF header has been modified, reboot the system to reflect the new swap space on the running system. At this point, the new boot programs can be installed and the system rebooted again to reflect the new boot programs on the running system. This is the safe method for accomplishing the capability of the **-f** option.  
  
This option is valid only if *device* has the Whole Disk layout.

|                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-H</b>                           | If this option is specified, <b>mkboot</b> treats <i>device</i> to be a Hard Partition layout disk. This option cannot be used along with the <b>-l</b> and <b>-W</b> options.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>-i</b> <i>included_lif_file</i>  | If the <b>-i</b> option is specified one or more times, <b>mkboot</b> copies each <i>included_lif_file</i> and ignores any other LIF files in the boot programs. The sole exceptions to this rule are the files <b>ISL</b> and <b>HPUX</b> , which are copied without regard to the <b>-i</b> options. If <i>included_lif_file</i> is also specified with the <b>-p</b> option, the <b>-i</b> option is ignored. If the <b>-i</b> option is used with <b>LABEL</b> as its argument and the file <b>LABEL</b> does not exist in the boot programs, and <i>device</i> is an LVM layout disk or the <b>-l</b> option is used, <b>mkboot</b> creates a minimal <b>LABEL</b> file on <i>device</i> which will permit the system to boot on <i>device</i> , possibly without swap or dump.                                                                     |
| <b>-l</b>                           | If this option is used, <b>mkboot</b> treats <i>device</i> as an LVM layout disk, regardless of whether or not it is currently set up as one. This option cannot be used along with the <b>-H</b> and <b>-W</b> options.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>-p</b> <i>preserved_lif_file</i> | If the <b>-p</b> option is specified one or more times, <b>mkboot</b> keeps each specified <i>preserved_lif_file</i> intact on <i>device</i> . If <i>preserved_lif_file</i> also appears as an argument to the <b>-i</b> option, that <b>-i</b> option is ignored. This option is typically used with the autoexecute file, <b>AUTO</b> , and with the LVM file, <b>LABEL</b> .<br><br>If <b>LABEL</b> is specified as an argument to the <b>-p</b> option and <b>LABEL</b> does not exist on the <i>device</i> , and if the layout is LVM, <b>mkboot</b> creates a minimal <b>LABEL</b> file. In general, if <i>preserved_lif_file</i> is not on the <i>device</i> , <b>mkboot</b> fails. An exception to this condition is if the <i>preserved_lif_file</i> is <b>LABEL</b> and the layout is not LVM, in which case the <b>LABEL</b> file is ignored. |
| <b>-u</b>                           | If <b>-u</b> is specified, <b>mkboot</b> uses the information contained in the LIF header to identify the location of the swap area, boot area, and raw I/O so that installation of the boot programs does not violate any user data.<br><br>Normally, the LIF header information is overwritten on each invocation of <b>mkboot</b> . This option is typically used with the <b>-W</b> option, to modify boot programs on a disk that is actively supporting swap and/or raw I/O.                                                                                                                                                                                                                                                                                                                                                                       |
| <b>-v</b>                           | If this option is specified, <b>mkboot</b> displays its actions, including the amount of swap space available on the specified device.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>-W</b>                           | If this option is specified, <b>mkboot</b> treats <i>device</i> as a disk having the Whole Disk layout. This option cannot be used along with the <b>-l</b> and <b>-H</b> options. This option will also fail on a disk having large-file enabled HFS filesystem.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <i>device</i>                       | Install the boot programs on the given device special file. The specified <i>device</i> can identify either a character-special or block-special device. However, <b>mkboot</b> requires that both the block and character device special files be present. <b>mkboot</b> attempts to determine whether <i>device</i> is character or block special by examining the specified path name. For this reason, the complete path name must be supplied. If <b>mkboot</b> is unable to determine the corresponding device file, a message is written to the display, and <b>mkboot</b> exits.                                                                                                                                                                                                                                                                 |

**rmboot** removes the boot programs from the boot area.

## EXAMPLES

Install default boot programs on the specified disk, treating it as an LVM disk:

```
mkboot -l /dev/dsk/c0t5d0
```

Use the existing layout, and install only SYSLIB and ODE files and preserve the EST file on the disk:

```
mkboot -i SYSLIB -i ODE -p EST /dev/rdsk/c0t5d0
```

Install only the SYSLIB file and retain the ODE file on the disk. Use the Whole Disk layout. Use the file **/tmp/boot1f** to get the boot programs rather than the default. (The **-i ODE** option will be ignored):

```
mkboot -b /tmp/boot1f -i SYSLIB -i ODE -p ODE -W /dev/rdsk/c0t5d0
```

**WARNINGS**

If *device* has a Whole Disk layout, a file system must reside on the device being modified.

When executing from a recovery system, the **mkboot** command (if used) must be invoked with the **-f** option; otherwise it will not be able to replace the boot area on your disk.

If *device* is, or is intended to become an LVM physical volume, *device* must specify the whole disk.

If *device* is, or is intended to become a Hard Partitioned disk, *device* must specify section 6.

**DEPENDENCIES**

**mkboot** and **rmboot** fail if file system type on *device* is not HFS.

**LVM and Hard Partition Layouts**

The **-f**, **-h**, and **-u** options are not supported.

**AUTHOR**

**mkboot** and **rmboot** were developed by HP.

**FILES**

|                          |                                                                |
|--------------------------|----------------------------------------------------------------|
| <b>/usr/lib/uxbootlf</b> | file containing default boot programs                          |
| <b>ISL</b>               | initial system loader                                          |
| <b>HPUX</b>              | HP-UX bootstrap and installation utility                       |
| <b>AUTO</b>              | defines default/automatic boot behavior (see <i>hpux(1M)</i> ) |
| <b>LABEL</b>             | used by LVM                                                    |
| <b>RDB</b>               | diagnostics tool                                               |
| <b>IOMAP</b>             | diagnostics tool                                               |

**SEE ALSO**

*boot(1M)*, *hpux(1M)*, *isl(1M)*, *lif(4)*, *lvlnboot(1M)*, *mkfs(1M)*, *newfs(1M)*.

**NAME**

mkfs (generic) - construct a file system

**SYNOPSIS**

```
/usr/sbin/mkfs [-F FStype] [-o specific_options] [-V] special [operands]
/usr/sbin/mkfs [-F FStype] [-m] [-V] special
```

**DESCRIPTION**

The **mkfs** command creates a file system by writing on the special file *special*. *operands* are listed on file system specific manual pages (see "SEE ALSO").

**Options**

**mkfs** recognizes the following options:

- F *FStype*** Specify the file system type on which to operate (see *fstyp*(1M) and *fs\_wrapper*(5)). If this option is not included on the command line, then the file system type is determined from the file **/etc/fstab** by matching *special* with an entry in that file. If there is no entry in **/etc/fstab**, then the file system type is determined from the file **/etc/default/fs**.
- m** Display the command line that was used to create the file system. The file system must already exist. This option provides a means of determining the parameters used to construct the file system.
- o *specific\_options*** Specify options specific to the file system type. *specific\_options* is a list of suboptions and/or keyword/attribute pairs intended for an *FStype*-specific module of the command. See the file system specific manual entries for a description of the *specific\_options* that are supported, if any.
- V** Echo the completed command line, but perform no other action. The command line is generated by incorporating the specified options and arguments with other information derived from **/etc/fstab**. This option allows the user to verify the command line.

**EXAMPLES**

Execute the **mkfs** command to create a 32MB HFS file system on **/dev/dsk/c1t2d0**:

```
mkfs -F hfs /dev/dsk/c1t2d0 32768
```

Execute the **mkfs** command on an HFS file system, **/dev/dsk/c1t2d0**, to recreate the command that was used to create the file system on **/dev/dsk/c1t2d0**:

```
mkfs -F hfs -m /dev/dsk/c1t2d0
```

**AUTHOR**

**mkfs** was developed by HP and the University of California, Berkeley.

**FILES**

```
/etc/default/fs Specifies the default file system type.
/etc/fstab      Static information about the file systems.
```

**SEE ALSO**

chmod(1), bdf(1M), df(1M), fsadm(1M), fsck(1M), fstyp(1M), mkfs\_hfs(1M), mkfs\_vxfs(1M), newfs(1M), fstab(4), group(4), passwd(4), fs\_wrapper(5).

**STANDARDS CONFORMANCE**

**mkfs**: SVID3

**NAME**

mkfs (hfs) - construct an HFS file system

**SYNOPSIS**

```
/usr/sbin/mkfs [-F hfs] [-d] [-L|-S] [-V] [-o specific_options] special
[size [nsect ntrack blksize fragsize ncpg minfree rps nbp1]]

/usr/sbin/mkfs [-d] [-F hfs] [-L|-S] [-V] [-o specific_options]
special [proto [nsect ntrack blksize fragsize ncpg minfree rps nbp1]]

/usr/sbin/mkfs [-F hfs] [-m] [-V] special
```

**Remarks**

HFS file systems are normally created with the **newfs** command (see *newfs\_hfs(1M)*).

**DESCRIPTION**

The **mkfs** command constructs an HFS file system by writing on the special file *special*. The **mkfs** command builds the file system with a root directory and a **lost+found** directory (see *fsck\_hfs(1M)*). The **FS\_CLEAN** magic number for the file system is stored in the superblock.

The **mkfs** command creates the file system with a rotational delay value of zero (see *tunefs(1M)*).

**Options**

**mkfs** recognizes the following options:

- F hfs** Specify the HFS file system type.
- d** This option allows the **mkfs** command to make the new file system in an ordinary file. In this case, *special* is the name of an existing file in which to create the file system. When this option is used, the size of the new file system cannot be defaulted. It must either be specified on the command line following *special*, or if a prototype file is being used, it must be the second token in the prototype file as usual.
- L|-S** There are two types of HFS file systems, distinguished mainly by directory formats that place different limits on the length of file names.  
 If **-L** is specified, build a long-file-name file system that allows directory entries (file names) to be up to **MAXNAMLEN** (255) bytes long.  
 If **-S** is specified, build a short-file-name file system that allows directory entries (file names) to be up to **DIRSIZ** (14) bytes long.  
 If neither **-L** nor **-S** is specified, build a file system of the same type as the root file system.
- m** Display the command line that was used to create the file system. The file system must already exist. This option provides a means to determine the parameters used to construct the file system.
- V** Echo the completed command line, but perform no other action. The command line is generated by incorporating the user-specified options and other information derived from **/etc/fstab**. This option allows the user to verify the command line.
- o *specific\_options*** Specify a list of comma separated suboptions and/or keyword/attribute pairs from the list below.  
**largefiles|nolargefiles**  
 Controls the *largefile featurebit* for the file system. The default is **nolargefiles**. This means the bit is not set, and files created on the file system will be limited to less than 2 gigabytes in size. If **largefiles** is specified, the bit is set and the maximum size for files created on the file system is not limited to 2 gigabytes (see *mount\_hfs(1M)* and *fsadm\_hfs(1M)*).

**Arguments**

**mkfs** recognizes the following arguments:

*special* The file name of a special file.

One of the following arguments can be included after *special*:

|              |                                                                                                                                                                                                                                                                                                                                                                               |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>size</i>  | The number of <b>DEV_BSIZE</b> blocks in the file system. <b>DEV_BSIZE</b> is defined in <b>&lt;sys/param.h&gt;</b> . The default value is the size of the entire disk or disk section minus any swap or boot space requested.<br><br>The size of HFS file systems are limited by <b>UFS_MAXDEVLK</b> (defined in <b>&lt;sys/fs.h&gt;</b> ) to 256GB-1 or 268,435,455 blocks. |
| <i>proto</i> | The name of a file that can be opened. The <b>mkfs</b> command assumes it is a prototype file and takes its directions from that file. See "Prototype File Structure" below.                                                                                                                                                                                                  |

The following optional arguments allow fine-tune control over file system parameters:

|                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>nsect</i>    | The number of sectors per track on the disk. The default value is 32 sectors per track.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <i>ntrack</i>   | The number of tracks per cylinder on the disk. The default value is 16 tracks per cylinder.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <i>blksize</i>  | The primary block size for files on the file system. Valid values are: 4096, 8192, 16384, 32768, and 65536. The default value is 8192 bytes.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <i>fragsize</i> | The fragment size for files on the file system. <i>fragsize</i> represents the smallest amount of disk space to be allocated to a file. It must be a power of two no smaller than <b>DEV_BSIZE</b> and no smaller than one-eighth of the file system block size. The default value is 1024 bytes.                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <i>nctp</i>     | The number of disk cylinders per cylinder group. This number must be in the range 1 to 32. The default value is 16 cylinders per group.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <i>minfree</i>  | The minimum percentage of free disk space allowed. The default value is 10 percent.<br><br>Once the file system capacity reaches this threshold, only users with appropriate privileges can allocate disk blocks.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <i>rps</i>      | The number of disk revolutions per second. The default value is 60 revolutions per second.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <i>nbpi</i>     | The density of inodes in the file system specified as the number of bytes per inode. The default value is 6144 bytes per inode.<br><br>This number should reflect the expected average size of files in the file system. If fewer inodes are desired, a larger number should be used; if more inodes are desired, a smaller number should be used.<br><br><b>Note:</b> The number of inodes that will be created in each cylinder group of a file system is approximately the size of the cylinder group divided by the number of bytes per inode, up to a limit of 2048 inodes per cylinder group. If the size of the cylinder group is large enough to reach this limit, the default number of bytes per inode will be increased. |

### Prototype File Structure

A prototype file describes the initial file structure of a new file system. The file contains tokens separated by spaces or newline characters. It cannot contain comments.

The first token is the name of a file to be copied onto block zero as the bootstrap program (usually **/etc/BOOT**). If the file name is "", no bootstrap code is placed on the device. The second token is a number specifying the number of **DEV\_BSIZE** blocks in the file system.

The next three tokens specify the mode, user ID, and group ID of the root directory of the new file system, followed by the initial contents of the root directory in the format described for a directory file below, and terminated with a **\$** token.

A file specification consists of four tokens giving the name, mode, user ID, and group ID, and an initial contents field. The syntax of the initial contents field depends on the mode.

A name token is a file name that is valid for the file system. The root directory does not have a name token.

A mode token is a 6-character string. The first character specifies the type of the file. It can be one of the following characters:

- Regular file

**b** Block special file  
**c** Character special file  
**d** Directory  
**l** Symbolic link  
**L** Hard link

The second character of a mode token is either **u** or **-** to specify set-user-ID mode or not. The third character of a mode token is either **g** or **-** to specify the set-group-ID mode or not. The rest of a mode token is a three-digit octal number giving the *owner*, *group*, and *other* read, write, and execute permissions (see *chmod(1)*).

The user-ID and group-ID tokens define the owner of the file. These values can be specified numerically or with symbolic names that appear in the current password and group databases.

**Regular file.** The initial contents field is the path name of an existing file in the current file system whose contents and size are copied to the new file.

**Block or character special file.** The initial contents field is two numeric tokens that specify the major and minor device numbers.

**Directory file.** The initial contents field is a list of file specifications for the entries in the directory. The list is terminated with a **\$** token. Directories can be nested. For each directory, the **mkfs** command automatically makes the **.** and **..** entries.

**Symbolic link.** The initial contents field is a path name that is used as the path to which the symbolic link should point.

**Hard link.** The initial contents field is a path name that is used as the name of a file within the new file system to which the entry should be linked. The mode, user-ID and group-ID tokens of this entry are ignored; they are taken from the target of the link. The target of the link must be listed before the entry specifying the link. Hard links to directories are not permitted.

With the exception of the permissions field of the mode token (which is always an octal number), all numeric fields can be specified in hexadecimal (using a leading **0x**), octal (using a leading **0**), or decimal.

Here is a sample prototype specification. The indentation clarifies the directory recursion.

```

/etc/BOOT
12288
d--555 bin bin
sbin d--755 bin bin
    init ---555 bin bin /sbin/init
    savecore ---555 bin bin /sbin/savecore
    $
dev d--555 bin bin
    b0 b--640 root sys 0 0x0e0000
    c0 c--640 root sys 4 0x0e0000
    $
etc d--755 bin bin
    init l--777 bin bin /sbin/init
    passwd ---444 bin bin /etc/passwd
    group ---444 bin bin /etc/group
    $
usr d--755 bin bin
    bin d--755 bin bin
        sh ---555 bin bin /usr/bin/sh
        rsh L--555 bin bin /usr/bin/sh
        su -u-555 root bin /usr/bin/su
        mailq l--777 bin bin /usr/sbin/sendmail
        $
    sbin d--755 bin bin
        sendmail -ug555 root mail /usr/sbin/sendmail
        $
    $
$

```



**Access Control Lists**

Every file with one or more optional ACL entries consumes an extra (continuation) inode. If you anticipate significant use of ACLs on a new file system, you can allocate more inodes by reducing the value of *nbpi* appropriately. The small default value typically causes allocation of many more inodes than are actually necessary, even with ACLs. To evaluate your need for extra inodes, run the **bdf -i** command on existing file systems. For more information on access control lists, see *acl(5)*.

**EXAMPLES**

Execute the **mkfs** command to create a 32MB HFS file system on the non-LVM disk */dev/dsk/c1t2d0*:

```
mkfs -F hfs /dev/dsk/c1t2d0 32768
```

Display the command that was used to construct the file system on */dev/dsk/c1t2d0*:

```
mkfs -F hfs -m /dev/dsk/c1t2d0
```

Create an HFS file system within a logical volume */dev/vg01/my\_lvol* of a size equal to the size of *my\_lvol*:

```
mkfs -F hfs /dev/vg01/my_lvol
```

**WARNINGS**

The old **-F** option, from prior releases of *mkfs(1M)*, is no longer supported.

*mkfs\_hfs(1M)* cannot be executed specifying creation of a file system on a whole disk if that disk was previously used as an LVM disk. If you wish to do this, use *mediainit(1)* to reinitialize the disk first.

The **-o largefile** option should be used with care, since older applications will not react correctly when confronted with large files.

**AUTHOR**

**mkfs** was developed by HP and the University of California, Berkeley.

**FILES**

*/var/adm/sbtab* List of locations of the superblocks for the created file system. The **mkfs** command appends entries to this file.

**SEE ALSO**

*chmod(1)*, *bdf(1M)*, *df(1M)*, *fsadm\_hfs(1M)*, *fsck(1M)*, *fsck\_hfs(1M)*, *fsclean(1M)*, *mkfs(1M)*, *mount\_hfs(1M)*, *newfs(1M)*, *newfs\_hfs(1M)*, *dir(4)*, *fs(4)*, *fstab(4)*, *group(4)*, *passwd(4)*, *symlink(4)*, *acl(5)*.

**STANDARDS CONFORMANCE**

**mkfs**: SVID3

**NAME**

mkfs - construct a VxFS file system

**SYNOPSIS**

```
/usr/sbin/mkfs [-F vxfs] [-V] -m special
/usr/sbin/mkfs [-F vxfs] [-V]
[-o [N] [X] [ninode=n] [nau=n] [bsize=n] [logsize=n] [ausize=n] [aufirst=n]
[aupad=n] [version=n] [inosize=n] [largefiles | nolargefiles] ] special size
```

**DESCRIPTION**

**mkfs** creates a VxFS file system by writing on the *special* device file, unless either the **-o N** or **-m** option is specified. *special* must be the first argument after the options are given. The file system is created based on the options and *size* specified on the command line. The numeric *size* specifies the number of sectors in the file system. By default, *size* is specified in units of **DEV\_BSIZE** sectors (currently, 1024 bytes). If *size* is not specified, **mkfs** determines the size of the *special* device.

*size* can also be specified with a suffix to indicate a unit of measure other than sectors. Append **k** or **K** to indicate the value is in kilobytes, **m** or **M** to indicate megabytes, or **g** or **G** to indicate gigabytes. An appended letter can be separated from the number by a space. In that case, enclose the letter and number in a set of quotes, for example:

```
"512 k"
```

**mkfs** builds a file system with a root directory and a **lost+found** directory (see *fsck\_vxfs(1M)*). The file system can have disk layout Version 2, disk layout Version 3, or disk layout Version 4. Version 2 supports dynamic inode allocation. Version 3 adds support for large files and large UIDs. Version 4 adds support for Access Control Lists. On HP-UX 11.1x systems, the default is Version 4. On HP-UX 11.0 systems, the default is Version 3. You can choose the disk layout version with the **version=n** option (see below).

Inode allocation is done dynamically. There are a minimum number of inodes allocated to the file system by **mkfs**, and any other inode allocations are done on an as-needed basis during file system use.

**Options**

**mkfs** recognizes the following options:

- F vxfs** Specify the VxFS file system type.
- m** Display the command line which was used to create the file system. The file system must already exist.
- V** Echo the completed command line, but do not execute the command. The command line is generated by incorporating the user-specified options and other information derived from **/etc/fstab**. This option allows the user to verify the command line.
- o specific\_options**

Specify options specific to the VxFS file system type. *specific\_options* is a comma separated list of suboptions and/or keyword/attribute pairs.

The following *specific\_options* are valid on a VxFS file system:

- N** Do not write the file system to the *special* file. This option gives all the information needed to create a file system but does not create it.
- X** Create a file system in a file. Used for debugging only.

**aufirst=n**

*n* is the starting block number, in blocks of size *bsize*, of the first allocation unit. This option allows the allocation units to be aligned to a particular boundary, such as a cylinder boundary. This option is not applicable to Version 3 and 4 disk layouts, which always set the starting block number to zero.

**aupad=n**

*n* is the size, in blocks of size *bsize*, of the padding to leave between the end of the inode list and the first data block in each allocation unit. This option allows the data blocks of an allocation unit to be aligned to a particular boundary, such as a cylinder boundary. This option is ignored for Version 3 and 4 disk layouts.

**ausize=n**

*n* is the size, in blocks of size *bsize*, of an allocation unit. This option is not

applicable to the Version 3 and 4 disk layouts, which fix the allocation unit size at 32768. For disk layout Version 2, this is an alternate way of specifying the number of allocation units. This option cannot be used with the **nau** option. With **ausize**, the last allocation unit on the file system can be shorter than the others. If the last allocation unit on the file system is not long enough to contain an entire allocation unit header, the size of the resulting file system is shortened to the end of the last complete allocation unit. This parameter may not exceed 262144 blocks.

The algorithm used to choose the default value is rather complicated, but is intended to balance the number of allocation units (4 to 16 is a good range), the size of the allocation units (at least 32768 blocks), and other factors.

**bsize**=*n*

*n* is the block size for files on the file system and represents the smallest amount of disk space allocated to a file. *n* must be a power of 2 selected from the range 1024 to 8192. The default is 1024 for file systems less than 8 gigabytes, 2048 for file systems less than 16 gigabytes, 4096 for file systems less than 32 gigabytes, and 8192 for larger file systems.

**inosize**=*n*

*n* is the on-disk inode structure size for files on the file system. The valid values are 256 and 512 bytes. The default is 256. There is usually no reason to increase the inode size, and not using the default value can adversely affect file system performance.

**largefiles** | **nolargefiles**

Valid only for the Version 3 and 4 disk layouts. Controls the *largefiles* flag for the file system. If **largefiles** is specified, the bit is set and files two gigabytes or larger can be created. If **nolargefiles** is specified, the bit is cleared and files created on the file system are limited to less than two gigabytes. The default is **nolargefiles**. See *fsadm\_vxfs(1M)*.

NOTE: Large files are supported on HP-UX 10.20 systems and above. Be careful when implementing large file system capability. System administration utilities such as backup may not operate correctly if they are not large-file aware.

**logsize**=*n*

*n* is the number of file system blocks to allocate for an activity logging area. The minimum value for Version 2 and 3 disk layouts is 32 blocks. The minimum value for Version 4 disk layouts is the number of blocks that make the log no less than 256K. The maximum value for *n* is the number of blocks that make the log no greater than 16384K. This means that for a *bsize* of 1024, 2048, 4096, or 8192 bytes the maximum value of *logsize* is 16384, 8192, 4096, or 2048 blocks, respectively. To avoid wasting space, the default *logsize* is 1024 blocks for a file system 8 megabytes or larger, 128 blocks for a file system 2 megabytes or larger but less than 8 megabytes, and 32 blocks for a file system less than 2 megabytes.

A large log provides better performance on metadata-intensive workloads. A small log uses less space on the disk and leaves more room for file data. For example, and NFS-intensive workload performs better with a large log; a small floppy device requires a small log.

NOTE: The amount of virtual memory required by **fsck** (see *fsck\_vxfs(1M)*) to check a VxFS file system is proportional to the size of the log. The maximum amount of virtual memory used is twice the size of the log. Therefore, the sum of physical memory and swap space must be at least 32 MB to ensure that a file system with a 16384K log can be cleaned. On small systems, take care not to create a file system with a log larger than half the available swap space. A maximum log size of one third the total of memory and swap space is a good rule of thumb (see *swapinfo(1M)*).

**nau**=*n*

*n* is the number of allocation units on the file system. This option is not applicable to the Version 3 and 4 disk layouts. For disk layout Version 2, if *nau* is specified, then *ausize* is determined by evenly dividing the sectors among the allocation units. By default, the number of allocation units is based on the value

of *ausize*.

**ninode**=*n*

*n* is the maximum number of inodes in the file system. The actual maximum number of inodes is *n* rounded up to an appropriate boundary. The digit 0 and the string **unlimited** both mean that the number of inodes is unlimited. The default is **unlimited**.

**version**=*n*

*n* is the VxFS disk layout version number. Valid values are 2, 3, and 4.

## EXAMPLES

To use **mkfs** to create a VxFS file system on **/dev/rdisk/c0t6d0**:

```
mkfs -F vxfs /dev/rdisk/c0t6d0 1024
```

To use **mkfs** to determine the command that was used to create the VxFS file system on **/dev/rdisk/c0t6d0**:

```
mkfs -F vxfs -m /dev/rdisk/c0t6d0
```

To create a VxFS file system on **/dev/vgqa/lvol1**, with a Version 4 disk layout and largefiles capability:

```
mkfs -F vxfs -o version=4,largefiles /dev/vgqa/lvol1
```

## WARNINGS

If you want to reuse a special device that was previously used by LVM, you must first wipe out all the LVM information remaining on the disk. Use *pvremove*(1M) to remove the LVM information before executing *mkfs\_vxfs*(1M). (You can also remove the LVM information by initializing the device with *mediainit*(1), but that is slower.)

The **-o largefiles** option should be used with care, since older applications will not react correctly when confronted with large files.

## RETURN VALUE

Upon successful completion, the **mkfs** command returns a value of 0. The return value is 1 if a syntax error occurs. Other errors return a value of 32.

## SEE ALSO

*bdf*(1M), *chmod*(1), *chown*(2), *df\_vxfs*(1M), *fs\_vxfs*(4), *fsadm\_vxfs*(1M), *fsck\_vxfs*(1M), *group*(4), *mkfs*(1M), *mount\_vxfs*(1M), *newfs\_vxfs*(1M), *passwd*(4), *pvremove*(1M).

## STANDARDS CONFORMANCE

**mkfs** : SVID3

**NAME**

mklost+found - make a lost+found directory for *fsck*(1M)

**SYNOPSIS**

**/usr/sbin/mklost+found**

**DESCRIPTION**

The **mklost+found** command creates a directory named **lost+found** in the current directory. It also creates several empty files which are then removed to provide empty slots for the **fsck** command (see *fsck*(1M)).

For an HFS file system, the **mklost+found** command is not normally needed since the **mkfs** command automatically creates the **lost+found** directory when a new file system is created (see *mkfs*(1M)).

**AUTHOR**

**mklost+found** was developed by the University of California, Berkeley.

**SEE ALSO**

*fsck*(1M), *mkfs*(1M).

**NAME**

mknod - create special files

**SYNOPSIS**

`/sbin/mknod name c major minor`

`/sbin/mknod name b major minor`

`/sbin/mknod name p`

**DESCRIPTION**

The **mknod** command creates the following types of files:

- Character device special file (first SYNOPSIS form),
- Block device special file (second SYNOPSIS form),
- FIFO file, sometimes called a named pipe (third SYNOPSIS form).

*name* is the path name of the file to be created. The newly created file has a default mode that is readable and writable by all users (0666), but the mode is modified by the current setting of the user's file mode creation mask (see *umask(1)*).

**Character and Block Special Files**

Character device special files are used for devices that can transfer single bytes at a time, such as nine-track magnetic tape drives, printers, plotters, disk drives operating in "raw" mode, and terminals. To create a character special file, use the **c** argument.

Block device special files are used for devices that usually transfer a block of data at a time, such as disk drives. To create a block device special file, use the **b** argument.

The remaining arguments specify the device that will be accessible through the new special file:

|              |                                                                                                                          |
|--------------|--------------------------------------------------------------------------------------------------------------------------|
| <i>major</i> | The major number specifies the major device type (for example, the device driver number).                                |
| <i>minor</i> | The minor number specifies the device location, which is typically, but not always, the unit, drive, and/or line number. |

The *major* and *minor* values can each be specified in hexadecimal, octal, or decimal, using C language conventions (decimal: no leading zero; octal: leading zero; hexadecimal: leading 0x).

The assignment of major and minor device numbers is specific to each HP-UX system. Refer to the System Administrator manuals supplied with your system for details.

Only users who have appropriate privileges can use **mknod** to create a character or block device special file.

**FIFO files**

To create a FIFO (named pipe or buffer) file, use the **p** argument. You can also use the **mkfifo** command for this purpose (see *mkfifo(1)*). All users can use **mknod** to create FIFO files.

**WARNINGS****Access Control Lists**

In HFS file systems, optional ACL entries can be added to special files and FIFOs with the **chacl** command (see *chacl(1)*). In JFS file systems, optional ACL entries can be added to special files and FIFOs with the **setacl** command (see *setacl(1)*). However, system programs are likely to silently change or eliminate the optional ACL entries for these files.

**SEE ALSO**

*chacl(1)*, *mkdir(1)*, *mkfifo(1)*, *setacl(1)*, *umask(1)*, *lsdev(1M)*, *sam(1M)*, *mknod(2)*, *acl(5)*, *aclv(5)*, *mknod(5)*.

HP-UX System Administrator manuals.

**STANDARDS CONFORMANCE**

*mknod*: SVID2, SVID3, XPG2

**NAME**

mkpdf - create a Product Description File from a prototype PDF

**SYNOPSIS**

**mkpdf** [-c *comment\_string*] [-n] [-r *alternate\_root*] *prototype\_PDF* *new\_PDF*

**DESCRIPTION**

The **mkpdf** program reads a prototype PDF and generates a new PDF (see *pdf(4)*) that reflects the current status of the file system files defined by path names in the prototype file.

If *pathname* is a directory, the *size*, *version*, *checksum*, and *linked\_to* target fields are forced to be empty. If the file is a device, the *version*, *checksum*, and *linked\_to* fields are forced to be empty and the *size* field contains the major and minor device numbers.

If a path name in *prototype\_PDF* is prefaced with a question mark (?), the file is assumed to be an optional file. This file is processed in the same manner as all other files except that, if the file does not exist, values provided in the prototype are reproduced, and the ?, is passed through to *new\_PDF*. If a path name is not preceded with ?, and the file does not exist on the file system, an error is reported and no entry is added to *new\_PDF*.

If a dash (-) is used for *prototype\_PDF* or *new\_PDF*, **mkpdf** assumes that standard input and/or standard output, respectively, is being used for the appropriate value.

Comments in *prototype\_PDF* are supported as follows: Lines beginning with the percent character (%) are generally passed through, in order, to *new\_PDF*, except that any "% Product Description File" and "% total size is ..." lines are removed to prevent duplication of these automatically generated lines in *new\_PDF* when *prototype\_PDF* is a PDF. Lines beginning with a pound character (#), and lines containing only the newline character (\n) are not passed through to *new\_PDF*. Note that blank space preceding these special characters is not allowed and will generally result in error messages about files not found.

A size summary is produced as a comment at the end of the PDF.

**Options**

- c *comment\_string* Insert a string that contains a comment about the product for which this PDF is being generated. This is used as a second comment line of the PDF. See *pdf(4)* for a description of the first comment line. If this option is not specified, no second comment line is produced.
- n Record numerical representation of user ID from */etc/passwd* and group ID from */etc/group* for each file instead of the usual text representation.
- r *alternate\_root* Prefix the string *alternate\_root* to each path name in the prototype (after removing the optional ?) to form a modified path name to be used to gather attributes for the entry. Default is an empty string.

**EXAMPLES**

Given a file **Proto** with contents:

```
/usr/bin/basename
/usr/bin/cat
/usr/bin/ccat
/usr/bin/dirname
/usr/bin/grep
/usr/bin/ls
/usr/bin/ll:::::::::/usr/bin/ls
/usr/bin/su
```

the command:

```
mkpdf -c "fileset TEST, Release 1.0" Proto -
```

produces the PDF shown in the EXAMPLE section of *pdf(4)*.

The following example creates a totally new PDF for the fileset **ALBA\_CORE**. The *pathname* and *linked\_to* are taken from the prototype PDF. All other fields are generated from the file system.

```
mkpdf /tmp/ALBA_CORE /system/ALBA_CORE/new.pdf
```

The next example shows how to create a completely new PDF from just a list of files. The PDF for the files under the `/PRODUCT` directory is created by executing the `find` command (see *find(1)*) on all the files in the directory structure under `/PRODUCT`. A `/` is edited onto the beginning of each path name to make it absolute. The path names are then piped to `mkpdf`. The `-r` option specifies that a root of `/PRODUCT` should be prefixed to each path name while the directory is being searched. A `-` in the *prototype\_PDF* position specifies that `stdin` is being used for the prototype PDF file. The resulting PDF does not contain the `/PRODUCT` prefix. Note that, with only a list of path names, the *linked\_to* field of linked files will not conform to the convention explained in *pdf(4)*.

```
cd /PRODUCT
find * -print | sed -e 's:^:/' |
mkpdf -r /PRODUCT - PDF
```

## RETURN VALUE

Upon completion, `mkpdf` returns one of the following values:

- 0 Successful completion.
- 1 Nonoptional files in the prototype file were not found.
- 2 `mkpdf` encountered other problems.

## DIAGNOSTICS

*filename: no such file or directory*

A nonoptional file was not found on the file system and will not appear in the new PDF.

## WARNINGS

Sizes reported do not reflect blocks allocated to directories.

Use of PDFs is discouraged since this functionality is obsolete and is being replaced with Software Distributor (see *sd(4)*).

## AUTHOR

`mkpdf` was developed by HP.

## SEE ALSO

*pdfck(1M)*, *pdfdiff(1M)*, *pdf(4)*.



**NAME**

mksf - make a special (device) file

**SYNOPSIS**

```
/sbin/mksf [-C class | -d driver] [-D directory] [-H hw-path] [-I instance] [-q|-v]
[driver-options] [special-file]

/sbin/mksf [-C class | -d driver] [-D directory] [-H hw-path] -m minor [-q|-v] [-r]
special-file
```

**DESCRIPTION**

The **mksf** command makes a special file in the devices directory, normally **/dev**, for an existing device, a device that has already been assigned an instance number by the system. The device is specified by supplying some combination of the **-C**, **-d**, **-H**, and **-I** options. If the options specified match a unique device in the system, **mksf** creates a special file for that device; otherwise, **mksf** prints an error message and exits. If required, **mksf** creates any subdirectories relative to the device installation directory that are defined for the resulting special file.

For most drivers, **mksf** has a set of built-in driver options, *driver-options*, and special-file naming conventions. By supplying some subset of the driver options, as in the first form above, the user can create a special file with a particular set of characteristics. If a *special-file* name is specified, **mksf** creates the special file with that special file name; otherwise, the default naming convention for the driver is used.

In the second form, the *minor* number and *special-file* name are explicitly specified. This form is used to make a special file for a driver without using the built-in driver options in **mksf**. The **-r** option specifies that **mksf** should make a character (raw) device file instead of the default block device file for drivers that support both.

**Options**

**mksf** recognizes the following options:

- C *class*** Match a device that belongs to a given device class, *class*. Device classes can be listed with the **lsdev** command (see **lsdev(1M)**). They are defined in the files in the directory **/usr/conf/master.d**. This option is not valid for pseudo devices. This option cannot be used with **-d**.
- d *driver*** Match a device that is controlled by the specified device driver, *driver*. Device drivers can be listed with the **lsdev** command (see **lsdev(1M)**). They are defined in the files in the directory **/usr/conf/master.d**. This option cannot be used with **-C**.
- D *directory*** Override the default device installation directory **/dev** and install the special files in *directory* instead. *directory* must exist; otherwise, **mksf** displays an error message and exits. See WARNINGS.
- H *hw-path*** Match a device at a given hardware path, *hw-path*. Hardware paths can be listed with the **ioscan** command (see **ioscan(1M)**). A hardware path specifies the addresses of the hardware components leading to a device. It consists of a string of numbers separated by periods (.), such as **52** (a card), **52.3** (a target address), and **52.3.0** (a device). If a hardware component is a bus converter, the following period, if any, is replaced by a slash (/) as in **2**, **2/3**, and **2/3.0**. This option is not valid for pseudo devices.
- I *instance*** Match a device with the specified *instance* number. Instances can be listed with the **-f** option of the **ioscan** command (see **ioscan(1M)**). This option is not valid for pseudo devices.
- m *minor*** Create the special file with the specified minor number *minor*. The format of *minor* is the same as that given in **mknod(1M)** and **mknod(5)**.
- q** Quiet option. Normally, **mksf** displays a message as each driver is processed. This option suppresses the driver message, but not error messages. See the **-v** option.
- r** Create a character (raw) special file instead of a block special file.
- v** Verbose option. In addition to the normal processing message, display the name of each special file as it is created. See the **-q** option.

**Naming Conventions**

Many special files are named using the `c`*card*`t`*target*`d`*device* naming convention. These variables have the following meaning wherever they are used.

- card* The unique interface card identification number from `ioscan` (see `ioscan(1M)`). It is represented as a decimal number with a typical range of 0 to 255.
- target* The device target number, for example the address on a HP-FL or SCSI bus. It is represented as a decimal number with a typical range of 0 to 15.
- device* A address unit within a device, for example, the unit in a HP-FL device or the LUN in a SCSI device. It is represented as a decimal number with a typical range of 0 to 15.

**Special Files**

The driver-specific options (*driver-options*) and default special file names (*special-file*) are listed below.

`asio0`

- `-a access-mode` Port access mode (0-2). The default access mode is 0 (Direct connect). The *access-mode* meanings are:

| <i>access-mode</i> | <b>Port Operation</b> |
|--------------------|-----------------------|
| 0                  | Direct connect        |
| 1                  | Dial out modem        |
| 2                  | Dial in modem         |

- `-c` CCITT.
- `-f` Hardware flow control (RTS/CTS).
- `-i` Modem dialer. Cannot be used with `-l`.
- `-l` Line printer. Cannot be used with `-i`.
- `-p port` Multiplexer port number (0 for **built-in serial port**; 0-1 for **SAS console ports**). The default port number is 0.
- `-r fifo-trigger` *fifo-trigger* should have a value between 0 and 3. The following table shows the corresponding FIFO trigger level for a given *fifo-trigger* value.

| <i>fifo-trigger</i> | <b>Receive FIFO Trigger Level</b> |
|---------------------|-----------------------------------|
| 0                   | 1                                 |
| 1                   | 4                                 |
| 2                   | 8                                 |
| 3                   | 14                                |

- `-t` Transparent mode (normally used by diagnostics).
- `-x xmit-limit` *xmit-limit* should have a value between 0 and 3. The following table shows the corresponding transmit limit for a given *xmit-limit* value.

| <i>xmit-limit</i> | <b>Transmit Limit</b> |
|-------------------|-----------------------|
| 0                 | 1                     |
| 1                 | 4                     |
| 2                 | 8                     |
| 3                 | 12                    |

- special-file* The default special file name depends on the *access-mode* and whether the `-i` and `-l` options are used.

| <i>access-mode</i> | <i>-i</i> | <i>-l</i> | <b>Special File Name</b> |
|--------------------|-----------|-----------|--------------------------|
| —                  | no        | yes       | <i>ccardp0_lp</i>        |
| 2                  | no        | no        | <i>tttydcardp0</i>       |
| 1                  | no        | no        | <i>culcardp0</i>         |
| 0                  | yes       | no        | <i>cuacardp0</i>         |
| 0                  | no        | no        | <i>tttycardp0</i>        |

audio

*-f format* Audio format (0-3). The *format* meanings are:

| <i>format</i> | <b>Audio Format</b>       | <b>File Name Modifier</b><br><i>format-mod</i> |
|---------------|---------------------------|------------------------------------------------|
| 0             | No change in audio format |                                                |
| 1             | 8-bit Mu-law              | <b>U</b>                                       |
| 2             | 8-bit A-law               | <b>A</b>                                       |
| 3             | 16-bit linear             | <b>L</b>                                       |

*-o output-dest* Output destination (0-4). The *output-dest* should have a value between 0 and 4. The following table shows the corresponding output destinations for a given *output-dest* value.

| <i>output-dest</i> | <b>Output Destinations</b> | <b>File Name Modifier</b><br><i>output-mod</i> |
|--------------------|----------------------------|------------------------------------------------|
| 0                  | All outputs                | <b>B</b>                                       |
| 1                  | Headphone                  | <b>E</b>                                       |
| 2                  | Internal Speaker           | <b>I</b>                                       |
| 3                  | No output                  | <b>N</b>                                       |
| 4                  | Line output                | <b>L</b>                                       |

*-r* Raw, control access. This option cannot be used with either the *-f* or *-o* options.  
*special-file* The default special file name depends on the options specified.

| <b>Options</b> | <b>Special File Name</b>              |
|----------------|---------------------------------------|
| <i>-r</i>      | <i>audioCtl_card</i>                  |
| <i>-f 0</i>    | <i>audio_card</i>                     |
| all others     | <i>audiooutput-modformat-mod_card</i> |

The optional *output-mod* and *format-mod* values are given in the tables above. Note the underscore ( *\_* ) before *card* in each special file name. Also note that for *card 0*, each file will be linked to a simpler name without the trailing *\_card*.

autox0 schgr

Note that *-i* cannot be used with either *-r* or *-p*.

*-i* Ioctl; create picker control special file.

*-p optical-disk[:last-optical-disk]* The optical disk number (starts with 1). If the optional *:last-optical-disk* is given then special files for the range of disks specified will be created.

*-r* Raw; create character, not block, special file.

*special-file* A special file cannot be given if a range of optical disks is given with the *-p* option. If one is given for the single disk case, the name will have an **a** appended to the end for the A-side device and a **b** appended to the end for the B-side device. The default special file name depends on whether the *-r* option is used.

| -r  | Special File Name                                                                                      |
|-----|--------------------------------------------------------------------------------------------------------|
| yes | <code>rac/ccardttargetdevice_optical-diska</code><br><code>rac/ccardttargetdevice_optical-diskb</code> |
| no  | <code>ac/ccardttargetdevice_optical-diska</code><br><code>ac/ccardttargetdevice_optical-diskb</code>   |

Note the underscore ( `_` ) between *device* and *optical-disk*.

CentIf

`-h handshake-mode`

Handshake mode. Valid values range from 1 to 6:

| <i>handshake-mode</i> | Handshake operation                        |
|-----------------------|--------------------------------------------|
| 1                     | Automatic NACK/BUSY handshaking            |
| 2                     | Automatic BUSY only handshaking            |
| 3                     | Bidirectional read/write                   |
| 4                     | Stream mode (NSTROBE only, no handshaking) |
| 5                     | Automatic NACK/BUSY with pulsed NSTROBE    |
| 6                     | Automatic BUSY with pulsed NSTROBE         |

*special-file*

The default special file name is `ccardt0d0_1p` for *handshake-mode* 2 and `ccardt0d0hhandshake-mode_1p` for all others. For SCentIf, the only valid handshake value is 2 (Automatic BUSY only handshaking).

consp1

`-r fifo-trigger` *fifo-trigger* should have a value between 0 and 3. The following table shows the corresponding FIFO trigger level for a given *fifo-trigger* value.

| <i>fifo-trigger</i> | Receive FIFO Trigger Level |
|---------------------|----------------------------|
| 0                   | 1                          |
| 1                   | 4                          |
| 2                   | 8                          |
| 3                   | 14                         |

`-t`

Transparent mode (normally used by diagnostics).

`-x xmit-limit` *xmit-limit* should have a value between 0 and 3. The following table shows the corresponding transmit limit for a given *xmit-limit* value.

| <i>xmit-limit</i> | Transmit Limit |
|-------------------|----------------|
| 0                 | 1              |
| 1                 | 4              |
| 2                 | 8              |
| 3                 | 12             |

*special-file*

The default special file name is as follows:

| Special File Name      |
|------------------------|
| <code>ttycardp0</code> |

disc3

`-f`

Floppy.

`-r`

Raw; create character, not block, special file.

`-s section`

The section number.

*special-file*

The default special file name depends on whether the `-r` and `-s` options are used:

| -r  | -s  | Special File Name                                                                                                                 |
|-----|-----|-----------------------------------------------------------------------------------------------------------------------------------|
| yes | no  | <b>r</b> dsk/ <i>c</i> card <i>t</i> target <i>d</i> device and<br><b>r</b> floppy/ <i>c</i> card <i>t</i> target <i>d</i> device |
| yes | yes | <b>r</b> dsk/ <i>c</i> card <i>t</i> target <i>d</i> devices <i>section</i>                                                       |
| no  | no  | <b>d</b> sk/ <i>c</i> card <i>t</i> target <i>d</i> device and<br><b>f</b> loppy/ <i>c</i> card <i>t</i> target <i>d</i> device   |
| no  | yes | <b>d</b> sk/ <i>c</i> card <i>t</i> target <i>d</i> devices <i>section</i>                                                        |

hil

Note that only one of **-a**, **-k**, or **-r** is allowed.

- a** *address*      The link address (1-7).
- k**                Cooked keyboard.
- n**                The hil controller device.
- special-file*      The default special file name depends on the **-a**, **-k**, and **-r** options:

| Option    | Special File Name                        |
|-----------|------------------------------------------|
| <b>-a</b> | <b>hil_</b> <i>card</i> . <i>address</i> |
| <b>-k</b> | <b>hil</b> kbd_ <i>card</i>              |
| <b>-r</b> | <b>r</b> hil_ <i>card</i>                |

Note the underscore ( **\_** ) before *card*. Also note that for *card* 0, each file will be linked to a simpler name without *\_card*, either **hil***address*, **hil**kbd, or **r**hil.

lan0 lan1 lan2 lan3

Note that only one of **-e** or **-i** is allowed.

- e**                Ethernet protocol.
- i**                IEEE 802.3 protocol.
- t**                Transparent mode (normally used by diagnostics).
- special-file*      The default special file name depends on the **-e**, **-i**, and **-t** options:

| Option    | -t  | Special File Name             |
|-----------|-----|-------------------------------|
| <b>-e</b> | no  | <b>ether</b> <i>card</i>      |
| <b>-e</b> | yes | <b>diag/ether</b> <i>card</i> |
| <b>-i</b> | no  | <b>lancard</b>                |
| <b>-i</b> | yes | <b>diag/lancard</b>           |

lantty0

- e**                Exclusive access.
- special-file*      The default special file name depends on whether the **-e** option is used:

| -e  | Special File Name              |
|-----|--------------------------------|
| no  | <b>lantty</b> <i>card</i>      |
| yes | <b>diag/lantty</b> <i>card</i> |

lpr2 lpr3

- c**                Capital letters. Convert all output to uppercase.
- e**                Eject page after paper-out recovery.
- n**                No form-feed.
- o**                Old paper-out behavior (abort job).
- r**                Raw.
- t**                Transparent mode (normally used by diagnostics).
- w**                No wait. Don't retry errors on open.
- special-file*      The default special file name depends on whether the **-r** option is used:



| -r  | Special File Name                   |
|-----|-------------------------------------|
| no  | <code>ccardttargetdevice_lp</code>  |
| yes | <code>ccardttargetdevice_rlp</code> |

`mux0 mux2 mux4 eisa_mux0 pci_mux0`

**-a** *access-mode* Port access mode (0-2). The default access mode is 0 (Direct connect). The *access-mode* meanings are:

| <i>access-mode</i> | Port Operation |
|--------------------|----------------|
| 0                  | Direct connect |
| 1                  | Dial out modem |
| 2                  | Dial in modem  |

**-c** CCITT.  
**-f** Hardware flow control (RTS/CTS).  
**-i** Modem dialer. Cannot be used with **-l**.  
**-l** Line printer. Cannot be used with **-i**.  
**-p** *port* Multiplexer port number (0-15 for `mux0` and `mux2`; 0-1 for `mux4`; a1 - a16, b1 - b16, c1 - c16 & etc for the `eisa_mux0` or `pci_mux0`). Some MUX cards controlled by a particular driver have fewer than the maximum supported ports.  
**-t** Transparent mode (normally used by diagnostics).  
*special-file* The default special file name depends on the *access-mode* and whether the **-i** and **-l** options are used. The term "card" below refers to the Instance number of the mux card.

| <i>access-mode</i> | -i  | -l  | Special File Name           |
|--------------------|-----|-----|-----------------------------|
| —                  | no  | yes | <code>ccardpport_lp</code>  |
| 2                  | no  | no  | <code>tttydcardpport</code> |
| 1                  | no  | no  | <code>culcardpport</code>   |
| 0                  | yes | no  | <code>cuaocardpport</code>  |
| 0                  | no  | no  | <code>tttycardpport</code>  |

`pflop sflop`

**-r** Raw; create character, not block, special file.  
*special-file* The default special file name depends on whether the **-r** option is used:

| -r  | Special File Name                       |
|-----|-----------------------------------------|
| no  | <code>floppy/ccardttargetdevice</code>  |
| yes | <code>rfloppy/ccardttargetdevice</code> |

`ps2`

Note that only one of **-a**, or **-p** is allowed.

**-a** *auto\_device* Autosearch device. An *auto\_device* value of 0 means first mouse; a value of 1 means first keyboard.  
**-p** *port* PS2 port number.  
*special-file* The default special file name depends on the **-a**, and **-p** options:

| Option      | Special File Name     |
|-------------|-----------------------|
| <b>-a</b> 0 | <code>ps2mouse</code> |
| <b>-a</b> 1 | <code>ps2kbd</code>   |
| <b>-p</b>   | <code>ps2_port</code> |

Note the underscore ( `_` ) before *port*.

`SAS console ports` See `asio0`.

`SCentIf` See `CentIf`.

scc1

**-a** *access-mode* Port access mode (0–2). The default access mode is 0. The *access-mode* meanings are:

| <i>access-mode</i> | Port Operation |
|--------------------|----------------|
| 0                  | Direct connect |
| 1                  | Dial out modem |
| 2                  | Dial in modem  |

- b** Port B.
- c** CCITT.
- i** Modem dialer. Cannot be used with **-l**.
- l** Line printer. Cannot be used with **-i**.

*special-file* The default special file name depends on the *access-mode* and whether the **-i** and **-l** options are used.

| <i>access-mode</i> | <b>-i</b> | <b>-l</b> | Special File Name               |
|--------------------|-----------|-----------|---------------------------------|
| —                  | no        | yes       | <i>ccardport_lp</i>             |
| 2                  | no        | no        | <i>tttyccardport</i>            |
| 1                  | no        | no        | <i>culccardport</i>             |
| 0                  | yes       | no        | <i>cua<del>ccard</del>port</i>  |
| 0                  | no        | no        | <i>ttty<del>ccard</del>port</i> |

**schgr** See **autox0**.

sdisk

- r** Raw; create character, not block, special file.
- s** *section* The section number.
- special-file* The default special file name depends on whether the **-r** and **-s** options are used:

| <b>-r</b> | <b>-s</b> | Special File Name                                           |
|-----------|-----------|-------------------------------------------------------------|
| yes       | no        | <i>rdsk/ccard<del>target</del>device</i>                    |
| yes       | yes       | <i>rdsk/ccard<del>target</del>device<del>s</del>section</i> |
| no        | no        | <i>dsk/ccard<del>target</del>device</i>                     |
| no        | yes       | <i>dsk/ccard<del>target</del>device<del>s</del>section</i>  |

**sflop** See **pflop**.

stape

- a** AT&T-style rewind/close.
- b** *bpi* Bits per inch or tape density. The recognized values for *bpi* are:  
BEST, D1600, D3480, D3480C, D3590, D3590C, D6250, D6250C, D800, D8MM\_8200, D8MM\_8200C, D8MM\_8500, D8MM\_8500C, DDS1, DDS1C, DDS2, DDS2C, NOMOD, QIC\_1000, QIC\_11, QIC\_120, QIC\_1350, QIC\_150, QIC\_2100, QIC\_24, QIC\_2GB, QIC\_525, QIC\_5GB, DLT\_42500\_24, DLT\_42500\_56, DLT\_62500\_64, DLT\_81633\_64, DLT\_62500\_64C, DLT\_81633\_64C,  
or a decimal number density code.
- c** [*code*] Compression with optional compression code. The optional decimal code is used to select a particular compression algorithm on drives that support more than one compression algorithm. This option must be specified at the end of an option string. See *mt(7)* for more details.
- e** Exhaustive mode. This option allows the driver to experiment with multiple configuration values in an attempt to access the media. The default behavior is to use only the configuration specified.
- n** No rewind on close.
- p** Partition one.

- s** [*block-size*] Fixed block size mode. If a numeric *block-size* is given, it is used for a fixed block size. If the **-s** option is used alone, a device-specific default fixed block size is used. This option must be specified at the end of an option string.
- u** UC Berkeley-style rewind/close.
- w** Wait (disable immediate reporting).
- x** *index* Use the *index* value to access the tape device driver property table entry. Recognized values for *index* are decimal values in the range 0 to 30.
- special-file* Put all tape special files in the `/dev/rmt` directory. This is required for proper maintenance of the Tape Property Table (see *mt(7)*). Device files located outside the `/dev/rmt` directory may not provide consistent behavior across system reboots. The default special file names are dependent on the tape drive being accessed and the options specified. All default special files begin with `rmt/c card t target d device`. See *mt(7)* for a complete description of the default special file naming scheme for tapes.

## tape2

- a** AT&T-style rewind/close.
- b** *bpi* Bits per inch or tape density. The recognized values for *bpi* are:  
BEST, D1600, D3480, D3480C, D6250, D6250C, D800, D8MM\_8200,  
D8MM\_8200C, D8MM\_8500, D8MM\_8500C, DDS1, DDS1C, DDS2, DDS2C,  
NOMOD, QIC\_1000, QIC\_11, QIC\_120, QIC\_1350, QIC\_150, QIC\_2100,  
QIC\_24, QIC\_2GB, QIC\_525, QIC\_5GB, DLT\_42500\_24, DLT\_42500\_56,  
DLT\_62500\_64, DLT\_81633\_64, DLT\_62500\_64C, DLT\_81633\_64C,  
or a decimal number density code.
- c** [*code*] Compression with optional compression code. The optional decimal code is used to select a particular compression algorithm on drives that support more than one compression algorithm. This option must be specified at the end of an option string. See *mt(7)* for more details.
- n** No rewind on close.
- o** Console messages disabled.
- t** Transparent mode, normally used by diagnostics.
- u** UC Berkeley-style rewind/close.
- w** Wait (disable immediate reporting).
- x** *index* Use the *index* value to access the tape device driver property table entry. The recognized values for *index* are decimal values in the range 0 to 30.
- z** RTE compatible close.
- special-file* Put all tape special files in the `/dev/rmt` directory. This is required for proper maintenance of the Tape Property Table (see *mt(7)*). Device files located outside the `/dev/rmt` directory may not provide consistent behavior across system reboots. The default special file names are dependent on the tape drive being accessed and the options specified. All default special files begin with `rmt/c card t target d device`. See *mt(7)* for a complete description of the default special file naming scheme for tapes.

## RETURN VALUE

**mksf** exits with one of the following values:

- 0 Successful completion.
- 1 Failure. An error occurred.

## DIAGNOSTICS

Most of the diagnostic messages from **mksf** are self-explanatory. Listed below are some messages deserving further clarification. Errors cause **mksf** to abort immediately.

### Errors

Ambiguous device specification



Matched more than one device in the system. Use some combination of the **-d**, **-C**, **-H**, and **-I** options to specify a unique device.

**No such device in the system**

No device in the system matched the options specified. Use **ioscan** to list the devices in the system (see *ioscan(1M)*).

**Device driver *name* is not in the kernel**

**Device class *name* is not in the kernel**

The indicated device driver or device class is not present in the kernel. Add the appropriate device driver and/or device class to the **config** input file and generate a new kernel (see *config(1M)*).

**Device has no instance number**

The specified device has not been assigned an instance number. Use **ioscan** to assign an *instance* to the device.

**Directory *directory* doesn't exist**

The *directory* argument of the **-D** option doesn't exist. Use **mkdir** to create the directory (see *mkdir(1)*).

**EXAMPLES**

Make a special file named **/dev/printer** for the line printer device associated with instance number 2.

```
mksf -C printer -I 2 /dev/printer
```

Make a special file, using the default naming convention, for the tape device at hardware path 8.4.1. The driver-specific options specify 1600 bits per inch and no rewind on close.

```
mksf -C tape -H 8.4.1 -b D1600 -n
```

**WARNINGS**

Many commands and subsystems assume their device files are in **/dev**; therefore, the use of the **-D** option is discouraged.

**AUTHOR**

**mksf** was developed by HP.

**FILES**

```
/dev/config      I/O system special file
/etc/mtconfig    Tape driver property table database
```

**SEE ALSO**

**mkdir(1)**, **config(1M)**, **insf(1M)**, **ioscan(1M)**, **lsdev(1M)**, **mknod(1M)**, **rmsf(1M)**, **mknod(2)**, **ioconfig(4)**, **mknod(5)**, **mt(7)**.

**NAME**

mk\_kernel - build a bootable HP-UX kernel and/or kernel modules

**SYNOPSIS**

```
usr/sbin/mk_kernel [-o pathname] [-s system_file] [-S] [-v]
/usr/sbin/mk_kernel -M module_name [[-M module_name]...] [-v]
```

**DESCRIPTION**

**mk\_kernel** builds an executable file which can be used as a bootable kernel and kernel modules if any are configured. If the build succeeds, the newly built kernel is called **vmunix\_test**, and the kernel function set directory (where the function set directory is the directory structure containing the set of modules that correspond to the kernel) is called **dlkm.vmunix\_test**. The file and directory are placed in the build directory, as defined below.

The build directory is the target directory where **mk\_kernel** places files and directories. In addition to the kernel and kernel modules, files such as **conf.c**, **conf.o**, and **tune.h** are also placed in the build directory.

If the path used to designate the system file is **/stand/system**, the build directory is **/stand/build**. If another path is used to designate the system file, the build directory is the current working directory. System files for the kernel modules are expected to be found in **/stand/system.d**. Libraries for the kernel are expected to be found in **/usr/conf/lib**. The master file used is the composite of files found under **/usr/conf/master.d**.

If the **-o** option is not specified, the kernel file and kernel function set directory remain in the working directory. If **-o /stand/vmunix** is specified, the target kernel file and kernel function set directory are not overwritten. The new kernel file and the kernel function set directory are moved to the default path as the system shuts down or starts up. The previous versions of the file and directory are renamed to **/stand/vmunix.prev** and **/stand/dlkm.vmunix.prev**. Until the system reboots, the new kernel file and the directory must be kept as **vmunix\_test** and **dlkm.vmunix\_test**, respectively.

If the **-o** option is specified with other than **/stand/vmunix**, the kernel file and kernel function set directory is created or updated immediately. In case the administrator needs to place these targets to the system default path, the **kmupdate** command must be used to trigger the replacement. Manually replacing the default kernel (**/stand/vmunix**) or any file under the kernel function set directory (**/stand/dlkm**) must be avoided.

**mk\_kernel** exits with no action if the environment variable **SW\_INITIAL\_INSTALL** has the value of 1. **SW\_INITIAL\_INSTALL** is exported by SD with that value only when the system is undergoing its initial software system installation.

**Options**

**mk\_kernel** recognizes the following options.

**-M module\_name**

Specify the module to configure. No kernel image will be generated. For details see *config(1M)*.

**-o pathname**

Specify the target file path. The created kernel file, **vmunix\_test**, is moved from the build directory to the path specified by the option argument. The associated kernel function set directory, **dlkm.vmunix\_test**, is moved to the same destination directory.

If the default kernel, **/stand/vmunix**, is specified or the **-o** option is not specified, the created kernel file does not replace **/stand/vmunix** and remains as **vmunix\_test**.

The kernel file and associated kernel function set directory are automatically moved to **/stand/vmunix** and **/stand/dlkm** during either shutdown or startup.

**-s system\_file**

Specify the kernel template file. If this option is not specified, the system file **/stand/system** is used.

**-S**

Specify that all configured kernel modules are to be statically linked into the kernel. For details see *config(1M)*.

**-v**

Verbose mode.

**RETURN VALUE**

**mk\_kernel** returns 0 upon normal completion, and 1 if an error occurred.

**DIAGNOSTICS**

Messages and warnings are sent to **stdout**. Messages from **config** and other commands are displayed when invoked from **mk\_kernel**. Errors cause **mk\_kernel** to halt immediately; warnings allow the program to continue.

**EXAMPLES**

```
mk_kernel -o /stand/vmunix
```

Uses the file **/stand/system** to build a new kernel and kernel module(s). The new kernel file is placed in **/stand/build/vmunix\_test** upon success. Kernel function set directory is placed in **/stand/build/dlkm.vmunix\_test**. These files are moved automatically to **/stand/vmunix** and **/stand/dlkm** during shutdown or startup. The current set is saved as **/stand/vmunix.prev** and **/stand/dlkm.vmunix.prev**.

```
mk_kernel -s /mnt/altsys/stand/system.new
```

Uses the file **/mnt/altsys/stand/system.new** to build a new kernel and kernel module(s). The new kernel is named **vmunix\_test** in the present working directory. The kernel function set directory, **dlkm.vmunix\_test**, is placed in the current working directory.

```
mk_kernel -s /stand/system -o /tmp/new_kernel
```

Uses the file **/stand/system** to build a new kernel and kernel module(s). The new kernel file is placed in **/tmp/new\_kernel**. The kernel function set directory is in **/tmp/dlkm.new\_kernel**. If the administrator wants to use this kernel as the default kernel, the **kmupdate** command can be used.

**WARNINGS**

System administrators are expected to treat the kernel and dlkm, *kernel\_name*, as a set. Do not manually copy the kernel or manually update the current kernel file with its associated kernel function set directory. To update the default kernel, always use the **kmupdate** command.

Kernel modules are separate objects to be independently configured into the system without requiring a reboot. To accomplish this, the kernel relies on several files under the kernel function set directory.

- kernel file: *kernel\_name* or **/stand/vmunix**
- kernel function set directory: **dlkm.kernel\_name** or **/stand/dlkm**

The kernel function set directory contains kernel modules, a module database file, and a kernel symbol table file. These files and directories are expected to be found in a directory whose name matches the booted kernel. If the kernel function set directory is not found, the dynamically loadable kernel module feature is disabled.

**FILES**

|                                      |                                                         |
|--------------------------------------|---------------------------------------------------------|
| <b>/stand/vmunix</b>                 | Default kernel                                          |
| <b>/stand/dlkm</b>                   | Default kernel function set directory                   |
| <b>/stand/system</b>                 | Default system file                                     |
| <b>/stand/build/vmunix_test</b>      | Kernel built by <b>mk_kernel</b>                        |
| <b>/stand/build/dlkm.vmunix_test</b> | Kernel function set directory build by <b>mk_kernel</b> |
| <b>/stand/vmunix.prev</b>            | Saved kernel                                            |
| <b>/stand/dlkm.vmunix.prev</b>       | Saved kernel function set directory                     |

**SEE ALSO**

**config(1M)**, **kmupdate(1M)**.

**NAME**

modprpw - modify protected password database

**SYNOPSIS**

**modprpw** [-E|-V] [-l|-n [*domain*]]

**modprpw** [-x] [-l|-n [*domain*]] *username*

**modprpw** [-A|-e|-v|-k] [-m *field=value,...*] [-l|-n [*domain*]] *username*

**DESCRIPTION**

**modprpw** updates the user's protected password database settings. This command is available only to the superuser in a trusted system.

Usage other than via SAM, and/or modifications out of sync with `/etc/passwd` or NIS+ tables, may result in serious database corruption and the inability to access the system.

All updated values may be verified using *getprpw*(1M).

The database contains information for both local and NIS+ users. However, some NIS+ information is kept on the master. Since a user may be both local and NIS+, **modprpw** uses the *nsswitch.conf*(4) default if neither `-l` nor `-n` are specified.

**Options**

**modprpw** sets user's parameters as defined by the options specified. At least one option is required. If a field is not specified in the option then its value remains unchanged in the database.

**modprpw** recognizes the following options...

**-A** To add a new user entry and to return a random password which the new user must use to login the first time. This entry has to be created with the given username and the **-m uid=value**.

Error is returned if the user already exists.

May be combined with one of the `-l` or `-n` options. It also adds entries to the NIS+ tables, if `-n` is specified.

Unlike *useradd*(1M), it does not create nor populate the home directory, and it does not update `/etc/passwd`.

**-E** This option is specified WITHOUT a user name to expire all user's passwords. It goes through the protected password database and zeroes the successful change time of all users. The result is all users will need to enter a new password at their next login.

May be combined with one of `-l` or `-n` options.

**-e** This option is specified with a user name to expire the specified user's password. It zeroes the successful change time.

May be combined with options `-l`, `-m`, `-n`.

**-k** To unlock/enable a user's account that has become disabled, except when the lock is due to a missing password or \* password.

May be combined with options `-l`, `-m`, `-n`.

**-l** This option specifies to modify data for a local user. It cannot be specified with the `-n` option. This option must be specified with other options.

**-m** Modify the database field to the specified value and/or resets locks. Valid with one of `-A`, `-e`, `-v`, `-k` options; and one of `-l`, `-n` options.

A list of database fields may be used with comma as a delimiter. An "invalid-opt" is printed, and processing terminates, if a list of database fields passed to `-m` contains an invalid database field.

Boolean values are specified as YES, NO, or DFT for system default values (`/etc/passwd/auth/system/default`). Numeric values are specified as positive numbers, 0, or -1. If the *value* -1 is specified, the numeric value in the database is removed, allowing the system default value to be used. Time values are specified in days, although the database keeps them in seconds.

No aging is present if the following 4 database parameters are all zero: `u_minchg`, `u_exp`, `u_life`, `u_pw_expire_warning`.

Unless specified by **n/a**, all database fields can be set. They are listed below in the order shown in **prot.h**. The database fields are fully explained in *prpwd(4)*.

| <b>FIELD=VALUE</b>    | <b>DATABASE FIELD</b>                                                                                                                                                                             |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>n/a</b>            | database <b>u_name</b> .                                                                                                                                                                          |
| <b>uid=value</b>      | database <b>u_id</b> .<br>Set the <i>uid</i> of the user. No sanity checking is done on this value.                                                                                               |
| <b>n/a</b>            | database <b>u_pwd</b> .                                                                                                                                                                           |
| <b>n/a</b>            | database <b>u_owner</b> .                                                                                                                                                                         |
| <b>bootpw= value</b>  | database <b>u_bootauth</b> .<br>Set boot authorization privilege, <b>YES/NO/DFT</b> . <b>NO</b> removes it from the user file.                                                                    |
| <b>audit= value</b>   | database <b>u_auditid</b> .<br>Set audit id. Automatically limited not to exceed the next available id.                                                                                           |
| <b>audflg= value</b>  | database <b>u_auditflag</b> .<br>Set audit flag.                                                                                                                                                  |
| <b>mintm= value</b>   | database <b>u_minchg</b> =(value*86400).<br>Set the minimum time interval between password changes (days). 0 = none. Same as non-trusted mode minimum time.                                       |
| <b>maxpwn= value</b>  | database <b>u_maxlen</b> .<br>Set the maximum password length for system generated passwords.                                                                                                     |
| <b>exptm= value</b>   | database <b>u_exp</b> =(value*86400).<br>Set password expiration time interval (days). 0 = expired. Same as non-trusted mode maximum time.                                                        |
| <b>lftm= value</b>    | database <b>u_life</b> .<br>Set password life time interval (days). 0 = infinite.                                                                                                                 |
| <b>n/a</b>            | database <b>u_succhg</b> .<br>Modified by options <b>e</b> , <b>E</b> , <b>v</b> , <b>V</b> , maybe <b>k</b> .                                                                                    |
| <b>n/a</b>            | database <b>u_unsucchg</b> .                                                                                                                                                                      |
| <b>acctexp= value</b> | database <b>u_acct_expire</b> =(value*86400+now).<br>Set account expiration time interval (days). This interval is added to "now" to form the value in the database (database 0 = no expiration). |
| <b>llog= value</b>    | database <b>u_llogin</b> .<br>Set the last login time interval (days). Used with <b>u_succlog</b> .                                                                                               |
| <b>expwarn= value</b> | database <b>u_pw_expire_warning</b> =(value*86400).<br>Set password expiration warning time interval (days). 0 = none.                                                                            |
| <b>n/a</b>            | database <b>u_pswduser</b> . Obsoleted field.                                                                                                                                                     |
| <b>usrpick= value</b> | database <b>u_pickpw</b> .<br>Set whether User Picks Password, <b>YES/NO/DFT</b> .                                                                                                                |
| <b>sysnpnw= value</b> | database <b>u_genpwd</b> .<br>Set whether system generates pronounceable passwords, <b>YES/NO/DFT</b> .                                                                                           |
| <b>rstrpw= value</b>  | database <b>u_restrict</b> .<br>Set if generated password is restricted, <b>YES/NO/DFT</b> . If <b>YES</b> , password will be checked for triviality.                                             |
| <b>nullpw= value</b>  | database <b>u_nullpw</b> .                                                                                                                                                                        |

m

Set whether null passwords are allowed, **YES/NO/DFT**. **YES is not recommended!**

**n/a** database **u\_pwchanger**. Obsolescent field.

**admnum= value** database **u\_pw\_admin\_num**. Obsoleted field.

**syschpw= value** database **u\_genchars**.  
Set whether system generates passwords having characters only, **YES/NO/DFT**.

**sysltpw= value** database **u\_genletters**.  
Set whether system generates passwords having letters only, **YES/NO/DFT**.

**timeod= value** database **u\_tod**.  
Set the time-of-day allowed for login.  
The format is:  
*key0Starttime-Endtime, key1Starttime-Endtime,...*  
*keynStarttime-Endtime*

Where *key* has the following values:

**Mo** - Monday  
**Tu** - Tuesday  
**We** - Wednesday  
**Th** - Thursday  
**Fr** - Friday  
**Sa** - Saturday  
**Su** - Sunday  
**Any** - everyday  
**Wk** - Monday -> Friday

and *Starttime* and *Endtime* are in military format: *HHMM*, where:  
 00 <= *HH* <= 23, and 00 <= *MM* <= 59.

**n/a** database **u\_suclog**.

**n/a** database **u\_unsuclog**.

**n/a** database **u\_suctty**.

**n/a** database **u\_numunsuclog**.

**n/a** database **u\_unsuctty**.

**umaxlntr= value** database **u\_maxtries**.  
Set Maximum Unsuccessful Login tries allowed. 0 = infinite.

**alock= value** database **u\_lock**.  
Set the administrator lock, **YES/NO/DFT**.

**-n** Can be specified with or without domain name; i.e., **-n [domain]**. If **-n [domain]** is specified, modifies data for the NIS+ user. The *domain* name must be fully qualified, with a terminating period. If *domain* name is not specified, the local domain will be used.

It cannot be specified with the **-l** option. This option must be specified with other options.

**-v** This option is specified WITHOUT a user name to "validate/refresh" all user's passwords. It goes through the protected password database and sets the successful change time to the current time for all users. The result is that all user's password aging restarts at the current time.

May be combined with one of **-l** or **-n** options.

**-v** This option is specified with a user name to "validate/refresh" the specified user's password. It sets the successful change time to the current time.

May be combined with options **-l**, **-m**, **-n**.

**-x** Delete the user's password and return a random password that the user must later supply to the login process to login and pick a new password. Not valid for root. Also resets locks.

May be combined with one of **-l** or **-n** options.

**RETURN VALUE**

- 0 Success.
- 1 User not privileged.
- 2 Incorrect usage.
- 3 Can not find the entry or file.
- 4 Can not change the entry.
- 5 Not a Trusted System.
- 6 Not a NIS+ user.

**EXAMPLES**

Set the Minimum time between password changes to 12 (days), set the System generates pronounceable password flag to NO, and set the System generates password having characters only flag to YES.

```
modprpw -m mintm=12,sysnpw=NO,syschpw=YES someusr
```

The following example is to restrict the times that user joeblow can get on the system on Mondays and Fridays to 5PM-9PM, and Sundays from 5AM-9AM. Other days are not restricted.

```
modprpw -m timeod=Mo1700-2100,Fr1700-2100,Su0500-0900 joeblow
```

**WARNINGS**

This command is intended for SAM use only. It may change with each release and can not be guaranteed to be backward compatible.

Several database fields interact with others. Side effects may not be apparent until much later.

Special meanings may apply in the following cases:

- an absent field,
- a field without a value,
- a field with a zero value.

Very little, if any checking is done to see if values are valid. It is the user's responsibility to range check values.

**FILES**

|                                       |                             |
|---------------------------------------|-----------------------------|
| <b>/etc/passwd</b>                    | System Password file        |
| <b>/tcb/files/auth/*/*</b>            | Protected Password Database |
| <b>/tcb/files/auth/system/default</b> | System Defaults Database    |

**AUTHOR**

**modprpw** was developed by HP.

**SEE ALSO**

getprpw(1M), prpwd(4), nsswitch.conf(4).

m

**NAME**

mount, umount - mount and unmount file systems

**SYNOPSIS**

```
/usr/sbin/mount [-l] [-p|-v]
/usr/sbin/mount -a [-F FStype] [-eQ]
/usr/sbin/mount [-F FStype] [-eQrV] [-o specific_options] {special|directory}
/usr/sbin/mount [-F FStype] [-eQrV] [-o specific_options] special directory
/usr/sbin/umount [-v] [-V] {special|directory}
/usr/sbin/umount -a [-F FStype] [-h host] [-v]
```

**DESCRIPTION**

The **mount** command mounts file systems. Only a superuser can mount file systems. Other users can use **mount** to list mounted file systems.

The **mount** command attaches *special*, a removable file system, to *directory*, a directory on the file tree. *directory*, which must already exist, will become the name of the root of the newly mounted file system. *special* and *directory* must be given as absolute path names. If either *special* or *directory* is omitted, **mount** attempts to determine the missing value from an entry in the `/etc/fstab` file. **mount** can be invoked on any removable file system, except `/`.

If **mount** is invoked without any arguments, it lists all of the mounted file systems from the file system mount table, `/etc/mnttab`.

The **umount** command unmounts mounted file systems. Only a superuser can unmount file systems.

**Options (mount)**

The **mount** command recognizes the following options:

- a** Attempt to mount all file systems described in `/etc/fstab`. All optional fields in `/etc/fstab` must be included and supported. If the **-F** option is specified, all file systems in `/etc/fstab` with that *FStype* are mounted. If **noauto** is specified in an entry's option list, this entry is skipped. File systems are not necessarily mounted in the order listed in `/etc/fstab`.
- e** Verbose mode. Write a message to the standard output indicating which file system is being mounted.
- F FStype** Specify *FStype*, the file system type on which to operate. See *fstyp(1M)*. If this option is not included on the command line, then it is determined from either `/etc/fstab`, by matching *special* with an entry in that file, or from file system statistics of *special*, obtained by `statfsdev()` (see *statfsdev(3C)*).
- l** Limit actions to local file systems only.
- o *specific\_options*** Specify options specific to each file system type. *specific\_options* is a list of comma separated suboptions and/or keyword/attribute pairs intended for a *FStype*-specific version of the command. See the *FStype*-specific manual entries for a description of the *specific\_options* supported, if any.
- p** Report the list of mounted file systems in the `/etc/fstab` format.
- Q** Prevent the display of error messages that result from an attempt to mount already mounted file systems.
- r** Mount the specified file system as read-only. Physically write-protected file systems must be mounted in this way or errors occur when access times are updated, whether or not any explicit write is attempted.
- h** Unmount only those file systems listed in `/etc/mnttab` that are remote-mounted from *host*.
- v** Report the regular output with file system type and flags; however, the *directory* and *special* fields are reversed.



- v           Echo the completed command line, but perform no other action. The command line is generated by incorporating the user-specified options and other information derived from `/etc/fstab`. This option allows the user to verify the command line.

### Options (umount)

The **umount** command recognizes the following options:

- a           Attempt to unmount all file systems described in `/etc/mnttab`. All optional fields in `/etc/mnttab` must be included and supported. If *FStype* is specified, all file systems in `/etc/mnttab` with that *FStype* are unmounted. File systems are not necessarily unmounted in the order listed in `/etc/mnttab`.
- F *FStype*   Specify *FStype*, the file system type on which to operate. If this option is not included on the command line, then it is determined from `/etc/mnttab` by matching *special* with an entry in that file. If no match is found, the command fails.
- v           Verbose mode. Write a message to standard output indicating which file system is being unmounted.
- v           Echo the completed command line, but perform no other action. The command line is generated by incorporating the user-specified options and other information derived from `/etc/fstab`. This option allows the user to verify the command line.

### EXAMPLES

List the file systems currently mounted:

```
mount
```

Mount the HFS file system `/dev/dsk/c1t2d0` at directory `/home`:

```
mount -F hfs /dev/dsk/c1t2d0 /home
```

Unmount the same file system:

```
umount /dev/dsk/c1t2d0
```

### AUTHOR

**mount** was developed by HP, AT&T, the University of California, Berkeley, and Sun Microsystems.

### FILES

|                          |                                      |
|--------------------------|--------------------------------------|
| <code>/etc/fstab</code>  | Static information about the systems |
| <code>/etc/mnttab</code> | Mounted file system table            |

### SEE ALSO

fsadm(1M), mount\_cdfs(1M), mount\_hfs(1M), mount\_lofs(1M), mount\_nfs(1M), mount\_vxfs(1M), setmnt(1M), mount(2), fstab(4), mnttab(4), fs\_wrapper(5), quota(5).

### STANDARDS CONFORMANCE

**mount**: SVID3

**umount**: SVID3

**NAME**

mountall, umountall - mount and unmount multiple file systems

**SYNOPSIS**

```
/sbin/mountall [-F FStype] [-l|-r] [file_system_table | -]
/sbin/mountall [-l|-r] [-m]
/sbin/mountall [-n]
/sbin/umountall [-F FStype] [-k] [-l|-r]
```

**DESCRIPTION**

**mountall** is used to mount file systems according to *file\_system\_table*. By default, */etc/fstab* is the *file\_system\_table*. If a dash (-) is specified, **mountall** reads *file\_system\_table* from the standard input; the standard input must be in the same format as the */etc/fstab*.

Before each file system is mounted, a check is done using **fsck** (see *fsck(1M)*) to ensure that the file system is mountable. If the file system is not mountable, it is repaired by **fsck** before the mount is attempted.

**umountall** causes all mounted file systems except the non-removable file systems such as **root** to be unmounted.

**Options**

**mountall** and **umountall** recognize the following options:

- F *FStype* Specify the file system type (*FStype*) to be mounted or unmounted.
- l Specify action on local file systems only.
- r Specify action on remote file systems only.
- k Send a **SIGKILL** signal to processes that have files opened.
- m Attempt to mount all the unmounted file systems. This option will not perform the file system consistency check and repair.
- n Perform the file system consistency check and repair on all unmounted file system. This option will not mount the file systems.

**DIAGNOSTICS**

Error and warning messages may originate from **fsck**, **mount**, **fuser**, or **umount**. See *fsck(1M)*, *mount(1M)*, or *fuser(1M)* to interpret the error and warning messages.

**EXAMPLES**

Mount all unmounted file systems listed in */etc/fstab*:

```
mountall
```

Mount all local file systems listed in */etc/fstab*:

```
mountall -l
```

Mount all remote file systems listed in */etc/fstab*:

```
mountall -r
```

Mount all local hfs file systems:

```
mountall -F hfs -l
```

Unmount all NFS file systems and kill any processes that have files opened in the file system:

```
umountall -F nfs -k
```

**WARNINGS**

**umountall**, especially with the **-k** option, should be used with extreme caution, because it can cause severe damage.

The **-n** option may not be available in future releases.

**mountall** may not be effective with some cases of LOFS file systems.

**FILES**

|                          |                                           |
|--------------------------|-------------------------------------------|
| <code>/etc/fstab</code>  | Static information about the file systems |
| <code>/etc/mnttab</code> | Mounted file system table                 |

**SEE ALSO**

`fck(1M)`, `mount(1M)`, `fuser(1M)`, `mnttab(4)`, `fstab(4)`, `signal(2)`.

## NAME

mountd - NFS mount request server

## SYNOPSIS

`/usr/sbin/rpc.mountd [-l log_file] [-t n] [-p]`

## DESCRIPTION

**mountd** is an RPC server that answers file system mount requests. It reads file `/etc/xtab` (described in *exports(4)*) to determine which directories are available to which machines. It also provides information on what file systems are mounted by which clients. This information can be printed using the **showmount** command (see *showmount(1M)*).

**rpc.mountd** can be started at boot time by setting the variable **NFS\_SERVER** to 1 in the file `/etc/rc.config.d/nfsconf`.

## Options

**mountd** recognizes the following options:

- l *log\_file*** Log any errors to the named log file, *log\_file*. Errors are not logged if the **-l** option is not specified.  
The information logged to the file includes the date and time of the error, the host name, process ID and name of the function generating the error, and the error message. Note that different services can share a single log file since enough information is included to uniquely identify each error.
- p** Run from unreserved ports. This option restores the old default behavior on HP-UX. The default has been changed for the mount daemon to run from reserved ports unless this option is set.
- t *n*** Specify tracing level *n*, where *n* can have one of the following values:
  - 1 Errors only (default)
  - 2 Errors, mount requests and mount failures

## WARNINGS

The default behavior of the mount daemon is to run from reserved ports. If the daemon needs to be run from unreserved ports, use the **-p** option.

If a client crashes, executing **showmount** on the server will show that the client still has a file system mounted; i.e., the client's entry is not removed from `/etc/rmtab` until the client reboots and executes **umount -a** (see *showmount(1M)*).

Also, if a client mounts the same remote directory twice, only one entry appears in `/etc/rmtab`. Doing a **umount** of one of these directories removes the single entry and **showmount** no longer indicates that the remote directory is mounted.

## AUTHOR

**mountd** was developed by Sun Microsystems, Inc.

## FILES

`/etc/rmtab` List of all hosts having file systems mounted from this machine

## SEE ALSO

*inetd(1M)*, *mount(1M)*, *portmap(1M)*, *showmount(1M)*, *exports(4)*, *inetd.conf(4)*, *inetd.sec(4)*, *rmtab(4)*, *services(4)*.

**NAME**

mount\_cacheofs - mount CacheFS file systems

**SYNOPSIS**

```
mount -F cacheofs [ generic_options ] -o backfstype= file_system_type
    [ other_cacheFS_options ] special_mount_point
```

**DESCRIPTION**

The CacheFS-specific version of the **mount** command mounts a cached file system; if necessary, it NFS-mounts its back file system. It also provides a number of CacheFS-specific options for controlling the caching process.

**Options**

To mount a CacheFS file system, use the generic **mount** command with the **-F** option followed by the argument **cacheofs**. The following generic **mount** options are available:

- m** Mount the file system without making an entry in the **/etc/mnttab** file.
- O** Overlay mount. Allows the filesystem to be mounted over an existing mount point, making the underlying filesystem inaccessible. If a mount is attempted on a pre-existing mount point without setting this flag, mount will fail with the error:  
**mount -F cacheofs: mount failed Device busy.**
- r** Mount the file system read-only.

The following arguments to the **-o** option are specifically for CacheFS mounts. Use commas to separate multiple options. Note: the **backfstype** argument must be specified.

**backfstype= file\_system\_type**

The file system type of the back file system (for example, **nfs**).

**backpath= path**

Specifies where the back file system is already mounted. If this argument is not supplied, CacheFS determines a mount point for the back file system. The back file system must be read-only.

**cachedir= directory**

The name of the cache directory.

**cacheid= ID** *ID* is a string specifying a particular instance of a cache. If you do not specify a cache ID, CacheFS will construct one.

**write-around | non-shared**

Write modes for CacheFS. The **write-around** mode (the default) handles writes the same as NFS does; that is, writes are made to the back file system, and the affected file is purged from the cache. You can use the **non-shared** mode when you are sure that no one else will be writing to the cached file system. In this mode, all writes are made to both the front and the back file system, and the file remains in the cache.

**noconst** Disables cache consistency checking. By default, periodic consistency checking is enabled. Specify **noconst** only when you know that the back file system will not be modified. Trying to perform cache consistency check using **cfsadmin -s** will result in error. **demandconst** and **noconst** are mutually exclusive.

**demandconst**

Verifies cache consistency only when explicitly requested, rather than the periodic checking that is done by default. A consistency check is requested by using the **-s** option of the **cfsadmin(1M)** command. This option is useful for back file systems that change infrequently, for example, **/usr/bin**. **demandconst** and **noconst** are mutually exclusive.

**local-access**

Causes the front file system to interpret the mode bits used for access checking instead of having the back file system verify access permissions. Do not use this argument with secure NFS.

**purge** Purge any cached information for the specified file system.

**rw | ro** Read-write (default) or read-only.

**suid | nosuid**

Allow (default) or disallow set-uid execution.

**acregmin=*n*** Specifies that cached attributes are held for at least *n* seconds after file modification. After *n* seconds, CacheFS checks to see if the file modification time on the back file system has changed. If it has, all information about the file is purged from the cache and new data is retrieved from the back file system. The default value is 30 seconds.

**acregmax=*n*** Specifies that cached attributes are held for no more than *n* seconds after file modification. After *n* seconds, all file information is purged from the cache. The default value is 30 seconds.

**acdirmin=*n*** Specifies that cached attributes are held for at least *n* seconds after directory update. After *n* seconds, CacheFS checks to see if the directory modification time on the back file system has changed. If it has, all information about the directory is purged from the cache and new data is retrieved from the back file system. The default value is 30 seconds.

**acdirmax=*n*** Specifies that cached attributes are held for no more than *n* seconds after directory update. After *n* seconds, all directory information is purged from the cache. The default value is 30 seconds.

**actimeo=*n*** Sets **acregmin**, **acregmax**, **acdirmin**, and **acdirmax** to *n*.

## EXAMPLES

The following example CacheFS-mounts the file system **server1:/user2**, which is already NFS-mounted on **/usr/abc** as **/xyz**.

```
example# mount -F cacheefs -o backfstype=nfs,backpath=/usr/abc,
        cachedir=/cache1 server1:/user2 /xyz
```

The lines similar to the following appear in the **/etc/mnttab** file after the **mount** command is executed:

```
server1:/user2      /usr/abc      nfs
/usr/abc           /cache1/xyz   cacheefs      backfstype=nfs
```

## AUTHOR

**mount\_cacheefs** was developed by Sun Microsystems, Inc.

## SEE ALSO

cfsadmin(1M), fsck\_cacheefs(1M), mount(1M).

**NAME**

mount, umount - mount and unmount an CDFS file systems

**SYNOPSIS**

```
/usr/sbin/mount [-l] [-p|-v]
/usr/sbin/mount -a [-F cdfs] [-eQ]
/usr/sbin/mount [-F cdfs] [-eQrV] [-o specific_options] {special|directory}
/usr/sbin/mount [-F cdfs] [-eQrV] [-o specific_options] special directory
/usr/sbin/umount -a [-F cdfs] [-v]
/usr/sbin/umount [-v] [-V] {special|directory}
```

**DESCRIPTION**

The **mount** command mounts file systems. Only a superuser can mount file systems. Other users can use **mount** to list mounted file systems.

The **mount** command attaches *special*, a removable file system, to *directory*, a directory on the file tree. *directory*, which must already exist, will become the name of the root of the newly mounted file system. *special* and *directory* must be given as absolute path names. If either *special* or *directory* is omitted, **mount** attempts to determine the missing value from an entry in the **/etc/fstab** file. **mount** can be invoked on any removable file system, except **/**.

If **mount** is invoked without any arguments, it lists all of the mounted file systems from the file system mount table, **/etc/mnttab**.

The **umount** command unmounts mounted file systems. Only a superuser can unmount file systems.

**Options (mount)**

**mount** recognizes the following options:

- a** Attempt to mount all file systems described in **/etc/fstab**. All optional fields in **/etc/fstab** must be included and supported. If **-F cdfs** is specified, all CDFS file systems in **/etc/fstab** are mounted. If **noauto** is specified in an entry's option list, this entry is skipped. File systems are not necessarily mounted in the order listed in **/etc/fstab**.
- e** Verbose mode. Write a message to standard output indicating which file system is being mounted.
- F cdfs** Specify the CDFS file system type (see *fstyp*(1M)).
- l** Limit actions to local file systems only.
- o *specific\_options***  
Specify options specific to the CDFS file system type. *specific\_options* is a list of comma separated suboptions and/or keyword/attribute pairs intended for the CDFS specific module of the command.  
The following *specific\_options* are valid on CDFS file systems.
  - cdcase** Suppress the display of version numbers. Show and match file names as lower case.
  - defaults** Use all default options. When given, this must be the only option specified.
  - ro** Mount read-only (default).
  - suid** Allow set-user-ID execution (default).
  - nosuid** Do not allow set-user-ID execution.
- p** Report the list of mounted file systems in the **/etc/fstab** format.
- Q** Prevent the display of error messages resulting from an attempt to mount already mounted file systems.
- r** Mount the specified file system as read-only. This option is equivalent to the **-o ro** *specific\_option*. For CDFS file systems this is a default option.

- v Report the regular output with file system type and flags; however, *directory* and *special* fields are reversed.
- V Echo the completed command line, but performs no other action. The command line is generated by incorporating the user-specified options and other information derived from */etc/fstab*. This option allows the user to verify the command line.

**Options (umount)**

**umount** recognizes the following options:

- a Attempt to unmount all file systems described in */etc/mnttab*. All optional fields in */etc/mnttab* must be included and supported. If **-F cdfs** is specified, all CDFS file systems in */etc/mnttab* are unmounted. File systems are not necessarily unmounted in the order listed in */etc/mnttab*.
- F cdfs Specify the CDFS file system type (see *fstyp(1M)*).
- v Verbose mode. Write a message to standard output indicating which file system is being unmounted.
- V Echo the completed command line, but performs no other action. The command line is generated by incorporating the user-specified options and other information derived from */etc/fstab*. This option allows the user to verify the command line.

**DIAGNOSTICS**

**umount** complains if the special file is not mounted or if it is busy. The file system is busy if it contains an open file or some logged-in user's working directory.

**EXAMPLES**

Mount a local CDFS disk:

```
mount -F cdfs /dev/dsk/c0t0d4 /cdrom
```

Unmount a local CDFS disk:

```
umount /dev/dsk/c0t0d4
```

**WARNINGS**

Some degree of validation is done on the file system, however, it is generally unwise to mount file systems that are defective, corrupt, or of unknown origin.

**NOTES**

Additional CD-ROM formats are supported using PFS (Portable File System) utilities. See *pfs(4)* for more details.

**AUTHOR**

**mount** was developed by HP, AT&T, the University of California, Berkeley, and Sun Microsystems.

**FILES**

|                    |                                           |
|--------------------|-------------------------------------------|
| <i>/etc/fstab</i>  | Static information about the file systems |
| <i>/etc/mnttab</i> | Mounted file system table                 |

**SEE ALSO**

*fsclean(1M)*, *mount(1M)*, *quotaon(1M)*, *mount(2)*, *fstab(4)*, *mnttab(4)*, *pfs(4)*, *fs\_wrapper(5)*, *quota(5)*.

**STANDARDS CONFORMANCE**

**mount**: SVID3

**umount**: SVID3



**NAME**

mount, umount - mount and unmount an HFS file systems

**SYNOPSIS**

```
/usr/sbin/mount [-l] [-p|-v]
/usr/sbin/mount -a [-F hfs] [-eQ] [-f]
/usr/sbin/mount [-F hfs] [-eQrV] [-f] [-o specific_options] {special|directory}
/usr/sbin/mount [-F hfs] [-eQrV] [-f] [-o specific_options] special directory
/usr/sbin/umount -a [-F hfs] [-v]
/usr/sbin/umount [-v] [-V] {special|directory}
```

**DESCRIPTION**

The **mount** command mounts file systems. Only a superuser can mount file systems. Other users can use **mount** to list mounted file systems.

The **mount** command attaches *special*, a removable file system, to *directory*, a directory on the file tree. *directory*, which must already exist, will become the name of the root of the newly mounted file system. *special* and *directory* must be given as absolute path names. If either *special* or *directory* is omitted, **mount** attempts to determine the missing value from an entry in the `/etc/fstab` file. **mount** can be invoked on any removable file system, except `/`.

If **mount** is invoked without any arguments, it lists all of the mounted file systems from the file system mount table, `/etc/mnttab`.

The **umount** command unmounts mounted file systems. Only a superuser can unmount file systems.

**Options (mount)**

**mount** recognizes the following options:

- a** Attempt to mount all file systems described in `/etc/fstab`. All optional fields in `/etc/fstab` must be included and supported. If **-F hfs** is specified, all HFS file systems in `/etc/fstab` are mounted. If **noauto** is specified in an entry's option list, this entry is skipped. File systems are not necessarily mounted in the order listed in `/etc/fstab`.
- e** Verbose mode. Write a message to standard output indicating which file system is being mounted.
- f** Force the file system to be mounted, even if the file system clean flag indicates that the file system should have **fsck** run on it before mounting (see *fsck(1M)*). This option is valid only on HFS file systems.
- F hfs** Specify the HFS file system type (see *fstyp(1M)*).
- l** Limit actions to local file systems only.
- o *specific\_options*** Specify options specific to the HFS file system type. *specific\_options* is a list of comma separated suboptions and/or keyword/attribute pairs intended for the HFS specific module of the command.

The following *specific\_options* are valid on HFS file systems.

- defaults** Use all default options. When given, this must be the only option specified.
- rw** Mount read-write (default).
- ro** Mount read-only.
- suid** Allow set-user-ID execution (default).
- nosuid** Do not allow set-user-ID execution.
- behind** Enable, where possible, asynchronous writes to disk. This is the default on 700 systems.

|                     |                                                                                                                                                                                                       |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>delayed</b>      | Enable delayed or buffered writes to disk. This is the default on 800 systems.                                                                                                                        |
| <b>fs_async</b>     | Enable relaxed posting of file system metadata.                                                                                                                                                       |
| <b>no_fs_async</b>  | Enable rigorous posting of file system metadata. This is the default.                                                                                                                                 |
| <b>largefiles</b>   | Attempt to enable the creation of files greater than 2 gigabytes in size. File systems have to be created or configured to enable large files (see <i>mkfs_hfs(1M)</i> and <i>fsadm_hfs(1M)</i> ).    |
| <b>nolargefiles</b> | Attempt to disable the creation of files greater than 2 gigabytes in size. File systems have to be created or configured to disable large files. (see <i>mkfs_hfs(1M)</i> and <i>fsadm_hfs(1M)</i> ). |
| <b>quota</b>        | Enable disk quotas (valid only for <b>rw</b> file systems).                                                                                                                                           |
| <b>noquota</b>      | Disable disk quotas (default).                                                                                                                                                                        |

Mounting with the **quota** option also enables quotas for the file system, unlike some other systems, which require the additional invocation of the **quotaon** command after the file system has been mounted (see *quotaon(1M)*). Running **quotaon** does no harm, but it is not necessary.

|           |                                                                                                                                                                                                                                                                                         |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-p</b> | Report the list of mounted file systems in the <i>/etc/fstab</i> format.                                                                                                                                                                                                                |
| <b>-Q</b> | Prevent the display of error messages resulting from an attempt to mount already mounted file systems.                                                                                                                                                                                  |
| <b>-r</b> | Mount the specified file system as read-only. This option is equivalent to the <b>-o ro</b> <i>specific_option</i> . Physically write-protected file systems must be mounted in this way or errors occur when access times are updated, whether or not any explicit write is attempted. |
| <b>-v</b> | Report the regular output with file system type and flags; however, <i>directory</i> and <i>special</i> fields are reversed.                                                                                                                                                            |
| <b>-V</b> | Echo the completed command line, but performs no other action. The command line is generated by incorporating the user-specified options and other information derived from <i>/etc/fstab</i> . This option allows the user to verify the command line.                                 |

### Options (umount)

**umount** recognizes the following options:

|               |                                                                                                                                                                                                                                                                                                                                 |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-a</b>     | Attempt to unmount all file systems described in <i>/etc/mnttab</i> . All optional fields in <i>/etc/mnttab</i> must be included and supported. If <b>-F hfs</b> is specified, all HFS file systems in <i>/etc/mnttab</i> are unmounted. File systems are not necessarily unmounted in the order listed in <i>/etc/mnttab</i> . |
| <b>-F hfs</b> | Specify the HFS file system type (see <i>fstyp(1M)</i> ).                                                                                                                                                                                                                                                                       |
| <b>-v</b>     | Verbose mode. Write a message to standard output indicating which file system is being unmounted.                                                                                                                                                                                                                               |
| <b>-V</b>     | Echo the completed command line, but performs no other action. The command line is generated by incorporating the user-specified options and other information derived from <i>/etc/fstab</i> . This option allows the user to verify the command line.                                                                         |

### DIAGNOSTICS

**umount** complains if the special file is not mounted or if it is busy. The file system is busy if it contains an open file or some logged-in user's working directory.

### EXAMPLES

Mount a local HFS disk:

```
mount -F hfs /dev/dsk/c0t0d4 /usr
```

Unmount a local HFS disk:

```
umount /dev/dsk/c0t0d4
```

**WARNINGS**

Some degree of validation is done on the file system, however, it is generally unwise to mount file systems that are defective, corrupt, or of unknown origin.

**AUTHOR**

**mount** was developed by HP, AT&T, the University of California, Berkeley, and Sun Microsystems.

**FILES**

|                    |                                           |
|--------------------|-------------------------------------------|
| <b>/etc/fstab</b>  | Static information about the file systems |
| <b>/etc/mnttab</b> | Mounted file system table                 |

**SEE ALSO**

fsclean(1M), mount(1M), mkfs\_hfs(1M), fsadm\_hfs(1M), quotaon(1M), mount(2), fstab(4), mnttab(4), fs\_wrapper(5), quota(5).

**STANDARDS CONFORMANCE**

**mount**: SVID3

**umount**: SVID3

**NAME**

mount, umount - mount and unmount an LOFS file system

**SYNOPSIS**

```
/usr/sbin/mount [-p|-v]
/usr/sbin/mount -a [-F lofs] [-eQ]
/usr/sbin/mount [-F lofs] [-eQrV] [-o specific_options] {special_directory|directory}
/usr/sbin/mount [-F lofs] [-eQrV] [-o specific_options] special_directory directory
/usr/sbin/umount [-v] [-V] {special_directory|directory}
/usr/sbin/umount -a [-F lofs] [-v]
```

**DESCRIPTION**

The **mount** command mounts LOFS file systems. Only superuser can mount LOFS file systems. Other users can use **mount** to list mounted file systems.

**mount**, attaches *special\_directory*, a directory from one of the mounted file systems, to *directory*, an another directory in one of the mounted file systems. This enables new file systems to be created, which provide access to existing directories or file systems using alternate path names. Both *special\_directory* and *directory* should already exist. *directory* will become the root of the newly mounted LOFS file system, containing the file system hierarchy under *special\_directory*. *special\_directory* and *directory* must be specified as absolute path names. If either *special\_directory* or *directory* is omitted, **mount** attempts to determine the missing value from an entry in the **/etc/fstab** file. **mount** can be invoked on any removable file system, except **/**.

If **mount** is invoked without any arguments, it lists all the mounted file systems from the file system mount table, **/etc/mnttab**.

The **umount** command unmounts mounted file systems. Only a superuser can unmount file systems.

**Options (mount)**

**mount** recognizes the following options:

- a Attempt to mount all file systems described in **/etc/fstab**. All optional fields in **/etc/fstab** must be included and supported. If **-F lofs** is specified, all LOFS file systems in **/etc/fstab** are mounted. If **noauto** is specified in an entry's option list, this entry is skipped. File systems are not necessarily mounted in the order listed in **/etc/fstab**.
- e Verbose mode. Write a message to standard output indicating which file system is being mounted.
- F lofs Specify the LOFS file system type (see *fstyp*(1M)).
- l Limit actions to local file systems only. LOFS is a local file system.
- o *specific\_options* Specify options specific to the LOFS file system type. *specific\_options* is a list of comma separated suboptions and/or keyword/attribute pairs intended for the LOFS specific module of the command.  
The following *specific\_options* are valid on an LOFS file system:  
  - defaults Use all default options. When used, this must be the only option specified.
  - ro Read-only (see *WARNINGS* below).
- p Report the list of mounted file systems in the **/etc/fstab** format.
- Q Prevent display of error messages resulting from an attempt to mount already mounted file systems.
- r Mount the specified file system as read-only (see *WARNINGS* below).
- v Report the output in a new style. The new style has the file system type and flags displayed in addition to the old output. The *directory* and *special\_directory* fields are reversed.

- v           Echo the completed command line, but perform no other action. The command line is generated by incorporating the user-specified options and other information derived from `/etc/fstab`. This option allows the user to verify the command line.

### Options (umount)

The **umount** command recognizes the following options:

- a           Attempt to unmount all file systems described in `/etc/mnttab`. All optional fields in `/etc/mnttab` must be included and supported. If **-F lofs** file system type is specified, all the LOFS file systems in `/etc/mnttab` are unmounted. File systems are not necessarily unmounted in the order listed in `/etc/mnttab`.
- F lofs      Specify the LOFS file system type (see *fstyp*(1M)).
- v           Verbose mode. Write a message to standard output indicating which file system is being unmounted.
- v           Echo the completed command line, but perform no other action. The command line is generated by incorporating the user-specified options and other information derived from `/etc/fstab`. This option allows the user to verify the command line.

### EXAMPLES

Mount an LOFS file system:

```
mount /usr /tmp/usr
```

Mount another LOFS file system:

```
mount -F lofs /usr/sbin /tmp/sbin
```

### WARNINGS

LOFS file systems provide the user with numerous applications; however, they may be potentially confusing. LOFS file systems should generally be created by an experienced user.

For LOFS file systems which are mounted read-only, if the underlying file system is mounted writable, certain write operations on the LOFS will succeed. Thus LOFS should not be relied upon to provide a strictly write-only alternative image of a read-write file system.

### AUTHOR

**mount** was developed by HP, AT&T, the University of California, Berkeley, and Sun Microsystems.

### FILES

|                          |                                           |
|--------------------------|-------------------------------------------|
| <code>/etc/fstab</code>  | Static information about the file systems |
| <code>/etc/mnttab</code> | Mounted file system table                 |

### SEE ALSO

mount(1M), mount(2), fstab(4), mnttab(4).

### STANDARDS CONFORMANCE

**mount**: SVID3

**NAME**

mount, umount - mount and unmount an NFS file systems

**SYNOPSIS**

```
/usr/sbin/mount [-l] [-p|-v]
/usr/sbin/mount -a [-F nfs] [-eQ]
/usr/sbin/mount [-F nfs] [-eQrV] [-o specific_options] {host:path|directory}
/usr/sbin/mount [-F nfs] [-eQrV] [-o specific_options] host:path directory

/usr/sbin/umount -a [-F nfs] [-h host] [-v]
/usr/sbin/umount [-v] [-V] {host:path|directory}
```

**DESCRIPTION**

The **mount** command mounts file systems. Only a superuser can mount file systems. Other users can use **mount** to list mounted file systems.

The **mount** command attaches *host:path* to *directory*. *host* is a remote system, *path* is a directory on this remote system and *directory* is a directory on the local file tree. *directory* must already exist, be given as an absolute path name and will become the name of the root of the newly mounted file system. If either *host:path* or *directory* is omitted, **mount** attempts to determine the missing value from an entry in the */etc/fstab* file. **mount** can be invoked on any removable file system, except */*.

If **mount** is invoked without any arguments, it lists all of the mounted file systems from the file system mount table, */etc/mnttab*. The **umount** command unmounts mounted file systems. Only a superuser can unmount file systems.

**OPTIONS**

**-r** Mount the specified file system read-only.

**-o *specific\_options***

Set file system specific options according to a comma-separated list chosen from words below.

**rw** | **ro** *resource* is mounted read-write or read-only. The default is **rw**.

**suid** | **nosuid**

Setuid execution allowed or disallowed. The default is **suid**.

**remount** If a file system is mounted read-only, remounts the file system read-write.

**bg** | **fg** If the first attempt fails, retry in the background, or, in the foreground. The default is **fg**.

**quota** Enables **quota(1M)** to check whether the user is over quota on this file system; if the file system has quotas enabled on the server, quotas will still be checked for operations on this file system. The default is **quota**.

**noquota** Prevent **quota(1M)** from checking whether the user exceeded the quota on this file system; if the file system has quotas enabled on the server, quotas will still be checked for operations on this file system.

**retry=*n*** The number of times to retry the mount operation. The default is 1.

**vers=<NFS version number>**

By default, the version of NFS protocol used between the client and the server is the highest one available on both systems. If the NFS server does not support NFS Version 3, then the NFS mount will use NFS Version 2.

**port=*n*** Set server UDP port number to *n* (the default is the port customarily used for NFS servers).

**proto=<transp>**

Use the transport protocol *<transp>* for this mount. Valid values for *<transp>* are *tcp* (connection-oriented) and *udp* (connectionless). The default behavior is to attempt a *tcp* connection. If the *tcp* connection attempt fails when defaulting to *tcp*, a *udp* connection will be attempted.

- grpid** By default, the GID associated with a newly created file will obey the System V semantics; that is, the GID is set to the effective GID of the calling process. This behavior may be overridden on a per-directory basis by setting the set-GID bit of the parent directory; in this case, the GID of a newly created file is set to the GID of the parent directory (see **open(2)** and **mkdir(2)**). Files created on file systems that are mounted with the **grpid** option will obey BSD semantics independent of whether the set-GID bit of the parent directory is set; that is, the GID is unconditionally inherited from that of the parent directory.
- rsize=*n*** Set the read buffer size to *n* bytes. The default value is set by kernel.
- wsiz=*n*** Set the write buffer size to *n* bytes. The default value is set by kernel.
- timeo=*n*** Set the NFS timeout to *n* tenths of a second. The default value is set by kernel.
- retrans=*n*** Set the number of NFS retransmissions to *n*. The default value is 5.
- soft | hard** Return an error if the server does not respond, or continue the retry request until the server responds. The default value is **hard**.
- intr | nointr** Allow (do not allow) keyboard interrupts to kill a process that is hung while waiting for a response on a hard-mounted file system. The default is **intr**.
- noac** Suppress attribute caching.
- nocto** Suppress fresh attributes when opening a file.
- devs | nodevs** Allow (do not allow) access to local devices. The default is **devs**.
- acdirmax=*n*** Hold cached attributes for no more than *n* seconds after directory update. The default value is 60.
- acdirmin=*n*** Hold cached attributes for at least *n* seconds after directory update. The default value is 30.
- acregmax=*n*** Hold cached attributes for no more than *n* seconds after file modification. The default value is 60.
- acregmin=*n*** Hold cached attributes for at least *n* seconds after file modification. The default value is 3.
- actimeo=*n*** Set *min* and *max* times for regular files and directories to *n* seconds. **actimeo** has no default; it sets **acregmin**, **acregmax**, **acdirmin**, and **acdirmax** to the value specified.
- O** Overlay mount. Allow the file system to be mounted over an existing mount point, making the underlying file system inaccessible. If a mount is attempted on a pre-existing mount point without setting this flag, the mount will fail, producing the error **device busy**.

### Options (umount)

**umount** recognizes the following options:

- a** Attempt to unmount all file systems described in **/etc/mnttab**. All optional fields in **/etc/mnttab** must be included and supported. If **-F nfs** option is specified, all NFS file systems in **/etc/mnttab** are unmounted. File systems are not necessarily unmounted in the order listed in **/etc/mnttab**.
- F nfs** Specify the NFS file system type (see **fstyp(1M)**).
- h host** Unmount only those file systems listed in **/etc/mnttab** that are remote-mounted from *host*.
- v** Verbose mode. Write a message to standard output indicating which file system is being unmounted.
- V** Echo the completed command line, but performs no other action. The command line is generated by incorporating the user-specified options and other information derived from **/etc/fstab**. This option allows the user to verify the command line.

## NFS File Systems

### Background vs. Foreground

File systems mounted with the **bg** option indicate that **mount** is to retry in the background if the server's mount daemon (**mountd**(1M)) does not respond. **mount** retries the request up to the count specified in the **retry=*n*** option. Once the file system is mounted, each NFS request made in the kernel waits **timeo=*n*** tenths of a second for a response. If no response arrives, the time-out is multiplied by 2 and the request is retransmitted. When the number of retransmissions has reached the number specified in the **retrans=*n*** option, a file system mounted with the **soft** option returns an error on the request; one mounted with the **hard** option prints a warning message and continues to retry the request.

### Hard vs. Soft

File systems that are mounted read-write or that contain executable files should always be mounted with the **hard** option. Applications using **soft** mounted file systems may incur unexpected I/O errors.

To improve NFS read performance, files and file attributes are cached. File modification times get updated whenever a write occurs. However, file access times may be temporarily out-of-date until the cache gets refreshed. The attribute cache retains file attributes on the client. Attributes for a file are assigned a time to be flushed. If the file is modified before the flush time, then the flush time is extended by the time since the last modification (under the assumption that files that changed recently are likely to change soon). There is a minimum and maximum flush time extension for regular files and for directories. Setting **actimeo=*n*** sets flush time to *n* seconds for both regular files and directories.

## EXAMPLES

To mount an NFS file system:

```
mount serv:/usr/src /usr/src
```

To mount an NFS file system readonly with no suid privileges:

```
mount -r -o nosuid serv:/usr/src /usr/src
```

To mount an NFS file system over Version 3:

```
mount -o vers=3 serv:/usr/src /usr/src
```

To unmount all file systems imported from a given host, enter the following command as root:

```
umount -h mysystem.home.work.com -a
```

The hostname must match what is in **/etc/mnttab** exactly (as shown by the **bdf** command). For example, if **bdf** shows:

```
mysystem:/projects,
```

the **umount** command would be

```
umount -h mysystem -a.
```

## FILES

|                    |                                                  |
|--------------------|--------------------------------------------------|
| <b>/etc/mnttab</b> | table of mounted file systems.                   |
| <b>/etc/fstab</b>  | list of default parameters for each file system. |

## SEE ALSO

fsck(1M), mount(1M), quotaon(1M), mount(2), fstab(4), mnttab(4), fs\_wrapper(5), quota(5).

## STANDARDS COMPLIANCE

**mount**: SVID3

**umount**: SVID3



**NAME**

mount, umount - mount and unmount a VxFS file system

**SYNOPSIS**

```
/usr/sbin/mount [-l] [-v|-p]
/usr/sbin/mount [-F vxfs] [-eQ] -a
/usr/sbin/mount [-F vxfs] [-eQrV] [-o specific_options] {special|directory}
/usr/sbin/mount [-F vxfs] [-eQrV] [-o specific_options] special directory
/usr/sbin/umount [-V] [-v] {special | directory}
/usr/sbin/umount [-F vxfs] [-v] -a
```

**DESCRIPTION**

**mount** attaches *special*, a removable file system, to *directory*, a directory on the file tree. (This directory is also known as the mount point). *directory*, which must already exist, will become the name of the root of the newly mounted file system. If either *special* or *directory* is omitted, **mount** attempts to determine the missing value from an entry in */etc/fstab*. **mount** can be invoked on any removable file system, except */*. *special* and *directory* must be given as absolute path names.

If **mount** is invoked with no arguments it lists all the mounted file systems from the mounted file system table, */etc/mnttab*.

The **umount** command unmounts mounted file systems.

Only the superuser can **mount** and **umount** file systems. Other users can use **mount** to list mounted file systems.

**Options**

**mount** recognizes the following options:

- a Attempt to mount all file systems described in */etc/fstab*. All optional fields in */etc/fstab* must be included and supported. If **-F vxfs** is specified, all VxFS file systems in */etc/fstab* are mounted. If **noauto** is specified in an entry's option list (in */etc/fstab*), that entry is skipped (not mounted). File systems are not necessarily mounted in the order listed in */etc/fstab*.
- e Verbose mode. Write a message to the standard output indicating which file system is being mounted.
- F vxfs Specify the file system type (**vxfs**).
- l Limit actions to local file systems only.
- o *specific\_options* Specify options specific to the VxFS file system type. *specific\_options* is a list of comma separated suboptions and/or keyword/attribute pairs intended for the VxFS-specific module of the command. Unlike some file system commands, multiple **-o** options do not accumulate; only the last option is used.

The following *specific\_options* are valid on a VxFS file system:

**blkclear**

Clear all data extents before allocating them to a file (requires synchronous zeroing, on disk, of certain newly allocated extents). This prevents uninitialized data from being written to a file at the time of a system crash.

**convosync=direct|dsync|unbuffered|closesync|delay**

Alter the caching behavior of the file system for *O\_SYNC* and *O\_DSYNC* I/O operations.

The **direct** value handles any reads or writes with the *O\_SYNC* or *O\_DSYNC* flag as if the **VX\_DIRECT** caching advisory is set.

The **dsync** value handles any writes with the *O\_SYNC* flag as if the **VX\_DSYNC** caching advisory is set. It does not modify behavior for writes with *O\_DSYNC* set.

The **unbuffered** value handles any reads or writes with the **O\_SYNC** or **O\_DSYNC** flag as if the **VX\_UNBUFFERED** caching advisory is set.

The **closesync** value delays **O\_SYNC** or **O\_DSYNC** writes so that they do not take effect immediately.

The **closesync**, **dsync**, **direct**, and **unbuffered** values all run the equivalent of an *fsync(2)* to be run when any file accessed with the **O\_SYNC** or **O\_DSYNC** flag is closed.

The **delay** value delays **O\_SYNC** or **O\_DSYNC** writes so that they do not take effect immediately. With this option, VxFS changes **O\_SYNC** or **O\_DSYNC** writes into delayed writes. No special action is performed when closing a file. This option effectively cancels data integrity guarantees normally provided by opening a file with **O\_SYNC** or **O\_DSYNC**.

NOTE: The **convosync** option is available only with the HP OnLineJFS product.

#### **datainlog|nodatainlog**

Generally, VxFS does **O\_SYNC** or **O\_DSYNC** writes by logging the data and the time change to the inode (**datainlog**). If the **nodatainlog** option is used, the logging of synchronous writes is disabled; **O\_SYNC** writes the data into the file and updates the inode synchronously before returning to the user.

NOTE: The **datainlog** option is available only with the HP OnLineJFS product.

#### **largefiles|nolargefiles**

These options do not turn largefiles capability on and off (use **mkfs\_vxfs** or **fsadm\_vxfs** to set and clear the largefiles flag), but they do verify whether a file system is largefiles capable. If **nolargefiles** is specified and the mount succeeds, then the file system does not contain any files whose size is two gigabytes or larger, and such files cannot be created. If **largefiles** is specified and the mount succeeds, then the file system may contain files whose size is two gigabytes or larger, and large files can be created. For a mount to succeed, the option must match the largefiles flag as specified by **mkfs\_vxfs** or **fsadm\_vxfs**.

NOTE: Large files are supported on HP-UX 10.20 systems and above. Be careful when enabling large file system capability. System administration utilities such as backup may experience problems if they are not large-file aware.

#### **log|delaylog|tmplog|nolog**

Control intent logging. To maintain file system integrity after a system failure, logging must be enabled. The default is **log**. In **log** mode, file system structural changes are logged to disk before the system call returns to the application. If the system crashes, *fsck\_vxfs(1M)* completes logged operations that did not complete.

In **delaylog** mode, some system calls return before the intent log is written. This improves the performance of the system, but some changes are not guaranteed until a short time later when the intent log is written. This mode approximates traditional UNIX system guarantees for correctness in case of system failures.

In **tmplog** mode, the intent log is almost always delayed. This improves performance, but recent changes may disappear if the system crashes. This mode is only recommended for temporary file systems.

**nolog** is an alias for **tmplog**.

#### **mincache=direct|dsync|unbuffered|closesync|tmpcache**

Alter the caching behavior of the file system.

The **direct** value handles any reads without the **O\_SYNC** flag, or any writes without the **O\_SYNC** flag, **VX\_DSYNC**, **VX\_DIRECT**, and **VX\_UNBUFFERED** caching advisories, as if the **VX\_DIRECT** caching advisory was set.

The **dsync** value handles any writes without the **O\_SYNC** flag or one of the **VX\_DIRECT**, **VX\_DSYNC**, or **VX\_UNBUFFERED** caching advisories as if the **VX\_DSYNC** caching advisory was set.

The **unbuffered** value handles any reads without the **O\_SYNC** flag, or any writes without the **O\_SYNC** flag, **VX\_DSYNC**, **VX\_DIRECT**, and

**VX\_UNBUFFERED** caching advisories, as if the **VX\_UNBUFFERED** caching advisory was set.

The **closesync**, **dsync**, **unbuffered**, and **direct** values all cause the equivalent of an *fsync(2)* to be run when the file is closed.

The **tmpcache** value disables delayed extending writes, trading off integrity for performance. When this option is chosen, VxFS does not zero out new extents allocated as files are sequentially written. Uninitialized data may appear in files being written at the time of a system crash. See *vxfsio(7)* for an explanation of **VX\_DIRECT**, **VX\_DSYNC**, and **VX\_UNBUFFERED**.

NOTE: **mincache=direct**, **mincache=dsync**, **mincache=unbuffered**, and **mincache=tmpcache** are available only with the HP OnLineJFS product.

**quota** Enable disk quotas (valid only for **rw** type file systems). VxFS maintains quota information in a private area of the file system. If the file system is mounted with quotas enabled, and the file system was previously mounted with quotas disabled and was modified, then the quota information is rebuilt. This may take a while.

#### **remount**

Change the mount options for a mounted file system. In particular, **remount** changes the logging and caching policies. It also changes a file system from read-only to read/write.

**remount** cannot change a file system from read/write to read-only, nor can it set the **snapof** or **snapsize** attributes.

**rw|ro** Read/write or read-only. The default is **rw**.

**snapof=filesystem**

Mount the file system as a snapshot of *filesystem*, where *filesystem* is either the directory on which a VxFS file system is mounted, or is the block special file containing a mounted VxFS file system. An explicit **-F vxfs** option is required to mount a snapshot file system.

NOTE: **snapof=filesystem** is available only with the HP OnLineJFS product.

**snapsize=size**

Used in conjunction with **snapof**. *size* is the size in sectors of the snapshot file system being mounted. This option is required only when the device driver is incapable of determining the size of *special*, and defaults to the entire device if not specified.

NOTE: **snapsize=size** is available only with the HP OnLineJFS product.

**suid|nosuid**

setuid execution allowed or setuid execution not allowed. The default is **suid**.

- p** Report the list of mounted file systems in the **/etc/fstab** format.
- Q** Prevent display of error messages, resulting from an attempt to mount already mounted file systems.
- r** Mount the specified file system as read-only. Physically write-protected file systems must be mounted in this way or errors occur when access times are updated, whether or not any explicit write is attempted.
- v** Reports the regular output with file system type and flags, however, *directory* and *special* fields are reversed.
- V** Echo the completed command line, but do not execute the command. The command line is generated by incorporating the user-specified options and other information derived from **/etc/fstab**. This option allows the user to verify the command line.

**umount** recognizes the following options:

- a** Attempt to unmount all file systems described in **/etc/mnttab**. All optional fields in **/etc/mnttab** must be included and supported. If **-F vxfs** is specified, all VxFS file systems in **/etc/mnttab** are unmounted. File systems are not necessarily unmounted in the

order listed in `/etc/mnttab`.

- F vxfs** Specify the file system type (**vxfs**).
- v** Verbose mode. Write a message to the standard output indicating which file system is being unmounted.
- V** Echo the completed command line, but do not execute the command. The command line is generated by incorporating the user-specified options and other information derived from `/etc/fstab`. This option allows the user to verify the command line.

## EXAMPLES

List the file systems currently mounted:

```
mount
```

Mount a VxFS file system `/dev/dsk/c1t2d0` at directory `/home`

```
mount -F vxfs /dev/dsk/c1t2d0 /home
```

Unmount the same file system:

```
umount /dev/dsk/c1t2d0
```

## NOTES

Only a privileged user can mount file systems.

Large files (over two gigabytes) are supported on HP-UX 10.20 systems and above.

## FILES

|                          |                                           |
|--------------------------|-------------------------------------------|
| <code>/etc/fstab</code>  | Static information about the file systems |
| <code>/etc/mnttab</code> | Mounted file system table                 |

## SEE ALSO

`fsadm_vxfs(1M)`, `fsck_vxfs(1M)`, `mkfs_vxfs(1M)`, `mount(1M)`, `mount(2)`, `fsync(2)`, `fstab(4)`, `mnttab(4)`, `quota(5)`, `vxfsio(7)`.

## STANDARDS CONFORMANCE

**mount:** SVID3

**umount:** SVID3

**NAME**

mrinfo - Multicast Routing Configuration Information Tool

**SYNOPSIS**

```
/usr/sbin/mrinfo [-d debuglevel] [-r retries] [-t timeout] [ multicast-router ]
```

**DESCRIPTION**

**mrinfo** requests the configuration information from the *multicast-router*, and prints the information to the standard out. *multicast-router* can be either an IP address or a system name. **mrinfo** sends out the *ASK\_NEIGHBORS* igmp message to the specified *multicast-router*, when the router receives the request, it sends back its configuration information. If the *multicast-router* is not specified, the request is sent the local router.

The the configuration information for each interface is printed in the following format:

```
interface_addr -> neighbor_addr (neighbor_name) [metrics/thresh/flags]
```

If there are multiple neighbor routers on one interface, they will all be reported on the output. The possible values for *flag* are:

**tunnel**        Neighbors are reached via tunnel.  
**srcrt**        The tunnel uses IP source routing.  
**down**        The interface is down.  
**disabled**    The interface is administratively disabled for multicast routing.  
**querier**    The local router is the querier of the subnet.

Please see *mrouted*(1M) for **metrics** and **thresh**.

The command line options are:

**-d debuglevel** Sets the level for printing out the debug message. The default is 0, only error and warning messages will be printed. Debug level three prints most the messages.  
**-r retries**    Sets the retry times to pull the routing daemon for information. The default is 3.  
**-t timeout**    Specifies the timeout value in seconds for waiting the reply. The default value is 4.

**EXAMPLE**

The following is an example of quering the multicasting configuration from the local routing daemon.

```
mrinfo
127.0.0.1 (localhost) [version 3.3]:
15.13.106.144 -> 15.13.106.145 (hpntcbs.cup.hp.com) [10/1/querier]
193.2.1.39 -> 0.0.0.0 (all-zeros-broadcast) [1/1/disabled]
15.13.106.144 -> 15.255.176.33 (matmos.hpl.hp.com) [10/1/tunnel]
15.13.106.144 -> 15.17.20.7 (hpspddc.vid.hp.com) [10/1/tunnel/down]
```

**Note**

**mrinfo** must be run as root.

**AUTHOR**

**mrinfo** was developed by Van Jacobson.

**SEE ALSO**

*mrouted*(1M), *map-mbone*(1M).

**NAME**

mouted - IP multicast routing daemon

**SYNOPSIS**

```
/usr/sbin/mouted [-p] [-c config_file] [-d debug_level]
```

**DESCRIPTION**

The **mouted** command is an implementation of the Distance-Vector Multicast Routing Protocol (DVMRP), an earlier version of which is specified in RFC-1075. It maintains topological knowledge via a distance-vector routing protocol (like RIP, described in RFC-1058), upon which it implements a multicast datagram-forwarding algorithm called Reverse Path Multicasting.

**mouted** forwards a multicast datagram along a shortest (reverse) path tree rooted at the subnet on which the datagram originates. The multicast delivery tree may be thought of as a broadcast delivery tree that has been pruned back so that it does not extend beyond those subnetworks that have members of the destination group. Hence, datagrams are not forwarded along those branches which have no listeners of the multicast group. The IP time-to-live of a multicast datagram can be used to limit the range of multicast datagrams.

In order to support multicasting among subnets that are separated by (unicast) routers that do not support IP multicasting, **mouted** includes support for "tunnels", which are virtual point-to-point links between pairs of **mouted**s located anywhere in an internet. IP multicast packets are encapsulated for transmission through tunnels, so that they look like normal unicast datagrams to intervening routers and subnets. The encapsulation is added on entry to a tunnel and stripped off on exit from a tunnel. By default, the packets are encapsulated using the IP-in-IP protocol (IP protocol number 4).

The tunnelling mechanism allows **mouted** to establish a virtual internet for the purpose of multicasting only, which is independent of the physical internet and which may span multiple Autonomous Systems.

**mouted** handles multicast routing only; there may or may not be unicast routing software running on the same machine as **mouted**. With the use of tunnels, it is not necessary for **mouted** to have access to more than one physical subnet in order to perform multicast forwarding.

**Invocation**

If the **-d** option is not specified or if the debug level is specified as 0, **mouted** detaches from the invoking terminal. Otherwise, it remains attached to the invoking terminal and responsive to signals from that terminal. If **-d** is specified with no argument, the debug level defaults to 2. Regardless of the debug level, **mouted** always writes warning and error messages to the system log demon. Non-zero debug levels have the following effects:

- level 1    all **syslog** messages are also printed to **stderr**.
- level 2    all level 1 messages plus notifications of "significant" events are printed to **stderr**.
- level 3    all level 2 messages plus notifications of all packet arrivals and departures are printed to **stderr**.

Upon startup, **mouted** writes its *pid* to the file **/var/tmp/mouted.pid**.

**Configuration**

**mouted** automatically configures itself to forward on all multicast-capable interfaces (i.e., interfaces that have the IFF\_MULTICAST flag set, excluding the loopback "interface"). **mouted** finds other **mouted**s directly reachable via those interfaces. To override the default configuration or to add tunnel links to other **mouted**s, configuration commands may be placed in **/etc/mouted.conf** (or an alternative file, specified by the **-c** option). There are four types of configuration commands:

```

phyint <local-addr> [disable] [metric <m>]
        [threshold <t>] [rate_limit <b>]
        [boundary (<boundary-name>|<scoped-addr>/<mask-len>)]
        [altnet <network>/<mask-len>]

tunnel <local-addr> <remote-addr> [metric <m>]
        [threshold <t>] [rate_limit <b>]
        [boundary (<boundary-name>|<scoped-addr>/<mask-len>)]

cache_lifetime <ct>

pruning <off/on>

name <boundary-name> <scoped-addr>/<mask-len>

```

The file format is free-form; white space (including newlines) is not significant. The *boundary* and *altnet* options may be specified as many times as necessary.

The **phyint** command can be used to disable multicast routing on the physical interface identified by local IP address <local-addr>, or to associate a non-default metric or threshold with the specified physical interface. The local IP address <local-addr> may be replaced by the interface name (such as **lan0**). If **phyint** is attached to multiple IP subnets, describe each additional subnet with the *altnet* option. **phyint** commands must precede **tunnel** commands.

The **tunnel** command can be used to establish a tunnel link between local IP address <local-addr> and remote IP address <remote-addr>, and to associate a non-default metric or threshold with that tunnel. The local IP address <local-addr> may be replaced by the interface name (such as **lan0**). The remote IP address <remote-addr> may be replaced by a host name, if and only if the host name has a single IP address associated with it. The tunnel must be set up in the **mrouted.conf** files of both routers before it can be used.

**cache\_lifetime** is a value that determines the amount of time that a cached multicast route stays in kernel before timing out. The value of this entry should lie between 300 (5 min) and 86400 (1 day). It defaults to 300.

The **pruning** command is provided for **mrouted** to act as a non-pruning router. It is also possible to start **mrouted** in a non-pruning mode using the **-p** option on the command line. It is expected that a router would be configured in this manner for test purposes only. The default mode is pruning enabled.

You may assign names to boundaries to make configuration easier with the **name** command. The *boundary* option on **phyint** or **tunnel** commands can accept either a name or a boundary.

The *metric* option is the "cost" associated with sending a datagram on the given interface or tunnel; it may be used to influence the choice of routes. The metric defaults to 1. Metrics should be kept as small as possible because **mrouted** cannot route along paths with a sum of metrics greater than 31.

The threshold is the minimum IP time-to-live required for a multicast datagram to be forwarded to the given interface or tunnel. It is used to control the scope of multicast datagrams. (The TTL of forwarded packets is only compared to the threshold; it is not decremented by the threshold. Every multicast router decrements the TTL by 1.) The default threshold is 1.

In general, all **mrouted**s connected to a particular subnet or tunnel should use the same metric and threshold for that subnet or tunnel.

The *rate\_limit* option allows the network administrator to specify a certain bandwidth in Kbits/second which would be allocated to multicast traffic. It defaults to 500Kbps on tunnels and 0 (unlimited) on physical interfaces.

The *boundary option* allows an interface to be configured as an administrative boundary for the specified scoped address. Packets belonging to this address will not be forwarded on a scoped interface. The boundary option accepts either a name or a boundary spec.

**mrouted** will not initiate execution if it has fewer than two enabled **vifs** (virtual interface), where a **vif** is either a physical multicast-capable interface or a tunnel. It will log a warning if all of its **vifs** are tunnels; such an **mrouted** configuration would be better replaced by more direct tunnels.

**Example Configuration**

This is an example configuration for a multicast router at a large school.

```
#
# mrouted.conf example
#
# Name our boundaries to make it easier
name LOCAL 239.255.0.0/16
name EE 239.254.0.0/16
#
# lan1 is our gateway to compsci, don't forward our
# local groups to them
phyint lan1 boundary EE
#
# lan2 is our interface on the classroom net, it has four
# different length subnets on it.
# note that you can use either an ip address or an
# interface name
phyint 172.16.12.38 boundary EE altnet 172.16.15.0/26
      altnet 172.16.15.128/26 altnet 172.16.48.0/24
#
# atm0 is our ATM interface, which doesn't properly
# support multicasting.
phyint atm0 disable
#
# This is an internal tunnel to another EE subnet
# Remove the default tunnel rate limit, since this
# tunnel is over ethernet
tunnel 192.168.5.4 192.168.55.101 metric 1 threshold 1
      rate_limit 0
#
# This is our tunnel to the outside world.
# Careful with those boundaries, Eugene.
tunnel 192.168.5.4 10.11.12.13 metric 1 threshold 32
      boundary LOCAL boundary EE
```

**Signals**

**mrouted** responds to the following signals:

|      |                                                                                                                            |
|------|----------------------------------------------------------------------------------------------------------------------------|
| HUP  | restarts <b>mrouted</b> . The configuration file is reread every time this signal is evoked.                               |
| INT  | terminates execution gracefully (i.e., by sending good-bye messages to all neighboring routers).                           |
| TERM | same as INT                                                                                                                |
| USR1 | dumps the internal routing tables to <code>/usr/tmp/mrouted.dump</code> .                                                  |
| USR2 | dumps the internal cache tables to <code>/usr/tmp/mrouted.cache</code> .                                                   |
| QUIT | dumps the internal routing tables to <code>stderr</code> (only if <b>mrouted</b> was invoked with a non-zero debug level). |

For convenience in sending signals, **mrouted** writes its *pid* to `/var/tmp/mrouted.pid` upon startup.



EXAMPLES

The routing tables look like this:

| Virtual Interface Table                |               |                        |        |        |          |  |
|----------------------------------------|---------------|------------------------|--------|--------|----------|--|
| Vif                                    | Local-Address |                        | Metric | Thresh | Flags    |  |
| 0                                      | 36.2.0.8      | subnet: 36.2           | 1      | 1      | querier  |  |
|                                        |               | groups: 224.0.2.1      |        |        |          |  |
|                                        |               | 224.0.0.4              |        |        |          |  |
|                                        |               | pkts in: 3456          |        |        |          |  |
|                                        |               | pkts out: 2322323      |        |        |          |  |
| 1                                      | 36.11.0.1     | subnet: 36.11          | 1      | 1      | querier  |  |
|                                        |               | groups: 224.0.2.1      |        |        |          |  |
|                                        |               | 224.0.1.0              |        |        |          |  |
|                                        |               | 224.0.0.4              |        |        |          |  |
|                                        |               | pkts in: 345           |        |        |          |  |
|                                        |               | pkts out: 3456         |        |        |          |  |
| 2                                      | 36.2.0.8      | tunnel: 36.8.0.77      | 3      | 1      |          |  |
|                                        |               | peers: 36.8.0.77 (2.2) |        |        |          |  |
|                                        |               | boundaries: 239.0.1    |        |        |          |  |
|                                        |               | : 239.1.2              |        |        |          |  |
|                                        |               | pkts in: 34545433      |        |        |          |  |
|                                        |               | pkts out: 234342       |        |        |          |  |
| 3                                      | 36.2.0.8      | tunnel: 36.6.8.23      |        | 3      | 16       |  |
| Multicast Routing Table (1136 entries) |               |                        |        |        |          |  |
| Origin-Subnet                          | From-Gateway  | Metric                 | Tmr    | In-Vif | Out-Vifs |  |
| 36.2                                   |               | 1                      | 45     | 0      | 1* 2 3*  |  |
| 36.8                                   | 36.8.0.77     | 4                      | 15     | 2      | 0* 1* 3* |  |
| 36.11                                  |               | 1                      | 20     | 1      | 0* 2 3*  |  |
| .                                      |               |                        |        |        |          |  |
| .                                      |               |                        |        |        |          |  |
| .                                      |               |                        |        |        |          |  |

In this example, there are four **vifs** connecting to two subnets and two tunnels. The **vif** 3 tunnel is not in use (no peer address). The **vif** 0 and **vif** 1 subnets have some groups present; tunnels never have any groups. This instance of **mrouted** is the one responsible for sending periodic group membership queries on the **vif** 0 and **vif** 1 subnets, as indicated by the "querier" flags. The list of boundaries indicate the scoped addresses on that interface. A count of the number of incoming and outgoing packets is also shown at each interface.

Associated with each subnet from which a multicast datagram can originate is the address of the previous hop router (unless the subnet is directly connected), the metric of the path back to the origin, the amount of time since an update was received for this subnet, the incoming **vif** for multicasts from that origin, and a list of outgoing **vifs**. The asterisk ( \* ) indicates that the outgoing **vif** is connected to a leaf of the broadcast tree rooted at the origin, and a multicast datagram from that origin will be forwarded on that outgoing **vif** only if there are members of the destination group on that leaf.

The **mrouted** command also maintains a copy of the kernel forwarding cache table. Entries are created and deleted by **mrouted**.



The cache tables look like this:

```

Multicast Routing Cache Table (147 entries)
  Origin          Mcast-group    CTmr  Age  Ptmr  IVif  Forwvifs
13.2.116/22      224.2.127.255    3m   2m   -    0    1
>13.2.116.19
>13.2.116.196
138.96.48/21     224.2.127.255    5m   2m   -    0    1
>138.96.48.108
128.9.160/20     224.2.127.255    3m   2m   -    0    1
>128.9.160.45
198.106.194/24   224.2.135.190    9m  28s  9m   0P
>198.106.194.22

```

Each entry is characterized by the origin subnet number, mask, and the destination multicast group. The **CTmr** field indicates the lifetime of the entry. The entry is deleted from the cache table when the timer decrements to zero. The **Age** field is the time since this cache entry was originally created. Since cache entries get refreshed if traffic is flowing, routing entries can grow very old. The **Ptmr** field is simply a dash if no prune was sent upstream, or the amount of time until the upstream prune will time out. The **IVif** field indicates the incoming **vif** for multicast packets from that origin. Each router also maintains a record of the number of prunes received from neighboring routers for a particular source and group. If there are no members of a multicast group on any downward link of the multicast tree for a subnet, a prune message is sent to the upstream router. They are indicated by a **P** after the **vif** number. The **Forwvifs** field shows the interfaces along which datagrams belonging to the source-group are forwarded. A **p** indicates that no datagrams are being forwarded along that interface. An unlisted interface is a leaf subnet with no members of the particular group on that subnet. A **b** on an interface indicates that it is a boundary interface; that is, traffic will not be forwarded on the scoped address on that interface. An additional line with a **>** as the first character is printed for each source on the subnet. Note that there can be many sources in one subnet.

## m

### FILES

```

/etc/mrouted.conf
/var/run/mrouted.pid
/var/tmp/mrouted.dump
/var/tmp/mrouted.cache

```

### SEE ALSO

mrinfo(1M), map-mbone(1M).

DVMRP is described, along with other multicast routing algorithms, in the paper "Multicast Routing in Internetworks and Extended LANs" by S. Deering, in the *Proceedings of the ACM SIGCOMM '88 Conference*.

### AUTHORS

Steve Deering, Ajit Thyagarajan, Bill Fenner.

**NAME**

mtail - tails the mail log file

**SYNOPSIS**

`/usr/sbin/mtail` [*n*]

**DESCRIPTION**

**mtail** displays the last part of the mail log, typically `/var/adm/syslog/mail.log`. By default, it displays the last 20 lines of this log.

**Options**

*n* Display last *n* lines of `/var/adm/syslog/mail.log` instead of just 20.

**SEE ALSO**

sendmail(1M).

**NAME**

mvdir - move a directory

**SYNOPSIS**

`/usr/sbin/mvdir dir newdir`

**DESCRIPTION**

**mvdir** moves one directory tree into another existing directory (within the same file system), or renames a directory without moving it.

*dir* must be an existing directory.

If *newdir* does not exist but the directory that would contain it does, *dir* is moved and/or renamed to *newdir*. Otherwise, *newdir* must be an existing directory not already containing an entry with the same name as the last pathname component of *dir*. In this case, *dir* is moved and becomes a subdirectory of *newdir*. The last pathname component of *dir* is used as the name for the moved directory.

**mvdir** refuses to move *dir* if the path specified by *newdir* would be a descendent directory of the path specified by *dir*. Such cases are not allowed because cyclic sub-trees would be created as in the case, for example, of **mvdir** *x/y x/y/z/t* which is prohibited.

**mvdir** does not allow directory *.* to be moved.

Only users who have appropriate privileges can use **mvdir**.

**EXTERNAL INFLUENCES****International Code Set Support**

Single- and multi-byte character code sets are supported.

**AUTHOR**

**mvdir** was developed by OSF and HP.

**SEE ALSO**

cp(1), mkdir(1), mv(1).

**STANDARDS CONFORMANCE**

**mvdir**: SVID2, SVID3