

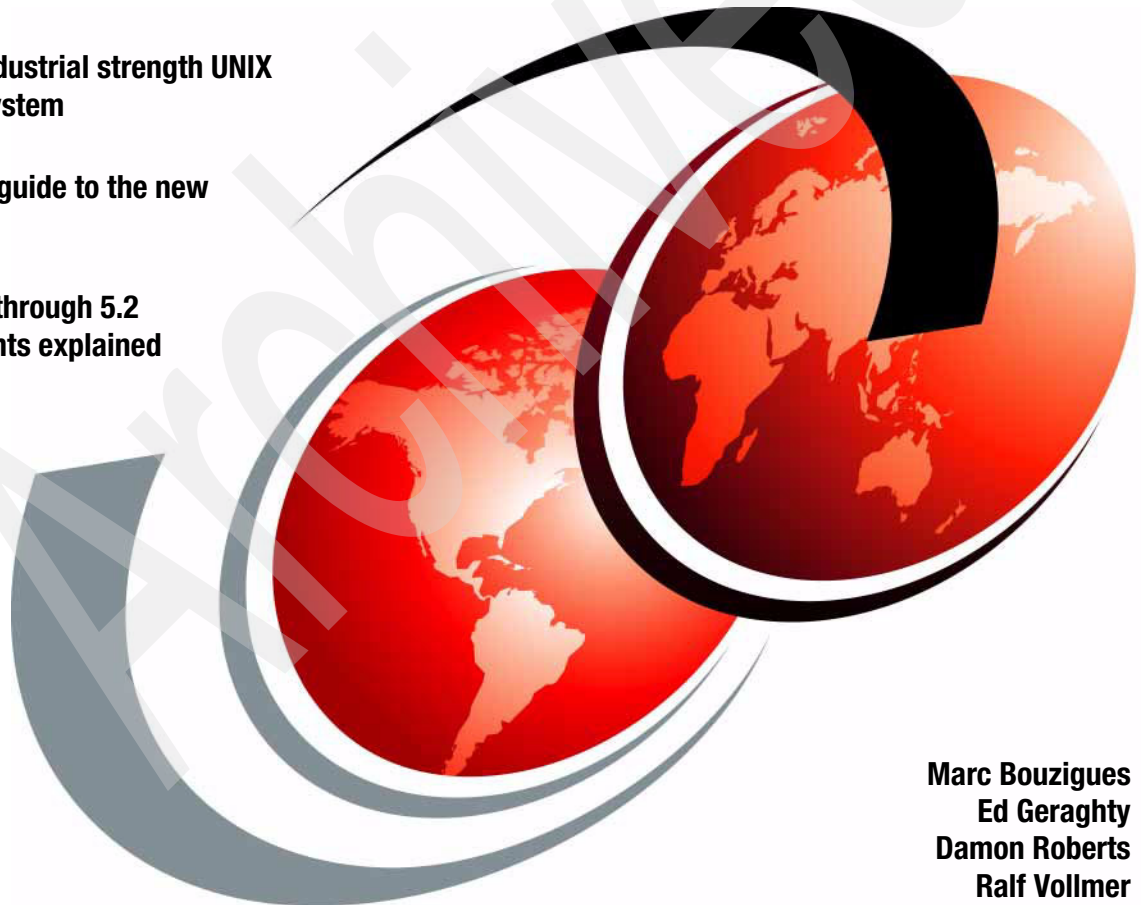


AIX 5L Differences Guide Version 5.2 Edition

AIX - The industrial strength UNIX
operating system

An expert's guide to the new
release

Version 5.0 through 5.2
enhancements explained



Marc Bouzigues
Ed Geraghty
Damon Roberts
Ralf Vollmer

ibm.com/redbooks

Redbooks



International Technical Support Organization

AIX 5L Differences Guide Version 5.2 Edition

December 2002

Archived

Note: Before using this information and the product it supports, read the information in “Notices” on page xxix.

Third Edition (December 2002)

This edition applies to AIX 5L for POWER Version 5.2, program number 5765-E62.

© Copyright International Business Machines Corporation 2001, 2002. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	xvii
Tables	xxv
Notices	xxix
Trademarks	xxx
Preface	xxxi
The team that wrote this redbook	xxxi
Become a published author	xxxviii
Comments welcome	xxxix
Chapter 1. Introduction to the enhancements	1
1.1 AIX 5L Version 5.2 enhancements	3
1.2 AIX 5L Version 5.1 enhancements	6
Chapter 2. Application development	11
2.1 Large data type support - binary compatibility	12
2.2 Very large program support (5.2.0)	12
2.2.1 The very large address space model	12
2.2.2 Enabling the very large address space model (5.2.0)	13
2.3 Malloc enhancements	14
2.3.1 Malloc multiheap	14
2.3.2 Malloc buckets	14
2.3.3 Malloc enhancement (5.2.0)	15
2.4 pthread differences and enhancements	15
2.4.1 Debug library	15
2.4.2 Unregister atfork handler	16
2.4.3 atfork and cancellation cleanup handler support (5.1.0)	16
2.4.4 Wait list and pthread state information enhancements (5.1.0)	17
2.4.5 Signal context support enhancements (5.1.0)	17
2.4.6 Deadlock detection (5.1.0)	18
2.4.7 Resource query support (5.1.0)	19
2.4.8 Multiple read/write lock read owners	20
2.4.9 Thread level resource collection (5.1.0)	20
2.5 POSIX-compliant AIO (5.2.0)	20
2.6 Context switch avoidance	21
2.7 Defunct process harvesting (5.2.0)	22
2.7.1 Zombie harvesting	22

2.7.2	Zombie harvesting in versions prior to Version 5.2	22
2.7.3	Zombie harvesting in Version 5.2	22
2.8	Software-vital product data (5.1.0)	23
2.9	KornShell enhancements	25
2.9.1	ksh93	25
2.9.2	New value for shell attribute	25
2.10	Perl 5.6 (5.1.0)	26
2.10.1	Installing more than one Perl version	26
2.10.2	Security considerations	26
2.11	Java currency	27
2.12	Common Information Model	27
2.12.1	CIM base support (5.1.0)	27
2.12.2	Common Information Model (5.2.0)	28
2.13	OpenGL 64-bit support in DWA mode (5.1.0)	30
Chapter 3.	Resource management	33
3.1	Workload Manager (WLM)	34
3.1.1	Workload Manager enhancements history	35
3.1.2	Concepts and architectural enhancements	37
3.1.3	Automatic assignment	45
3.1.4	Manual assignment	46
3.1.5	Resource sets	51
3.1.6	WLM configuration enhancements	55
3.1.7	Monitoring WLM with wlmmon and wlmpref (5.1.0)	71
3.1.8	Workload Manager enhancements (5.2.0)	83
3.2	Logical partitioning	100
3.2.1	Hardware Management Console (HMC)	101
3.2.2	LPAR minimum requirements	102
3.2.3	Memory guidelines for LPAR	102
3.2.4	Dynamic LPAR (5.2.0)	104
3.2.5	Using the AIX DLPAR Framework	113
3.3	Capacity Upgrade on Demand	131
3.3.1	The chcod command (5.1.0)	132
3.3.2	Enhancement to the lsvpd command (5.2.0)	132
3.4	Dynamic CPU sparing and CPU Guard (5.2.0)	133
3.4.1	Change CPU Guard default (5.2.0)	134
3.5	UE-Gard (5.2.0)	136
3.6	Resource set scheduling and affinity services	136
3.6.1	Memory affinity	142
3.6.2	Large page support	143
3.7	Resource Monitoring and Control	145
3.7.1	Packaging and installation	146
3.7.2	Concepts of RMC	146

3.7.3	How to set up an efficient monitoring system	152
3.7.4	Web-based System Manager enhancements (5.1.0)	153
3.7.5	Resources	158
3.7.6	Command line interface (5.1.0)	160
3.7.7	RSCT NLS enablement (5.2.0)	162
3.8	Cluster System Management	162
3.8.1	Overview	162
3.8.2	Hardware control and integration	164
3.8.3	AIX consumability	164
3.8.4	Interoperability between AIX and Linux	165
Chapter 4.	Storage management	167
4.1	Multipath I/O (5.2.0)	168
4.1.1	MPIO device driver overview	168
4.1.2	MPIO concepts	169
4.1.3	Detecting an MPIO-capable device	172
4.1.4	ODM changes for MPIO device	173
4.1.5	Path management	174
4.1.6	The rmpath command	175
4.1.7	The lspath command	177
4.1.8	The chpath command	179
4.1.9	Device management	181
4.1.10	The iostat command enhancements	185
4.2	LVM enhancements	187
4.2.1	The redefinevg command	187
4.2.2	Read-only varyonvg	187
4.2.3	LVM hot spare disk in a volume group	187
4.2.4	Support for different logical track group sizes	196
4.2.5	LVM hot-spot management	198
4.2.6	The migratelp command	207
4.2.7	The recreatevg command	208
4.2.8	The mkvg command (5.1.0)	208
4.2.9	Passive mirror write consistency check	209
4.2.10	Thread-safe liblvm.a	211
4.2.11	Advanced RAID support (5.2.0)	211
4.2.12	Bad block configuration	212
4.2.13	Snapshot support for mirrored VGs (5.2.0)	213
4.2.14	Performance improvement of LVM commands (5.2.0)	214
4.2.15	Unaligned I/O support in LVM (5.2.0)	215
4.2.16	Logical Volume serialization (5.2.0)	215
4.2.17	The mklv and extendlv commands (5.1.0)	216
4.3	JFS enhancements	218
4.3.1	The root file system ownership (5.1.0)	218

4.3.2	Directory name lookup cache (5.2.0)	218
4.3.3	The .indirect for JFS (5.1.0)	219
4.3.4	Complex inode lock (5.1.0)	220
4.3.5	The defragfs command enhancement (5.2.0)	220
4.3.6	du and df command enhancements (5.2.0)	221
4.3.7	rmfs command enhancement (5.2.0)	221
4.3.8	Increased file descriptor limit (5.2.0)	221
4.3.9	importvg command enhancement (5.2.0)	221
4.3.10	RAM disk enhancement (5.2.0)	222
4.3.11	Megabyte and Gigabyte file systems (5.2.0)	222
4.4	The enhanced Journaled File System	224
4.4.1	New in JFS2	224
4.4.2	Compatibility	227
4.4.3	Commands and utilities changes	229
4.4.4	JFS2 rootvg support for 64-bit systems (5.1.0)	236
4.4.5	JFS2 performance enhancements (5.1.0)	238
4.4.6	JFS2 support for filemon and fileplace (5.2.0)	239
4.4.7	JFS2 large file system (5.2.0)	240
4.4.8	JFS and JFS2 file system sizes (5.2.0)	240
4.4.9	JFS2 log sizes (5.2.0)	240
4.4.10	JFS2 performance enhancements (5.2.0)	240
4.4.11	JFS2 snapshot image (5.2.0)	242
4.5	VERITAS Foundation Suite for AIX (5.1.0)	250
4.5.1	VERITAS Foundation Suite on the AIX Bonus Pack	251
4.5.2	Why use VERITAS Foundation Suite on AIX	251
4.5.3	Support for LVM and JFS for AIX	252
4.6	AIX iSCSI Initiator Version 1.0 (5.2.0)	252
4.7	NFS enhancements	253
4.7.1	NFS statd multithreading	253
4.7.2	Multithreaded AutoFS	253
4.7.3	Cache file system enhancements	254
4.7.4	The cachefslog command (5.1.0)	254
4.7.5	NFS cache enhancement	255
4.7.6	Netgroups for NFS export (5.1.0)	255
4.7.7	unmount command enhancement (5.2.0)	256
4.8	CD-ROM/DVD-RAM automount facility (5.2.0)	257
4.8.1	The cdromd daemon	257
4.8.2	User commands for the automount facility	258
4.9	Uppercase mapping for ISO CD-ROM (5.1.0)	260
4.10	Common HBA API support (5.2.0)	261
	Chapter 5. Reliability, availability, and serviceability	263
5.1	Error log enhancements	264

5.1.1	Elimination of duplicate errors	264
5.1.2	The errpt command enhancements	264
5.1.3	Link between error log and diagnostics	265
5.1.4	Error log enhancements (5.2.0)	266
5.2	Trace facility (5.1.0)	267
5.2.1	The trace command enhancements	267
5.2.2	The trcrpt command enhancements	269
5.2.3	Trace event groups	269
5.3	Trace Report GUI (5.2.0)	272
5.4	Loader trace hooks (5.2.0)	275
5.5	System dump enhancements	276
5.5.1	The dumpcheck command	276
5.5.2	The coredump() system call	277
5.5.3	The snap command enhancements	277
5.5.4	Dedicated dump device (5.1.0)	277
5.5.5	System dump facility enhancements (5.2.0)	279
5.6	The adump command enhancement (5.2.0)	280
5.7	System hang detection	280
5.7.1	Priority management (5.2.0)	282
5.7.2	Lost I/O management (5.2.0)	284
5.8	Fast device configuration enhancement	284
5.9	Boot LED displays (5.2.0)	285
5.10	Improved PCI FRU isolation (5.2.0)	287
5.10.1	EEH overview	287
5.10.2	Detailed description of EEH	288
5.10.3	EEH-supported adapters	289
5.10.4	AIX error logging	291
5.10.5	Error log entries	291
5.11	DBX enhancements	292
5.11.1	The dbx command enhancements (5.2.0)	294
5.12	KDB kernel and kdb command enhancements	297
5.12.1	Kernel debugger introduction	297
5.12.2	New functions and enhancements (5.1.0)	297
5.12.3	New functions and enhancements (5.2.0)	301
5.13	Lightweight core file support	307
5.14	Core file naming enhancements (5.1.0)	308
5.14.1	File naming	308
5.14.2	Error log entry (5.2.0)	309
5.15	Gathering core files (5.1.0)	310
5.15.1	Using the snapcore command	310
5.15.2	Using the check_core utility	311
5.16	Non-sparseness support for the restore command	312
5.17	The pax command enhancements	313

5.18 The snap command enhancements (5.1.0)	313
5.18.1 Flag enhancements	313
5.18.2 The -T flag	315
5.19 The tar command enhancements (5.2.0)	315
Chapter 6. System management	317
6.1 Installation and migration	318
6.1.1 Alternate disk install enhancement (5.2.0)	318
6.1.2 NIM enhancement (5.2.0)	321
6.1.3 Version 5.2 AIX migration (5.2.0)	323
6.2 Web-based System Manager	329
6.2.1 Web-based System Manager architecture	329
6.2.2 Web-based System Manager enhancements for AIX 5L	340
6.2.3 Web-based System Manager PC Client (5.1.0)	346
6.2.4 Web-based System Manager Client for Linux (5.2.0)	352
6.2.5 Accessibility for Web-based System Manager	353
6.3 Documentation search-engine enhancement	354
6.4 Information Center (5.2.0)	355
6.4.1 AIX online message database	356
6.5 Software license agreement enhancements (5.1.0)	357
6.5.1 The inulag command	358
6.5.2 The installp command enhancements	359
6.5.3 The lspp command enhancements	360
6.5.4 Additional information in the bosinst.data file	361
6.5.5 System installation (BOS install)	362
6.5.6 Accepting licenses after reboot	362
6.5.7 SMIT function enhanced	363
6.5.8 Islicense and chlicense enhancement (5.2.0)	364
6.6 The bffcreate and lppmgr enhancement (5.2.0)	365
6.7 Comparison reports for LPPs (5.2.0)	368
6.8 mksysb on CD or DVD (5.1.0)	372
6.8.1 Personal system backup	372
6.8.2 Generic backup	372
6.8.3 Non-bootable volume group backup	372
6.8.4 Tested software and hardware	372
6.9 The mkcd command enhancement (5.2.0)	373
6.9.1 ISO9660 format	374
6.9.2 UDF format	375
6.9.3 Additional flags for the mkcd command	375
6.10 Enhanced restore command (5.2.0)	376
6.10.1 Overview	377
6.11 Paging space enhancements	378
6.11.1 Deactivating a paging space	378

6.11.2	Decreasing the size of a paging space	380
6.12	The dd command enhancement (5.1.0)	383
6.13	shutdown enhancements	384
6.14	Crontab enhancements (5.1.0)	385
6.15	Sendmail upgrade enhancements (5.1.0)	386
6.15.1	Sendmail 8.11.0 supports the Berkeley DB	386
6.16	NCARGS value configuration (5.1.0)	387
6.17	Extended host name support (5.1.0)	389
6.18	OpenType font support (5.1.0)	390
6.18.1	TrueType rasterizer	390
6.18.2	AGFA rasterizer enhancement (5.2.0)	390
6.19	Terminal support enhancements (5.1.0)	391
6.19.1	ANSI terminal support	391
6.20	New utmpd daemon (5.2.0)	392
6.21	System information command (5.2.0)	392
Chapter 7.	Performance management	395
7.1	Performance tools	396
7.1.1	Performance tools repackaging (5.1.0)	396
7.1.2	Emulation and alignment detection	397
7.1.3	Performance monitor API	397
7.1.4	The locktrace command (5.1.0)	398
7.1.5	Cmdstat tools enhancement (5.1.0)	399
7.1.6	The vmstat command enhancements	400
7.1.7	The iostat command enhancements	400
7.1.8	The netpmn and filemon command enhancements	401
7.1.9	The gennames command (5.1.0)	402
7.1.10	The svmon command enhancements	402
7.1.11	The svmon command enhancements (5.2.0)	406
7.1.12	The topas command enhancements	406
7.1.13	FDPR binary optimizer	409
7.1.14	The tprof command	409
7.1.15	The gensyms command (5.2.0)	416
7.1.16	The pstat command (5.2.0)	416
7.1.17	CPU Utilization Reporting Tool (curt) (5.2.0)	416
7.1.18	Simple Performance Lock Analysis Tool (splat) (5.2.0)	419
7.1.19	Perfstat API library (5.1.0 and 5.2.0)	421
7.1.20	Xprofiler analysis tool (5.2.0)	422
7.2	AIX tuning framework (5.2.0)	424
7.2.1	The /etc/tunables commands	425
7.2.2	Tuning commands enhancement	425
7.2.3	Web-based System Manager access	428
7.2.4	SMIT access	429

Chapter 8. Networking	431
8.1 Quality of Service support	432
8.1.1 QoS manager overlapping policies	432
8.1.2 QoS manager command line support	435
8.1.3 Quality of Service enhancements (5.2.0)	436
8.2 BIND 9 enhancements (5.2.0)	439
8.3 TCP/IP routing subsystem enhancements	460
8.3.1 Multipath routing	460
8.3.2 Dead gateway detection	466
8.3.3 User interface for multipath routing and DGD	477
8.4 TCP/IP general enhancements	481
8.4.1 Split-connection proxy systems (5.1.0)	481
8.4.2 TCP splicing (5.1.0)	482
8.4.3 UDP fragmentation (5.1.0)	483
8.4.4 TCB headlock (5.1.0)	484
8.4.5 Explicit Congestion Notification (5.1.0)	484
8.4.6 IPv6 API upgrade (5.1.0)	486
8.4.7 Performance enhancements (5.2.0)	487
8.4.8 TCP/UDP inpcb hash table tunable enhancements (5.2.0)	488
8.4.9 TCP keep alive enhancements (5.2.0)	488
8.4.10 Asynchronous accept() routine supported (5.2.0)	489
8.4.11 IPv6 functional update (5.2.0)	489
8.5 TCP/IP RAS enhancements (5.1.0)	491
8.5.1 Snap enhancement	491
8.5.2 Network option enhancements	491
8.5.3 The iptrace command enhancement	493
8.5.4 Trace enhancement	494
8.6 Virtual IP address support	495
8.6.1 Virtual IP address enhancement (5.2.0)	499
8.7 Mobile IPv6 (5.2.0)	502
8.8 DHCP enhancements (5.2.0)	504
8.9 FTP server enhancements (5.2.0)	509
8.10 Network buffer cache dynamic data support	512
8.10.1 Dynamic data buffer cache	513
8.10.2 Cache object-specific expiration time	515
8.11 Direct I/O and callbacks for NBC (5.2.0)	516
8.11.1 Callback for NBC	516
8.11.2 Direct I/O for NBC	518
8.12 HTTP GET kernel extension enhancements	518
8.12.1 HTTP 1.1 persistent connections support	519
8.12.2 External 64-bit FRCA API	520
8.12.3 Memory-based HTTP entities caching	521
8.13 Packet capture library	522

8.14 Firewall hooks enhancements	523
8.15 Fast Connect enhancements	525
8.15.1 Locking enhancements	526
8.15.2 Per-share options	526
8.15.3 PC user name to AIX user name mapping	526
8.15.4 Windows Terminal Server support	528
8.15.5 Search caching	528
8.15.6 Memory-mapped I/O (5.1.0)	529
8.15.7 send_file API	530
8.16 SMB file system support (5.2.0)	530
8.16.1 Installing SMBFS	531
8.16.2 Mounting a file system	531
8.17 SNMPv3 (5.2.0)	532
8.17.1 AIX SNMP subagent for enterprise MIB	535
8.18 Internet Key Exchange enhancements (5.1.0)	536
8.18.1 Security enhancements	536
8.18.2 New serviceability features	541
8.18.3 System management enhancements	541
8.18.4 Notify messages	543
8.18.5 The syslog enhancements	544
8.19 Dynamic Feedback Protocol (5.1.0)	545
8.19.1 The dfpd agent	545
8.19.2 Configuration file	546
8.19.3 Reports	546
8.20 ATM LANE and MPOA enhancements	547
8.20.1 Debug option (5.1.0)	548
8.20.2 IP fragmentation (5.1.0)	548
8.20.3 Token-ring support for MPOA	552
8.20.4 ATM communications support for UNI and ILMI V4.0 (5.2.0)	553
8.21 ATM network performance enhancements (5.2.0)	553
8.21.1 Changes to LANE2 timers design	554
8.21.2 Changes to checksum offload design	555
8.21.3 Changes to dynamic MTU design	555
8.22 EtherChannel enhancements (5.1.0)	556
8.22.1 Network interface backup mode	556
8.23 EtherChannel backup (5.2.0)	560
8.23.1 EtherChannel overview	560
8.23.2 EtherChannel backup adapter	561
8.23.3 netif_backup mode	561
8.23.4 Configuration	562
8.24 Virtual Local Area Network (5.1.0)	563
8.25 AIX Web browser support (5.2.0)	566

Chapter 9. Security, authentication, and authorization	569
9.1 Java security enhancements (5.1.0)	570
9.1.1 Certificate Management Protocol	570
9.1.2 Java Cryptography Extension	570
9.1.3 Java Secure Sockets Extension	570
9.1.4 Public-Key Cryptography Standards	571
9.2 User and group integration	571
9.2.1 Existing authentication methods	571
9.2.2 Identification and authentication architecture	573
9.2.3 Native Kerberos Version 5 support	575
9.3 Concurrent groups enhancement (5.1.0)	579
9.4 IBM SecureWay Directory Version 3.2	579
9.4.1 LDAP overview	580
9.5 IBM Directory Server Version 4.1 (5.2.0)	582
9.5.1 LDAP 64-bit client and C API (5.2.0)	583
9.6 LDAP name resolution enhancement	583
9.6.1 IBM SecureWay Directory schema for LDAP name resolution	583
9.6.2 LDIF file for LDAP host database	585
9.6.3 LDAP configuration file for local resolver subroutines	586
9.6.4 LDAP-based name resolution configuration	588
9.6.5 Performance and limitations	589
9.7 LDAP security audit plug-in (5.1.0)	589
9.7.1 Implementation	590
9.7.2 Configuration files	590
9.7.3 Audit information	592
9.8 Overall AIX directory integration (5.2.0)	592
9.9 Directory-enabled printing (5.2.0)	594
9.10 AIX security LDAP integration (5.2.0)	599
9.10.1 Host login restrictions for LDAP users	609
9.11 Updating password maps in NIS (5.1.0)	611
9.12 NIS/NIS+ integration into LDAP (5.2.0)	611
9.13 Pluggable Authentication Module support	614
9.13.1 PAM services (5.1.0)	614
9.13.2 PAM enhancements (5.2.0)	614
9.14 Public Key Infrastructure enhancements (5.2.0)	621
9.14.1 Overview of PKI and Certificate Authentication Service	622
9.14.2 LDAP server installation and configuration	624
9.14.3 Certificate Authentication Service configuration	629
9.14.4 Common user and administrator tasks using PKI	638
9.14.5 Process authentication group commands	640
9.15 CAPP and EAL4+ security install (5.2.0)	642
9.15.1 Packaging summary	642
9.15.2 Installation steps	643

9.16	Tivoli readiness	648
9.17	TCB integration with Tivoli Risk Manager (5.2.0)	648
9.18	Enterprise Identity Mapping (5.2.0)	649
9.19	Enhanced login privacy (5.2.0)	649
9.20	Cryptographically secure pseudo-random numbers	651
9.21	IP security enhancements (5.2.0)	652
9.21.1	IKE components using /dev/random	652
9.21.2	Diffie-Hellman group 5 supported	652
9.21.3	Generic data management tunnel support	654
9.21.4	SMIT IKE support (5.2.0)	656
9.21.5	Web-based System Manager for IP security enhancements	658
9.21.6	IP Security static filter description	659
9.21.7	Cryptographic Library	660
9.22	Secure rcmds enhancements (5.2.0)	661
Chapter 10. System V affinity		663
10.1	Weak symbol support (5.2.0)	664
10.1.1	AIX C++ compiler	664
10.1.2	GNU C++ compiler and templates	664
10.1.3	Differences between weak and global links	665
10.2	System V commands (5.2.0)	665
10.2.1	atrm	666
10.2.2	cpio	666
10.2.3	date	667
10.2.4	df	667
10.2.5	dfshares	668
10.2.6	dfmounts	669
10.2.7	dircmp	669
10.2.8	dispgid	670
10.2.9	dispuid	670
10.2.10	getconf	670
10.2.11	getdev	671
10.2.12	getdgrp	672
10.2.13	groups	673
10.2.14	last	673
10.2.15	ldd	674
10.2.16	listdgrp	674
10.2.17	ln	675
10.2.18	logins	675
10.2.19	mach	676
10.2.20	ps	677
10.2.21	pwck	677
10.2.22	quot	678

10.2.23	settime	679
10.2.24	setuname	679
10.2.25	swap	679
10.2.26	umountall	680
10.2.27	wall	680
10.2.28	whodo	681
10.2.29	zdump	681
10.2.30	zic	682
10.3	The /proc file system	684
10.3.1	The /proc file system enhancements (5.2.0)	688
10.3.2	/proc/pid#/cwd	688
10.3.3	/proc/pid#/fd	688
10.4	New proctools (5.2.0)	688
10.4.1	procwdx	689
10.4.2	procfiles	689
10.4.3	procflags	689
10.4.4	proccred	689
10.4.5	procmap	690
10.4.6	procldd	690
10.4.7	procsig	690
10.4.8	procstack	691
10.4.9	procstop	691
10.4.10	procrun	692
10.4.11	procwait	692
10.4.12	proctree	692
10.5	Process system call tracing with truss	693
10.5.1	Truss enhancements (5.2.0)	695
10.6	User API for Sun threaded applications (5.2.0)	696
10.6.1	Application binary interface (ABI)	696
10.6.2	AIX LPP packaging	697
10.7	System V Release 4 print subsystem	697
10.7.1	Understanding the System V print service	698
10.7.2	Packaging and installation	701
10.7.3	System V print subsystem management	711
10.7.4	User interface specifications	713
10.7.5	User interface for AIX and System V print subsystems	715
10.7.6	Terminfo and supported printers	720
10.7.7	Switching between AIX and System V print subsystems	723
10.7.8	Enable debugging for qdaemon	726
10.7.9	Enable debugging for JetDirect backend	726
10.8	SMIT System V print (5.2.0)	727
10.8.1	Installation	727
10.8.2	SMIT integration	728

Chapter 11. Linux affinity	733
11.1 The geninstall command (5.1.0)	734
11.1.1 Install RPM packages	735
11.1.2 Install AIX LPPs	736
11.2 The gencopy command (5.1.0)	738
11.2.1 Examples	739
11.3 Install Wizard for applications (5.1.0)	740
11.3.1 Invoking the Wizard	741
11.3.2 Example of the Install Wizard	741
11.4 The devinstall command enhancement (5.1.0)	747
11.4.1 The previous structure of devinstall	747
11.4.2 Structure of the new version of devinstall	748
11.5 BOS installation allows different desktops (5.1.0)	750
11.5.1 Using a TTY console	750
11.5.2 Using a LFT console	751
11.5.3 Using NIM for BOS installation	752
11.6 AIX Toolbox for Linux Applications	753
11.6.1 Basic Linux commands	754
11.6.2 System management tools	754
11.6.3 Red Hat Package Manager	756
11.6.4 Graphical framework	758
11.7 AIX source affinity for Linux applications (5.1.0)	762
11.7.1 Compiling open source software	764
 Chapter 12. Hardware support	 765
12.1 AIX 5L 64-bit kernel overview	766
12.1.1 Why a 64-bit kernel is needed	766
12.1.2 64-bit kernel considerations	767
12.1.3 External page table scaling for 64-bit kernel (5.2.0)	767
12.2 Interrupt saturation avoidance (5.2.0)	767
12.3 Hardware Multithreading enabling (5.1.0)	768
12.4 DVD-ROM support (5.2.0)	769
12.5 Kernel scalability for SMP machines (5.1.0)	769
12.5.1 Proch callouts implementation	769
12.6 Audio support for the 64-bit kernel (5.1.0)	770
12.7 The millicode functions (5.2.0)	770
12.8 Ultimeia and PCMCIA device restrictions	771
12.9 Diagnostics enhancements	771
12.9.1 Turboways PCI ATM adapter diagnostic enhancements (5.1.0) ..	771
12.9.2 LS-120 floppy drive diagnostic support (5.1.0)	774
12.9.3 Physical location codes (5.2.0)	774
12.10 Common Character Mode support for AIX (5.1.0)	775
12.10.1 PCI Common Character Mode	775

12.10.2 Device driver configuration	775
12.11 AIX configuration commands (5.2.0)	775
12.11.1 The prtftconf command	776
12.11.2 The lsconf command	776
12.12 Hardware support (5.2.0)	776
Chapter 13. National language support	783
13.1 Input methods for Chinese locales (5.1.0)	784
13.1.1 Input methods window	784
13.1.2 Intelligent ABC Input Method	785
13.1.3 BiaoXing Ma Input Method	786
13.1.4 Zheng Ma Input Method	786
13.1.5 PinYin Input Method	787
13.1.6 Internal Code Input Method	788
13.2 Euro support for non-European countries (5.1.0)	789
13.2.1 Testing the Euro glyph	790
13.3 National language support Euro (5.2.0)	791
13.4 Korean keyboard enablement (5.1.0)	793
13.5 NLS: Unicode Extension B Enhancement (5.2.0)	793
13.5.1 Enhancements to Version 5.2	794
13.6 Unicode XOM enhancement (5.2.0)	794
13.7 Additional locale support (5.2.0)	795
13.8 Removal of obsolete locales (5.2.0)	797
13.9 Unicode 3.1 support (5.2.0)	797
13.10 NLS JISX0213 compliance (5.2.0)	799
Abbreviations and acronyms	801
Related publications	811
IBM Redbooks	811
Other resources	811
Referenced Web sites	811
How to get IBM Redbooks	814
IBM Redbooks collections	814
Index	815

Figures

2-1	CIM logical flow diagram	29
3-1	Web-based System Manager Overview and Tasks dialog	35
3-2	Basic Workload Manager elements in AIX Version 4.3	36
3-3	Hierarchy of classes.	39
3-4	Resources cascading through tiers	42
3-5	SMIT with the class creation attributes screen	43
3-6	SMIT panel shows the additional localshm attribute	45
3-7	Resource set definition to a specific class	52
3-8	SMIT main panel for resource set management	53
3-9	SMIT panel for rset registry management	54
3-10	SMIT panel to add a new resource set	55
3-11	SMIT main panel for Workload Manager configuration	56
3-12	Web-based System Manager options for Workload Manager	57
3-13	An example of adding a Subclass to a Superclass	59
3-14	Example of SMIT panel for creating a new rule	63
3-15	Fields that can be modified for a specific rule	65
3-16	SMIT panel for Update Workload Management.	66
3-17	SMIT panel for manual assignment of processes	67
3-18	WLM_Console tab-down menu	72
3-19	Report browser	73
3-20	Bar view	74
3-21	Snapshot view	75
3-22	Table view	76
3-23	Report properties	76
3-24	Times menu	78
3-25	Example of trend display, Bar View	78
3-26	Example of trend display, Snapshot View	79
3-27	Tier/Class menu	79
3-28	Advanced menu	80
3-29	Example of the Advanced menu	81
3-30	Select the configuration to add the attribute value group to.	84
3-31	Right-click the Attribute Value Groups option	85
3-32	Attribute Value configuration screen	85
3-33	Adding an attribute value group	86
3-34	New Condition menu option for monitoring	87
3-35	New condition configuration box	88
3-36	Time-based configuration drop-down menu	90
3-37	Drop-down to create the configuration set	91

3-38	Defining the configuration set	92
3-39	Selecting the configuration file and setting the times it is functional . . .	93
3-40	Time-based configurations.	94
3-41	WLM Overview and Tasks submenu, Total Limits section	96
3-42	Selecting the properties of a configuration.	97
3-43	Process Limits configuration screen	98
3-44	Class member limits.	99
3-45	IBM eServer pSeries DLPAR system architecture.	105
3-46	DLPAR system architecture.	107
3-47	HMC slot removal	113
3-48	DLPAR operation phases	115
3-49	HMC memory profile	126
3-50	Output of the ps -lemo THREAD	131
3-51	UE-Gard logic	135
3-52	Condition Properties dialog - General tab	147
3-53	Condition Properties dialog - Monitored Resources tab	148
3-54	Response Properties dialog - General tab.	149
3-55	Action Properties dialog - General tab.	150
3-56	Action Properties dialog - When in effect tab.	151
3-57	Web-based System Manager, Host Overview plug-in	154
3-58	Web-based System Manager, Host menu of the Overview plug-in . . .	155
3-59	Web-based System Manager, audit log panel.	156
3-60	Web-based System Manager, condition property panel	157
3-61	Web-based System Manager, conditions panel	157
4-1	Three adapters connected to a single device without MPIO facility. . .	170
4-2	Three adapters connected to a single MPIO device	170
4-3	This panel shows the device of hdisk9	182
4-4	Selection of a parent	183
4-5	List the all the devices under a parent.	183
4-6	Selection of a device, hdisk9 in this example	184
4-7	Displays the parent of hdisk9.	184
4-8	Volume Group Properties dialog	189
4-9	Physical Volumes notebook tab.	190
4-10	Advanced Method of volume group creation	191
4-11	New Volume Group dialog	192
4-12	New Volume Group, second panel in dialog	193
4-13	New Volume Group, third panel in dialog	194
4-14	New Volume Group, fourth panel in dialog	195
4-15	New Volume Group, fifth panel in dialog	196
4-16	Volume Group Properties dialog	198
4-17	Volume Group Properties Hot Spot Reporting tab.	201
4-18	Logical Volumes Properties notebook	202
4-19	Manage Hot Spots sequential dialog	203

4-20	Hot Spot Management dialog	204
4-21	Hot Spot Management statistics	205
4-22	Hot Spot selection	206
4-23	Physical destination partition	207
4-24	Logical volume serialization	216
4-25	File system list panel	223
4-26	File system Size panel	224
4-27	Example of a server importing and mounting JFS volumes.	227
4-28	AIX 5L JFS2 machine NFS mounting a JFS file system	228
4-29	AIX 4.X JFS machine NFS mounting a JFS2 file system	229
4-30	SMIT panel for JFS2 management	230
4-31	SMIT panel for adding a JFS2 file system	231
4-32	SMIT panel for adding a logical volume and assigning as JFS2	232
4-33	SMIT panel for showing the logical volume selection	233
4-34	Web-based System Manager panel for file system creation	234
4-35	SMIT panel for adding a logical volume as a jfs2log device	235
4-36	Advanced Options installation menu	237
4-37	Selecting snapshot in the Journaled File Systems submenu.	244
4-38	Snapshot creation screen, click Create	245
4-39	Snapshot creation screen with options configured	246
4-40	It is possible to changes its size, back it up, or unmount it	247
4-41	Possible to delete unmounted snapshots	248
4-42	Snapshot image screen	248
5-1	SMIT panel for START Trace	268
5-2	SMIT panel for Trace Report	269
5-3	SMIT panel for Manage Event Groups	270
5-4	SMIT panel for creating a new event group	271
5-5	SMIT panel for creating a new event group	272
5-6	tgiv view window	274
5-7	tgiv filter window	275
5-8	SMIT panel for lost I/O management.	284
6-1	Selecting alternate disk install from the Install More Software screen.	318
6-2	SMIT Alternate Disk Installation panel	320
6-3	NIM Alternate Disk Migration screen	320
6-4	SMIT nim _lppmgr panel for the lppsource lppsource234	322
6-5	BOS Installation and Maintenance menu.	325
6-6	Installation and Settings screen.	326
6-7	Method of Installation screen.	326
6-8	Disks to install screen	327
6-9	Installation and Settings screen, install method set to migrate	328
6-10	Install Options for migration install	328
6-11	Web-based System Manager user interface	331
6-12	Container plug-in example	332

6-13	Example of logical volumes container in detail view	333
6-14	Overview plug-in example, users and groups overview.	334
6-15	Web-based System Manager icon on CDE user interface	339
6-16	An example of output from a session log.	341
6-17	An example of session log detailed entry	342
6-18	Command tool creation dialog	343
6-19	Example of result type Show result panel	344
6-20	Tips bar example	344
6-21	SNMP monitor configuration through Web-based System Manager . .	346
6-22	Configassist: Configuration task manager	347
6-23	Web server to run Web-based System Manager in a browser	348
6-24	Configure Web-based System Manager Applet mode.	349
6-25	InstallShield Multi-Platform for PC Client	350
6-26	Installation of Web-based System manager PC Client	350
6-27	Log On screen for Web-based System Manager PC Client	351
6-28	Web-based System Manager PC Client	352
6-29	Accessibility example.	354
6-30	Information Center	356
6-31	View of search interface of the AIX message database	357
6-32	SMIT panel for accepting new software agreements using installp . .	360
6-33	Configuration assistant, software license after reboot	363
6-34	SMIT panel for license management	364
6-35	Licenses Web-based System Manager dialog.	365
6-36	SMIT software maintenance and utilities panel	366
6-37	Rename software image repository	366
6-38	SMIT Clean Up Software Images in Repository panel.	368
6-39	SMIT Comparison Reports panel.	370
6-40	SMIT Compare Installed Software to Fix Repository panel	371
6-41	SMIT Compare Installed Software to Fix Repository panel results . .	371
6-42	SMIT Deactivate a Paging Space panel	379
6-43	Selected pull-down for volume management.	380
6-44	SMIT panel for decreasing the size of a paging space	381
6-45	Properties dialog to increase page space	382
6-46	SMIT System Environment panel	388
6-47	SMIT Change/Show Characteristics of Operating System panel . . .	389
6-48	Telnet session from Microsoft Windows 2000	391
7-1	Topas main screen	407
7-2	Workload Manager screen using the W subcommand	408
7-3	topas with per-CPU usage enabled	409
7-4	Logic flow for post-process mode and manual offline mode	415
7-5	Xprofiler applications	424
7-6	System performance main panel	428
7-7	I/O parameters	429

7-8	The smitty tuning fast path	429
7-9	Tuning Network Option Parameters dialog	430
7-10	Change/Show Network Current Option Parameters dialog	430
8-1	DGD sample configuration	476
8-2	Add Static Route SMIT menu	480
8-3	Web-based System Manager menu for static route management.	481
8-4	Basic architecture of split-connection application layer proxies	482
8-5	The previous definition of bytes 13 and 14 of the TCP header	485
8-6	The new definition of bytes 13 and 14 of the TCP header	485
8-7	Add a Virtual IP Address Interface SMIT menu	496
8-8	SMIT Add a Virtual IP Address Interface panel	500
8-9	SMIT Change/Show a Virtual IP address Interface panel	502
8-10	The different mobile IPv6 nodes	503
8-11	SMIT Configure Mobile IPv6 panel	504
8-12	FRCA GET data flow	519
8-13	SMIT panel with user name mapping option highlighted	527
8-14	Map a Client User Name to a Server User Name panel	528
8-15	SMIT panel with Enable search caching option highlighted.	529
8-16	Send_file attributes	530
8-17	Web-based System Manager VPN screen	542
8-18	Web-based System Manager VPN Overview and Tasks panel	543
8-19	Level of IKE components to be logged	545
8-20	System environment ATM LAN Emulation	547
8-21	An example of an MPOA network	549
8-22	SMIT panel for Change/Show an MPOA client	551
8-23	SMIT panel for adding an ATM LE client	552
8-24	SMIT panel for adding a token ring ATM LE client	553
8-25	SMIT panel to add a new EtherChannel	557
8-26	SMIT panel for choosing the adapters that belong to the channel.	557
8-27	SMIT panel for configuring the EtherChannel	558
8-28	SMIT screen showing changes to allow EtherChannel backup	562
8-29	SMIT panel for adding a VLAN	564
8-30	SMIT Available Network Interfaces panel	565
8-31	SMIT Change/Show a Standard Ethernet Interface panel	565
8-32	AIX Netscape 7 Web browser	568
9-1	Implementation detail of the LDAP security audit plug-in	590
9-2	LDAP hierarchy for AIX directory-enabled subsystems	593
9-3	LDAP hierarchy for AIX System V directory-enabled printing	596
9-4	Web-based Systems Manager - Directory Enabled Printers	599
9-5	LDAP Hierarchy for AIX security database and NIS maps	601
9-6	AIX Security Service to PAM module path	616
9-7	PAM Module to AIX Security Service Path	619
9-8	LDAP hierarchy for myexample.company PKI example	624

9-9	IBM directory administration GUI - suffixes	627
9-10	SMIT screen of PKI - Change/Show a Certificate Authority.	634
9-11	SMIT screen of PKI - Change/Show a CA Account.	635
9-12	SMIT screen of PKI - Add/Change/Show an LDAP Account	635
9-13	SMIT screen of PKI - Change/Show the Policy	636
9-14	BOS Installation and Maintenance screen.	643
9-15	Installation and Settings screen	644
9-16	Change method of installation to new and complete overwrite	644
9-17	Change disks to BOS install	645
9-18	Installation and Settings screen, selecting option 3, More Options . . .	646
9-19	Install Options screen	646
9-20	Selecting CAPP and EAL4+.	647
9-21	SMIT Use Internet Key Exchange Refresh Method dialog	657
9-22	SMIT Advanced IP Security Configuration IKE enhancements	657
9-23	IP security Overview and Tasks dialog	658
9-24	IP Security Basic IKE Tunnel Connection wizard	659
10-1	Overview of print request processing.	700
10-2	Web-based System Manager menu for System V print subsystem. . .	717
10-3	Add new printer Web-based System Manager wizard: Step 4 of 4 . .	718
10-4	Print Spooling menu of SMIT.	720
10-5	Selecting System V print spooling menus	728
10-6	System V print spooling options.	729
10-7	System V print request management screen.	730
10-8	System V destination management screen	731
11-1	Installation Wizard invoked by the command line	742
11-2	Installation Wizard for selecting source of installation	743
11-3	Installation Wizard for selecting the software to install	744
11-4	Installation Wizard for selecting software from product	745
11-5	Installation Wizard to begin installation	746
11-6	Installation Wizard task panel	747
11-7	BOS installation while using a TTY console	751
11-8	BOS installation menu while using a LFT console.	751
11-9	Warning messages during desktop install	752
11-10	User administration provided by KDE	755
11-11	System V init editor provided by KDE	756
11-12	AIX Toolbox for Linux Applications graphical framework.	759
11-13	Gnome Desktop running on AIX 5L Version 5.1	760
11-14	KDE 1.1.2 desktop running on AIX 5L Version 5.1	761
11-15	Glade running on AIX 5L Version 5.1	762
12-1	Diagnostic panel for running DMA test	772
12-2	Diagnostic panel for running external wrap test.	773
12-3	Diagnostic panel for test complete.	773
12-4	Diagnostics panel.	774

13-1	Window of Chinese input method	784
13-2	ABC Input Method setting window	785
13-3	BiaoXing Ma Input Method setting window	786
13-4	Zheng Ma Input Method setting window	787
13-5	PinYin Input Method setting window	788
13-6	Internal Code Input Method setting window	788
13-7	Korean keyboard	793

Archived

Tables

2-1	The vpdadd command flags	23
2-2	The vpdcl command flags	24
2-3	Supported adapters and required filesets	31
2-4	New packaging information	31
3-1	List of process types	49
3-2	Examples of class assignment rules	50
3-3	The drmgr command flags	117
3-4	DLPAR script error and logging	118
3-5	DLPAR script commands	119
3-6	Input variables for memory add/remove operations	120
3-7	Input variables for processor add/remove operations	121
3-8	LED processor indicator codes	127
3-9	DR-related error log entries	129
3-10	The chcod command flags	132
3-11	RMC commands	160
3-12	ERRM commands	160
4-1	The mkpath command flags	175
4-2	The rmpath command flags	176
4-3	The lspath command flags	178
4-4	The chpath command flags	180
4-5	The lvmstat command flags	199
4-6	The splitvg command flags	214
4-7	Journalized file system specifications	225
4-8	Old JFS names versus new JFS2 interface names	239
4-9	CD-ROM/DVD-RAM automount flags	259
5-1	Loader trace hooks	275
5-2	System memory to dump device size ratios	279
5-3	Second line of front panel display information	285
5-4	EEH adapter support	289
6-1	List of standard plug-ins in Web-based System Manager	335
6-2	Comparison chart with the new enhancements	340
6-3	Components that are saved in the preferences file	345
6-4	Most common flags of the lppmgr command	367
6-5	Required hardware and software for backup CDs	373
6-6	Additional flags of the mkcd command	376
6-7	Most common flags for restore with -P option	377
6-8	System-wide configuration names	393
7-1	Performance tools packaging versus platform	396

7-2	The locktrace command flags	398
7-3	The curt command flags	417
7-4	The splat command flags	419
7-5	New performance APIs	422
7-6	Common flags of the tuning commands	426
8-1	Network options for dead gateway detection	474
8-2	The route command parameters for multipath routing and DGD	478
8-3	New netstat command flag	479
8-4	Static Route and Add an IPv6 Static Route SMIT menu new fields	479
8-5	Parameters of getipnodebyname	486
8-6	Parameters of getipnodebyaddr subroutine	487
8-7	The iptrace command flags	494
8-8	Per-share value options	526
8-9	The mount command flags for SMBFS	532
8-10	Linux versus AIX VPN function mapping	539
8-11	Web-based System Manager tunnel daemons	544
9-1	Java enhancements versus fileset	570
9-2	Mapping of the AIX Security Services calls and the PAM API	618
9-3	Cryptographic Library algorithms and key lengths	661
10-1	Most common flags for atrm	666
10-2	Most common flags for cpio	666
10-3	Most common flags for /usr/sysv/bin/df	667
10-4	Most common flags for dircmp	669
10-5	Most common flags for the logins command	675
10-6	Flags not found in AIX for ps	677
10-7	Most common flags for quot	678
10-8	Most common flags for the swap command	680
10-9	Most common flags for umountall	680
10-10	Most common flags for zdump	682
10-11	Most common flags for zic	682
10-12	Function of pseudo files in /proc/<pid> directory	685
10-13	Filesets for Sun user thread library	697
10-14	AIX print subsystem backend support	704
10-15	Print service commands available to all users	711
10-16	Administrative print service commands	712
10-17	System V printing: User and administrative commands	714
10-18	Supported printers in the terminfo database	720
10-19	Printer support by the System V print subsystem in AIX 5L	721
11-1	The geninstall command flags	734
11-2	The gencopy command flags	738
11-3	Form number for AIX Toolbox for Linux Applications CD	753
12-1	New flags for prtconf	776
12-2	Version 5.2 withdrawn PCI adapter support	777

12-3	Version 5.2 withdrawn PReP-specific ISA adapter support	778
12-4	Version 5.2 withdrawn ISA adapter support	779
12-5	Version 5.2 PCI RS/6000 withdrawn support listing	780
12-6	Version 5.2 MCA RS/6000 withdrawn support listing	780
12-7	Version 5.2 MCA-based SP nodes withdrawn support	782
12-8	Version 5.2 device support withdrawn	782
13-1	Modified locales for using Euro	789
13-2	Locale settings versus font fileset	789
13-3	List of euro-enabled locales	791
13-4	Additional locales	795
13-5	Obsolete locales	797
13-6	Unicode encoding as UTF-8	798

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.


This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AFP™	GXT1000™	RACF®
AFS®	IBM®	Redbooks™
AIX®	Micro Channel®	Redbooks(logo)™ 
AIX 5L™	Netfinity®	RMF™
AS/400®	NetView®	RS/6000®
DB2®	OS/390®	S/390®
DB2 Universal Database™	PAL®	SecureWay®
developerWorks™	Perform™	Sequent®
DFS™	PowerPC®	SP™
DPI®	PowerPC Reference Platform®	Tivoli®
@server	pSeries™	Versatile Storage Server™
ESCON®	PTX®	WebSphere®
FlashCopy®	QMF™	

The following terms are trademarks of International Business Machines Corporation and Lotus Development Corporation in the United States, other countries, or both:

Approach®	Lotus®	Word Pro®
-----------	--------	-----------

The following terms are trademarks of other companies:

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

C-bus is a trademark of Corollary, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.

Preface

This IBM redbook focuses on the differences introduced in AIX 5L through Version 5.2 when compared to AIX Version 4.3.3. It is intended to help system administrators, developers, and users understand these enhancements and evaluate potential benefits in their own environments.

AIX 5L introduces many new features, including Linux and System V affinity, dynamic LPAR, multipath I/O, 32- and 64-bit kernel and application support, virtual IP, quality of service enhancements, enhanced error logging, dynamic paging space reduction, hot-spare disk management, advanced Workload Manager, JFS2 snapshot image, and others. The availability of Web-based System Manager for Linux continues AIX's move towards a standard, unified interface for system tools. There are many other enhancements available with AIX 5L, and you can explore them in this redbook.

This publication is a companion publication to the previously published *AIX Version 4.3 Differences Guide*, SG24-2014, Third Edition, which focused on the enhancements introduced in AIX Version 4.3.3.

For customers who are familiar with AIX 5L Version 5.1, features that are new in AIX 5L Version 5.2 are indicated by a version number (5.2.0) in the title of the section.

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, Austin Center.

Marc Bouzigues is the benchmarking manager at the PSSC center in Montpellier, France. His responsibilities include overseeing the pSeries benchmarking (AIX environment) for complex customer environments. He architects overall solutions to meet his customer needs while handling the project management responsibilities of the center. He has over seven years of experience with the AIX and UNIX platform.

Ed Geraghty is a senior software engineer in IBM's Advanced Internet Technology group in Boston, MA. He was technical Web master on several high-profile Web sites such as the IBM Olympic Web sites, Sydney Olympic Web Store, IBM Intellectual Patent Network, and other sports sites. His areas of expertise include high-volume Web infrastructures, IT security, networks,

pSeries and RS/6000 SP systems, and Linux. He has written extensively on LDAP, PKI, PAM, and other network and security enhancements.

Damon Roberts works in IBM Global Services as an IT Specialist for e-Business hosting delivery support in Warwick, United Kingdom. He provides third-level technical support for customer machines running in IBM server farms and is currently seconded to the solution team as a Technical Solution Designer. He has over six years of experience with AIX and pSeries machines. His areas of expertise include pSeries systems, RS/6000 SP systems, Oracle, SAP Basis, performance tuning and HACMP/ES. He is both a SAP Basis and AIX certified professional.

Ralf Vollmer is an IT Specialist in pre-sales technical support for IBM @server pSeries and RS/6000, part of the Web Server Sales Organization in Stuttgart, Germany. He holds a degree in Computer Science from the University of Ulm. Ralf is a member of the AIX Technology Focus Group, supporting IBM sales, IBM Business Partners, and customers with pre-sales consultation and implementation of client/server environments. He has written extensively on Dynamic LPAR and LVM and file system enhancements.

The authors of the first edition are:

Erwin Behnen	IBM Germany
Mauro Minomizaki	IBM Brazil
Armin Olaf Roell	IBM Germany

The authors of the second edition are:

René Akeret	IBM Switzerland
Anke Hollanders	IBM Belgium
Stuart Lane	IBM South Africa
Antony Peterson	IBM Australia

The project that produced this publication was managed by:

Scott Vetter	IBM Austin
---------------------	------------

Thanks to the following people for their invaluable contributions to this project. Without their help, this publication would have been impossible.

Ackermann, Jack	IBM Austin
Adkins, Janet	IBM Austin
Alagarsamy, Gowthaman	IBM India
Albot, Andre L.	IBM Austin

Alford, Jack	IBM Austin
Allen, James P.	IBM Austin
Amin, Sandy	IBM Austin
Anderson, Ray	IBM Austin
Anglin, Debbie	IBM Austin
Anthony, Craig	IBM Austin
Arbab, Reza	IBM Austin
Arbeitman, Jane	IBM Austin
Baratakke, Kavitha	IBM Austin
Baregar, Vijayeendra N	IBM India
Batten, David	IBM Austin
Batten, Pamela	IBM Austin
Beals, Stephanie	IBM Poughkeepsie
Bekdache, Bassel	IBM Austin
Birgen, Greg	IBM Austin
Bohy, Anne	IBM Austin
Borunda, Maritza	IBM Austin
Brandyberry, Matthew	IBM Austin
Brenner, Larry	IBM Austin
Brown, Deanna Quigg	IBM Austin
Brown, Joe	IBM Austin
Brown, Mark	IBM Austin
Brown, William	IBM Austin
Browning, Luke M.	IBM Austin
Bsaibes, Mounir	IBM Austin
Burdick, Dean	IBM Austin
Buros, Bill	IBM Austin
Carroll, Scott	IBM Austin
Castillo, George	IBM Austin
Cawlfeld, Kevin J	IBM Austin
Celikkan, Ufuk	IBM Austin

Chaky, Joseph	IBM Poughkeepsie
Chang, Daisy	IBM Austin
Christensen, Carol	IBM Austin
Clar, Jean-Jacques	IBM Austin
Clissold, David	IBM Austin
Cossmann, Helmut	IBM Heidelberg
Craft, Julie	IBM Austin
Cuan, Elizabeth	IBM Austin
Cunningham, Jim	IBM Austin
Cyr, Michael	IBM Austin
Dai, Zhi-wei	IBM Austin
Date, Medha	IBM Austin
De Leon, Baltazar	IBM Austin
Dea, Frank	IBM Austin
Devendran, Saravanan	IBM Austin
Doshi, Bimal	IBM Austin
Echols, Walter	IBM Austin
Emmons, John	IBM Austin
Epsztein, Sara Dominique	IBM Austin
Fernandes, Lillian S	IBM Austin
Flaig, Greg	IBM Austin
Fontenot, Nathan	IBM Austin
Fontenot, Shevaun	IBM Austin
Freimuth, Douglas M.	IBM Watson Research
Fumery, Pierre	IBM Austin
Furutera, Masahiro	IBM Japan
Gajjar, Pankaj	IBM Austin
Geise, David	IBM Austin
Genty, Denise	IBM Austin
Greenberg, Randy	IBM Austin
Griffiths, Nigel	IBM U.K.

Grubbs, Mark	IBM Austin
Gu, Dixin	IBM Austin
Guelorget, Jacqueline	Bull France
Hall, Lon	IBM Austin
Harrell, Michael S.	IBM Austin
Haugh, Julianne	IBM Austin
Hausmann, Sally	IBM Austin
Hepkin, David A	IBM Austin
Hezari, Emilia	IBM Austin
Hoetzel, Mary	IBM Austin
Horton, Joshua	IBM Poughkeepsie
Hsiao, Duen-wen	IBM Austin
Irwin, Frank	IBM Austin
Iwata, Megumi	IBM Japan
Jain, Vinit	IBM Austin
Jones, Corradino	IBM Austin
K, Uma	IBM Austin
Kamat, Naveen	IBM India
Kitamorn, Alongkorn	IBM Austin
Kline, Nyralin	IBM Austin
Knop, Felipe	IBM Poughkeepsie
Kovacs, Bob G	IBM Austin
Laib, Greg	IBM Poughkeepsie
Lentz, Jim	IBM Austin
Liu, Su	IBM Austin
Loafman, Zachary M	IBM Austin
Lowe, Suanne	IBM Austin
Lu, Yantian (Tom)	IBM Austin
Machutt, Susan	IBM Austin
Madan, Sheena	IBM Austin
Mall, Michael	IBM Austin

Marion, Neal	IBM Austin
McBrearty, Gerald	IBM Austin
McCorkle, Brian	IBM Austin
McCracken, Dave	IBM Austin
McCreary, Hye-Young	IBM Austin
McNichol, Dan	IBM Austin
Meenakshisundaram, Subramaniam	IBM India
Messing, Jeff	IBM Austin
Michel, Larry A.	IBM Austin
Mishra, Rajeev	IBM Austin
Mita, Hajime	IBM Tokyo
Molis, Steve	IBM Austin
Morton, Beth	IBM Austin
Nakagawa, Toru	IBM Japan
Narasimhan, Rashmi	IBM Austin
Nasypany, Stephen	IBM Austin
Nema, A	IBM India
Neuman, Grover	IBM Austin
Nguyen, Dac	IBM Austin
Nichols III, Frank L.	IBM Austin
Nogueras, Jorge Rafael	IBM Austin
Nott, Norman	IBM Poughkeepsie
Olesen, Mark	IBM Austin
Pafumi, Jim	IBM Austin
Pargaonkar, Shirish	IBM Austin
Parichhah, Subhrata	IBM India
Partridge, Jim	IBM Austin
Patel, Jayant	IBM Austin
Patwari, Veena	IBM Austin
Payne, Marilyn	IBM Austin
Peckham, Steve	IBM Austin

Poston, Rick	IBM Austin
Potluri, Prasad V.	IBM Austin
Potter, Bruce M	IBM Poughkeepsie
Qureshi, Sameer	IBM Austin
Ramirez, Ruben	IBM Austin
Ramirez, Tony	IBM Austin
Rosas, Jeff	IBM Austin
Rothaupt, Krystal	IBM Poughkeepsie
Rozendal, Kenneth	IBM Austin
Rubio, Pete	IBM Austin
Scherrer, Carolyn	IBM Austin
Segura, Ernest	IBM Austin
Shaffer, Jim	IBM Austin
Shankar, Manjunatha	IBM Austin
Shankar, Ravi A	IBM Austin
Sharma, Rakesh	IBM Austin
Shi, Amy	IBM Austin
Shi, Danling	IBM Austin
Shieh, Johnny	IBM Austin
Shvartsman, Edward	IBM Austin
Simpson, John	IBM Poughkeepsie
Smolders, Luc	IBM Austin
Springen, Nancy L.	IBM Austin
Srinivasaraghavan, Subhathra	IBM India
Stephenson, Marc J.	IBM Austin
Sugarbroad, Jean-philippe	IBM Austin
Suzuki, Masato	IBM Japan
Swanberg, Randy	IBM Austin
Taylor, Kurt	IBM Austin
Thompson, Robert	IBM Austin
Tong, Duyen	IBM Austin

Toungate, Marvin	IBM Austin
Tran, Kim	IBM Austin
Tran, Scott	IBM Austin
Unnikrishnan, Rama	IBM Austin
Unruh, Steve	IBM Austin
Vaidyanathan, Basu	IBM Austin
Valentin, Patrick	Bull France
Vallabhaneni, Vasu	IBM Austin
Vazzalwar, Girish	IBM Austin
Veeramalla, Ramesh	IBM Austin
Venkatsubra, Venkat	IBM Austin
Vidya, R	IBM India
Vinit, Jain	IBM Austin
Vo, Patrick	IBM Austin
Walters, Drew	IBM Austin
Warrier, Suresh	IBM Austin
Wheeler, Wayne	IBM Austin
Wigginton, Ann	IBM Austin
Wong, Andy	IBM Austin
Wu, Jason	IBM Austin
Xie, Linda	IBM Austin
Xu, Cheng	IBM China
Yang, Rae	IBM Austin
Yerneni, Lakshmi	IBM Austin
Younghaus, Jim	IBM Austin
Yuan, Gina	IBM Poughkeepsie

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our Redbooks to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- ▶ Use the online **Contact us** review redbook form found at:

ibm.com/redbooks

- ▶ Send your comments in an Internet note to:

redbook@us.ibm.com

- ▶ Mail your comments to:

IBM Corporation, International Technical Support Organization
Dept. JN9B Building 003 Internal Zip 2834
11400 Burnet Road
Austin, Texas 78758-3493

Introduction to the enhancements

AIX 5L represents the next generation of AIX. Built on a proven code base, AIX 5L is designed to exploit advanced 64-bit system and software architectures while introducing:

- ▶ Logical partitioning
- ▶ Improved workload management
- ▶ Integrated Linux affinity
- ▶ Network performance improvement
- ▶ System security enhancements
- ▶ Reliability, availability, and serviceability (RAS) enhancements and performance-tuning tools
- ▶ Cluster Systems Management

AIX 5L Version 5.2 moves the operating system into the next stage of IT infrastructure self-management with innovative scalability technology while continuing to offer application flexibility with Linux, tools that simplify systems management, leadership security mapping between heterogeneous platforms, and affinity with pSeries focus market segments. The addition of dynamic logical partitioning and keyed Capacity Upgrade on Demand significantly improves flexibility, workload management, and system resource use in the datacenter.

AIX 5L Version 5.2 reliability and scalability, combined with application binary compatibility across all AIX Version 5 releases and concurrent 32/64-bit functionality, make it the best choice for customers who:

- ▶ Require a robust platform for business-critical applications
- ▶ Want to leverage their IT investments in technology and skills
- ▶ Have network interoperability requirements with heterogeneous systems
- ▶ Need components and tools to build tailored solutions
- ▶ Want to reduce the cost of computing through improved systems and network management
- ▶ Need security at all levels of their operating, application, and network environments
- ▶ Deploy applications worldwide requiring multilingual support

1.1 AIX 5L Version 5.2 enhancements

The following set of topics was taken from the AIX 5L Version 5.2 announcement materials. The goal is to provide you with a correlation between the announcement and the organization within this guide. This list is not an exhaustive list of enhancements to AIX 5L Version 5.2, but a list of the key features recently introduced.

- ▶ Flexibility
 - Affinity with Linux

Helps deliver services across technology boundaries by allowing portable Linux applications to be combined with the scalability and robustness of AIX. For more information, see Chapter 11, “Linux affinity” on page 733.
- ▶ System scalability
 - JFS2 file system

Efficient storage of large (16 Terabyte) files assists deployment of advanced applications and databases (see 4.4.7, “JFS2 large file system (5.2.0)” on page 240).
 - Large pages

16 MB pages help improve throughput for p670/p690 compute-intensive workloads that require large amounts of data to be transferred between memory and storage (3.6.2, “Large page support” on page 143).
- ▶ Logical partition support for p670/p690
 - Dynamic LPAR

Enables addition or removal of processors, adapters or memory without system reboot, improving system availability and resource utilization (3.2.4, “Dynamic LPAR (5.2.0)” on page 104).
 - Dynamic Capacity Upgrade on Demand (CUoD)

Allows activation of additional processors when needed—without a system or partition reboot, for greater flexibility and improved workload throughput (3.3, “Capacity Upgrade on Demand” on page 131).
 - Dynamic processor sparing (with CUoD)

Supports dynamic substitution of failing processors with spare, inactive processors to help keep systems available and processing their assigned workloads (3.4, “Dynamic CPU sparing and CPU Guard (5.2.0)” on page 133).

- ▶ e-business and network performance
 - Virtual IP address (VIPA)

Helps applications remain available if a network connection is lost (8.6, “Virtual IP address support” on page 495).
 - IP multipath routing

Improves network availability by providing multiple routes to a destination (8.3.1, “Multipath routing” on page 460).
 - Multiple default gateways

Keeps traffic moving through a network by detecting and routing around dead gateways (8.3.2, “Dead gateway detection” on page 466).
 - Mobile IPv6

Extends Internet connectivity to small, hand-held devices (8.7, “Mobile IPv6 (5.2.0)” on page 502).
 - Network tuning interface

Helps reduce administrative effort associated with managing and tuning networks (7.2.1, “The /etc/tunables commands” on page 425).
- ▶ Security
 - Kerberos Version 5 authentication

Helps administrators simplify password authentication for users connecting to several machines (9.2.3, “Native Kerberos Version 5 support” on page 575).
 - Pluggable Authentication Module (PAM)

Permits the use of distributed security services to reduce administrative effort associated with linking users to multiple applications (9.13, “Pluggable Authentication Module support” on page 614).
 - Enterprise identity mapping

Allows a user single-point access to a network comprised of heterogeneous server platforms (9.18, “Enterprise Identity Mapping (5.2.0)” on page 649).
- ▶ Java
 - Included in base AIX

Delivers a popular cross-platform programming language for e-business applications (2.11, “Java currency” on page 27).

- ▶ Systems and resource management
 - Fix Manager

Provides reports that compare fix levels on a system to a reference system or base level of fixes for easier administration (6.7, “Comparison reports for LPPs (5.2.0)” on page 368).
 - RSCT resource monitoring and control

Delivers clustering technology to automate resource monitoring, improving system availability and performance (3.7, “Resource Monitoring and Control” on page 145).
 - Dynamic Workload Manager

Adds time-based resource policies to allocate resources to applications within a whole system or in a partition (3.1.1, “Workload Manager enhancements history” on page 35).
- ▶ Storage
 - Split mirror support for Logical Volume Manager

Helps reduce any impact to system performance due to re-integrating the split mirror (4.2.13, “Snapshot support for mirrored VGs (5.2.0)” on page 213).
 - JFS2 file system snapshot

Helps administrators monitor and manage file system for action as needed (4.4.11, “JFS2 snapshot image (5.2.0)” on page 242).
 - I/O size and alignment for Logical Volume Manager

Removes size and alignment restrictions to help improve file system and overall system performance (4.2.15, “Unaligned I/O support in LVM (5.2.0)” on page 215).
 - Migration via Alternate Disk Install

Improves management of multiple operating system migrations in environments where downtime is critical (6.1.3, “Version 5.2 AIX migration (5.2.0)” on page 323).
- ▶ Reliability, Availability, Serviceability (RAS)
 - Automated system hang recovery

Helps systems remain available without administrator intervention (5.7, “System hang detection” on page 280).

- CPU-Gard

Proactively checks processor integrity and removes failing processors so that systems are more available (3.4.1, “Change CPU Guard default (5.2.0)” on page 134).
- System UE-Gard

Improves system uptime by proactively managing checkstop errors at a thread level (3.5, “UE-Gard (5.2.0)” on page 136).
- Multipath I/O

Enhances internal reliability of SCSI disk connections and permits maintenance deferral (4.1, “Multipath I/O (5.2.0)” on page 168).
- ▶ Debugging and performance tools
 - Xprofiler

Helps developers identify the most processor-intensive software functions via a graphical interface (7.1.20, “Xprofiler analysis tool (5.2.0)” on page 422).
 - Template-based performance tuning

Allows administrators the capability to capture system tuning schemes via stanza files and export them to multiple servers (7.2, “AIX tuning framework (5.2.0)” on page 424).

1.2 AIX 5L Version 5.1 enhancements

The following list is a quick description of the enhancements and differences available in this release. For further information, consult the references provided.

- ▶ AIX 5L kernel and application development differences

A summary of these differences can be found in 12.1, “AIX 5L 64-bit kernel overview” on page 766.
- ▶ Development environment and tool enhancements
 - An improved print function for DBX that provides more legible output is explained in 5.11, “DBX enhancements” on page 292.
 - Pthread enhancements, including application-level access to the pthread debug library, a new method to unregister atfork handlers, and a read/write locking enhancement are explained in 2.4, “pthread differences and enhancements” on page 15.
 - Core file enhancements that allow an application to core dump without termination are discussed in 5.13, “Lightweight core file support” on page 307.

- Enhancements to the KDB kernel debugger, including a new way to load it and additional subcommands, are discussed in 5.12, “KDB kernel and kdb command enhancements” on page 297.
- Enhancements that allow application level control over the scheduler during critical sections to prevent loss of context are explained in 2.6, “Context switch avoidance” on page 21.
- A new Korn shell, ksh93, is discussed in 2.9, “KornShell enhancements” on page 25.
- Enhancements in malloc that provide faster access to free memory for applications are discussed in 2.3, “Malloc enhancements” on page 14.
- An improved **restore** command helps you recover sparse database files, as explained in 5.16, “Non-sparseness support for the restore command” on page 312.
- The **pax** command includes support for large files, such as dumps greater than 2 GB, as discussed in 5.17, “The pax command enhancements” on page 313.
- AIX 5L introduces the IBM AIX Developer Kit, JAVA 2 Technology Edition Version 1.3.0, as discussed in 2.11, “Java currency” on page 27.
- ▶ Storage management enhancements
 - The /proc file system is discussed in 10.3, “The /proc file system” on page 684.
 - The JFS2 is introduced in 4.4, “The enhanced Journaled File System” on page 224. It provides the capability to store much larger files than JFS, in a more efficient manner.
 - NFS statd, AutoFS, and CacheFS enhancements are discussed in 4.7.1, “NFS statd multithreading” on page 253, 4.7.2, “Multithreaded AutoFS” on page 253, and 4.7.3, “Cache file system enhancements” on page 254.
 - A new passive mirror write consistency check can improve disk mirroring performance, as discussed in 4.2.9, “Passive mirror write consistency check” on page 209.
 - Updates to LVM libraries for multithreaded applications are discussed in 4.2.10, “Thread-safe liblvm.a” on page 211.
- ▶ System and resource management
 - An expanded set of devices that allow for simultaneous multiple device configuration during system startup is discussed in 5.8, “Fast device configuration enhancement” on page 284.

- New ways for you to dynamically manage your paging areas, such as deactivating a paging space with the **swapoff** command or decreasing its size, are discussed in 6.11, “Paging space enhancements” on page 378.
- Updates to the error log provide a more concise view of system errors, such as a link between the error log and diagnostics or the elimination of duplicate errors, and are described in 5.1, “Error log enhancements” on page 264.
- AIX 5L provides a set of resources to be monitored and actions to be taken at defined events providing automatic monitoring and recovery of select critical system resources. For more information, see 3.7, “Resource Monitoring and Control” on page 145.
- Shutdown logging is available, as described in 6.13, “shutdown enhancements” on page 384.
- New methods to diagnose system errors through dump improvements are described in 5.5, “System dump enhancements” on page 276.
- The ability to recover from certain system hangs is covered in 5.7, “System hang detection” on page 280.
- Enhancements to performance tools, including the **truss**, **iostat**, and **vmstat** commands, are discussed in 7.1, “Performance tools” on page 396.
- Workload Manager continues to receive improvements, as discussed in Chapter 3, “Resource management” on page 33.
- The new System V Release 4 print subsystem is discussed in 10.7, “System V Release 4 print subsystem” on page 697.
- Web-based System Manager receives major usability improvements with a much improved architecture and usability enhancements, such as accelerator keys. A discussion of all the enhancements can be found in 6.2, “Web-based System Manager” on page 329.
- Security and user authentication and LDAP enhancements are discussed in 9.2, “User and group integration” on page 571, 9.4, “IBM SecureWay Directory Version 3.2” on page 579, and 9.6, “LDAP name resolution enhancement” on page 583.
- A new documentation search engine to handle single- and double-byte searches together is discussed in 6.3, “Documentation search-engine enhancement” on page 354.
- AIX is Tivoli ready, as discussed in 9.16, “Tivoli readiness” on page 648.

► Networking enhancements

- The demand for QoS arises from applications such as digital audio/video or real-time applications and the need to manage bandwidth resources for arbitrary administratively-defined traffic classes. For more information, see 8.1, “Quality of Service support” on page 432.
- Together, multipath routing and dead gateway detection provide automatic selection of alternate network pathways that provide significant improvements in network availability. For more information, see 8.3, “TCP/IP routing subsystem enhancements” on page 460.
- With virtual IP address, the application is bound to a virtual IP address, not a real network interface that can fail. When a network or network interface failure is detected (using routing protocols or other schemes), a different network interface can be used by modifying the routing table without affecting application operation. For more information, see 8.6, “Virtual IP address support” on page 495.
- Dynamic Feedback Protocol (DFP) is a way to provide load statistics to a Load Manager so that load can be balanced by sending future connections to available servers. For more information, see 8.19, “Dynamic Feedback Protocol (5.1.0)” on page 545.
- Sendmail Version 8.11 improves performance by having multiple queues, memory-buffered pseudo-files, and more control over resolver time-outs. For more information, see 6.15, “Sendmail upgrade enhancements (5.1.0)” on page 386.
- TCP/IP performance over congested networks is improved through increased initial windows, explicit congestion notification, and limited transmit mechanism functions, which are configurable by a system administrator. For more information, see 8.3, “TCP/IP routing subsystem enhancements” on page 460.
- TCP splicing helps push the data-relaying function of a proxy application (from server-side socket to the client-side socket or vice versa) into the kernel. For more information, see 8.4.2, “TCP splicing (5.1.0)” on page 482.
- Network Interface Takeover is a new option allowing the configuration of multiple adapters, including IBM 10/100 Mbps Ethernet PCI adapter, Gigabit Ethernet-SX PCI adapter, and 10/100/1000 Base-T Ethernet PCI adapter, allowing one or more to be designated as a backup. For more information, see 8.22, “EtherChannel enhancements (5.1.0)” on page 556.
- Virtual LAN (VLAN) provides the ability to create virtual LANs across multiple physical LANs or segment and/or divide physical LAN segments into virtual LANs. For more information, see 8.24, “Virtual Local Area Network (5.1.0)” on page 563.

- Enhancements to the network buffer cache and HTTP GET kernel extension provide class-leading Web server performance. For more information, see 8.10, “Network buffer cache dynamic data support” on page 512, and 8.12, “HTTP GET kernel extension enhancements” on page 518.
- Applications can be modified to capture network data packets through a new interface, as explained in 8.13, “Packet capture library” on page 522.
- To allow more flexible development of firewall software, AIX provides additional hooks, as described in 8.14, “Firewall hooks enhancements” on page 523.
- PC Interoperability using Fast Connect file and print services provides support for Windows 2000, improved user and name mapping, share options, WTS support, better performance, and more, as discussed in 8.15, “Fast Connect enhancements” on page 525.
- ▶ Enhancements to increase affinity with Linux
 - A set of Linux-compatible routines has been added to AIX 5.1 so that Linux applications using these routines do not have to supply their own libraries. For more information, see 11.7, “AIX source affinity for Linux applications (5.1.0)” on page 762.
 - AIX Toolbox for Linux Applications is delivered on a supplemental CD that contains a collection of open source and GNU software built for AIX and packaged in RPM format. For more information, see 11.6, “AIX Toolbox for Linux Applications” on page 753.

Application development

AIX 5L provides several enhancements that assist you in developing your own software. Topics in this chapter include pthread libraries, memory access, shell environment, Java, Perl, OpenGL, and the Common Information Model. There is also information on how to avoid a context switch, and what happens to defunct processes.

2.1 Large data type support - binary compatibility

To support further application growth and scalability and the new 64-bit kernel, some data types, such as `time_t`, have been enlarged from 32 bit to 64 bit.

Therefore, 64-bit applications compiled under AIX Version 4.3 will not run under AIX 5L and have to be recompiled. The reverse is true as well; that means in a mixed environment of machines running AIX Version 4.3 and 5L, you must have two versions of your 64-bit applications available and a means to select the correct binary for each platform. 32-bit applications are not affected by this change.

2.2 Very large program support (5.2.0)

Very large program support allows 32-bit applications to grow their data heap beyond the eight segment limit (2 GB) of the large program support to thirteen segments (3.25 GB).

It allows a Dynamic Segment Allocation (DSA) program to grow dynamically as needed, rather than to be restricted to the pre-allocated (static) data heap provided with the implementation of large program support. It also changes the behavior of `shmat()`, `mmap()`, `rmmap_create()`, and `as_att()` for very large programs such that segment allocation begins at the top and works down rather than working from the bottom up.

2.2.1 The very large address space model

The very large address space model enables large data applications in much the same way as the large address space model. There are several differences between the two address space models though. To allow a program to use the very large address space model, you must set the `o_maxdata` field in the XCOFF header to indicate the amount of data needed and set the `F_DSA` flag in the file header.

The data in the very large address space model is laid out beginning in segment 3 when the `o_maxdata` value is greater than zero. The program is then allowed to use as many segments as needed to hold the amount of data indicated by the `o_maxdata` field, up to a maximum of 13 segments. In the very large address space model, these data segments for the data are created dynamically instead of all at exec time as in the large address space model.

Using the very large address space model changes the way in which the segments for a program are managed. A program's data is laid out starting in

segment 3. The data then consumes as many segments as needed for the initial data heap. The remaining segments are available to use for other purposes such as `shmat()` or `mmap()`. Once a segment has been allocated for the data heap, it can no longer be used for any other purposes, even if the size of the heap is reduced.

Use of the very large address space model also changes the default behavior of system calls such as `shmat()` and `mmap()`. The behavior of these system calls in the very large address space model changes, so that they start placing files in segment 15 and work down instead of starting in segment 3 and working up. The system calls can use any of the available segments as long as they have not been allocated for the data heap.

The very large address space model will allow programs to specify a `maxdata` value of `0xD0000000`, the largest allowable value, and still use all of the available segments above segment 3 until they are allocated for the data heap. In the large address space model these additional segments would have been allocated for the data heap at exec and thus unavailable for other purposes.

It is important to note here that applications can see different behaviors when switching between the large address space model and the very large address space model.

2.2.2 Enabling the very large address space model (5.2.0)

There are two ways to enable the very large program support behavior for an executable. One is to link the executable with the new `maxdata` option and the other is to have the keyword `DSA` in the value of the `LDR_CNTRL` environment variable at exec time.

Enabling with linker option

The very large address-space model is used if any non-zero value is given for the `maxdata` keyword and the `dsa` keyword is used also.

For example, to link a program with the very large address space model enabled and that will have the maximum 13 segments reserved to it, the following command line could be used:

```
cc sample.o -bmaxdata:0xD0000000/dsa
```

The number `0xD0000000` is the number of bytes, in hexadecimal format, equal to thirteen 256 MB segments respectively. The value following the `-bmaxdata` flag can also be specified in decimal or octal format.

Enabling with environment variable

The very large address space model is used if the keyword DSA is in the value of the LDR_CNTRL environment variable at exec time.

For example, to execute a program with the very large address space model enabled and that will have the maximum 13 segments reserved to it, the following command line could be used:

```
export LDR_CNTRL=MAXDATA=0xD0000000@DSA
```

The DSA keyword signals that the executable is to behave as a very large program if the value of its maxdata field is non-zero.

This applies to 32-bit processes only. The DSA keyword for the LDR_CNTRL environment variable and the extended maxdata option is ignored for 64-bit processes.

2.3 Malloc enhancements

The following sections discuss new ways for applications to access memory.

2.3.1 Malloc multiheap

The multiheap malloc was introduced in AIX Version 4.3.3 as part of the service stream and it may not be well known.

A single free memory pool (or heap) is provided, by default, by malloc. In AIX Version 4.3.3, the capability to enable the use of multiple heaps of free memory was introduced, which reduces thread contention for access to memory. This feature may be enabled by setting the MALLOCMULTIHEAP environment variable to the number of heaps. Setting MALLOCMULTIHEAP in this manner enables malloc multiheap to use any of 32 heaps and the fast heap selection algorithm. The applications that benefit the most by this setting are multithreaded applications on multiprocessor systems.

2.3.2 Malloc buckets

Malloc buckets was introduced in AIX Version 4.3.3 as part of the service stream.

Malloc buckets provides an optional buckets-based extension of the default allocator. It is intended to improve malloc performance for applications that issue large numbers of small allocation requests. When malloc buckets is enabled, allocation requests that fall within a predefined range of block sizes are

processed by malloc buckets. All other requests are processed in the usual manner by the default allocator.

Malloc buckets is not enabled by default. It is enabled and configured prior to process startup by setting the `MALLOCTYPE` and `MALLOCBUCKETS` environment variables.

The default configuration for malloc buckets should be sufficient to provide a performance improvement for many applications that issue large numbers of small allocation requests. However, it may be possible to achieve additional gains by setting the `MALLOCBUCKETS` environment variable to modify the default configuration. Developers who wish to modify the default configuration should first become familiar with the application's memory requirements and usage. Malloc buckets can then be enabled with the `bucket_statistics` option to fine tune the buckets configuration.

2.3.3 Malloc enhancement (5.2.0)

A new optional malloc subsystem capability, malloc trace, enables users to use the AIX **trace** command or the `trcstart()` subroutine to gather statistics on the malloc subsystem. Malloc trace can be enabled through the `MALLOCDEBUG` environment variable.

A new optional facility, malloc log, allows the user to obtain information about the malloc subsystem showing the number of active allocations for a given size and stack traceback of each `malloc()`, `realloc()`, and `free()` call. The malloc log can be enabled through the `MALLOCDEBUG` environment variable.

Malloc error reporting provides an optional error reporting and detection extension to the malloc subsystem. Error reporting can be enabled through the `MALLOCDEBUG` environment variable.

2.4 pthread differences and enhancements

The following sections discuss the major changes in the area of pthreads.

Note that any calls ending in `_np` signify that a library routine is non-portable and should not be used in code that will be ported to other UNIX-based systems.

2.4.1 Debug library

In AIX Version 4.3.3 and previous releases, `dbx` was the only debugger that could access information about pthread library objects. In AIX 5L, the pthread

debug library (libpthdebug.a) provides a set of functions that allows application developers to examine and modify pthread library objects.

This library can be used for both 32-bit and 64-bit applications and is thread safe. The pthread debug library provides applications with access to the pthread library information. This includes information on pthreads, pthread attributes, mutexes, mutex attributes, condition variables, condition variable attributes, read/write locks, read/write lock attributes, and information about the state of the pthread library.

2.4.2 Unregister atfork handler

The pthread API is enhanced to support unregistering atfork handlers. This is needed for times when the module in which an atfork handler resides is unloaded but the application continues and later calls fork.

A new pthread API function, pthread_atfork_unregister_np(), is provided to unregister handlers installed with either of the pthread_atfork() and pthread_atfork_np() calls.

2.4.3 atfork and cancellation cleanup handler support (5.1.0)

The pthread API library has been enhanced to support debugging for atfork handlers and cancellation cleanup handlers. The new enhancements allow debuggers to get information about all active atfork and cancellation cleanup handlers in a process.

The following new functions make the debugging enhancements available:

- ▶ pthdb_atfork()
- ▶ pthdb_atfork_arg()
- ▶ pthdb_atfork_child()
- ▶ pthdb_atfork_parent()
- ▶ pthdb_atfork_prepare()
- ▶ pthdb_atfork_type()
- ▶ pthdb_cleanup()
- ▶ pthdb_cleanup_arg()
- ▶ pthdb_cleanup_func()

The definitions of the new functions are similar to the following:

```
int pthdb_atfork(pthdb_session_t session, pthdb_atfork_t *atforkp, int cmd);
```

```

int pthread_atfork_arg(pthread_session_t session, pthread_atfork_t atfork,
pthread_addr_t *argp);

int pthread_atfork_child(pthread_session_t session, pthread_atfork_t atfork,
pthread_addr_t *funcp);

int pthread_atfork_parent(pthread_session_t session, pthread_atfork_t atfork,
pthread_addr_t *funcp);

int pthread_atfork_prepare(pthread_session_t session, pthread_atfork_t atfork,
pthread_addr_t *funcp);

int pthread_atfork_type(pthread_session_t session, pthread_atfork_t atfork,
pthread_atfork_type_t *typep);

int pthread_cleanup(pthread_session_t session, pthread_thread_t pthread,
pthread_cleanup_t *cleanup, int cmd);

int pthread_cleanup_func(pthread_session_t session, pthread_thread_t pthread,
pthread_cleanup_t cleanup, pthread_addr_t *funcp);

int pthread_cleanup_arg(pthread_session_t session, pthread_thread_t pthread,
pthread_cleanup_t cleanup, pthread_addr_t *argp);

```

2.4.4 Wait list and pthread state information enhancements (5.1.0)

This enhancement provides the ability of the pthread library to be debugged using the pthread debug library. Using the new enhancement increases the accuracy with which the pthread debug library can detect hangs and deadlocks in threaded applications.

When a pthread must wait on a pthread object (mutex, condition variable, read-write lock, and so forth), there are times when its wait/wakeup scheduling responsibilities are handled completely within the kernel as opposed to in the pthread library. In such cases, for performance reasons, the wait list associated with the object and the state of the pthread are not always updated to accurately reflect the pthread's true condition while it is waiting in the kernel. This feature ensures that wait list and state information is accurate for pthreads waiting on process private pthread objects.

2.4.5 Signal context support enhancements (5.1.0)

In AIX 5L Version 5.0, an extension of the pthread library function `pthread_getthrds_np()` was introduced to support signal handler contexts on the stack. In AIX 5L Version 5.1, the pthread library is enhanced with a new API to support a similar function.

Just like the pthread library feature, this feature enables debuggers to access the signal stacks and initial stack of a given pthread. It returns either the current context of the pthread or the pthread context at the time of a specific signal delivery. This function also supplies the number of frames in the requested stack.

The new feature consists of one new pthread debug library API routine. This routine requests the following input:

- ▶ pthread
- ▶ Request signal level

The output, based on your input, is as follows:

- ▶ Total number of signal levels on the pthreads stack
- ▶ Number of frames in the requested signal stack
- ▶ A context (only one of the following):
 - The context at the time of signal delivery (if a signal level is different from the current level that is requested and exists).
 - The current context (if signal level zero is requested or the pthread has no signal contexts).
- ▶ Return code indicating either success or failure

The new function in the pthread library has the following definition:

```
int
pthdb_pthread_sigcontext(pthdb_session_t session,
                        pthdb_pthread_t pthread,
                        int *siglevelp,
                        int *frame_countp,
                        pthdb_context_t *context);
```

2.4.6 Deadlock detection (5.1.0)

The pthread deadlock detection function has been added to the public interface of the pthread debug library. This enables the debugger, such as **dbx**, to present information to the user, which uniquely describes any deadlocks within the debugged process, or *debuggee*.

The deadlock detection provides value to the debugger user by streamlining debugging scenarios that call for computing when the debuggee is in a deadlock. Without this new pthread debug library-level of support for deadlock detection, the debugger visually presents the current state of lock objects and lets you manually compute dependency relationships between all lock objects.

The following are new lock objects types:

- ▶ `spinlock_t`
- ▶ `pthread_mutex_t`
- ▶ `rec_mutex`
- ▶ `pthread_cond_t`
- ▶ `pthread_rwlock_t`

New definitions that have been added to pthread debug library are as follows:

```
pthdb_hang_node(session_t, pthdb_hang_node_t *owner, int cmd);
phdb_hang_node_waiter(session_t, pthdb_hang_node_t, pthdb_pthread_t *);
phdb_hang_node_owner(session_t, pthdb_hang_node_t, pthdb_pthread_t *);
pthdb_hang_node_resource(session_t, pthdb_hang_node_t, pthdb_resource_t *);
pthdb_hang_resource_type(session_t, pthdb_resource_t, pthdb_resource_type_t *);
pthdb_hang_resource(session_t, pthdb_resource_t, pthdb_handle_t *);
pthdb_hang_cycle(session_t, pthdb_hang_cycle_t *, int cmd);
pthdb_hang_cycle_node(session_t, pthdb_hang_cycle_t, pthdb_hang_node_t *, int cmd);
```

2.4.7 Resource query support (5.1.0)

The pthread resource query support provides a pthread debug library interface to query a pthread for the resource it owns or the resource it is waiting on.

Four new API functions have been added to the pthread debug library:

- ▶ `pthdb_pthread_owner_resource()`
- ▶ `pthdb_pthread_waiter_resource()`
- ▶ `pthdb_resource_type()`
- ▶ `pthdb_resource_handle()`

Upon the first call to `pthdb_pthread_owner_resource()`, since the pthread debug library session has been updated, the mutex and rwlock debug lists will be traversed and all locked resources will be stored in a list associated with the pthread that owns the specific resource. The resource at the head of the list corresponding to the pthread in the request will be returned.

Subsequent calls to `pthdb_pthread_owner_resource()` will result in the remainder of owned resources being returned to the user. As long as the pthread debug session is not updated, the information will be retrieved from the lists created on the first call.

2.4.8 Multiple read/write lock read owners

The X/Open Standard (XPG 5) read/write locks allow a single write owner or multiple read owners of the lock. This improves critical section performance for data, which is read much more often than it is written. AIX 5L enables the pthread library to save multiple read owners for process-private read/write locks. By default, the pthread library will save multiple read owners.

These read/write locks are made available through the `pthread.h` header file using the `pthread_rwlock_t` data type and several `pthread_rwlock_*`() functions.

2.4.9 Thread level resource collection (5.1.0)

The Dynamic Probe Class Library (DPCL) tool is designed to collect a target application's performance data, including resource usage, hardware counter information, and so forth. Previously, the `getrusage()` system call was used, but this facilitates the entire process scope resource usage only, therefore it cannot be used to query the resource usage per thread. Because it is also necessary to monitor threaded applications, the DPCL tool will call the `pthread_getrusage_np()` library call. This pthread library call supports both 32-bit and 64-bit applications and 32-bit and 64-bit kernels. In the instance where old binaries make use of this pthread library call, it will be necessary to recompile the source code.

For additional information on DPCL, the following Web site is available.

<http://www.cs.wisc.edu/~paradyn/DPCL>

2.5 POSIX-compliant AIO (5.2.0)

With AIX 5L Version 5.2, two different asynchronous I/O (AIO) kernel extensions are available, the legacy AIO and the new POSIX-compliant AIO. The legacy AIO was created before the POSIX standard was fully developed so it differs in how parameters are passed and in some of the function definitions. The functions defined by both have the same names because of backward compatibility for the legacy AIO and for POSIX compliance for the new AIO. Although the two extensions have the same symbol names, redefinitions are done in `aio.h` so that both extensions can use the `libc.a` interface. POSIX AIO can also be accessed through the new real time library `librt.a`. The POSIX version will be the default version for compiling, so a new `_AIO_AIX_SOURCE` macro is available to use in compiling for the legacy version.

For example, to use the POSIX AIO extension load it is as follows:

```
mkdev -l posix_aio0
```

To compile the AIO application with the POSIX AIO function definition loaded, include the `aiio.h` file as follows:

```
#include sys/aiio.h
```

To compile using the new real time library, do the following:

```
cc ... -lrt posix_aiio_program.c
```

To use the legacy AIO extension load it is as follows:

```
mkdev -l aiio0
```

To compile the AIO application with the legacy AIO function definition loaded add the following definition to the source code:

```
#define _AIO_AIX_SOURCE  
#include sys/aiio.h
```

Or add the definition on the command line:

```
xlcc ... -D_AIO_AIX_SOURCE ... legacy_aiio_program.c
```

To have the POSIX AIO extension loaded *at boot time* enter **smit chgposixaio**, change the state from defined to available, and press Enter. For the legacy AIO, run **smit chgaio** and change the state as described previously.

2.6 Context switch avoidance

For application programs that are using their own thread control or locking code, it is helpful to signal to the dispatcher that the program is in a critical section and should not to be preempted or stopped.

AIX 5L now allows an application to specify the beginning and ending of a critical section. The prototypes for these functions are listed in `/usr/include/sys/thread_ctl.h`. After an initial call of `EnableCriticalSection()`, a call to `BeginCriticalSection()` increments a memory location in the process data structure. The memory location is decremented again by a call to `EndCriticalSection()`. This location is checked by the dispatcher, and if it is positive, the process receives another time slice (up to 10 ms). If the process sleeps, calls `yield()`, or is checked by the dispatcher a second time, this behavior is automatically disabled. If the process is preempted by a higher priority process, it is again queued in the priority queue, but at the beginning instead of the end of the queue.

If a thread is still in a critical section at the end of the extra time slice, it loses its scheduling benefit for one time slice. At the end of that time slice, it is eligible

again for another slice benefit. If a thread never leaves a critical section, it cannot be stopped by a debugger or Ctrl+Z from the parent shell.

This feature works on a per-thread basis. In multithreaded applications, each thread can declare critical sections and each thread doing so must call the `EnableCriticalSections()` function. If a process, even a multithreaded process, has one of its threads in a critical section, the process cannot be stopped.

2.7 Defunct process harvesting (5.2.0)

Version 5.2 introduces a new approach to handling child processes that are orphaned when their associated parent process exits. This enhancement improves the performance of this process, and provides better control of the way defunct processes are handled.

2.7.1 Zombie harvesting

Zombie harvesting in Version 5.2 is no longer handled exclusively by the init process if a child's parent process exits. The following sections describe how this was handled prior to Version 5.2 and also in the new release.

2.7.2 Zombie harvesting in versions prior to Version 5.2

A zombie process is created when a process exits. A zombie process is preserved by the kernel in order for the parent process to retrieve information about that process, for example, its exit code. If the parent ignores the signal generated by the process, this acts as a flag to the kernel that the zombie can be terminated and its resources can be reclaimed. In this case, the swapper harvests the zombie as it scans the process table, once every second. The reaper thread is awakened by the swapper as necessary to perform the cleanup.

If the parent does not either relinquish its interest in its child's exit value (by ignoring `SIGCHLD`) or retrieve that value using one of the `wait()` system calls, its child processes are reparented to the init process as the parent exits. The init process is then responsible for using the `use wait()` system call to clean up the orphaned child processes. Children that have already exited before the parent exits are already zombies, and init can clean them up immediately. Other children are cleaned up later, as they exit and become zombies.

2.7.3 Zombie harvesting in Version 5.2

Child processes that have already exited are harvested synchronously by the parent as part of its own exit. Any remaining active processes are still reparented

to init, but with a new flag so that they will not be visible to init. In particular, they will not generate a SIGCHLD to init when they exit. Instead they will be harvested by the swapper and reaper threads in the same way as a process that is being ignored by its parent, even though in this case its parent, init, is handling SIGCHLD. The init process is only responsible for handling its own child processes and restarting them as necessary. In rare cases, a child may still be reparented to init without being flagged. These child processes are handled by init with the same method employed prior to Version 5.2.

2.8 Software-vital product data (5.1.0)

The **vpdadd** and **vpdde1** commands in AIX 5L Version 5.1 are executables, whereas in earlier versions of AIX, they were shell scripts. The reason for this is to improve the performance of the commands and also because they are now APIs for the VPD. The **vpdadd** command is called to add entries to the product, lpp, history, and vendor databases of the ODM. **vpdadd** and **vpdde1** are only intended to be used to manipulate the SWVPD and not actually install or uninstall objects. The **vpdde1** command removes entries from the VPD and vendor databases.

The syntax of the **vpdadd** command is:

```
Usage: vpdadd -c component | -p product | -f feature -v v.r.m.f
        [-D destdir] [-U path_to_uninstaller] [-R prereq]
        [-S msg_set] [-M msg_number] [-C msg_catalog]
        [-I description] [-P parent] [-u]
```

The descriptions of the flags are provided in Table 2-1.

Table 2-1 The vpdadd command flags

Flags	Description
-c component	The component name to add to the VPD. This entry must be unique regarding the destination directory. If the entry already exists, no new entry will be added and no error will occur. This allows a force install.
-v v.r.m.f	Version, release, modification, and fix level.
-D destination directory	The prefix directory for the files being installed. The default is /usr/opt.
-I description	The description of the component being installed.
-R fileset name v.r.m.f	Requisite software. Must be specified in quotes. This flag can be used more than once.

Flags	Description
-U <i>uninstaller</i>	The command to launch the uninstaller for this component.
-C <i>message catalogue</i>	The message catalogue to search for a translated description of the component.
-S <i>message set</i>	The message set if more than one in the catalog.
-M <i>message number</i>	The message number for the description.
-p <i>product</i>	The product name to be added to the VPD. The entry is only added if it is unique insofar as v.r.m.f or destination directory. If it is not unique, no error occurs. This allows a force install.
-f <i>feature</i>	The feature name to add to the VPD. The entry is only added if it is unique insofar as v.r.m.f and destination directory. If it is not unique, no error occurs. This allows a force install.
-u	Specifies that the entry to be added is an update. If a base level fileset does not exist, then an error will occur.
-P <i>parent</i>	Specifies the parent software unit. For example, a component would specify either a feature or a product as its parent, depending on where it was in the tree. This flag is optional and is used to allow tree listings in Web-based System Manager.

The syntax of the **vpdde1** command is:

```
vpdde1 -c component | -p product | -f feature -v v.r.m.f -D destdir
```

The descriptions of the flags are provided in Table 2-2.

Table 2-2 The **vpdde1** command flags

Flags	Description
-c <i>component</i>	Removes the specified component.
-v <i>v.r.m.f</i>	The version, release, modification, and fix levels of the component to be deleted from the VPD or vendor database.
-f <i>feature</i>	The feature to be removed from the vendor database.
-p <i>product</i>	The product to be removed from the vendor database.

2.9 KornShell enhancements

In AIX 5L, the 1993 version of the **ksh** implementation of the KornShell command and scripting language is provided in addition to the 1988 version. In addition, the default value of the shell attribute for a user changed from `/bin/ksh` to `/usr/bin/ksh`.

2.9.1 ksh93

In AIX 5L, the default shell is still `/usr/bin/ksh`, which is hardlinked to `/usr/bin/psh`, `/usr/bin/sh`, and `/usr/bin/tsh`. This is an enhanced **ksh** implementation of the 1988 version of the KornShell, making it POSIX compliant. In addition to this shell, an unmodified version of the 1993 version of **ksh** is supplied as `/usr/bin/ksh93`. This version is also POSIX compliant.

With the exception of POSIX-specific items, the 93 version should be backward compatible with the 88 version. Therefore, no changes to shell scripts should be necessary. You should check your scripts for compatibility problems with this release.

This new version of **ksh** has the following functional enhancements:

- ▶ Key binding
- ▶ Associative arrays
- ▶ Complete ANSI-C `printf()` function
- ▶ Name reference variables
- ▶ New expansion operators
- ▶ Dynamic loading of built-in commands
- ▶ Active variables
- ▶ Compound variables

For a detailed description of the new features, consult the official KornShell Web site at:

<http://www.kornshell.com>

2.9.2 New value for shell attribute

The value of the shell attribute is changed to read `/usr/bin/ksh`. This is especially important for the root user. In previous versions of AIX, the value reads `/bin/ksh` and relies therefore on the existence of the link between `/bin` and `/usr/bin`. If this link is accidentally removed, the system becomes unbootable because there is no shell available for root and many of the system commands.

2.10 Perl 5.6 (5.1.0)

Perl 5.5.3 was shipped in AIX Version 4.3.3. In an effort to ship the latest code, Perl 5.6 is shipped in AIX 5L Version 5.1, as can be shown with the following command:

```
# perl -v
This is perl, v5.6.0 built for aix
Copyright 1987-2000, Larry Wall
```

The Perl environment is packaged and shipped in two filesets: perl.rte and perl.man.en_US.

Any changes made on the Perl source and how to compile it on AIX 5L Version 5.1 are documented in the `/usr/lpp/perl.rte/README.perl.aix` file.

2.10.1 Installing more than one Perl version

Perl is installed in `/usr/opt/perl5`, with the accompanying man pages in `/usr/share/man`. There is a link from `/usr/bin/perl` to the Perl executable `/usr/opt/perl5/bin/perl5.6.0`. The Perl libraries are in `/usr/opt/perl5/lib/5.6.0`, with a link to there from `/usr/lib/perl`. To support a different version of Perl (for example, Perl 5.5.3) on the same system, do not use the **installp** command, because the fileset name is not different and **installp** will only allow you to have one version of the same fileset installed. Instead of using **installp**, you can put the Perl executables and libraries on your system.

1. Mount the first AIX installation media and use the **restore** command to install another Perl version:

```
# mount -r -v cdrfs /dev/cd# /mnt
# cd / restore -xvf /mnt/usr/sys/inst.images/perl.rte 5.5.3.0
```

2. Make sure you remember to set up the links to point to whichever version of Perl you want to use.

Note: In the previous example, `/dev/cd#` is your CD drive (for example, `/dev/cd0`). You could also NFS mount the images if you do not have them available on CD.

2.10.2 Security considerations

Make sure that you do not have directories in the `LIBPATH` with write access to non-root users.

If the `/usr/opt/perl5/bin/perl` executable has its `LIBPATH` set to `/usr/local/lib:/usr/lib:/lib`, and if the `/usr/local/lib` directory exists on the system with

write access for non-root users, then a non-root user could put a Trojan horse copy of the `libc.a` or `libbsd.a` shared library into this directory. Then, if a system administrator were to run a system management command that uses Perl 5.6, the administrator would inadvertently execute the Trojan horse copy of the shared library. This would cause the Trojan horse code to execute with the system administrator's privileges.

2.11 Java currency

In AIX 5L, the default Java version installed is IBM AIX Developer Kit, Java2 Technology Edition, Version 1.3.0.

The default AIX Developer Kit is installed in `/usr/java130`. Please see the readme for instructions on how to set up the `PATH` environment variable prior to using the Developer Kit. When multiple versions of the Developer Kit are installed, setting the `PATH` selects the version of the Developer Kit that runs.

Java installed on AIX 5L is, by default, the 32-bit Java 1.3.0.

The Web site specifically for Java on AIX is:

<http://www.ibm.com/developerworks/java/jdk/aix/>

2.12 Common Information Model

Common Information Model (CIM) is a common data model by which systems, applications, networks, and devices are modeled in a common framework for use by managing applications. A CIM Object Manager (CIMOM) is developed to provide a mechanism for the exchange of information in order for systems management applications to leverage CIM technology.

2.12.1 CIM base support (5.1.0)

In AIX 5L Version 5.1, a CIM Object Manager (CIMOM) is available. The CIM Object Manager makes CIM objects available to Web-based Enterprise Management (WBEM) applications.

The CIMOM follows an open source standard. For more information on the CIMOM APIs, refer to:

<http://www.snia.org>

For more information about the Common Information Model, see:

<http://www.dmtf.org>

See Chapter 11, “Linux affinity” on page 733, for more information.

AIX 5L Version 5.1 does not provide any CIM objects; it just provides the CIM Object Manager service.

The CIM Schema

The CIM Schema provides the actual model descriptions. The CIM Schema supplies a set of classes with properties and associations that provide a well-understood conceptual framework within which it is possible to organize the available information about the managed environment.

Managed Object Format

The management information is described in a language based on the Interface Definition Language (IDL) called the Managed Object Format (MOF).

The following example illustrates MOF, the syntax of the CIM Schemas:

```
[Abstract, Description(
    "An abstraction or emulation of a hardware entity, that may "
    "or may not be Realized in physical hardware. ... ") ]
class CIM_LogicalDevice : CIM_LogicalElement
{
    ...
    [Key, MaxLen (64), Description (
        "An address or other identifying information to uniquely "
        "name the LogicalDevice.") ]
    string DeviceID;
    [Description (
        "Boolean indicating that the Device can power managed. ...") ]
    boolean PowerManagementSupported;
    [Description (
        "Requests that the LogicalDevice be enabled (\"Enabled\" "
        "input parameter = TRUE) or disabled (= FALSE). ...)"]
    unit32 EnableDevice ([IN] boolean Enabled);
    ...
};
```

2.12.2 Common Information Model (5.2.0)

AIX 5L Version 5.2 enables instrumentation using the Common Information Model (CIM). This is a common data model for describing the overall management data for network or an enterprise environment.

In Version 5.2, the open source CIMOM, called Pegasus, has been ported to AIX. Pegasus, written in C++, is highly portable and contains the client API and the provider API, along with a CIMOM engine.

Logical information flow

Figure 2-1 shows the information flow for the CIM model. This is discussed in more detail in the text that follows the diagram.

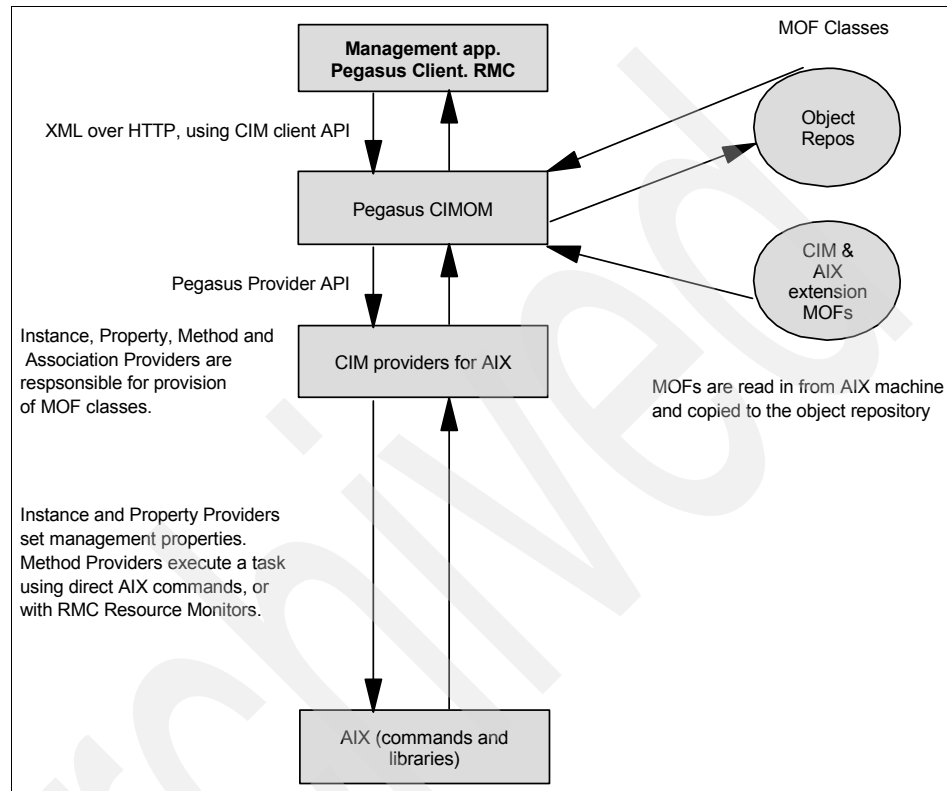


Figure 2-1 CIM logical flow diagram

The following are the main points regarding Figure 2-1.

- ▶ The CIM client API is used by the management application to request Pegasus to obtain an entire object or a set of an object.
- ▶ A request is made using XML over HTTP.
- ▶ CIMOM checks to see if the requested information is in the object repository. If it is there, CIMOM will give access to the management application.
- ▶ If the information is not in the repository, the MOF is used to determine the name of the provider for that managed object.
- ▶ Either the entire object will be obtained using the instance provider, or a specific dynamic property using the property provider.

- ▶ The providers then use AIX commands and libraries to obtain the information that they require and provide it to the CIMOM.
- ▶ CIMOM passes the information using XML over HTTP to the management application. If a task needs to be performed, CIMOM calls the appropriate method provider to call the required AIX commands and libraries. CIMOM again receives the results and passes it to the management application.

Installing CIM

Check the CSM software listing for the AIX release. The file is called `/opt/csm/install/csmfilelist_aixV52`. The software is included on the AIX CDs and the AIX toolbox. The following is also a useful link for RPM packages:

<http://www-1.ibm.com/servers/aix/products/aixos/linux/download.html>

There is a sample provider (AIX_OperatingSystem) included in this release that demonstrates how the Pegasus CIMOM works. The instructions on how to use this provider are contained in the readme files packaged with the RPM.

2.13 OpenGL 64-bit support in DWA mode (5.1.0)

OpenGL 3D graphics calls can be passed to the graphics adapter using the Direct Window Access (DWA) mode or the indirect mode. With DWA, OpenGL calls are passed directly to the graphics adapter device driver and are rendered. Indirect mode causes OpenGL calls to be passed to the GLX extension in the X Window server using a protocol, and rendering is performed by the GLX extension. The protocol-passing mechanism of indirect mode can result in much slower graphics performance than with DWA (DWA performance has been measured to be significantly faster than indirect for most operating scenarios).

Support for 64-bit indirect mode was first introduced in AIX Version 4.3.1. New 64-bit DWA support is introduced with AIX 5L Version 5.1.

The AIX 64-bit execution environment is important for certain data visualization applications that may require a larger memory address space, or increased precision for integer computations. It supports up to 2^{32} shared data segments. Note that 64-bit applications compiled for execution in the AIX Version 4.3 64-bit environment will need to be recompiled for execution in the AIX 5L Version 5.1 64-bit environment.

Applications that use 64-bit DWA may experience some performance differences compared to 32-bit DWA applications on POWER3-based systems. Degradations can be avoided by compiling the application into a shared library so that it resides in the same 4 GB region as the system's shared libraries.

The following graphics adapters will be 64-bit enabled:

- ▶ GTX6000P
- ▶ GTX4000P

OpenGL is packaged in device-dependent and device-independent filesets. The device-dependent software resides in separate filesets for 32-bit and 64-bit libraries. The device-independent software resides in a combined 32/64-bit library. Table 2-3 provides the adapters and their respective filesets that support DWA.

Table 2-3 Supported adapters and required filesets

Supported adapter	Required fileset
GTX4000P	OpenGL.OpenGL_X.dev.pci.14106e01.PPC64
GTX6000P	OpenGL.OpenGL_X.dev.pci.14107001.PPC64

Additional information about OpenGL support on AIX 5L Version 5.1 can be found in /usr/lpp/OpenGL/README.

OpenGL provides two new packages in order to fully support the 64-bit in DWA mode, as shown in Table 2-4.

Table 2-4 New packaging information

Package name	New fileset
OpenGL.OpenGL_X.dev	OpenGL.OpenGL_X.dev.pci.14106e01.PPC64 OpenGL.OpenGL_X.dev.pci.14107001.PPC64
OpenGL.OpenGL_X.rte	OpenGL.OpenGL_X.rte.pipe64++

Resource management

In this chapter the following topics are discussed:

- ▶ Workload Manager
- ▶ Logical partitioning
- ▶ Capacity Upgrade on Demand
- ▶ Dynamic CPU sparing
- ▶ CPU Guard and UE-Gard
- ▶ Resource Monitoring and Control
- ▶ Memory and system affinity services
- ▶ Cluster management software

3.1 Workload Manager (WLM)

WLM is designed to give the system administrator greater control over how the scheduler and Virtual Memory Manager (VMM) allocate CPU, physical memory, and I/O resources to processes. It can be used to prevent different jobs from interfering with each other and to allocate resources based on the requirements of different groups of users.

The major use of WLM is for large SMP systems, and it is typically used for server consolidation, where workloads from many different server systems, (print, database, general user, transaction processing systems, and so on) are combined. These workloads often compete for resources and have differing goals and service level agreements. At the same time, WLM can be used in uniprocessor workstations to improve responsiveness of interactive work by reserving physical memory. WLM can also be used to manage individual SP nodes.

WLM provides isolation between user communities with very different system behaviors. This can prevent effective starvation of workloads with certain characteristics, such as interactive or low CPU usage jobs, by workloads with other characteristics, such as batch or high CPU usage.

WLM offers the system administrator the ability to create different classes of service and specify attributes for those classes. The system administrator has the ability to classify jobs automatically into classes, based upon the user, group, or path name of the application.

WLM configuration is performed through the preferred interface, the Web-based System Manager (Figure 3-1 on page 35), through a text editor and AIX commands, or through the AIX administration tool SMIT.

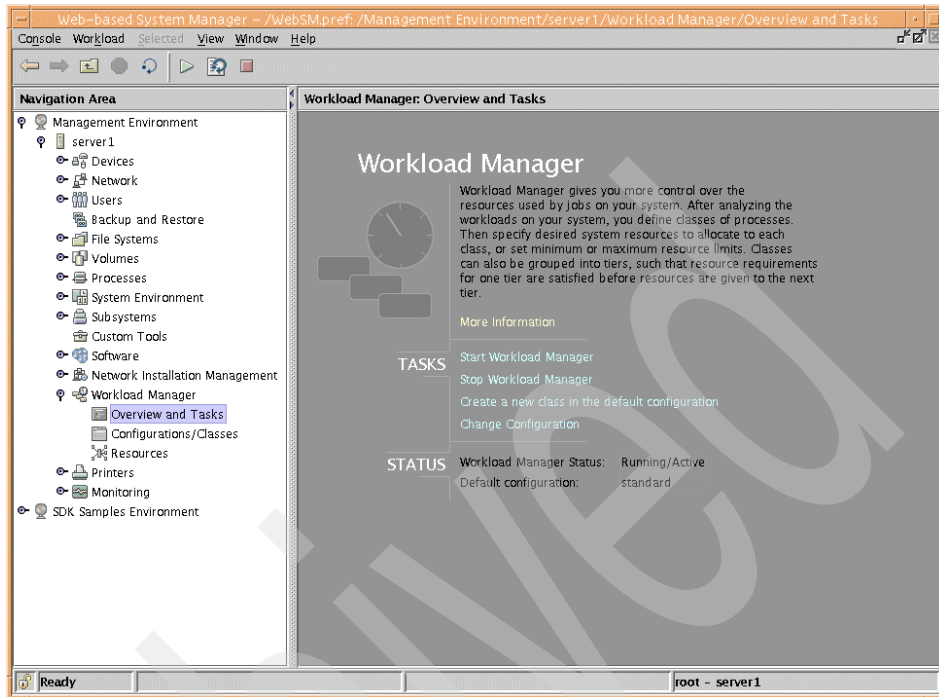


Figure 3-1 Web-based System Manager Overview and Tasks dialog

3.1.1 Workload Manager enhancements history

Since it was first released in AIX Version 4.3.3, Workload Manager (WLM) has gained new features and architectural improvements.

AIX Version 4.3.3

In AIX Version 4.3.3, WLM was able to allocate CPU and physical memory resources to classes of jobs and allowed processes to be assigned to classes based on user, group, or application (Figure 3-2 on page 36).

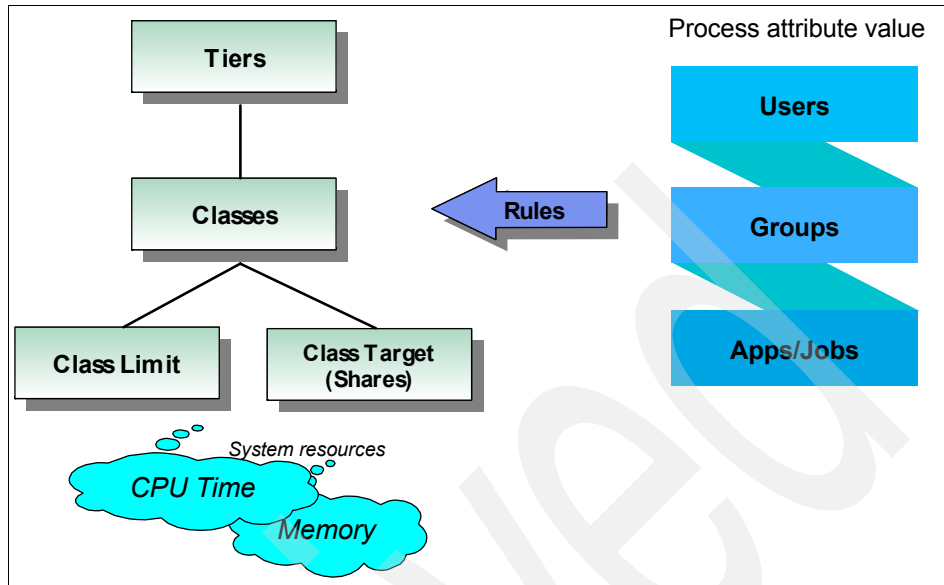


Figure 3-2 Basic Workload Manager elements in AIX Version 4.3

AIX Version 4.3.3 with Maintenance Level 2

With AIX Maintenance Level 2 (APAR IY06844), additional features were added to the first release of WLM, which were:

- ▶ Classification of existing processes to avoid stopping and starting applications when stopping and starting WLM.
- ▶ Passive mode to allow *before* and *after* WLM comparisons.
- ▶ Management of application file names, which allowed WLM to start even if some applications listed in the rules file could not be accessed.

AIX 5L

This section focuses on WLM functions that are available in AIX 5L, starting by outlining the enhancements it presents over its earlier release. The enhancements include:

- ▶ Management of disk I/O bandwidth, in addition to the already existing CPU cycles and real memory.
- ▶ Graphic display of resource utilization.
- ▶ Performance Toolbox integration with WLM classes, enabling the toolbox to display performance statistics.

- ▶ Fully dynamic configuration, including setting up new classes without restarting WLM.
- ▶ Application Programming Interface (API) to enable external applications to modify the system's behavior.
- ▶ Manual reclassification of processes, which provides the ability to have multiple instances of the same application in different classes.
- ▶ More application isolation and control:
 - New *Subclasses* add ten times the granularity of control (from 27 to 270 controllable classes).
 - Administrators can delegate Subclass management to other users and groups rather than root or system.
 - Possibility of inheritance of classification from parent to child processes.
- ▶ Application path name wildcard flexibility extended to user name and group name.
- ▶ Tier separation enforced for all resources, enabling a deeper prioritization of applications.

Note: For more information on previous Workload Manager architecture and features, refer to the following publications:

- ▶ *AIX Version 4.3 Differences Guide*, SG24-2014
- ▶ *AIX 5L Workload Manager (WLM)*, SG24-5977

3.1.2 Concepts and architectural enhancements

The following sections outline the concepts provided with WLM on AIX 5L.

Classes

The central concept of WLM is the class. A class is a collection of processes (jobs) that has a single set of resource limits applied to it. WLM assigns processes to the various classes and controls the allocation of system resources among the different classes. For this purpose, WLM uses class assignment rules and per-class resource shares and limits set by the system administrator. The resource entitlements and limits are enforced at the class level. This is a way of defining classes of service and regulates the resource utilization of each class of applications to prevent applications with very different resource utilization patterns from interfering with each other when they are sharing a single server.

Hierarchy of classes

WLM allows system administrators to set up a hierarchy of classes with two levels by defining Superclasses and Subclasses. In other words, a class can either be a *Superclass* or a *Subclass*. The main difference between Superclasses and Subclasses is the resource control (shares and limits):

- ▶ At the Superclass level, the determination of resource entitlement (based on the resource shares and limits) is based on the total amount of each resource managed by WLM available on the machine.
- ▶ At the Subclass level, the resource shares and limits are based on the amount of each resource allocated to the parent Superclass.

The system administrator (the root user) can delegate the administration of the Subclasses of each Superclass to a *Superclass administrator* (a non-root user), thus allocating a portion of the system resources to each Superclass and then letting Superclass administrators distribute the allocated resources among the users and applications they manage.

WLM supports 32 Superclasses (27 user defined plus five predefined). In turn, each Superclass can have 12 Subclasses (10 user defined and two predefined, as shown in Figure 3-3 on page 39). Depending on the needs of the organization, a system administrator can decide to use only Superclasses or both Superclasses and Subclasses. An administrator can also use Subclasses only for some of the Superclasses.

Each class is given a name by the WLM administrator who creates it. A class name can be up to 16 characters long and can only contain uppercase and lowercase letters, numbers, and underscores (_). For a given WLM configuration, the names of all the Superclasses must be different from one another, and the names of the Subclasses of a given Superclass must be different from one another. Subclasses of different Superclasses can have the same name. The fully qualified name of a Subclass is (*superclass_name.subclass_name*).

In the remainder of this section, whenever the term *class* is used, it is applicable to both Subclasses and Superclasses. The following subsections describe both super- and Subclasses in greater detail, as well as the backward compatibility WLM provides to configurations of its first release.

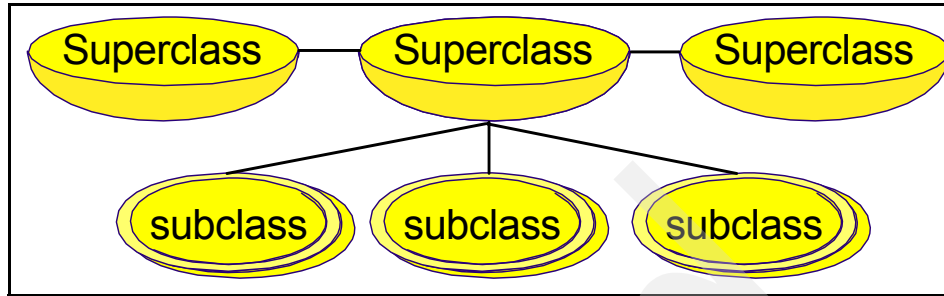


Figure 3-3 Hierarchy of classes

Superclasses

A Superclass is a class with Subclasses associated with it. No process can belong to the Superclass without also belonging to a Subclass, either predefined or user defined. A Superclass has a set of class assignment rules that determine which processes will be assigned to it. A Superclass also has a set of resource limitation values and resource target shares that determine the amount of resources that can be used by processes belonging to it. These resources will be divided among the Subclasses based on the resource limitation values and resource target shares of the Subclasses.

Up to 27 Superclasses can be defined by the system administrator. In addition, five Superclasses are automatically created to deal with processes, memory, and CPU allocation, as follows:

- ▶ *Default* Superclass: The default Superclass is named *Default* and is always defined. All non-root processes that are not automatically assigned to a specific Superclass will be assigned to the Default Superclass. Other processes can also be assigned to the Default Superclass by providing specific assignment rules.
- ▶ *System* Superclass: This Superclass has all privileged (root) processes assigned to it if they are not assigned by rules to a specific class, plus the pages belonging to all system memory segments, kernel processes, and kernel threads. Other processes can also be assigned to the System Superclass. This default is for this Superclass to have a memory minimum limit of one percent.
- ▶ *Shared* Superclass: This Superclass receives all the memory pages that are shared by processes in more than one Superclass. This includes pages in shared memory regions and pages in files that are used by processes in more than one Superclass (or in Subclasses of different Superclasses). Shared memory and files used by multiple processes that belong to a single Superclass (or Subclasses of the same Superclass) are associated with that Superclass. The pages are placed in the Shared Superclass only when a

process from a different Superclass accesses the shared memory region or file. This Superclass can have only physical memory shares and limits applied to it. It cannot have shares or limits for the other resource types, Subclasses, or assignment rules specified. Whether a memory segment shared by the processes in the different Superclasses is classified into the Shared Superclass, or remains in the Superclass it was initially classified into, depends on the value of the localshm attribute of the Superclass the segment was initially classified into.

- *Unclassified* Superclass: The processes in existence at the time WLM is started are classified according to the assignment rules of the WLM configuration being loaded. During this initial classification, all the memory pages attached to each process are charged either to the Superclass the process belongs to (when not shared, or shared by processes in the same Superclass) or to the Shared Superclass, when shared by processes in different Superclasses. However, there are a few pages that cannot be directly tied to any processes (and thus to any class) at the time of this classification, and this memory is charged to the *Unclassified* Superclass; for example, pages from a file that has been closed. The file pages will remain in memory, but no process *owns* these pages; therefore, they cannot be charged to a specific class. Most of this memory will end up being correctly reclassified over time, when it is either accessed by a process, or freed and reallocated to a process after WLM is started. There are a few kernel processes, such as wait or lrud, in the Unclassified Superclass. Even though this Superclass can have physical memory shares and limits applied to it, WLM commands do not allow you to set shares and limits or specify Subclasses or assignment rules on this Superclass.
- *Unmanaged* Superclass: A special Superclass named *Unmanaged* will always be defined. No processes will be assigned to this class. This class will be used to accumulate the memory usage for all pinned pages in the system that are not managed by WLM. The CPU utilization for the waitprocs is not accumulated in any class. This is deliberate; otherwise, the system would always seem to be at 100 percent CPU utilization, which could be misleading for users when looking at the WLM or system statistics. This Superclass cannot have shares or limits for any other resource types, Subclasses, or assignment rules specified.

Subclasses

A Subclass is a class associated with exactly one Superclass. Every process in the Subclass is also a member of the Superclass. Subclasses only have access to resources that are available to the Superclass. A Subclass has a set of class assignment rules that determine which of the processes assigned to the Superclass will belong to it. A Subclass also has a set of resource limitation values and resource target shares that determine the resources that can be used by processes in the Subclass. These resource limitation values and resource

target shares indicate how much of the Superclass's target (the resources available to the Superclass) can be used by processes in the Subclass.

Up to 10 out of a total of 12 Subclasses can be defined by the system administrator or by the Superclass administrator for each Superclass. In addition, two special Subclasses, Default and Shared, are always defined in each Superclass as follows:

- ▶ *Default* Subclass: The default Subclass is named Default and is always defined. All processes that are not automatically assigned to a specific Subclass of the Superclass will be assigned to the Default Subclass. You can also assign other processes to the Default Subclass by providing specific assignment rules.
- ▶ *Shared* Subclass: This Subclass receives all the memory pages used by processes in more than one Subclass of the Superclass. This includes pages in shared memory regions and pages in files that are used by processes in more than one Subclass of the same Superclass. Shared memory and files used by multiple processes that belong to a single Subclass are associated with that Subclass. The pages are placed in the Shared Subclass of the Superclass only when a process from a different Subclass of the same Superclass accesses the shared memory region or file. There are no processes in the Shared Subclass. This Subclass can only have physical memory shares and limits applied to it. It cannot have shares or limits for the other resource types or assignment rules specified.

Tiers

Tier configuration is based on the importance of a class relative to other classes in WLM. There are 10 available tiers from 0 to 9. Tier value 0 is the most important and value 9 is the least important. As a result, classes belonging to tier 0 will get resource allocation priority over classes in tier 1, classes in tier 1 will have priority over classes in tier 2, and so on. The default tier number, if the attribute is not specified, is 0.

The tier applies at both the Superclass and Subclass levels. Superclass tiers are used to specify resource allocation priority between Superclasses, and Subclass tiers are used to specify resource allocation priority between Subclasses of the same Superclass. There is no relationship between tier numbers of Subclasses of different Superclasses.

Tier separation, in terms of prioritization, is much more enforced in AIX 5L than in the previous release. A process in tier 1 will never have priority over a process in tier 0, since there is no overlapping of priorities in tiers. It is unlikely that classes in tier 1 will acquire any resources if the processes in tier 0 are consuming all the resources. This occurs because the control of leftover resources is much more

restricted than in the AIX Version 4.3.3 release of WLM, as shown in Figure 3-4 on page 42.

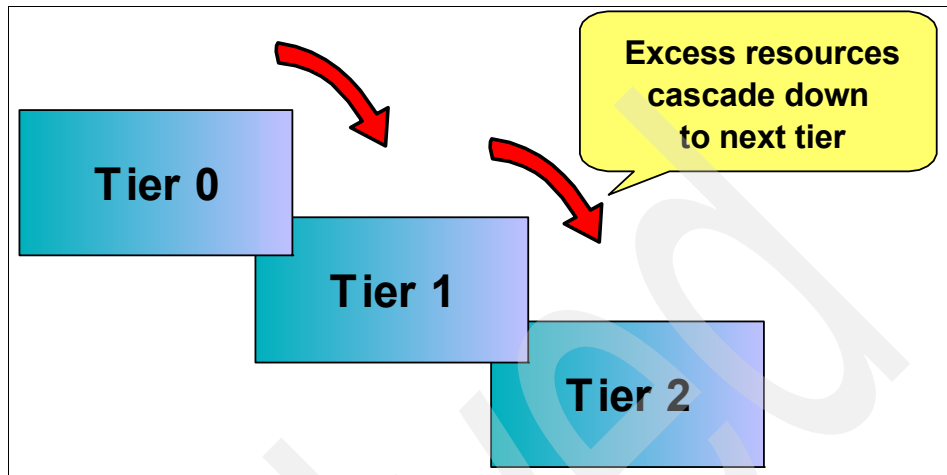


Figure 3-4 Resources cascading through tiers

Class attributes

In order to create a class, there are different attributes that are needed to have an accurate and well-organized group of classes. Figure 3-5 shows the SMIT panel for Class attributes.

General characteristics of a class

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

	[Entry Fields]	
* Class name	[]	
Description	[]	
Tier	[0]	+#
Resource Set		+
Inheritance	[No]	+
User authorized to assign its processes to this cl	[]	+
ass		
Group authorized to assign its processes to this c	[]	+
lass		
User authorized to administrate this class	[]	+
(Superclass only)		
Group authorized to administrate this class	[]	+
(Superclass only)		

F1=Help
F5=Reset
F9=Shell

F2=Refresh
F6=Command
F10=Exit

F3=Cancel
F7=Edit
Enter=Do

F4=List
F8=Image

Figure 3-5 SMIT with the class creation attributes screen

The sequence of attributes within a class (as shown in Figure 3-5 on page 43) is outlined below:

- ▶ **Class name**
A unique class name with up to 16 characters. It can contain uppercase and lowercase letters, numbers, and underscores (_).
- ▶ **Description**
An optional brief description about this class.
- ▶ **Tier**
A number between 0 and 9, for class priority ranking. It will be the tier that this class will belong to. An explanation about tiers can be found in “Tiers” on page 41.
- ▶ **Resource Set**
This attribute is used to limit the set of resources a given class has access to in terms of CPUs (processor set). The default, if unspecified, is *system*, which gives access to all the CPU resources available on the system.
- ▶ **Inheritance**
The inheritance attribute indicates whether a child process should inherit its parent’s class or get classified according to the automatic assignment rules

up on exec. The possible values are yes or no; the default is no. This attribute can be specified at both Superclass and Subclass level.

- User and Group authorized to assign its processes to this class

These attributes are valid for all the classes. They are used to specify the user name and the group name of the user or group authorized to manually assign processes to the class. When manually assigning a process (or a group of processes) to a Superclass, the assignment rules for the Superclass are used to determine which Subclass of the Superclass each process will be assigned to.

- User and Group authorized to administer this class

These attributes are valid only for Superclasses. They are used to delegate the Superclass administration to a user and group of users.

- Localshm

Specifies whether memory segments that are accessed by processes in different classes remain local to the class they were initially assigned to, or if they go to the Shared class.

Segment authorization to migrate to the Shared class (5.1.0)

With Workload Manager in earlier versions of AIX, whenever a memory segment is accessed by processes from different classes, the segment is reclassified as Shared. This occurs because one of the classes sharing the memory segment would otherwise be penalized as the user of this resource while the others are not. The consequence of the segment moving to Shared is that users partially lose control of it. In AIX 5L Version 5.1, an attribute has been added at the class level to avert the automatic reclassification of the class. This attribute, localshm, if set to no, allows the segment to be reclassified to the Shared class. If it is set to yes, then it is not reclassified. From the command line, the command will be similar to that shown in the example below:

```
# mkclass -a tier=2 -a adminuser=wlmua6 -a localshm=yes -c shares=2\  
-m shares=3 -d new_config super3
```

From the SMIT panels, general characteristics of a class panel will have the localshm option, as in the example shown in Figure 3-6.

General characteristics of a class

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

	[Entry Fields]	
Class name	super3	
Description	[]	
Tier	[2]	+#
Resource Set		+
Inheritance	[No]	+
User authorized to assign its processes to this cl	[]	+
ass		
Group authorized to assign its processes to this c	[]	+
lass		
User authorized to administrate this class	[wlm6]	+
(Superclass only)		
Group authorized to administrate this class	[]	+
(Superclass only)		
localshm	[Yes]	+

F1=Help
F5=Reset
F9=Shell

F2=Refresh
F6=Command
F10=Exit

F3=Cancel
F7=Edit
Enter=Do

F4=List
F8=Image

Figure 3-6 SMIT panel shows the additional localshm attribute

Classification process

There are two ways to classify processes in WLM:

- ▶ Automatic assignment when a process calls the system call exec, using assignment rules specified by a WLM administrator. This automatic assignment is always in effect (cannot be turned off) when WLM is active. This is the most common method of assigning processes to the different classes.
- ▶ Manual assignment of a selected process or group of processes to a class by a user with the required authority on both the process and the target class. This manual assignment can be done either by a WLM command, which could be invoked directly or through SMIT or Web-based System Manager, or by an application, using a function of the WLM Application Programming Interface. Manual assignment overrides automatic assignment.

3.1.3 Automatic assignment

The automatic assignment of processes to classes uses a set of class assignment rules specified by a WLM administrator. There are two levels of assignment rules:

- ▶ A set of assignment rules at the WLM configuration level used to determine which Superclass a given process should be assigned to

- ▶ A set of assignment rules at the Superclass level used to determine which Subclass of the Superclass the process should be assigned to

The assignment rules at both levels have exactly the same format.

When a process is created by fork, it remains in the same class as its parent. Usually, reclassification happens when the new process calls the system call `exec`. In order to classify the process, WLM starts by examining the top level rules list for the active configuration to find out which Superclass the process should belong to. For this purpose, WLM takes the rules one at a time, in the order they appear in the file, and checks the current values for the process attributes against the values and lists of values specified in the rule. When a match is found, the process will be assigned to the Superclass named in the first field of the rule. Then the rules list for the Superclass is examined in the same way to determine which Subclass of the Superclass the process should be assigned to. For a process to match one of the rules, each of its attributes must match the corresponding field in the rule. The rules to determine whether the value of a process attribute matches the values in the field of the rules list are as follows:

- ▶ If the field in the rule has a value of hyphen (-), any value of the corresponding process attribute is a match.
- ▶ If the value of the process attribute (for all the attributes except *type*) matches one of the values in the list in a rule, and it is not excluded (prefaced by an exclamation point (!)), it is considered a match.
- ▶ When one of the values for the *type* attribute in the rule is comprised of two or more values separated by a plus sign (+), a process will be a match for this value only if its characteristics match all the values mentioned above.

As previously mentioned, at both Superclass and Subclass levels, WLM goes through the rules in the order in which they appear in the rules list, and classifies the process in the class corresponding to the first rule for which the process is a match. This means that the order of the rules in the rules list is extremely important, and caution must be applied when modifying it in any way.

3.1.4 Manual assignment

Manual assignment is a feature introduced in AIX 5L WLM. It allows system administrators and applications to override, at any time, the traditional WLM automatic assignment (processes' automatic classification based on class assignment rules) and force a process to be classified in a specific class.

The manual assignment can be made or canceled separately at the Superclass level, the Subclass level, or both. In order to manually assign processes to a class or cancel an existing manual assignment, a user must have the right level

of privilege (that is, must be the root user, adminuser, or admingroup for the Superclass or authuser and authgroup for the Superclass or Subclass). A process can be manually assigned to a Superclass only, a Subclass only, or to a Superclass and a Subclass of the Superclass. In the latter case, the dual assignment can be done simultaneously (with a single command or API call) or at different times, possibly by different users.

A manual assignment will remain in effect (and a process will remain in its manually assigned class) until:

- ▶ The process terminates.
- ▶ WLM is stopped. When WLM is restarted, the manual assignments in effect when WLM was stopped are lost.
- ▶ The class the process has been assigned to is deleted.
- ▶ A new manual assignment overrides a prior one.
- ▶ The manual assignment for the process is canceled.

In order to assign a process to a class or cancel a prior manual assignment, the user must have authority both on the process and on the target class. These constraints translate into the following:

- ▶ The root user can assign any process to any class.
- ▶ A user with administration privileges on the Subclasses of a given Superclass (that is, the user or group name matches the attributes adminuser or admingroup of the Superclass) can manually reassign any process from one of the Subclasses of this Superclass to another Subclass of the Superclass.
- ▶ A user can manually assign their own processes (same real or effective user ID) to a Superclass or a Subclass for which he has manual assignment privileges (that is, the user or group name matches the attributes authuser or authgroup of the Superclass or Subclass).

This defines three levels of privilege among the persons who can manually assign processes to classes, root being the highest. In order for a user to modify or cancel a manual assignment, the user must be at the same or a higher level of privilege as the person who issued the last manual assignment.

Class assignment rules

After the definition of a class, it is time to set up the class assignment rules so that WLM can perform its automatic assignment. The assignment rules are used by WLM to assign a process to a class based on the user, group, application path name, type of process, and application tag, or a combination of these five attributes.

The next sections describe the attributes that constitute a class assignment rule. All these attributes can contain a hyphen, which means that this field will not be considered when assigning classes to a process.

Class name

This field must contain the name of a class which is defined in the class file corresponding to the level of the rules file we are configuring (either Superclass or Subclass). Class names can contain only uppercase and lowercase letters, numbers, and underscores (`_`), and can be up to 16 characters in length. No assignment rule can be specified for the system defined classes *Unclassified*, *Unmanaged*, and *Shared*.

Reserved

Reserved for future use. Its value *must* be a hyphen, and it must be present in the rule.

User

The user name (as specified in the `/etc/passwd` file, LDAP, or in NIS) of the user owning a process can be used to determine the class to which the process belongs. This attribute is a list of one or more user names, separated by a comma. Users can be excluded by using an exclamation point prefix. Patterns can be specified to match a set of user names using full Korn shell pattern matching syntax.

Applications that use the **setuid** permission to change the *effective* user ID they run under are still classified according to the user that invoked them. The processes are only reclassified if the change is done to the *real* user ID (UID).

Group

The group name (as specified in the `/etc/group` file, LDAP, or in NIS) of a process can be used to determine the class to which the process belongs. This attribute is a list composed of one or more groups, separated by a comma. Groups can be excluded by using an exclamation point prefix. Patterns can be specified to match a set of group names using full Korn shell pattern matching syntax.

Applications that use the **setgid** permission to change the *effective* group ID they run under are still classified according to the group that invoked them. The processes are only reclassified if the change is done to the *real* group ID (GID).

Application path names

The full path name of the application for a process can be used to determine the class to which a process belongs. This attribute is a list composed of one or more applications, separated by a comma. The application path names will be either full path names or Korn shell patterns that match path names. Application path names can be excluded by using an exclamation point prefix.

Process type

In AIX 5L, the process type attribute is introduced as one of the ways to determine the class to which a process belongs. This attribute consists of a comma-separated list, with one or more combination of values, separated by a plus sign (+). A plus sign provides a logical *and* function, and a comma provides a logical *or* function. Table 3-1 provides a list of process types that can be used. (Note: *32bit* and *64bit* are mutually exclusive.)

Table 3-1 List of process types

Attribute value	Process type
32bit	The process is a 32-bit process.
64bit	The process is a 64-bit process.
plock	The process called plock() to pin memory.
fixed	The process has a fixed priority (SCHED_FIFO or SCHED_RR).

Application tags

In AIX 5L, the application tag attribute is introduced as one of the forms of determining the class to which a process belongs. This is an attribute meant to be set by WLM's API as a way to further extend the process classification possibilities. This process was created to allow differentiated classification for different instances of the same application. This attribute can have one or more application tags, separated by commas. An application tag is a string of up to 30 alphanumeric characters.

The classification is done by comparing the value of the attributes of the process at exec time against the lists of class assignment rules to determine which rule is a match for the current value of the process attributes. The class assignment is done by WLM:

- ▶ When WLM is started for all the processes existing at that time
- ▶ Every time a process calls the system calls exec, setuid (and related calls), setgid (and related calls), setpri, and plock, once WLM is started

There are two *default* rules that are always defined (that is, hardwired in WLM). These are the default rules that assign all processes started by the user root to the System class, and all other processes to the Default class. If WLM does not find a match in the assignment rules list for a process, these two rules will be applied (the rule for System first), and the process will go to either System (UID root) or Default. These default rules are the only assignment rules in the standard configuration installed with AIX.

Table 3-2 is an example of classes with their respective attributes for assignment rules.

Table 3-2 Examples of class assignment rules

Class	Reserved	User	Group	Application	Type	Tag
System	-	root	-	-	-	-
db1	-	-	-	/usr/oracle/bin/db*	-	_db1
db2	-	-	-	/usr/oracle/bin/db*	-	_db2
devlt	-	-	dev	-	32bit	-
VPs	-	bob,!ted	-	-	-	-
acctg	-	-	acct*	-	-	-

In Table 3-2, the rule for Default class is omitted from display, though this class's rule is always present in the configuration. The rule for System is explicit, and has been put first in the file. This is deliberate so that all processes started by root will be assigned to the System Superclass. By moving the rule for the System Superclass further down in the rules file, the system administrator could have chosen to assign the root processes that would not be assigned to another class (because of the application executed, for example) to System only. In Table 3-2, with the rule for System on top, if root executes a program in /usr/oracle/bin/db* set, the process will be classified as System. If the rule for the System class was after the rule for the db2 class, the same process would be classified as db1 or db2, depending on the tag.

These examples show that the order of the rules in the assignment rules file is very important. The more specific assignment rules should appear first in the rules file, and the more general rules should appear last. An extreme example would be putting the default assignment rule for the Default class, for which every process is a match, first in the rules file. That would cause every process to be assigned to the Default class (the other rules would, in effect, be ignored).

You can define multiple assignment rules for any given class. You can also define your own specific assignment rules for the System or Default classes. The default rules mentioned previously for these classes would still be applied to processes that would not be classified using any of the explicit rules.

Backward compatibility issues

As mentioned earlier, in the first release of WLM, the system default for the resource shares was one share. In AIX 5L, it is -, which means that the resource consumption of the class for this particular resource is not regulated by WLM. This changes the semantics quite a bit, and it is advisable that system

administrators review their existing configurations and consider if the new default is good for their classes, or if they would be better off either setting up a default of one share (going back to the previous behavior) or setting explicit values for some of the classes.

In terms of limits, the first release of WLM only had one maximum, not two. This maximum limit was in fact a *soft* limit for CPU and a *hard* limit for memory. Limits specified for the old format, *min percent-max percent*, will have, in AIX 5L, the max interpreted as a softmax for CPU and both values of hardmax and softmax for memory. All interfaces (SMIT, AIX commands, and Web-based System Manager) will convert all data existing from its old format to the new one.

The disk I/O resource is new for the current version, so when activating the AIX 5L WLM with the configuration files of the first WLM release, the values for the shares and the limits will be the default ones for this resource. The system defaults are:

- ▶ shares = -
- ▶ min = 0 percent, softmax = 100 percent, hardmax = 100 percent

For existing WLM configurations, the disk I/O resource will not be regulated by WLM, which should lead to the same behavior for the class as with the first version.

3.1.5 Resource sets

WLM uses the concept of resource sets (or rsets) to restrict the processes in a given class to a subset of the system's physical resources. In AIX 5L, the physical resources managed are the memory and the processors. A valid resource set is composed of memory and at least one processor.

Figure 3-7 shows the SMIT panel where a resource set can be specified for a specific class.

General characteristics of a class

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

	[Entry Fields]	
Class name	Redbook	
Description	[Redbook example]	
Tier	[0]	+ #
Resource Set	sys/cpu.00003	+
Inheritance	[Yes]	+
User authorized to assign its processes to this class	[user_s]	+
Group authorized to assign its processes to this class	[system]	+
User authorized to administrate this class (Superclass only)	[user_s]	+
Group authorized to administrate this class (Superclass only)	[system]	+

F1=Help
F5=Reset
F9=Shell

F2=Refresh
F6=Command
F10=Exit

F3=Cancel
F7=Edit
Enter=Do

F4=List
F8=Image

Figure 3-7 Resource set definition to a specific class

By default, the system creates one resource set for all physical memory, one for all CPUs, and one separate set for each individual CPU in the system. The **lsrset** command lists all resource sets defined. A sample output for the **lsrset** command follows:

```
# lsrset -av
T Name Owner Group Mode CPU Memory Resources
r sys/sys0 root system r----- 4 511 sys/sys0
sys/node.00000 sys/mem.00000 sys/cpu.00003 sys/cpu.00002 sys/cpu.00001
sys/cpu.00000
r sys/node.00000 root system r----- 4 511 sys/sys0
sys/node.00000 sys/mem.00000 sys/cpu.00003 sys/cpu.00002 sys/cpu.00001
sys/cpu.00000
r sys/mem.00000 root system r----- 0 511 sys/mem.00000
r sys/cpu.00003 root system r----- 1 0 sys/cpu.00003
r sys/cpu.00002 root system r----- 1 0 sys/cpu.00002
r sys/cpu.00001 root system r----- 1 0 sys/cpu.00001
r sys/cpu.00000 root system r----- 1 0 sys/cpu.00000
```

rset registry

As mentioned previously, some resource sets in AIX 5L are created, by default, for memory and CPU. It is possible to create different resource sets by grouping two or more resource sets and storing the definition in the rset registry.

The rset registry services enable system administrators to define and name resource sets so that they can then be used by other users or applications. In order to alleviate the risks of name collisions, the registry supports a two-level naming scheme. The name of a resource set takes the form *name_space/rset_name*. Both the *name_space* and *rset_name* may each be 255 characters in size, are case-sensitive, and may contain only upper and lower case letters, numbers, underscores, and periods. The name space of sys is reserved by the operating system and used for rset definitions that represent the resources of the system.

The **SMIT rset** command has options to list, remove, or show a specific resource set used by a process and the management tools, as shown in Figure 3-8.

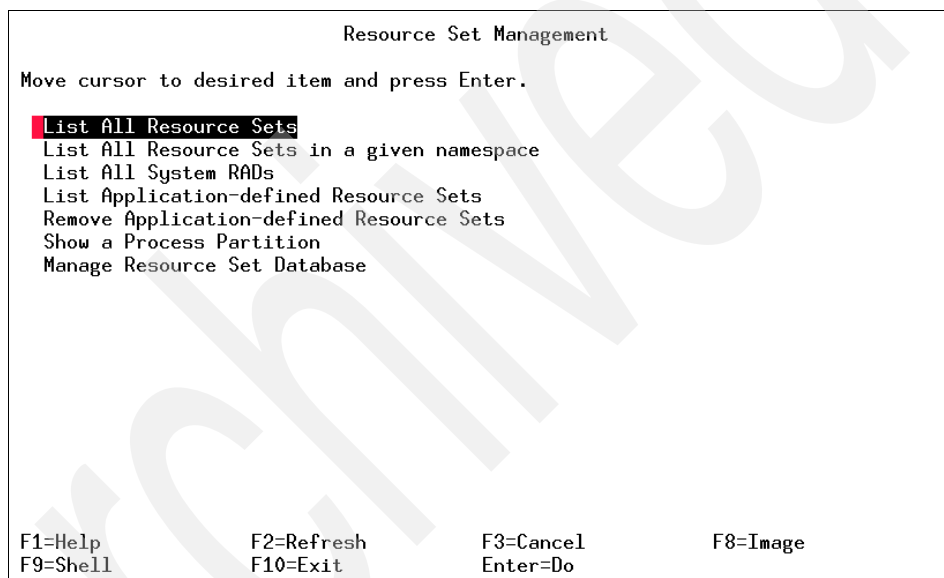


Figure 3-8 SMIT main panel for resource set management

To create, delete, or change a resource set in the rset registry, you must select the Manage Resource Set Database item in the SMIT panel. In this panel, it is also possible to reload the rset registry definitions to make all changes available to the system. Figure 3-9 on page 54 shows the SMIT panel for rset registry management.

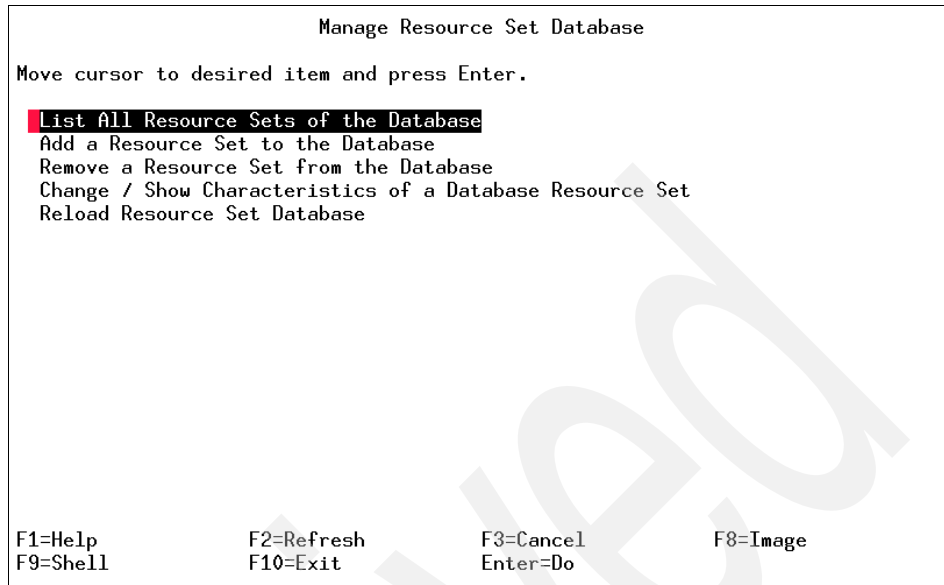


Figure 3-9 SMIT panel for rset registry management

To add a new resource set, you must specify a name space, a resource set name, and the list of resources. It is also possible to change the permissions for the owner and group of this rset. In addition, permissions for the owner, groups, and others can also be specified. Figure 3-10 on page 55 shows the SMIT panel for this task.

Add a Resource Set to the Database

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

* Entry Fields		
* Name Space		[Redbook] +
* Resource Set Name		[CPU0and1] +
* Owner		root +
* Group		system +
* Owner Permissions		rw +
* Group Permissions		r- +
* Others Permissions		r- +
* Resources		sys/cpu.00001.sys/cpu.> +

F1=Help
F5=Reset
F9=Shell

F2=Refresh
F6=Command
F10=Exit

F3=Cancel
F7=Edit
Enter=Do

F4=List
F8=Image

Figure 3-10 SMIT panel to add a new resource set

Whenever a new rset is created, deleted, or modified, a reload in the rset database is needed in order to make the changes effective.

3.1.6 WLM configuration enhancements

In AIX 5L, both the SMIT-based and the Web-based System Manager versions of WLM configuration are enhanced. Many new options are included because of the new features presented earlier in this section.

Figure 3-11 on page 56 shows a SMIT character-based main panel for Workload Manager.

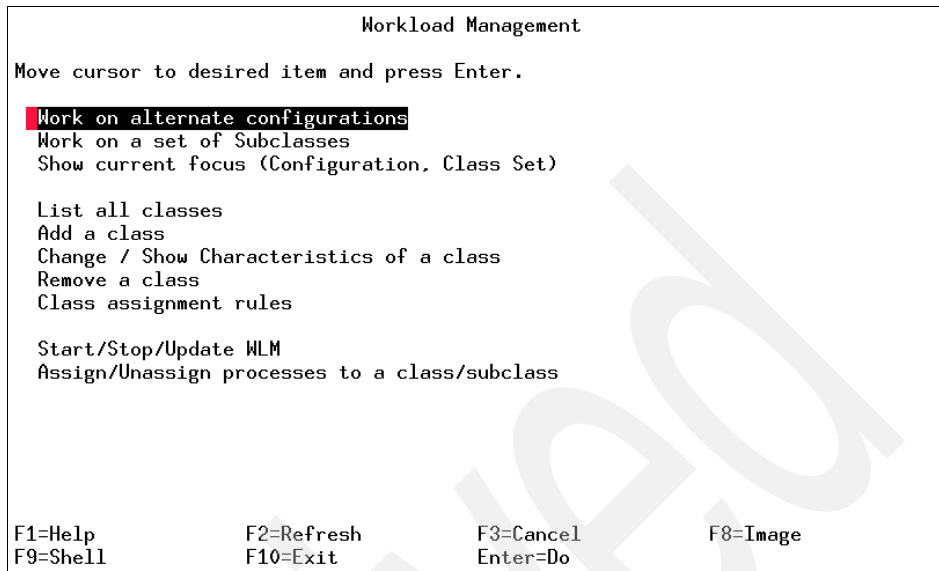


Figure 3-11 SMIT main panel for Workload Manager configuration

It is also possible to view, modify, or create Workload Manager through the Web-based System Manager, as shown on Figure 3-12 on page 57.

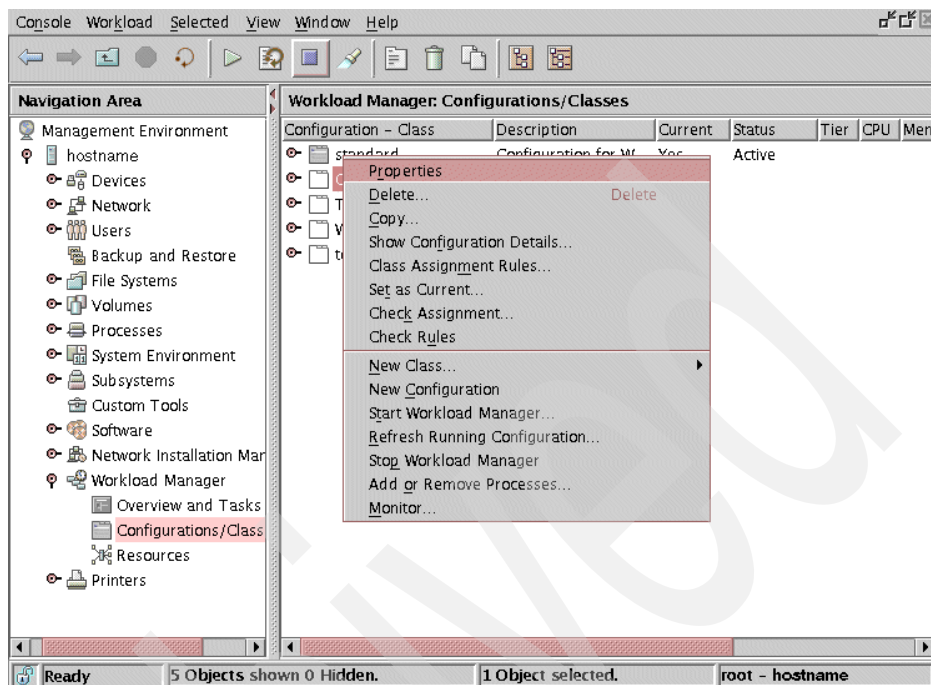


Figure 3-12 Web-based System Manager options for Workload Manager

Work on alternate configurations

This option allows you to create specific sets of configurations, each one with its own classes and rules. This is useful when different resources are needed for the same classes, or to provide a way to switch among different behaviors (for example, in a contingency situation).

When creating a new alternate configuration, WLM provides a sample configuration, called template, that defines the predefined Superclasses: Default, System, and Shared.

If this option is selected in the SMIT panel, it will open a new submenu with some additional options, which are discussed in the following sections.

Show all configurations

This option will display a list of all alternate configurations defined in the system. A sample output for this option is below:

```
COMMAND STATUS
```

```
Command: OK          stdout: yes          stderr: no
```

Before command completion, additional instructions may appear below.

```
redbook      : Redbook Configuration
standard     : Sample for Redbook
template     : Template to create a new configuration -
test : Template to create a new configuration -
```

Copy a configuration

This option copies an entire configuration to a different configuration set. It will preserve all definitions created or changed. It can be used, if you need to have multiple configuration sets, with slight differences on the attributes with the same, or almost the same, number and naming convention for Superclasses and Subclasses.

Create a configuration

A new configuration set will be created, using the default sample, which will create three basic classes: System, Default, and Shared. These classes are defined in the sample configuration called *Template* within WLM.

Select a configuration

In this option, you can switch to an alternate configuration. Keep in mind that this selection will be effective after the next WLM update or restart.

Enter configuration description

Each alternate configuration set has a label that can be modified to describe goals, or any other information.

Remove a configuration

This option allows you to completely remove a configuration from the system.

Work on a set of Subclasses

This option allows you to change the class set. A class set is needed when you need add, remove, or change attributes in Subclasses for a Superclass. If hyphen is selected, then any add, remove, or change class operations will be effective in the Superclass layer. On the other hand, if there is a Superclass assigned in this option, all the class operations will occur in the Subclass layer for this specific Superclass.

In Figure 3-13 on page 59, user in Superclasses was selected as the class set, and the operation created a new Subclass named DB for Superclass user.

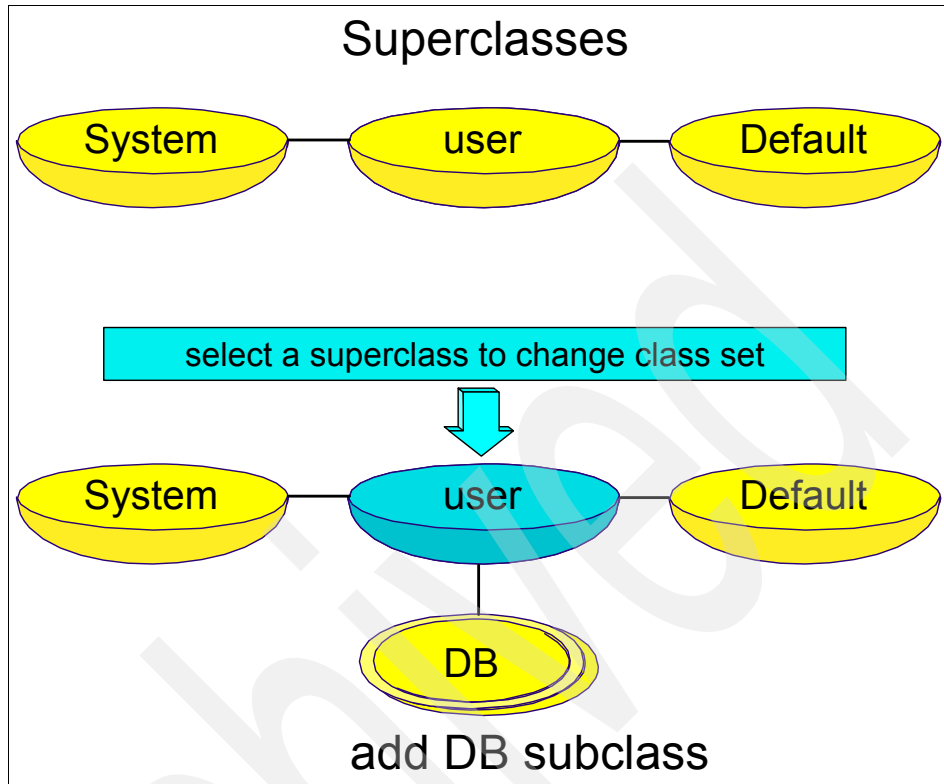


Figure 3-13 An example of adding a Subclass to a Superclass

Show current focus

This option provides output for two sets: The Configuration set and the Class set. This option is necessary when you do not know which configuration or class set you are pointing to.

COMMAND STATUS

Command: OK stdout: yes stderr: no

Before command completion, additional instructions may appear below.

Configuration: redbook

Class set: Subclasses of user/

current -> redbook

List all classes

This option shows a list of classes. If the class set is pointing to a specific Superclass, then all Subclasses for this specific Superclass will be listed. Otherwise, a list of Superclasses will be shown.

COMMAND STATUS

Command: OK stdout: yes stderr: no

Before command completion, additional instructions may appear below.

Default
Shared
db

Add a class

This option can be used to add a new Superclass or Subclass. “Class attributes” on page 42 gives a detailed description of all the fields for this panel.

Change/Show Characteristics of a class

This option allows you to change a class configuration. For example, tier, resource set, or administration users. But it also lets you change resource management characteristics for CPU, memory, and disk I/O. There is also a new option for limit.

General characteristics of a class

It is possible to change all the characteristics of a class; see “Class attributes” on page 42 for a list of attributes that can be modified with this option. Figure 3-5 on page 43 shows the SMIT panel for this option.

CPU resource management

It is possible to change the percentage of minimum and maximum CPU resources for a specific class. A new field introduced in this release is *Absolute maximum (%)*, which controls the enforced maximum CPU consumption for this class, even if there are CPU resources in idle.

A sample CPU resource management SMIT input screen for db class follows:

CPU resource management

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

Class name	[Entry Fields]
	db

Shares	[-]
#	
Minimum (%)	[0]
#	
Maximum (%)	[100]
#	
Absolute Maximum (%)	[100]
#	

Memory resource management

The total amount of physical memory available for processes at any given time is the total number of memory pages physically present on the system (minus the number of pinned pages). The pinned pages are not managed by WLM, since these pages cannot be stolen from a class and given to another class in order to regulate memory utilization. The memory utilization of a class is simply the ratio of the number of (non-pinned) memory pages being used by all the processes in the class to the number of pages available on the system (as defined above, expressed as a percentage). As in CPU resource management, there are minimum and maximum percentages (%) as soft limits, and absolute maximum as a hard limit.

A sample Memory resource management SMIT input screen for db class follows:

Memory resource management

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

	[Entry Fields]
Class name	db
Shares	[-]
Minimum (%)	[0]
Maximum (%)	[100]
Absolute Maximum (%)	[100]

Disk I/O resource management

For the disk I/O, the main difficulty is determining a meaningful available bandwidth for a device. When a disk is 100 percent busy, its throughput (in blocks per second) will be very different if one application is doing sequential I/Os than if several applications are doing random I/Os. If the maximum throughput measured for the sequential I/O case was used as a value of the I/O bandwidth available for the device to compute the percentage of utilization under random I/Os, statistical errors would be created. It would lead you to think that the device is, for example, 20 percent busy, when it is in fact at 100 percent utilization.

In order to get more accurate and reliable percentages of per class disk utilization, WLM uses the data provided by the disk drivers (which are displayed

with the **iostat** command), giving the percentage of the time the device has been busy during the last second for each disk device. WLM knows how many blocks in total have been read/written on a device during the last few seconds by all the classes accessing the device, how many blocks have been read/written by each class, and what was the percentage of utilization of the device, and can easily calculate what percentage of the disk throughput was consumed by each class. For example, if the total number of blocks read or written during the last second was 1000 and the device had been 70 percent busy, this means that a class reading or writing 100 blocks used 7 percent of the disk bandwidth. Similarly, to the CPU time (another renewable resource), the values used by WLM for its disk I/O regulation are also a decayed average over a few seconds of these per second percentages.

For the disk I/O resource, the shares and limits apply to each disk device accessed by the class individually, and the regulation is done independently for each device. Moreover, the same soft and hard limits apply to this resource.

A sample disk I/O resource management SMIT input screen for db class follows:

diskIO resource management

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[Entry Fields]		
Class name	db	
Shares	[-]	
#		
Minimum (%)	[0]	
#		
Maximum (%)	[100]	
#		
Absolute Maximum (%)	[100]	#

Remove a class

This option allows you to completely remove a class from the system.

Class assignment rules

After creating a class and setting the number of shares, soft and hard limits percentage for CPU, and memory and disk I/O, it is necessary to create the assignment rules. Class assignment rules will allow you to join all the class characteristics together within a specific application, user, and other types.

List all Rules

This option will show an output with all defined assignment rules set in the system with their specific characteristics, as in the following:

COMMAND STATUS

Command: OK stdout: yes stderr: no

Before command completion, additional instructions may appear below.

#	Class	User	Group	Application	Type	Tag
001	System	root	-	-		
002	Default	-	-	-		

By default, there are two predefined rules that will be available in any WLM class. The first rule is for the System class that causes any application started by *root* to be assigned to this rule. The second rule is for the Default class, and it defines the rules for any application issued in the system by any user other than *root*.

Create a new Rule

To create an assignment Rule in WLM, you must keep in mind that the order of the rule will be affected by or will affect other rules. WLM will follow the rules beginning with Rule number one (001). Then, for example, if rule number one states that all root user process will belong to System class, any root user process will never be affect by rule number two or later.

Figure 3-14 shows the SMIT panel for creating a new rule.

Create a new Rule

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

* Order of the rule

* Class name

* User

* Group

Application

Type

Tag

[1]

user

[wlmuser]

[]

[]

[]

[]

#

+

+

+

+

F1=Help

F2=Refresh

F3=Cancel

F4=List

F5=Reset

F6=Command

F7=Edit

F8=Image

F9=Shell

F10=Exit

Enter=Do

Figure 3-14 Example of SMIT panel for creating a new rule

A discussion of the fields to fill out for Rule Order follows. Order of the Rules and Class name are mandatory fields; all others are optional.

Order of the rule	Defines the rule order among other rules. The rule number one (001) is the first preferred order.
Class name	Specifies which class will be affected by the rule.
User	If specified, it will affect the user processes that match the pattern provided.
Group	If specified, it will affect the group processes that match the pattern provided.
Application	Affects a specific application, or you can use wildcards to affect a certain range of applications. For example, /tmp/wlm/* will affect any application under the /tmp/wlm directory.
Type	Only defined types of applications will be affected.
Tag	Affects specific applications that have a tag that matches.

Note: “Classification process” on page 45 has a detailed architectural approach about Assignment Rules.

Change/Show Characteristics of a Rule panel

It is possible to change all characteristics established for a Rule, including order and class. Figure 3-15 on page 65 shows a SMIT panel used for this item.

Change / Show Characteristics of a Rule

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

Order of the rule	[Entry Fields]
New Order of the rule	1 [1] #
* Class name	user +
* User	[root] +
* Group	[system] +
Application	[/tmp/wlm/sum.sh] +
Type	[-] +
Tag	[-]

F1=Help
F5=Reset
F9=Shell

F2=Refresh
F6=Command
F10=Exit

F3=Cancel
F7=Edit
Enter=Do

F4=List
F8=Image

Figure 3-15 Fields that can be modified for a specific rule

Delete a Rule

This option allows you to completely remove a Rule from the system.

Note: Note that any creations, deletions, or modifications in any kind of configuration within WLM will only be effective after you update WLM or restart WLM.

Start, Stop, or Update WLM

In this option, it is possible to Start and Stop WLM. Or, if you modified, created, or removed any component on WLM, you can update so that the changes take effect. Another function of this option is to show the WLM status.

Update Workload Management panel

The update function (as shown in Figure 3-16 on page 66) allows you to create classes, change assignment Rules, and perform many other functions that were not updated in earlier releases.

In this release, any action performed to change the configuration can be updated and be effective without needing to restart WLM.

Another enhancement for Update is the possibility of updating only a specific Superclass instead of the entire WLM.

Update Workload Management

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

Superclass name (restricted update,
applies only to 'current' configuration)

[Entry Fields] +

F1=Help
F5=Reset
F9=Shell

F2=Refresh
F6=Command
F10=Exit

F3=Cancel
F7=Edit
Enter=Do

F4=List
F8=Image

Figure 3-16 SMIT panel for Update Workload Management

Assign/Unassign processes to a class/Subclass

To assign or unassign processes to a class or Subclass, use the SMIT menu, as shown in Figure 3-17 on page 67, or see “Manual assignment” on page 46 for a description of the process from an architectural point of view.

Assign/Unassign processes to a class/subclass

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

Assign/Unassign to/from Superclass/Subclass/Both	[Entry Fields]	
Class name (for assignment)	Assign Superclass	+
List of PIDs	<input type="text"/>	+
List of PGIDs	<input type="text"/>	+

F1=Help
 F5=Reset
 F9=Shell

F2=Refresh
 F6=Command
 F10=Exit

F3=Cancel
 F7=Edit
 Enter=Do

F4=List
 F8=Image

Figure 3-17 SMIT panel for manual assignment of processes

Assign/Unassign to/from Superclass/Subclass/Both

This field is used to specify whether you are assigning or unassigning a process and if it belongs to a Superclass, Subclass, or both.

All the options for this field and their respective descriptions are:

Assign Superclass	All desired processes will be assigned to a specific Superclass.
Assign Subclass	All desired processes will be assigned to a specific Subclass.
Assign Both	All desired processes will be assigned to both Superclass and Subclass levels.
Unassign Superclass	All desired processes will be unassigned from a Superclass.
Unassign Subclass	All desired processes will be unassigned from a Subclass.
Unassign Both	All desired processes will be unassigned from both Superclass and Subclass.

Class name

This field must contain the Superclass or Subclass that will affect the processes listed to either Assign or Unassign.

List of PIDs

It is possible to select multiple processes at once. A comma must be used as a separator between each PID.

List of PGIDs

It is also possible to select a single PGID or a list of PGIDs instead of single PIDs.

WLM for accounting (5.1.0)

Starting with AIX 5L Version 5.1, WLM provides kernel support per class accounting, which means that accounting records can be gathered by WLM class. This new feature implies the enhancement of two new flags for the **acctcom** command: The **-w** and **-c** flags.

The accounting system utility allows you to collect and report on individual, group, and Workload Manager (WLM) class use of various system resources. This accounting information can be used to bill users for the system resources they utilize, and to monitor selected aspects of the system operation. To assist with billing, the accounting system provides the resource-usage totals defined by members of the adm group, and, if the **chargefee** command is included, factors in the billing fee.

The accounting system also provides data to assess the adequacy of current resource assignments, set resource limits and quotas, forecast future needs, and track supplies for printers and other devices.

The **acctcom** command displays selected process accounting record summaries. Each record represents one completed process. The default display consists of the command name, user name, TTY name, start time, end time, real seconds, CPU seconds, and mean memory size (in kilobytes). These default items have the following headings in the output:

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

Running the **acctcom** command with the **-w** flag will show all processes and their class name. Running the **acctcom** command with the **-c** flag displays all processes belonging to the specified class. A mechanism has been introduced to allow users to gather accounting information by class. A 64-bit key is generated from the Superclass and Subclass names to achieve this function. When the accounting records are processed, the signature of all the class names found in **/etc/wlm** is computed and stored in an internal table. For each record, the signature is compared to this table, and the class name is retrieved. The accounting command translates the key back into the class name.

For example, run the following command:

```
# acctcom -w
```

COMMAND			START	END	REAL	CPU	
MEAN							
NAME	USER	CLASS	TTYNAME	TIME	TIME	(SECS)	(SECS)
SIZE(K)							
#accton	root	System.Default	?	10:44:34	10:44:34	0.02	
0.02	0.00						
#bsh	root	System.Default	?	10:44:34	10:44:34	0.25	
0.00	248.00						
#setmaps	root	System.Default	?	10:49:26	10:49:26	0.02	
0.02	0.00						
#ls	root	System.Default	?	10:49:27	10:49:27	0.03	
0.02	80.00						
#more	root	System.Default	?	10:49:34	10:49:34	0.81	
0.09	60.00						
termdef	adm	Default.Default	?	10:49:42	10:49:42	0.02	
0.02	185.00						
ls	adm	Default.Default	?	10:49:43	10:49:43	0.02	
0.02	58.00						
nfssync_k	root	System.Default	?	10:49:44	10:49:44	0.00	
0.00	0.00						
nfssync_k	root	System.Default	?	10:49:44	10:49:44	0.00	
0.00	0.00						
ps	adm	Default.Default	?	10:49:45	10:49:45	0.05	
0.03	155.00						
#tsm	root	System.Default	?	10:49:26	10:49:51	25.61	
0.56	116.00						

You can see two different classes: The System.Default class and the Default.Default class. If you want to display all processes belonging to the Default.Default class, the -c flag has to be used:

```
# acctcom -c Default.Default
```

COMMAND			START	END	REAL	CPU	MEAN
NAME	USER	TTYNAME	TIME	TIME	(SECS)	(SECS)	SIZE(K)
termdef	adm	?	10:49:42	10:49:42	0.02	0.02	185.00
ls	adm	?	10:49:43	10:49:43	0.02	0.02	58.00
ps	adm	?	10:49:45	10:49:45	0.05	0.03	155.00

Also, a combination of the these two flags can be used:

```
# acctcom -wc Default
```

COMMAND			START	END	REAL	CPU
MEAN						
NAME	USER	CLASS	TTYNAME	TIME	TIME	(SECS)
(SECS)	SIZE(K)					
termdef	adm	Default.Default	?	10:49:42	10:49:42	0.02
0.02	185.00					

ls	adm	Default.Default	?	10:49:43	10:49:43	0.02
0.02	58.00					
ps	adm	Default.Default	?	10:49:45	10:49:45	0.05
0.03	155.00					

With the -c option, a Superclass name or a full class name can be passed. A Superclass name will display the records for all the Subclasses:

```
# acctcom -w -c class1
```

COMMAND				START	END	REAL
CPU	MEAN					
NAME	USER	CLASS	TTYNAME	TIME	TIME	(SECS)
(SECS)	SIZE(K)					
#date	wlmu1	class1.sub2	pts/0	05:26:05	05:26:05	0.09
0.09	95.00					
date	wlmu1	class1.sub2	tty0	05:26:40	05:26:40	0.02
0.02	0.00					
ls	wlmu1	class1.sub2	tty0	05:26:43	05:26:43	0.02
0.02	0.00					
vi	wlmu1	class1.sub2	tty0	05:26:48	05:26:55	7.38
0.03	432.00					
grep	wlmu1	class1.sub2	tty0	05:27:03	05:27:03	0.02
0.02	0.00					
#ksh	wlmu1	class1.sub2	tty0	05:26:36	05:27:05	29.91
0.08	214.00					
termdef	wlmu2	class1.Default	tty0	05:27:18	05:27:18	0.02
0.00	164.00					
find	wlmu2	class1.Default	tty0	05:27:31	05:27:31	0.09
0.00	0.00					
ls	wlmu2	class1.Default	tty0	05:27:39	05:27:39	0.02
0.02	213.00					
sleep	wlmu2	class1.Default	tty0	05:27:47	05:27:50	3.02
0.02	180.00					
#ksh	wlmu2	class1.Default	tty0	05:27:18	05:27:54	36.72
0.06	282.00					
who	wlmu0	class1.sub1	tty0	05:28:06	05:28:06	0.05
0.02	0.00					
df	wlmu0	class1.sub1	tty0	05:28:12	05:28:12	0.02
0.02	40.00					
cat	wlmu0	class1.sub1	tty0	05:28:19	05:28:19	0.02
0.02	122.00					
ls	wlmu0	class1.sub1	tty0	05:28:31	05:28:31	0.02
0.00	86.00					
cpio	wlmu0	class1.sub1	tty0	05:28:31	05:28:31	0.02
0.02	0.00					
#						

The following is the complete syntax of the **acctcom** command:

```
/usr/sbin/acct/acctcom [ [ -q | -o File ] | [ -a ] [ -b ] [ -c Classname ]  
[ -f ] [ -h ] [ -i ] [ -k ] [ -m ] [ -r ] [ -t ] [ -v ] [ -w ]] [ -C Seconds ]  
[ -g Group ] [ -H Factor ] [ -I Number ] [ -l Line ] [ -n Pattern ]  
[ -O Seconds ] [ -u User ] [ -e Time ] [ -E Time ] [ -s Time ] [ -S Time ]  
[ File ... ]
```

3.1.7 Monitoring WLM with **wlmmmon** and **wlmpperf** (5.1.0)

The new **wlmmmon** command in AIX 5L Version 5.1, and **wlmpperf** command, available with PTX Version 3.0 for AIX 5L and AIX Version 4.3.3, provides graphical views of Workload Manager (WLM) resource activities by class. While the **wlmstat** command provides a per-second fidelity view of WLM activity, it is not suited for long-term analysis. The **wlmmmon** and **wlmpperf** tools were created to supplement **wlmstat**.

These tools provide reports of WLM activity over much longer time periods. The **wlmmmon** tool is a disabled version of the **wlmpperf** tool, and the primary difference between the two tools is the period of WLM activity that may be analyzed. The recordings of **wlmpperf** are limited to one year; on the other hand, **wlmmmon** is limited to generating reports within the last 24 hour period. The recordings are generated by associated daemons that have minimal impact on overall system performance. In **wlmmmon**, this daemon is called **xmwlm**, and ships with the base AIX. For **wlmpperf**, the **xmtrend** daemon is used to collect and record WLM. These daemons sample WLM and system statistics at a very high rate (measured in seconds), but only record supersampled values at a low rate (measured in minutes). These values represent the minimum, maximum, mean, and standard deviation values for each collected statistic over the recording period. To execute **wlmmmon** and **wlmpperf**, you can enter **wlmmmon** or **wlmpperf** without any options. This section explains the execution of **wlmpperf**; any differences to **wlmmmon** are pointed out in the relevant sections.

Daemon recording and configuration

Both the **wlmmmon** and **wlmpperf** daemons create recordings in the **/etc/perf/wlm** directory.

For **wlmpperf**, the **xmtrend** daemon is used, and will utilize a configuration file for recording preferences. A sample of this configuration file for WLM-related recordings is located in **/usr/lpp/perfagent.server/xmtrend_wlm.cf**. Recording customization, startup, and operation is briefly described in the following section.

For **wlmmmon**, the **xmwlm** daemon is used, and cannot be customized. For recordings to be created, adequate disk allocations must be made for the **/etc/perf/wlm** directory, allowing at least 10 MB of disk space. Additionally, the daemon should be started from an **/etc/inittab** entry so that recordings will

automatically restart after system reboots. The daemon will operate whether the WLM subsystem is in active, passive, or disabled (off) mode. However, recording activity is limited when WLM is off.

In order to start the recording, the daemons have to be active. To start the graphic monitoring tool, run the **wlmon** command (base AIX) or the **wlperf** command (PTX).

Upon startup, a default Report Display is shown. To view recordings, use the WLM_Console menu, as described in the next section.

The WLM_Console menu

The tab down menu WLM_Console, shown in Figure 3-18, displays the following selections:

Open Log	Allows browsing to and viewing recordings.
Reports	Allows opening, copying, or deleting different reports (for wlperf only).
Print	Allows printing the current report.
Exit	Exits the wlmon tool.

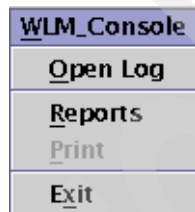


Figure 3-18 WLM_Console tab-down menu

The WLM report browser

When selecting the **Open Log** menu, the report browser is displayed, as shown in Figure 3-19 on page 73. The browser allows you to browse through the different directories and displays a list of reports.

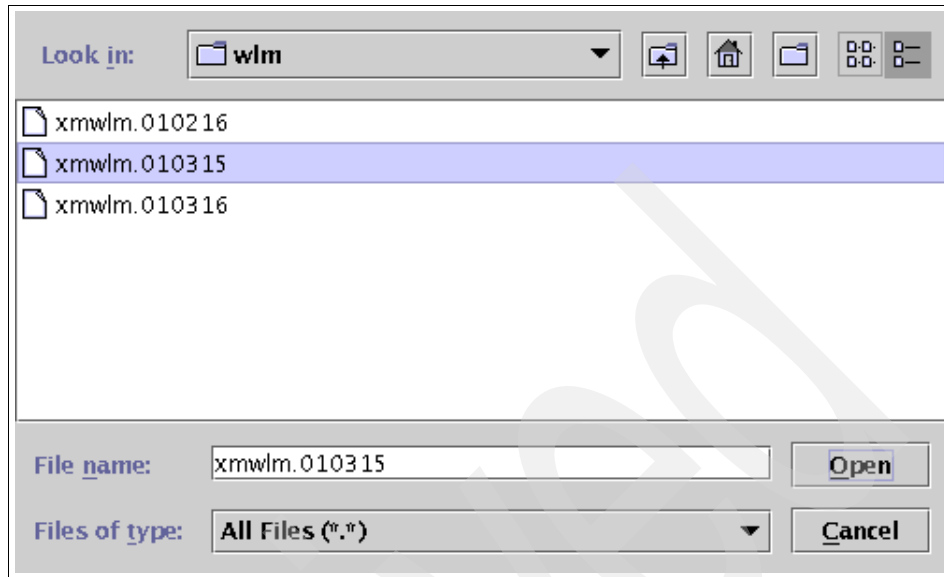


Figure 3-19 Report browser

Report displays

There are three types of report displays: Snapshot display, bar display, and tabulation display. The bar display is opened by default.

These three displays have the following common elements:

WLM Console	Tab-down menu that allow you to select open recordings (log file), open reports (wlmp erf only), print reports, and exit the tool.
Selected	Tab-down menu that allows you to select the report properties.
Tier column	Displays the tier number associated with a class.
Class column	Displays the class name.
Resource columns	Displays the resource information (CPU, memory, and disk I/O) based on the type of graphical report selection chosen.
Status area	Displays a set of global system performance metrics that are also recorded to aid in analysis. The set displayed may vary between AIX releases, but will include metrics such as run, queue, swap queue, and CPU busy.

Host	Displays the host name of the system on which the recording was made.
WLM State	Displays the state of WLM. This can be Active or Passive.
Time period	Displays the time period defined in the Times menu of the Report Properties panel. For trend reports comparing two time periods, two time displays are shown.

Bar display

As shown in Figure 3-20, the resource columns are displayed in bar-graph style, along with the percentage of measured resource activity over the time period specified. The percentage is calculated based on the total system resources defined by the WLM subsystem. If the detailed display is trended, the later (second) measurement is shown above the earlier (first) measurement interval.

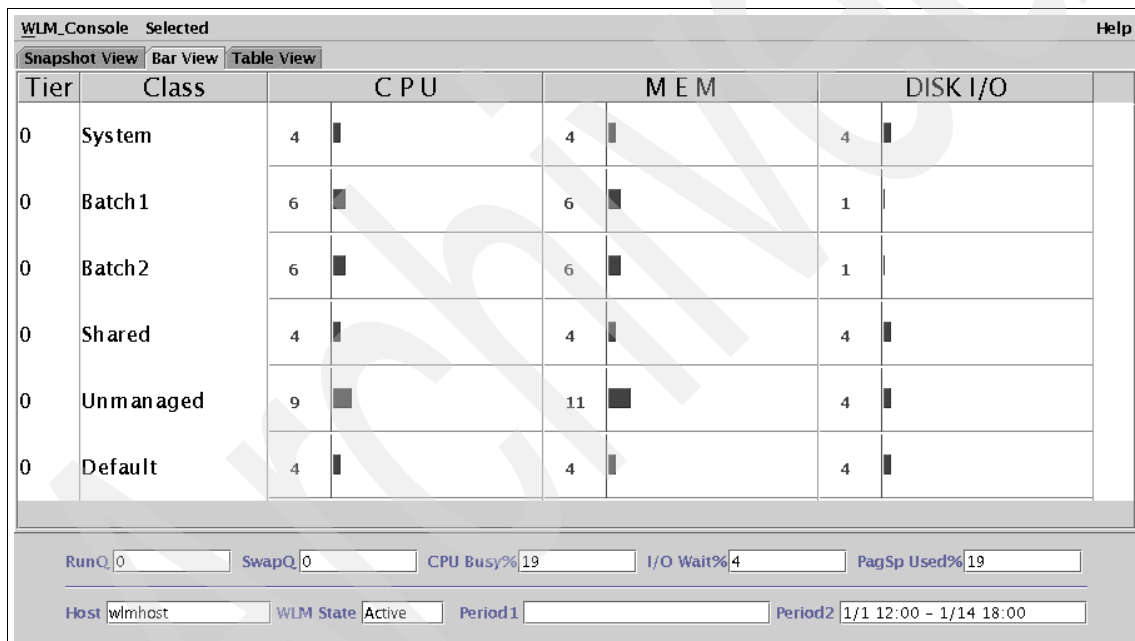


Figure 3-20 Bar view

Snapshot display

Figure 3-21 on page 75 shows the snapshot display, where it focuses on showing class resource relationships based on user-specified variation from the defined target shares. To select or adjust the variation parameters for this display, utilize the Report Properties panel Advanced menu, as shown in Figure 3-28 on page 80. If the Snapshot display is trended, the earlier (first) analysis period is shown by an arrow pointing from the earlier measurement to the later (second)

measurement. If there has been no change between the periods, no arrow is shown.

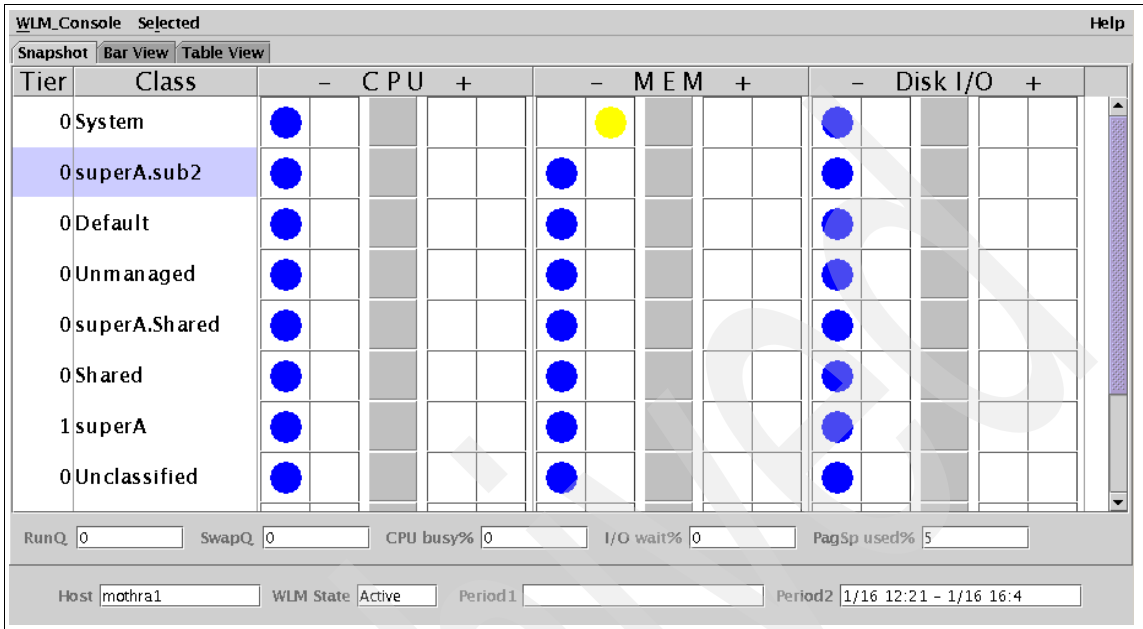


Figure 3-21 Snapshot view

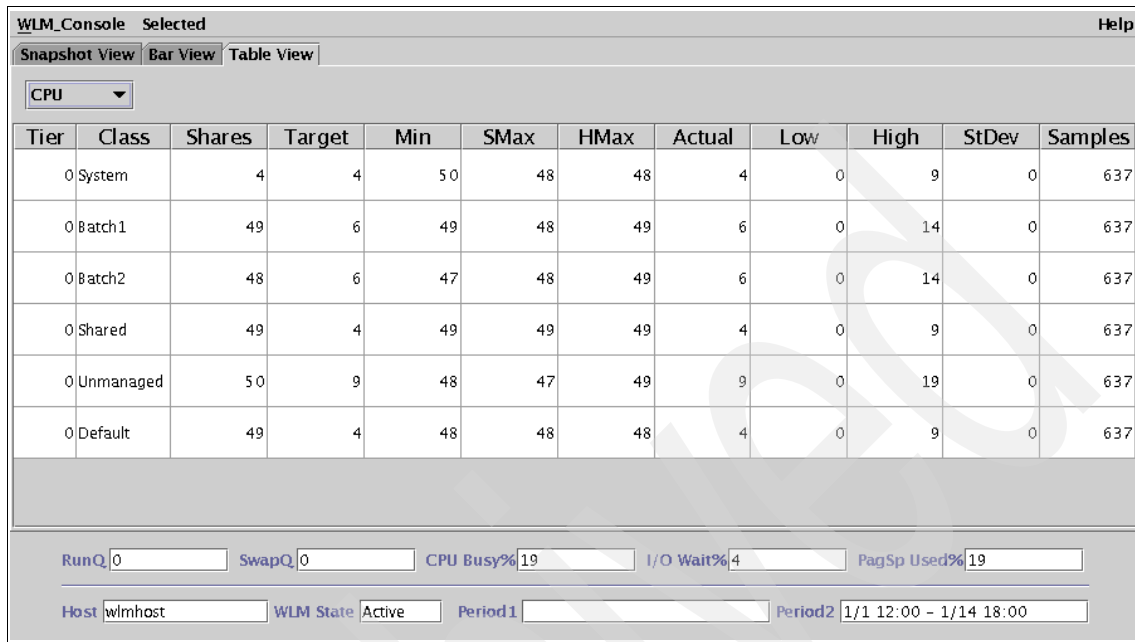
Tabulation display

The third type of display report is shown in Figure 3-22 on page 76. In this report, the following fields are provided:

Shares	Defined shares in WLM configuration.
Target	Computed share value target by WLM in percent. If the share is undefined, the target displays 100.
Min	Class minimum defined in WLM limits.
SMax	Class soft maximum defined in WLM limits.
HMax	Class hard maximum defined in WLM limits.
Actual	Calculated average over the sample period.
Low	Actual observed min across time period.
High	Actual observed max across time period.
Standard Deviation	Computed standard deviation of Actual, High, and Low. Indicates the variability of the Actual values during the recording period. Higher standard deviation means more variability; lower standard deviation means less variability.

Samples

Number of recorded samples for this period.



The image shows a screenshot of the WLM Console interface. At the top, there's a title bar 'WLM_Console' with a 'Selected' tab and a 'Help' button. Below the title bar are three tabs: 'Snapshot View', 'Bar View', and 'Table View', with 'Table View' being the active one. A dropdown menu shows 'CPU'. The main area is a table with 12 columns: Tier, Class, Shares, Target, Min, SMax, HMax, Actual, Low, High, StDev, and Samples. The table contains 6 rows of data for different classes. Below the table, there are several input fields for system metrics: RunQ (0), SwapQ (0), CPU Busy% (19), I/O Wait% (4), and PagSp Used% (19). At the bottom, there are fields for Host (wlmhost), WLM State (Active), Period1, and Period2 (1/1 12:00 - 1/14 18:00).

Tier	Class	Shares	Target	Min	SMax	HMax	Actual	Low	High	StDev	Samples
0	System	4	4	50	48	48	4	0	9	0	637
0	Batch1	49	6	49	48	49	6	0	14	0	637
0	Batch2	48	6	47	48	49	6	0	14	0	637
0	Shared	49	4	49	49	49	4	0	9	0	637
0	Unmanaged	50	9	48	47	49	9	0	19	0	637
0	Default	49	4	48	48	48	4	0	9	0	637

RunQ: 0 SwapQ: 0 CPU Busy%: 19 I/O Wait%: 4 PagSp Used%: 19

Host: wlmhost WLM State: Active Period1: Period2: 1/1 12:00 - 1/14 18:00

Figure 3-22 Table view

If the Table display is trended, the earlier (first) analysis is shown by the first number between the brackets and the later (second) analysis is shown by the second number between the brackets.

The report properties

The Report Properties panel allows the user to define the attributes that control the actual graphical representation of the WLM data. The report properties are displayed by selecting **Selected** at the top of the Report display, as shown in Figure 3-23.

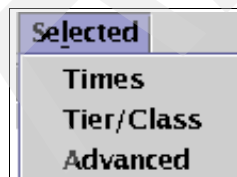


Figure 3-23 Report properties

Times menu

The first tabbed panel is displayed in Figure 3-24 on page 78. It allows the user to edit the time properties of a display.

Note: `wlmmmon` does not allow selection of days, weeks, and months.

The fields are:

Trend box	When checked, indicates that a trend report of the selected type will be generated. Trend reports allow the comparison of two different time periods on the same display. Selecting this box enables the End of first Period field for editing.
Width of Interval	Represents the period of time covered by any display type, measuring from user-input time selections. <i>Interval widths</i> are selected from this pull down menu. The selections available vary depending upon the tool being used. While <code>wlmmmon</code> only has selections for minutes and hours, <code>wlmpperf</code> has selections for minutes, hours, days, weeks, and months.
End of First Period	Represents the end time of a period of interest for generating a trend report. The first period always represents a time frame ending earlier than the last period. This field can only be edited if the <i>Trend box</i> is selected.
End of Last Period	Represents the end time of a period of interest for trend and non-trend reports.

Time Periods:

Time Range in recording: Jan 1, 12:00 PM - Jan 14, 18:00 PM

Trend : ☐

Width of Interval: 5 min

End of First Period (DD:hh:mm)

January 1 12 am 00

End of Last Period (DD:hh:mm)

January 1 12 am 00

Ok Cancel

Figure 3-24 Times menu

Figure 3-25 is an example of a trend selection. The display shows different usage of resources between the two time periods. The time periods are displayed in the fields called Period 1 and Period 2.

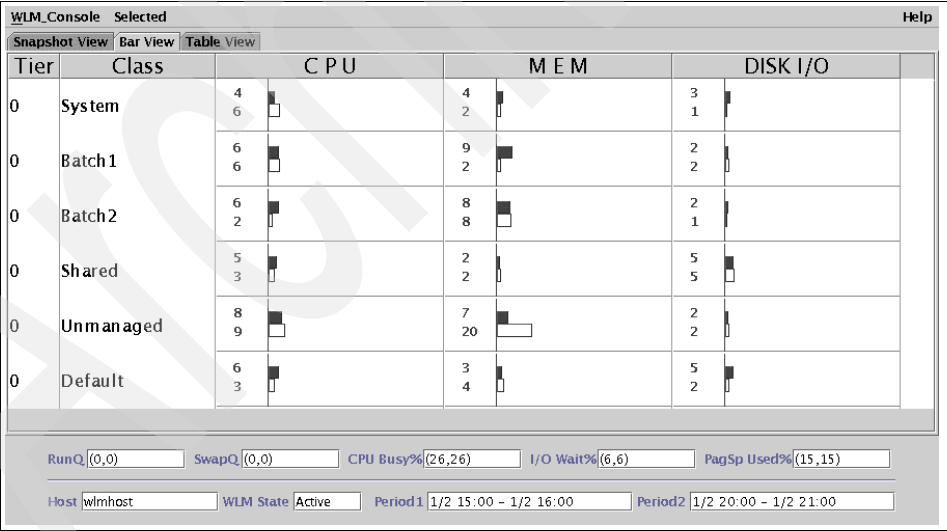


Figure 3-25 Example of trend display, Bar View

Figure 3-26 on page 79 also shows an example of a Snapshot display using the trend option.

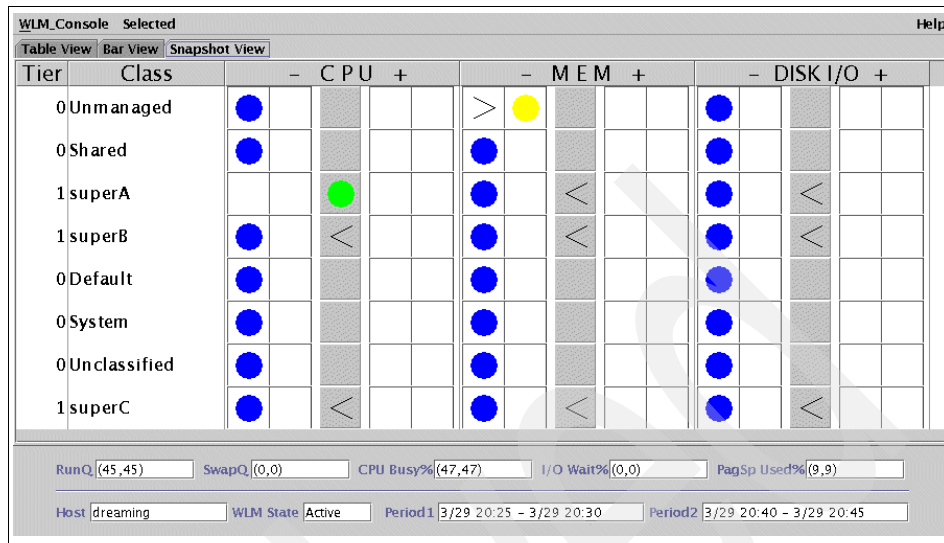


Figure 3-26 Example of trend display, Snapshot View

Tier/Class menu

The second tabbed pane is displayed in Figure 3-27. It allows users to define the set of WLM tiers or classes to be included in a report.

The pull-down menu at the top allows the user to select whether Superclasses or tiers are to be included or excluded in the Report display. The list on the bottom then allows the user to select specific tiers or specific Superclasses.

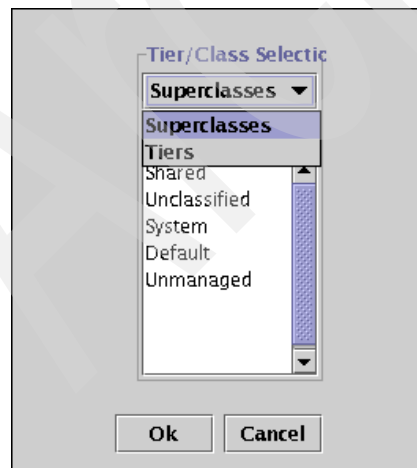
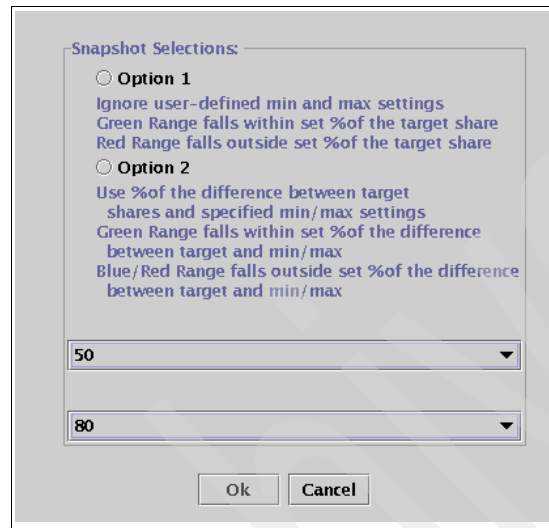


Figure 3-27 Tier/Class menu

Advanced menu (Snapshot option panel)

The third panel of the Report Properties panel is displayed, as shown in Figure 3-23 on page 76. It provides advanced options for the Snapshot display. For snapshots, exclusive methods for coloring the display are provided for user selection. *Option 1* ignores the minimum and maximum settings defined in the configuration of the WLM environment, while *Option 2* utilizes the minimum and maximum settings provided for user selection (Figure 3-28).



The dialog box is titled "Snapshot Selections:". It contains two radio button options. "Option 1" is selected and includes the text: "Ignore user-defined min and max settings", "Green Range falls within set %of the target share", and "Red Range falls outside set %of the target share". "Option 2" is unselected and includes the text: "Use %of the difference between target shares and specified min/max settings", "Green Range falls within set %of the difference between target and min/max", and "Blue/Red Range falls outside set %of the difference between target and min/max". Below the options are two dropdown menus. The first dropdown menu shows "50" and the second shows "80". At the bottom are "Ok" and "Cancel" buttons.

Figure 3-28 Advanced menu

The following example describes the functions of the Advanced menu.

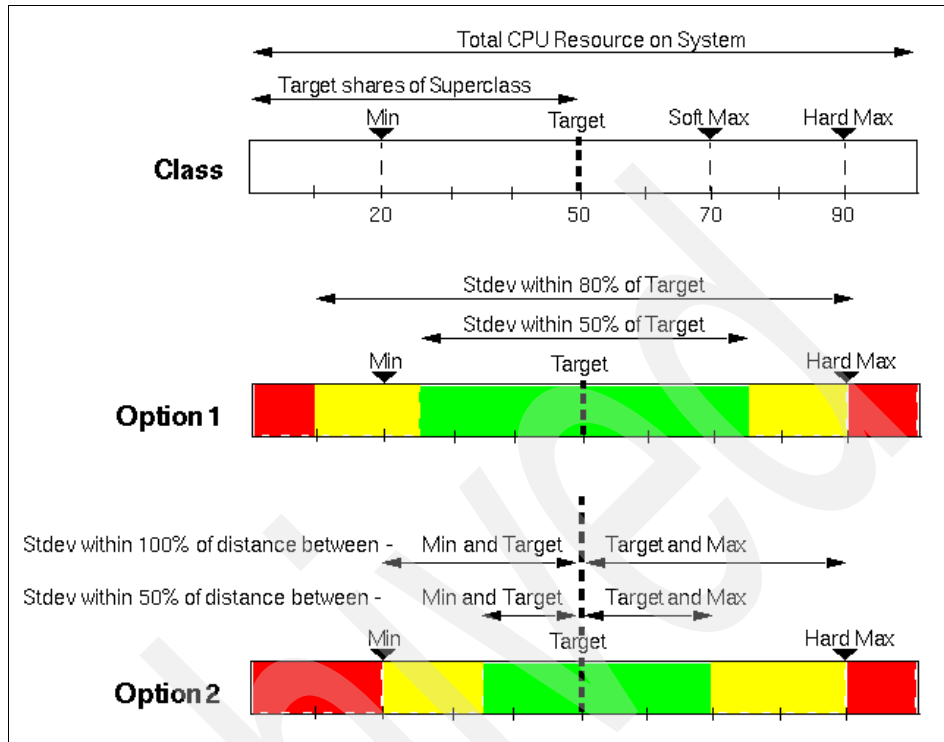


Figure 3-29 Example of the Advanced menu

Figure 3-29 shows a class definition with its soft and hard minimum and maximum. The class has as a target (share value) of 50 percent, a minimum limit (Min) of 20 percent, and maximum limit (Max) of 90 percent. The functions of the two advanced options are:

► Option 1

Ignores the user-defined min and max settings. In this example, we selected Option 1 with 50 percent as the green range percentage (green%) and 80 percent as the red range percentage (red%), as shown in Figure 3-28 on page 80.

To define the green range, the following formula is used:

- Low green range = Target - (Target x green%) = 50 - (50 x 50%) = 25
- High green range = Target + (Target x green%) = 50 + (50 x 50%) = 75

Figure 3-29 shows the green range from 25 percent to 75 percent, on a scale of 0 to 100 percent.

The red range is calculated with the same formula but with the red range percentage:

- Low red range = Target - (Target x red%) = 50 - (50 x 80%) = 10
- High red range = Target + (Target x red%) = 50 + (50 x 80%) = 90

The red range is shown in Figure 3-29 on page 81, Option 1, 0 to 10 percent and from 90 to 100 percent. The area between the red and green range is yellow.

► Option 2

takes in account the predefined minimum limit and maximum limit settings. If we use the same advanced options as in Figure 3-28 on page 80, the red and green range are interpreted between the target and the hard minimum and hard maximum definitions (here 20 and 90 percent).

- Low green range = Target - ((Target - MIN) x green%)
= 50 - ((50 - 20) x 50%) = 35 percent on the scale from 0 to 100 percent
- High green range = Target + ((MAX - Target) x green%)
= 50 + ((90 - 50) x 50%) = 70 percent on the scale from 0 to 100 percent
- Low red range = Target - ((Target - MIN) x red%)
= 50 - ((50 - 20) x 80%) = 26 percent on the scale from 0 to 100 percent
- High red range = Target + ((MAX - Target) x red%)
= 50 + ((90 - 50) x 80%) = 82 percent on the scale from 0 to 100 percent

Files and filesets for wlmmon and wlmperf

The following files and filesets are needed to run **wlmmon** or **wlmperf**.

Files

The files are:

/usr/bin/wlmmon	Base AIX, located in perfagent.tools
/usr/bin/xmwlm	Base AIX, located in perfagent.tools
/usr/bin/wlmperf	Performance Toolbox
/usr/lpp/perfagent.server/xmtrend.cf	Performance Toolbox

Prerequisite filesets

The following filesets are prerequisites for **wlmmon**:

- Java130.adt
- Java130.ext
- Java130.rte
- Java130.samples
- perfagent.tools

3.1.8 Workload Manager enhancements (5.2.0)

Version 5.2 introduces new features to Workload Manager that improve its ease of use and provide more control over resource usage. There are five new enhancements in Version 5.2 for Workload Manager (WLM). they include attribute value grouping, event notification, time-based configurations, limits on total resources in a class and an increase in the limit to the number of user-defined Superclasses and Subclasses.

Attribute value grouping

Attribute value groupings are essentially referenced lists whose names can be specified in the rules files for a configuration in WLM. In the rules file, located in `/etc/wlm/config_name/rules`, the attribute grouping name can be specified instead of listing out all the values for a specific rule. When referenced in a rules file, the grouping name must be preceded by a \$ (U.S. dollar sign) symbol. The grouping file by default is not defined, but once created it resides in `/etc/wlm/config_name/groupings`. Attribute value grouping is configuration specific, although it is possible to copy groupings files to the subdirectory of another configuration and then reference the same grouping names.

The format of an example grouping file is as follows:

```
adminusers=root,damo,edgy,marc,ralf,db2admin,db2inst1
shell=/bin/?sh,/bin/sh,/bin/tsh
admingroup=system,bin,sys,security,audit,cron
usergroup=staff,customer
appadminusers=appadm,appmaint
```

The grouping file has the following syntax rules and can be edited directly, although the recommendation would be to use either SMIT (fast path `wlmgroupings`) or Web-based System Manager:

- ▶ Comments are preceded with an asterisk (*).
- ▶ Attribute values can be continued onto multiple lines by the use of a backslash (\).
- ▶ A carriage return signifies the end of a list.
- ▶ An attribute name cannot have an empty string of values.
- ▶ An exclusion character (!) is not allowed, although wild cards are ([,],*,-,?,+)

Use of attribute value grouping

Once defined, the grouping names can be specified in the rules file for that configuration. To show how this works, the following is an example of a rules file that does not use attribute value groupings:

```
* class resvd user group application type tag
```

```

app      -      appadm,appmaint - - - -
app      --!staff,!customer - - -
db       -      appadm,appmaint - - - -
dbp      --!staff,!customer - - -
monitor  -      -      !staff,!customer      /bin/sh,/bin/csh,/bin/tsh
-        -
System   -      root   -      -      -      -
Default  -      -      -      -      -      -

```

Groupings enable the rules file to be easier to manage, both in terms of maintenance and when referencing the file. Using the values that have been input into the grouping file for this configuration, the rules file can be shown as follows:

```

* class resvd user group application type tag
app      -      $appadminusers  !$usergroup -      -      -
db       -      $adminusers     !$usergroup -      -      -
monitor  -      -      !$usergroup  $shell -      -
System   -      root   -      -      -      -
Default  -      -      -      -      -      -

```

The Web-based System Manager can be used to add, copy, edit, or delete attribute value groups. Select **Configurations/Classes** in the WLM submenu, as shown in the Figure 3-30.

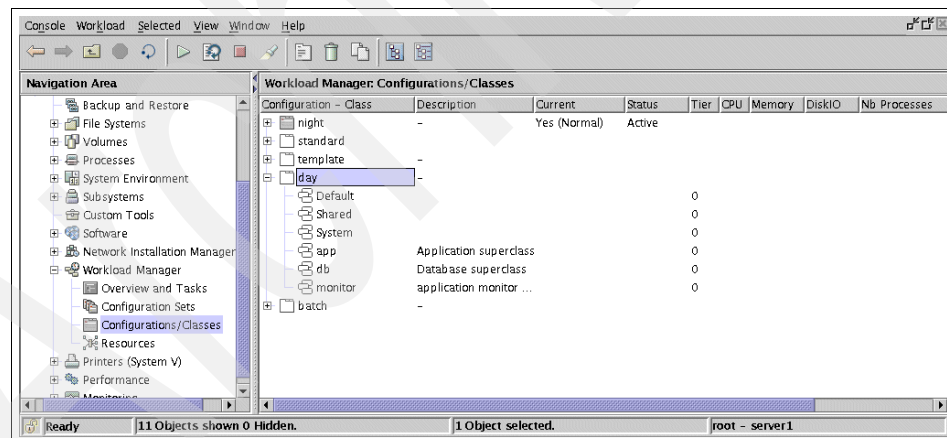


Figure 3-30 Select the configuration to add the attribute value group to

Right-click the configuration name and a menu will appear. From there click the **Attribute Value Groups** option, as shown in Figure 3-31 on page 85.

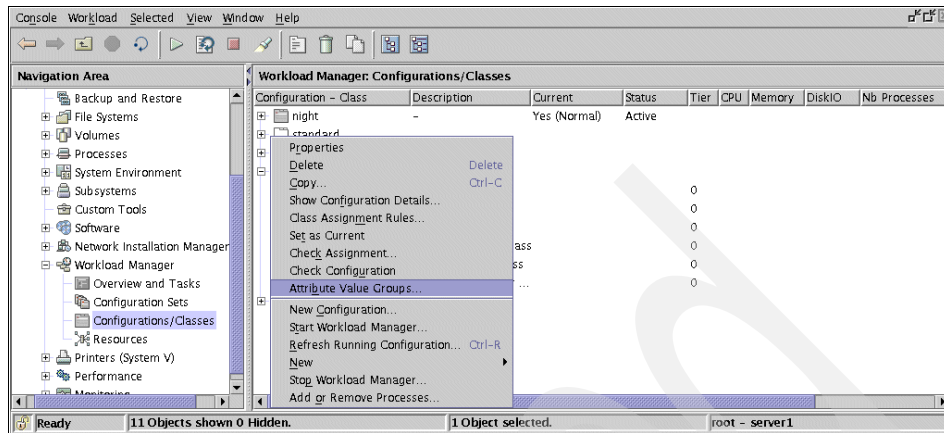


Figure 3-31 Right-click the Attribute Value Groups option

This will take the user into the initial configuration screen. To add a group, just click **New Group**, as shown in Figure 3-32.

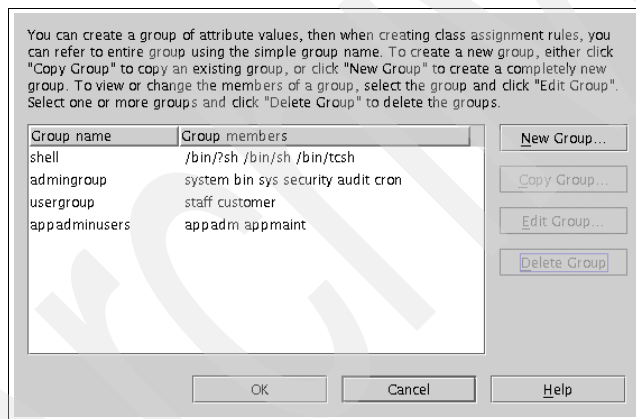


Figure 3-32 Attribute Value configuration screen

Type in the group name and also the group member and click **OK**, as shown in Figure 3-33 on page 86.

Specify the name of the new group in the top field, then list the values that you want to include in the group in the bottom field.
All members of this group must be the same type of object: for example, users.

New group name:

Group members:

Figure 3-33 Adding an attribute value group

Once created, the attribute value group is ready to be used for that configuration. If attribute value groups are to be used in more than one configuration, the most simple method to achieve this is to copy the contents of the `/etc/wlm/configuration/groupings` file to other configurations' subdirectories.

Event notification

Event notification enables the system administrator to be notified of WLM class-level related events based on configurable conditions. User-defined conditions and responses can be registered with the resource monitoring and control subsystem (RMC). The RMC then performs the defined action when a condition is met.

In previous versions, these alerts would be on a system-wide basis. Version 5.2 provides an additional level of granularity and reports alerts at the class level rather than at the host level.

Introduced with Version 5.2 is the WLM Resource Manager (WLMRM). WLMRM has been developed to allow RMC clients to monitor resources at the WLM class level and supports one resource class called IBM.WLM. Each WLM class is represented by a resource instance in this class and each resource (WLM class) can be monitored independently for one or more conditions.

WLMRM is contained in the `bos.rte.control` fileset.

Command line interface

WLMRM runs as a subsystem named IBM.WLMRM and supports the command line interface of the system resource controller. The following command can be used to view the status of the IBM.WLMRM subsystem:

```
lssrc -s IBM.WLMRM
```

WLMRM also supports the subset of the RMC command line interface that is related to querying resources and resource classes. The following command can be used to view resources in the IBM.WLM resource class:

```
lssrc IBM.WLM
```

Configuration

It is possible to define the new conditions to be monitored with the Web-based System Manager. If monitoring is selected from the left-hand menu submenu conditions, it is possible to select a new condition from the Conditions drop-down menu, as shown in Figure 3-34.

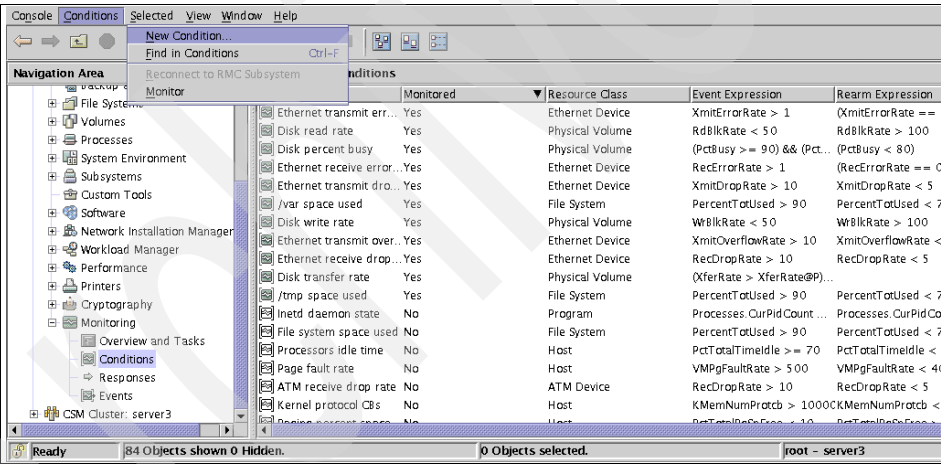


Figure 3-34 New Condition menu option for monitoring

The condition can then be configured in the New Condition box, as shown in Figure 3-35 on page 88.

The image shows a 'New condition configuration box' dialog. It has two tabs: 'General' and 'Monitored Resources'. The 'General' tab is active. The fields are as follows:

- Name:** Number of logins in class
- Management scope:** Local Machine
- Resource class:** IBM.WLM
- Monitored property:** NumLogins
- Event expression:** NumLogins > 20
- Event description:** Number of logins has exceeded 20
- Rearm expression:** NumLogins < 20
- Rearm description:** Number of logins has fallen below 20
- Severity:** Warning

Buttons include 'Details...', 'Use defaults', 'Responses to Condition...', 'OK', 'Cancel', and 'Help'.

Figure 3-35 New condition configuration box

Time-based configurations

Time based configurations provide the ability to assign a configuration to a time range. Time-based configurations are referred to as configuration sets or confsets. A confset is a collection of configurations, where each configuration is assigned to a time range.

Confsets allow the configuration to be changed depending on the expected system use at specific times of the day or days during a week. Essentially, each configuration is assigned one or more time ranges when they are active. Configurations created prior to Version 5.2 are compatible to be used within confsets.

So that partial changes are not implemented during a switch, a snapshot of all involved configurations of the set are written to `/etc/wlm/.running/.confset`. A directory for each configuration in the confset is created under this directory. The

existence of this directory indicates that this is a confset and its contents will be read by the WLM daemon. Only root users will be able to manage time ranges for the currently active configuration.

A Superclass update is allowed assuming that the user has the appropriate privileges to perform the change to the class. This will update the Superclass of the configuration of the current confset in the `/etc/wlm/.running` directory and refresh WLM if required. Once WLM is refreshed, and if the configuration is active, the changes will be immediate. Otherwise the changes will take place next time the configuration is active.

A confset includes a `.times` file, which details the time ranges and their associated configurations, together with a description file. If the configuration directory contains a `.times` file and no `classes` file, then the configuration is treated as a confset when it is loaded. When loaded into the kernel the `.times` file and all the configurations of the confset are also copied into the `/etc/wlm/.running/.confset` directory. These files are used for time range switches. WLM keeps track of time and loads the required configuration into the kernel as needed.

It is not mandatory to have time ranges to cover all times in the day, although a default configuration must be specified. The default configuration will be active during time ranges that have no other configuration specified.

New commands for time-based configurations

There are two new commands introduced to manage time based configurations, mainly for SMIT and Web-based System Manager use. They are:

► **lswlmconf**

The **lswlmconf** command shows current configuration, and lists regular WLM configurations and confsets. The syntax of the command is:

```
lswlmconf [ -r | -s | -c | -d config ] [ -l ]
```

The **lswlmconf** command is shown in the following example:

```
# lswlmconf
standard
template
day
night
batch
Normal
```

► **confsetcntrl**

The **confsetcntrl** command is used to manage the confset file. The syntax of the command follows.

To create configuration set `confset` with `defaultconfig` configuration, with default time range:

```
confsetcntrl -C confset defaultconfig
```

To delete `confset` or remove from `confset` all configurations and time ranges:

```
confsetcntrl { -D | -R } confset
```

To add or remove a time range for `config` to or from `confset` use the following. Reports warning if time ranges are not coherent:

```
confsetcntrl [ -d confset ] [ -a | -r ] config timerange
```

To lists and check all configurations and time ranges in the `confset` for their existence, syntax, and time range coherence:

```
confsetcntrl [ -d confset ] [ -l | -c ]
```

Time-based configurations can be set up both through Web-based System Manager and SMIT. For this example, Web-based System Manager has been used.

The configurations must be created before it is possible to allocate them to a `confset`. In the following example, an assumption has been made that the configurations are already defined to WLM. Figure 3-36 shows where to start from the drop-down menu, although the same options can be reached by right-clicking the configuration class.

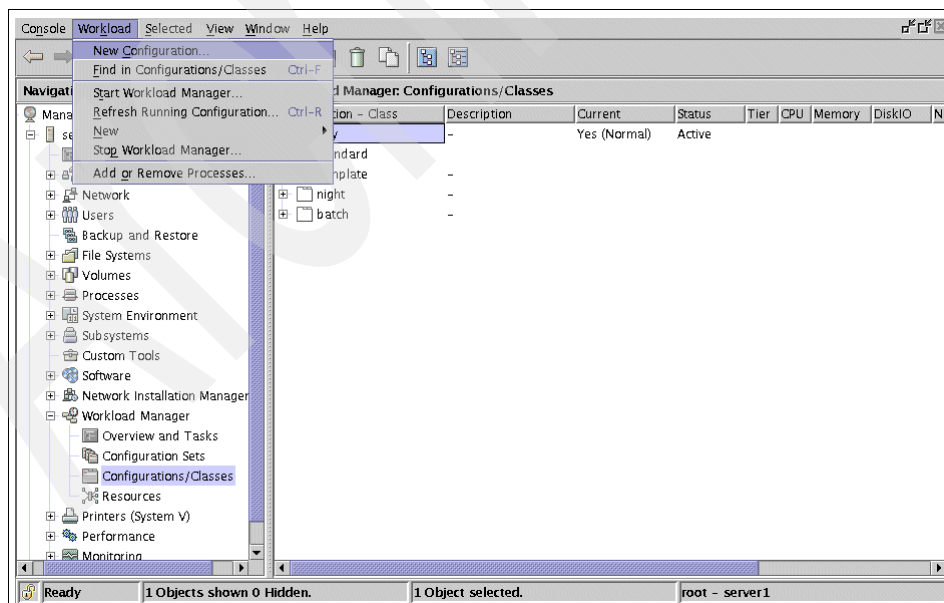


Figure 3-36 Time-based configuration drop-down menu

The configuration set must now be configured using the configuration classes that are already defined or defined in the previous set, as shown in Figure 3-37.

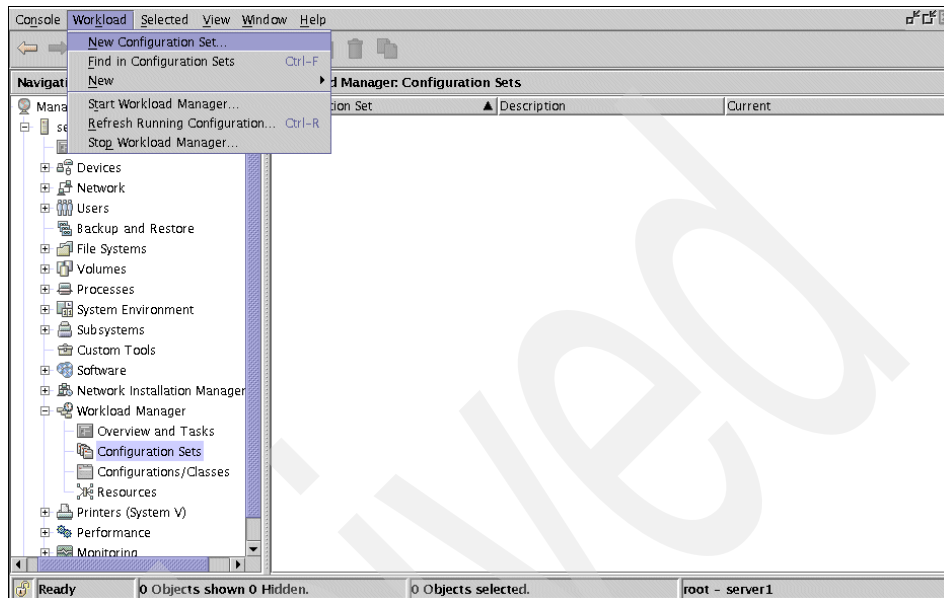


Figure 3-37 Drop-down to create the configuration set

The new configuration set is now defined and is ready for the configurations to be added to the set, as shown in Figure 3-38 on page 92.

General

Specify the name of the new set of time-based configurations, and an optional description. Then select the default configuration, which is the configuration in effect at any time not specified by other configuration definitions. Then use the Add, Remove and Change buttons to specify other configurations, and the days and times during which each configuration is in effect.

New configuration set:

Description:

Default configuration:

Configuration	Day Range	Time Range
---------------	-----------	------------

Figure 3-38 Defining the configuration set

The configurations are added to the configuration set by clicking the **Add** button on the right-hand side. This takes the user into the following screen where the configuration and the times for the configuration to run are set, as shown in Figure 3-39 on page 93.

Modify the day and time range during which the selected configuration applies.

Configuration set: Normal

Configuration: day

Day Range

☐ All week

☒ Selected days

From: Monday

To: Friday

Time Range

☐ All day

☒ Selected hours

From: Hour: 08 Minute: 00

To: Hour: 19 Minute: 00

OK Cancel Help

Figure 3-39 Selecting the configuration file and setting the times it is functional

After adding all the valid configurations a summary is provided. Note that configurations do not have to apply for every hour or day of the week. If there is no time range, the default configuration is used, as shown in Figure 3-40 on page 94.

General

To modify a set of time-based configurations, use the Add, Remove and Change buttons to specify other configurations, and the days and times during which each configuration is in effect.

Configuration Set: Normal

Description:

Default configuration: day

Configuration	Day Range	Time Range
night	Monday-Thursday	19:00-24:00
night	Tuesday-Friday	0:00-8:00
day	Monday-Friday	8:00-19:00
batch	Saturday-Sunday	All day
batch	Friday-Friday	19:00-24:00
batch	Monday-Monday	0:00-8:00

Add... Remove Change...

OK Cancel Help

Figure 3-40 Time-based configurations

All of these actions can be performed using the SMIT menus (fast path wlmconfset) assuming you are using already defined configurations. If the classes need to be defined then SMIT can be used (fast path wlmconfig).

Limits on total resources in a class

There are six new limits that can be specified at a class level. These are grouped into process total resources and class total resources.

Process total resources

Process total resources give the ability to limit the total resource consumption of a process. The process total resources include the following resource limits:

- totalCPU** Maximum CPU time limit
- totalDiskIO** Total disk I/O for a process (expressed in KB, MB, TB, PB, or EB)
- totalConnectTime** Time a login session in a class can remain active

These limits are specified at the class level but apply to each process in the class. When the limit is exceeded, the process is terminated with a SIGTERM and then a SIGKILL. These limits should only be specified on processes that should be killed when they consume excessive resource. The total limits, if used, are specified in the existing limits file. Normally resource type limits at the Subclass level are represented in percentage terms. The new resource types specified have absolute limits.

Class total resources

Class total resources give the ability to limit the number of processes, threads and login sessions at the class level. The class total resources include the following resource limits:

totalProcesses	Maximum number of processes allowed in the class
totalThreads	Maximum number of threads allowed in the class
total Logins	Total number of simultaneous logged-in user sessions

When class limits are reached for a resource, any attempt to create a new resource of that type in the class will fail. The existing limits file can be used for these new limits. These new resource types have absolute limits as opposed to limits expressed in percentage terms.

Enhanced commands for class total limits

The following commands were enhanced in Version 5.2.

► **wlmstat**

The **wlmstat** command with the -T flag displays total resource consumption values for a class. The syntax of the command is:

```
wlmstat -T
```

► **wlmcntrl**

The **wlmcntrl** command controls the state of WLM and can enable or disable it. Limits are enabled by default, if specified in the limits file, but can be disabled together with accounting using the -T flag. This is an enhancement to the **wlmcntrl** command. The syntax of the command is:

```
wlmcntrl -T [class|proc]
```

Using WLM, first ensure that the Total Limits box for the new limits is not checked. It is possible to configure this using the Web-based System Manager, by accessing the WLM section, Overview and Tasks submenu. This is illustrated in Figure 3-41 on page 96.

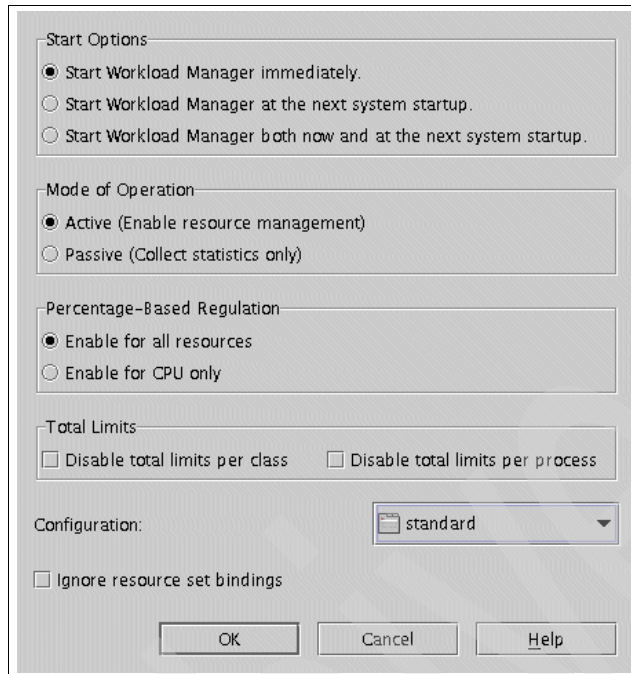


Figure 3-41 WLM Overview and Tasks submenu, Total Limits section

The new limits on total resources in a class are split into two section: Class member limits and process limits. In order to set these, select configuration classes from the WLM menu and right-click the configuration class attribute to access the pop-up menu, then select the **Properties** option, as shown in Figure 3-42 on page 97.

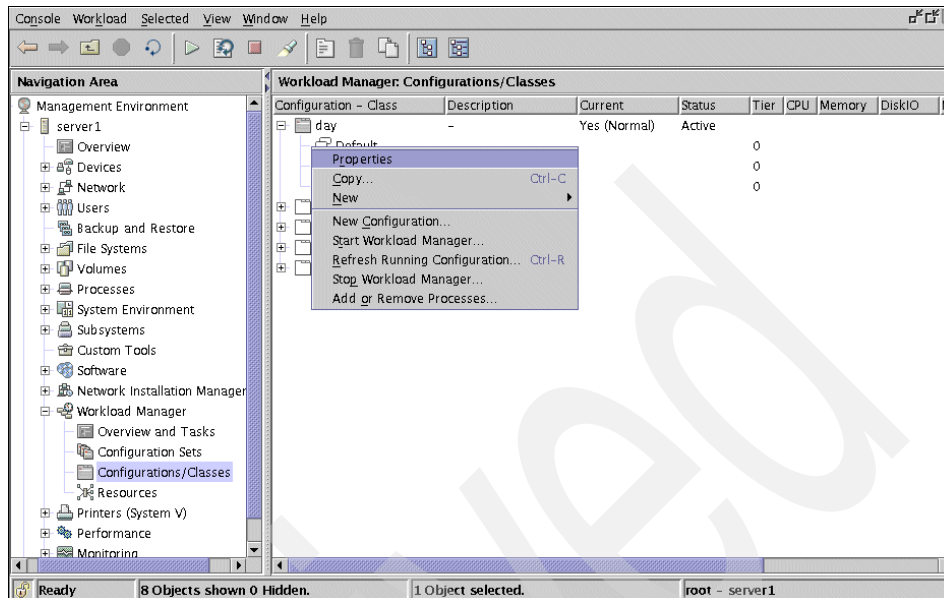
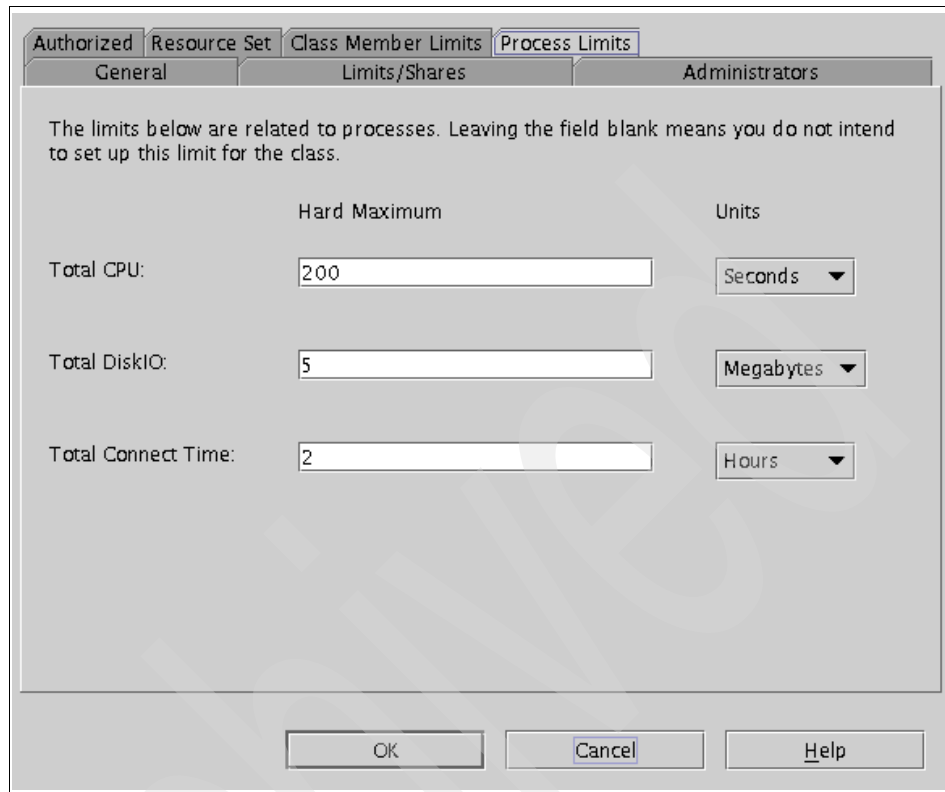


Figure 3-42 Selecting the properties of a configuration

This starts the properties menu box and from here, both the process limits and the class member limits configuration panels can be selected by clicking on the tabs, as shown in the screens following.

Once configured, the right hand side of the configurations/classes shows figures for the new limits of the classes and Subclasses that have just been configured. This is shown by simply scrolling right as shown in Figure 3-43 on page 98.



The limits below are related to processes. Leaving the field blank means you do not intend to set up this limit for the class.

	Hard Maximum	Units
Total CPU:	<input type="text" value="200"/>	<input type="text" value="Seconds"/>
Total DiskIO:	<input type="text" value="5"/>	<input type="text" value="Megabytes"/>
Total Connect Time:	<input type="text" value="2"/>	<input type="text" value="Hours"/>

OK Cancel Help

Figure 3-43 Process Limits configuration screen

Authorized Resource Set **Class Member Limits** Process Limits

General Limits/Shares Administrators

The limits below are related to classes. Leaving the field blank means you do not intend to set up this limit for the class.

Hard Maximum

Number of Processes: 15

Number of Threads: 30

Number of Logins: 3

OK Cancel Help

Figure 3-44 Class member limits

Further changes

The Overview and Tasks screen show the status of the WLM and also the current configuration.

Increase in the total limit on user-defined classes

There are two changes to user-defined class limits, one for Superclasses and one for Subclasses.

- ▶ Superclasses can now total 64, where previously the limit was 27.
- ▶ Subclasses can total 61 per Superclass, where previously the limit was 10.

3.2 Logical partitioning

LPAR stands for logical partitioning and is the ability to divide a physical server into *virtual* logical servers, each running in its own private copy of the operating system.

Though it may not seem practical, running a machine with a single LPAR, compared to full system partition mode (non-LPAR), provides for a faster system restart because the hypervisor has already provided some initialization, testing, and building of device trees. In environments where restart time is critical, it we recommend that you test the single LPAR scenario to see if it meets the system recycle time objectives.

Depending on the software installed on the server, dynamic LPAR may be available or unavailable:

Dynamic LPAR available

With dynamic LPAR available, the resources can be exchanged between partitions without stopping and rebooting the affected partitions. Dynamic LPAR requires AIX 5L Version 5.2 for all affected partitions, and the HMC recovery software must be at Release 3 Version 1 (or higher). In partitions running AIX 5L Version 5.1 or Linux, if available, the Dynamic Logical Partitioning menu is not available.

Dynamic LPAR unavailable

Without dynamic LPAR, the resources in the partitions are static. Dynamic LPAR is unavailable for partitions running AIX 5L Version 5.1 or Linux, when available. When you change or reconfigure your resource without dynamic LPAR, all the affected partitions must be stopped and rebooted in order to make resource changes effective.

A server can contain a mix of partitions that support dynamic LPAR along with those that do not.

Note: Rebooting a running partition only restarts the operating system and does not restart the LPAR. To restart an LPAR, the operating system should be shut down without reboot and afterwards restarted again.

3.2.1 Hardware Management Console (HMC)

With LPAR mode, an IBM Hardware Management Console for pSeries (HMC) is necessary. Either a dedicated 7315-C01 or an existing HMC from a p670 or p690 installation (FC 7316) can be used. If a server is used in full system partition mode (no LPARs) outside a cluster, an HMC is not required.

The HMC is a dedicated desktop workstation that provides a graphical user interface for configuring and operating pSeries servers functioning in either non-partitioned, LPAR, or clustered environments. It is configured with a set of hardware management applications for configuring and partitioning the server. One HMC is capable of controlling multiple pSeries servers. At the time of writing, a maximum of 16 non-clustered pSeries servers and a maximum of 64 LPARs are supported by one HMC.

The HMC is connected with special attachment cables to the HMC ports of the hardware. Only one serial connection to a server is necessary despite the number of LPARs.

With these cables, the maximum length from any server to the HMC is limited to 15 meters. To extend this distance, a number of possibilities are available:

- ▶ Another HMC could be used for remote access. This remote HMC must have a network connection to the HMC that is connected to the servers.
- ▶ AIX 5L Web-based System Manager Client could be used to connect to the HMC over the network or the Web-based System Manager PC client could be used, which runs on a Windows operating system-based or Linux operating system-based system.
- ▶ When a 128-Port Async Controller is used, the RS-422 cables connect to a RAN breakout box, which can be up to 330 meters. The breakout box is connected to the HMC port on the server using the attachment cable. When the 15 meter cable is used, the maximum distance the HMC can be is 345 meters, providing the entire cable length can be used.

The HMC provides a set of functions that are necessary to manage LPAR configurations. These functions include:

- ▶ Creating and storing LPAR profiles that define the processor, memory, and I/O resources allocated to an individual partition.
- ▶ Starting, stopping, and resetting a system partition.
- ▶ Booting a partition or system by selecting a profile.

- ▶ Displaying system and partition status.

In a non-partitionable system, the LED codes are displayed in the operator panel. In a partitioned system, the operator panel shows the word LPAR instead of any partition LED codes. Therefore all LED codes for system partitions are displayed over the HMC.

- ▶ Virtual console for each partition or controlled system.

With this feature, every LPAR can be accessed over the serial HMC connection to the server. This is a convenient feature when the LPAR is not reachable across the network or a remote NIM installation should be performed.

The HMC also provides a service focal point for the systems it controls. It is connected to the service processor of the system using the dedicated serial link. The HMC provides tools for problem determination and service support, such as call-home and error log notification through an analog phone line.

3.2.2 LPAR minimum requirements

Each LPAR must have a set of resources available. The minimum resources that are needed are the following:

- ▶ At least one processor per partition.
- ▶ At least 256 MB of main memory.
- ▶ At least one disk to store the operating system (for AIX, the rootvg).
- ▶ At least one disk adapter or integrated adapter to access the disk.
- ▶ At least one LAN adapter per partition to connect to the HMC.
- ▶ A partition must have an installation method, such as NIM, and a means of running diagnostics, such as network diagnostics.

3.2.3 Memory guidelines for LPAR

There are a few limitations that should be considered when planning for LPAR, as discussed in the following.

Memory

Planning the memory for logical partitioning involves additional considerations. These considerations are different when using AIX 5L Version 5.1, AIX 5L Version 5.2, or Linux.

When a machine is in full system partition mode (no LPARs) all of the memory is dedicated to AIX. When a machine is in LPAR mode, some of the memory used

by AIX is relocated outside the AIX-defined memory range. In the case of a single small partition on a p630 (256 MB), the first 256 MB of memory will be allocated to the hypervisor, 256 MB is allocated to translation control entries (TCEs) and to hypervisor per partition page tables, and 256 MB for the first page table for the first partition. TCE memory is used to translate the I/O addresses to system memory addresses. Additional small page tables for additional small partitions will fit in the page table block. Therefore, the memory allocated independently of AIX to create a single 256 MB partition is 768 MB (0.75 GB).

With the previous memory statements in mind, LPAR requires at least 2 GB of memory for two or more LPARs on a p630. It is possible to create a single 256 MB LPAR partition on a 1 GB machine; however, this configuration should be used for validation of minimum configuration environments for test purposes only. Other systems have different memory requirements.

You must close any ISA or IDE device before any dynamic LPAR memory is removed from the partition that owns the ISA or IDE I/O. This includes the diskette drive, serial ports, CD-ROM, or DVD-ROM, for example.

The following rules only apply for partitions with AIX 5L:

- ▶ The minimum memory for an LPAR is 256 MB. Additional memory can be configured in increments of 256 MB.
- ▶ The memory consumed outside AIX is from 0.75 GB up to 2 GB, depending on the amount of memory and the number of LPARs.
- ▶ For AIX 5L Version 5.1, the number of LPARs larger than 16 GB is limited to two in a system with 64 GB of installed memory, because of the memory alignment in AIX 5L Version 5.1.

LPARs that are larger than 16 GB are aligned on a 16 GB boundary. Because the hypervisor memory resides on the lower end of the memory and TCE resides on the upper end of the memory, there are only two 16 GB boundaries available.

The organization of the memory in a server must also be taken into account. Every processor card has its dedicated memory range. Processor card 1 has the range 0–16 GB, processor card 2 has the range 16–32 GB, processor card 3 32–48, and processor card 4 48–64 GB. If a processor card is not equipped with the maximum possible memory, there will be holes and the necessary 16 GB contiguous memory will not be present in the system. For example, in a system with three processor cards and 36 GB of memory, the memory is distributed into the ranges 0–12, 16–28, and 32–50. In this configuration, the only available 16 GB boundary (at 16 GB) has only 12 GB of memory, which is too small for a partition with more than 16 GB of memory and AIX 5L Version 5.1.

- ▶ With AIX 5L Version 5.2, there are no predefined limits concerning partitions larger than 16 GB, but the total amount of memory and hypervisor overhead remains a practical limit.

Note: To create LPARs running AIX 5L Version 5.2 or Linux larger than 16 GB, the checkbox **Small Real Mode Address Region** must be checked (on the HMC, LPAR Profile, Memory Options dialog). Do not select this box if you are running AIX 5L Version 5.1.

3.2.4 Dynamic LPAR (5.2.0)

With the availability of the IBM @server pSeries 690 server in December 2001, static logical partitioning (LPAR) was introduced to the pSeries platform. While LPAR provides a solution to logically remove and assign resources from one partition to another, the operating system in all affected partitions has to be rebooted, and the partitions have to be reset.

Dynamic LPAR (DLPAR) on IBM's @server pSeries servers enables the movement of hardware resources (such as processors, memory, and I/O slots) from one logical partition running an operating system instance to another partition without requiring reboots and resets.

With DLPAR technology the following features are enabled: Dynamic reconfiguration, Dynamic Capacity Upgrade on Demand (DCUoD), and CPU sparing.

As shown in the system architecture in Figure 3-45 on page 105, a DLPAR system is made up of several components. To provide the foundation for DLPAR, the following components were made DLPAR aware:

- ▶ HMC
- ▶ Hypervisor
- ▶ Global-Firmware
- ▶ Local-Firmware
- ▶ AIX

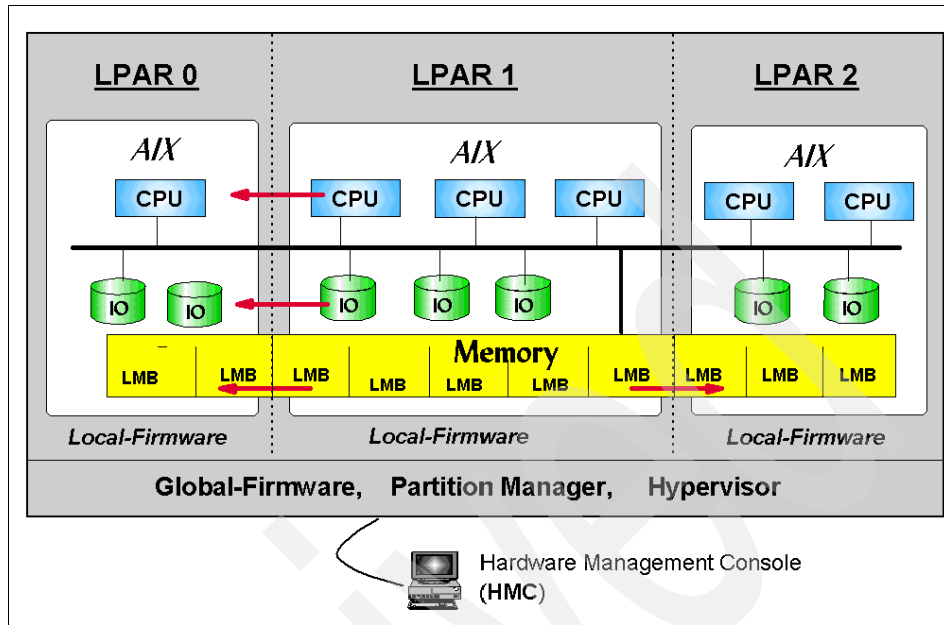


Figure 3-45 IBM eServer pSeries DLPAR system architecture

In this chapter, AIX as a component of the DLPAR environment and the implications of DLPAR on applications are described.

“DLPAR architecture (5.2.0)” on page 105 an introduction of the DLPAR architecture and how the components interact is given.

In “Introduction to AIX DLPAR Framework” on page 108 an introduction to the DLPAR Framework of AIX is given. The process of a dynamic reconfiguration is explained.

In 3.2.5, “Using the AIX DLPAR Framework” on page 113 the DLPAR application framework is described. The application framework allows applications and kernel extensions to be notified of DLPAR events so that they take appropriate action. Furthermore, methods to monitor DLPAR events are described.

DLPAR architecture (5.2.0)

Figure 3-46 on page 107 shows how DLPAR-aware components interact in an example where a user on the HMC initiates the movement of a resource from one partition to another.

A description of the involved components is provided as follows:

HMC	The Hardware Management Console (HMC) is the command center from which all decisions related to the movement of resources are made.
chhwres	The chhwres HMC command is where commands are issued to dynamically add and remove resources from partitions as well as move resources between partitions. This command can be issue using the HMC GUI or from the command line.
DRM	The Dynamic Reconfiguration Manager (DRM) is an agent that is designed to deal with DLPAR-specific issues. DRM invokes AIX commands to attach or detach DLPAR capable resources.
RMC	The Remote Monitoring and Control (RMC) handles monitoring and controlling distributed resource classes. It is a distributed framework that is designed to handle all security and connectivity issues related to networks. In conjunction with DRM, it enables the remote execution of commands to drive the configuration and unconfiguration of DLPAR-enabled resources.
RTAS	The Run-Time Abstraction Services (RTAS) is firmware that is replicated in each partition. It operates on objects in the Open Firmware Device Tree such as processors, logical memory blocks (LMBs), I/O slots, date chips, and NVRAM. Operations include query, allocate, electronically isolate, and free resources.
Global FW	One global firmware (FW) instance spanning the entire system. The global firmware is also known as the hypervisor. It contains the boot and partition manager, manages memory and I/O mappings, and provides a global name space for resources. It dictates the set of DLPAR-enabled resources and contains the Open Firmware device tree. AIX communicates with it through the RTAS layer.

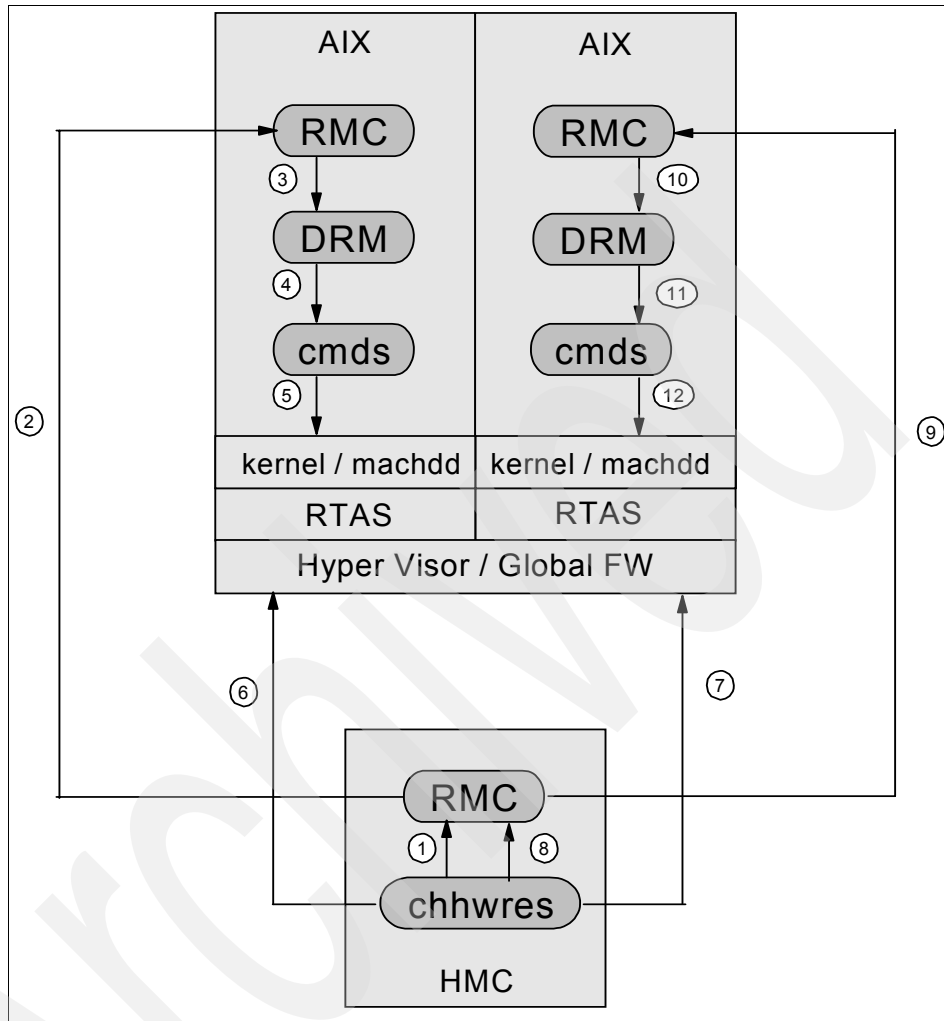


Figure 3-46 DLPAR system architecture

The sequence of operations for the given example as provided in Figure 3-46 is explained in the following:

1. The **chhwres** command on the HMC calls the RMC with the request to release the given resource.
2. RMC establishes a connection through the Ethernet network to the RMC on AIX and passes the request to release the resource. The RMC connection to the partition is established at boot time.
3. RMC then calls DRM with the request to release the resource.

4. DRM initiates the appropriate AIX commands to release the resource from the operating system (OS).
5. The AIX commands invoke the appropriate functions of the kernel. The OS attempts to stop using the specified resource. If it cannot stop using the resource, an error is returned to the user. If it can stop using the resource, the OS isolates the resource, powers it off and sets the status to unusable. Success is reported to the **chhwres** command on the HMC.
6. The **chhwres** command calls the global firmware and reclaims the resource.
7. The **chhwres** command calls the global firmware and assigns the resource to the partition.
8. The **chhwres** command calls RMC with the request to configure the resource.
9. RMC establishes a connection using the network to the RMC on the partition and passes on the request. The RMC connection is established at boot time.
10. RMC calls DRM with the configuration request.
11. The DRM calls the appropriate AIX commands with the request to add the resource to the operating system.
12. The AIX command initiates the appropriate OS functions and the OS attempts to make the specified resource usable using RTAS calls. If this operation is unsuccessful, an error is returned to the user. If the operation is successful, the OS takes ownership of the resource and firmware removes it from its resource pool. Then the resource is powered on, unisolated, and finally configured by the OS.

Introduction to AIX DLPAR Framework

This section describes the AIX DLPAR Framework support of the DLPAR architecture.

The RMC architecture provides a common abstraction for every resource in the system. This abstraction allows resources to be managed generically. Resources are represented through the definition of resource classes and are controlled through resource managers (the DRM). These are included in the `devices.chrp.base.dr` fileset.

As described in the previous example, the RMC-DRM is able to invoke a remote AIX command in a specific partition as a function of the HMC, and to receive the return code from this command. AIX provides a single DLPAR command (**drmgr**), through which all dynamic reconfiguration requests are funneled. The **drmgr** command should not be invoked directly from the AIX operating system prompt for DR operations. Doing this could result in inconsistent system behavior. However, the **drmgr** command can be used by the system root to configure and set up the DR framework for the applications as described in the next sections.

Time considerations

Time is an important factor for DLPAR operations, because a DLPAR operation could be quite lengthy. For example, it may take several minutes to reconfigure a large database so that it uses less memory. The amount of time that the system takes to perform a DLPAR operation depends on the size of the request and the state of the affected resources in the partition. In general, a CPU can be removed in time measured in seconds and 1 GB of memory can be removed in time measured in minutes.

To control time overruns, two time-out values are provided, which have to be considered in a DLPAR operation:

- ▶ The time limit for the overall operation
- ▶ The amount of time allotted for application reconfiguration

The overall timeout is set by the user at the HMC, which, by default, is set to a value of zero. A value of zero means that the operating system should take as long as it needs to complete the request without timing out. If a non-zero value is specified by the user, then the operating system stops reconfiguring resources at the appointed time; however, it may continue to call scripts and invoke signals to maintain a consistent application and operating system state. If a request times out, the resources are not automatically rolled back to the pre-request state and the user is notified that the command was partially completed.

Considering the time-out value for applications, you must distinguish between the two forms of application notification. The script-based mechanism (“DLPAR scripts” on page 116) is invoked synchronously, so the **drmgr** command that calls the scripts will wait either until the scripts have finished or up to the defined time-out. The default time-out value is 10 seconds. However, this value can be overwritten by the script vendor. This value can again be overwritten by the user that installs the script using the -w flag with the **drmgr** command.

The API-based handlers are called asynchronously. The caller always waits until the time of the time-out value is over, whether the handler has completed earlier or not at all. The default of this time-out value is 10 seconds also and cannot be explicitly overwritten. However, the time-out value scales with the overall time-out value, so if the overall time-out value is increased, the time-out value of the API-based handlers increases with it.

Note that the default time-out values are subject to change.

DLPAR flow for CPUs and memory

As described previously, the **drmgr** command handles all dynamic reconfiguration operations by calling the appropriate commands, and controls the process of the reconfiguration of resources.

The flow of the dynamic reconfiguration is generic and is described as follows:

1. The ODM lock is taken to guarantee that the ODM, Open Firmware (OF) device tree, and the kernel are atomically updated. This step can fail if the ODM lock is held for a long time and the user indicates that the DLPAR operation should have a time limit.
2. The dynamic reconfiguration command reads the OF device tree.
3. The dynamic reconfiguration command invokes the kernel to start the DR operation. The following steps are taken:
 - a. Requesting validation
 - b. Locking DR operation—only one can proceed at a time
 - c. Saving request in global kernel DR structure, which is used to pass information to signal handlers, which runs asynchronously to the DR command
 - d. Starting check phase
4. *Check* phase scripts are invoked.
5. Check phase signals are sent—conditional wait if signals were posted.
6. Check phase kernel extension callout. Callback routines of registered kernel extensions are called.

Note: The operation may fail in steps 4, 5, or 6 if any check phase handler signals an error. Once the *check* phase has passed without an error and the LPAR operation is in the *pre* phase, all pre phase application handlers will be called, even if they fail, and the dynamic reconfiguration is attempted.

7. The kernel marks the start of the pre phase.
8. Pre phase scripts are invoked.
9. Pre phase signals are sent—conditional wait if signals were posted.
10. The kernel marks *doit* phase start. This is an internal phase where the resource is either added or removed from the kernel.

Note: Steps 11–13 may be repeated depending on the request. Processor-based requests never loop; only one processor can be added or removed at a time in one DLPAR operation. If more than one processor needs to be added or removed, the HMC invokes AIX once for each processor.

Memory-based requests loop at the logical memory block (LMB) level, which represents 256 MB segments of memory, until the entire user request has been satisfied. The HMC remotely invokes AIX once for the complete memory request.

11. This step is only taken if adding a resource. The OF device tree is updated. The resource allocated, unisolated, and the connector configured. When unisolating the resource, it is assigned to the partition and ownership is transferred from FW to AIX.
 - For processors, the identity of the global and local interrupt server is discovered.
 - For memory, the physical address and size is discovered.
12. Invoke kernel to add or remove resource.
 - a. Callback functions of registered kernel extensions are called. Kernel extensions are told the specific resource that is being removed or added.
 - b. Resources in kernel are removed or added.
 - c. Kernel extension in post or posterr phase are invoked.If steps a or b fail, then the operation fails.
13. This step is only taken if removing a resource.

The OF is updated. Resources are isolated and unallocated for removal.

The OF device tree must be kept updated so that the config methods can determine the set of resources that are actually configured and owned by the OS.
14. Kernel marks post (or posterror) phase start, depending on the success of the previous steps.
15. Invoke configuration methods so that DR-aware applications and scripts will see state change in the ODM.
16. The post scripts are invoked.
17. The post signals are sent to registered processes—conditional wait if signals were posted.
18. The kernel clears the dynamic reconfiguration event.

19.ODM locks are unlocked.

In the following section a description of the changes made to AIX 5L Version 5.2 to support dynamic removal and addition of I/O slots is provided.

Dynamic I/O removal and addition

Dynamic removal and addition of I/O adapters has been provided by AIX prior to the dynamic reconfiguration support of processors and memory to utilize the hot plug capability of IBM RS/6000 and IBM @server pSeries systems.

To allow for the addition and removal of PCI slots and of integrated I/O devices of DLPAR systems such as the p690, p670 and p630, enhancements to the **lsslot** command have been made.

PCI slots and integrated I/O devices can be listed using the new connector type *slot* in the **lsslot** command, as shown in the following example:

```
lsslot -c slot
```

The output of this command looks similar to the following:

# Slot	Description	Device(s)
U1.5-P1-I1	DLPAR slot	pci13 ent0
U1.5-P1-I2	DLPAR slot	pci14 ent1
U1.5-P1-I3	DLPAR slot	pci15
U1.5-P1-I4	DLPAR slot	pci16
U1.5-P1-I5	DLPAR slot	pci17 ent2
U1.5-P1/Z1	DLPAR slot	pci18 scsi0

Before the slot can be removed though, the PCI device and all its children need to be deleted. Given that ent2 in the slot U1.5-P1-I5 in the previous example is not used, the devices could be removed using the following command:

```
rmdev -l pci17 -d -R
```

After the devices has been removed from AIX as described previously, the slot can be removed from the partition using the HMC GUI or command line interface. The GUI is shown in Figure 3-47 on page 113. Note that the slot must not be defined as *required* in the partition profile but only as *desired*, or the option to remove this slot on the HMC will not be given.

Choose a dynamic partition task below, then specify the adapter(s) and other required information for the specific task.

Tasks

☐ Move resource to a partition.

☐ Add resource to this partition.

☒ Remove resource from this partition.

Current adapter usage:

Drawer	Slot	Type
7040-61D*024458A-P1	Slot5/U1.5-P1-I5	Empty

Detail level: 0 ▼

Timeout setting (mins): 0 ▲▼

OK Close Help ?

Figure 3-47 HMC slot removal

To add the previously removed slot to the system, it needs to be added to the system using the HMC again first. Then the devices should be configured in the slot using the `cfgmgr` command.

3.2.5 Using the AIX DLPAR Framework

Prior to DLPAR, applications considered CPU and memory to be constant resources on a system. With DLPAR the number of CPUs and the amount of memory can change during the runtime of the applications.

Most applications are not aware of the number of CPUs and the memory in the system and are therefore most likely not affected by DLPAR operations. However, some applications are aware of the amount of these system resources, and they need to handle changes to the system configuration.

There are two types of applications with respect to DLPAR operations: DLPAR-safe and DLPAR-aware applications.

A DLPAR-safe application is one that does not fail as a result of a DLPAR operation. It may not be affected at all. Its performance may suffer or it may not scale with the addition of new resources. It may even prevent a DLPAR operation from succeeding, but it functions as expected.

A DLPAR-aware application is an application that adjusts its use of system resources in order to facilitate DLPAR operations. To participate in DLPAR operations, the application may either regularly poll the system topology to discover changes or it can register with the DLPAR application framework to receive notification of DLPAR events when they occur. The latter (registration) should be the preferred choice. The polling model should not be used if the application has a processor dependency, since it may need to unbind before the operating system attempts to reconfigure the resource and the polling model only provides notification after the DLPAR event.

Types of applications that should be made DLPAR aware are listed as follows:

- ▶ Enterprise level databases, because they scale with the system configuration. They typically use large pinned buffer pools that scale with the physical memory and the amount of threads scales with the number of CPUs.
- ▶ System tools (performance monitors, for example), because they report CPU and memory statistics.
- ▶ Multi-system level job schedulers, because they schedule jobs based on the number of CPUs and memory.
- ▶ License managers, because they license on a CPU basis.

DLPAR operations are non-destructive by design. That means DLPAR operations will fail if the resource to be removed is locked by applications or the kernel. A DLPAR CPU remove request will fail if an application is bound to the CPU being removed. This could be a **bindprocessor** command or WLM rset type binding. A DLPAR memory remove request will fail if most of the memory in the system is pinned. AIX has the capability to dynamically migrate pinned memory so that virtually any range of memory can be removed. However, if the system cannot acquire a new pinned page, the operation will fail. AIX allows approximately 80 percent of the system to be pinned. Therefore, programs that consume lots of pinned memory should be made DLPAR aware so that the system will have adequate resource to perform memory removal. Applications pin memory through the `plock()` and `shmget(SHM_PIN)` system calls.

Two interfaces are available to make an application DLPAR aware, a script-based and an API-based interface. Using the script-based approach, the administrator or software vendor installs a set of scripts that are called by the DLPAR application framework when a DLPAR event occurs. For the API-based approach, the new signal `SIGRECONFIG` is defined, which is sent during DLPAR events to all processes that are registered to catch this event.

Note that the SIGRECONFIG signal is also sent (along with the SIGCPUFAIL signal for backward compatibility) in the case of a CPU Guard event. Therefore the DLPAR application framework can also be utilized by CPU Guard aware applications.

In the first release of DLPAR support, the dynamic reconfiguration of I/O slots is not integrated into the DLPAR Framework in the same way that CPUs and memory is. The user cannot install DLPAR scripts or make their applications DLPAR aware by registering for a signal.

DLPAR operation phases

The DLPAR operation phases are independent of whether the approach is script- or API-based. Every DLPAR operation is divided into three phases:

- ▶ Check phase
- ▶ Pre phase
- ▶ Post phase

The check and pre phases occur before the actual dynamic reconfiguration is performed, whereas the post phase occurs after the dynamic reconfiguration is done. This process is shown in Figure 3-48.

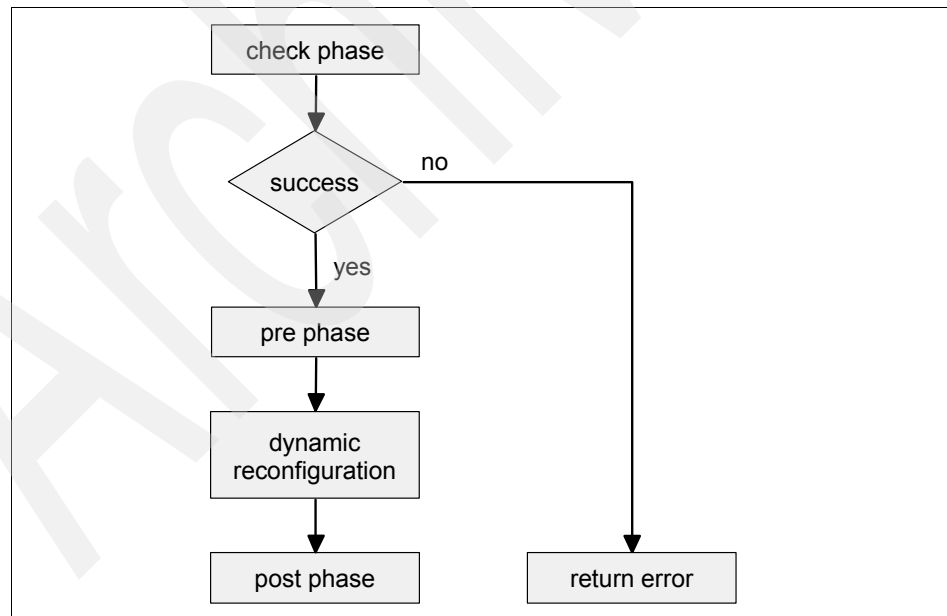


Figure 3-48 DLPAR operation phases

In the *check* phase the handler (script or signal) is called and requested to approve the DLPAR operation. If any handler declines this request, the operation fails before any changes to the system are done. This would be the opportunity for a non-DLPAR safe application to terminate the DLPAR operation, because it would fail after the DLPAR operation. Or a license manager could decline a CPU add request, because there are not enough CPU-based licenses purchased.

In the *pre* phase, the registered handlers are notified that the dynamic reconfiguration is about to occur. This is where the appropriate actions should be taken, to allow for a successful DLPAR operation. This will typically include tasks such as unbinding from CPUs or releasing pinned memory. A handler may still return an error, because he was not able to release the pinned memory for example, but all application handlers will be called anyway and the dynamic reconfiguration performed.

After the *pre* phase, the dynamic reconfiguration procedure is performed. The dynamic reconfiguration could fail for one of the reasons given earlier.

In the *post* phase, the registered handlers are notified that the dynamic reconfiguration has completed. Depending on whether the dynamic reconfiguration was successful, the handler can undo the changes done in the *pre* phase or adapt to the new system environment.

In the following an introduction to the script-based interface is given.

DLPAR scripts

As mentioned previously, DLPAR scripts are written by system administrators or software vendors. Scripts can be implemented in any scripting language such as Perl, shell, or it can be a compiled program. They are maintained by the system administrator using the **drmgr** command. The syntax of the command is as follows:

```
drmgr { -i script_name [-w minutes ] [ -f ] | -u script_name } [ -D hostname ]
drmgr [ -b ]
drmgr [ -R script_install_root_directory ]
drmgr [ -S syslog_ID ]
drmgr [ -l ]
```

A description of the most important flags for the **drmgr** command are provided in Table 3-3 on page 117. For a complete reference, refer to the man page or the documentation.

Table 3-3 The *drmgr* command flags

Flags	Description
-i <i>script_name</i>	This flag is used to install a script specified by the <i>script_name</i> parameter. By default scripts are installed to the <i>/usr/lib/dr/scripts/all</i> directory.
-w <i>minutes</i>	This flag is used to override the time limit value specified by the vendor for the script. The script will be ended if it exceeds the specified time limit.
-f	Using this flag forces an installed script to be overwritten.
-u <i>script_name</i>	This flag is used to uninstall a script specified by the <i>script_name</i> parameter.
-l	This option will display the details regarding the DLPAR scripts that are currently installed.

For example, to install the script */root/root_dlp_test.sh* in the default directory the following command could be used:

```
drmgr -i /root/root_dlp_test.sh
```

To list the details the **drmgr -l** command is used. The output is similar to the following:

```
DR Install Root Directory: /usr/lib/dr/scripts
Syslog ID: DRMGR
-----
/usr/lib/dr/scripts/all/root_lpar_test.sh          DLPAR test script
Vendor:IBM,      Version:1.0,      Date:19092002
Script Timeout:10,      Admin Override Timeout:0
Resources Supported:
      Resource Name: cpu      Resource Usage: root_dlp_test.sh
command [parameter]
-----
```

DLPAR scripts get notified at each of the DLPAR operation phases explained previously. Notifying DLPAR scripts involves invoking the scripts in the appropriate environment with the appropriate parameters.

The environment the script is executed in is as follows:

- ▶ Execution user ID and group ID are set to uid or gid of the script.
- ▶ The PATH environment is set to */usr/bin:/etc:/usr/sbin*.
- ▶ The working directory is */tmp*.
- ▶ Environment variables to describe the DLPAR event are set.

DLPAR scripts can write any necessary output to stdout. The format of the output should be name=value pair strings separated by newline characters to relay specific information to the **drmgr**. For example, the output DR_VERSION=1.0 could be produced with the following **ksh** command:

```
echo "DR_VERSION=1.0"
```

Error and logging messages are provided by DLPAR scripts in the same way as regular output by writing name=value pairs to stdout. The DR_ERROR=message pair should be used to provide error descriptions. The name=value pairs in Table 3-4 contain information to be used to provide error and debug output for the syslog.

Table 3-4 DLPAR script error and logging

name=value pair	Description
DR_LOG_ERR=message	Logs the message with the syslog level of the LOG_ERR environment variable.
DR_LOG_WARNING=message	Logs the message with the syslog level of the LOG_WARNING environment variable.
DR_LOG_INFO=message	Logs the message with the syslog level of the LOG_INFO environment variable.
DR_LOG_EMERG=message	Logs the message with the syslog level of the LOG_EMERG environment variable.
DR_LOG_DEBUG=message	Logs the message with the syslog level of the LOG_DEBUG environment variable.

DLPAR scripts can also write additional information to stdout that will be reflected to the HMC. The level of information that should be provided is based on the detail level passed to the script in the DR_DETAIL_LEVEL=N environment variable. N must be in the range of 0 to 5, where the default value of 0 signifies no information. A value of 1 is reserved for the operating system and is used to present the high-level flow. The remaining levels (2–5) can be used by the scripts to provide information with the assumption that larger numbers provide greater detail.

The syntax the DLPAR script is invoked with follows:

```
[ input_name1=value1 ... ] scriptname command [ input_parameter1 ... ]
```

Input variables are set as environment variables on the command line, followed by the script to be invoked that is provided with a command and with further parameters. A description of the function the commands should perform is provided in Table 3-5 on page 119. If the script is called with a command that is not implemented it should exit with a return code of 10.

Table 3-5 DLPAR script commands

Command and parameter	Description
scriptinfo	Identifies the version, date, and vendor of the script. It is called when the script is installed.
register	Identifies the resources managed by the script. If the script returns the resource name (cpu or mem), the script will be automatically invoked when DLPAR attempts to reconfigure processors and memory, respectively. The register command is called when the script is installed with the DLPAR subsystem.
usage <i>resource_name</i>	Returns information describing how the resource is being used by the application. The description should be relevant so that the user can determine whether to install or uninstall the script. It should identify the software capabilities of the application that are impacted. The usage command is called for each resource that was identified by the register command.
checkrelease <i>resource_name</i>	Indicates whether the DLPAR subsystem should continue with the removal of the named resource. A script might indicate that the resource should not be removed if the application is not DLPAR aware and the application is considered critical to the operation of the system.
prerelease <i>resource_name</i>	Reconfigures, suspends, or terminates the application so that its hold on the named resource is released.
postrelease <i>resource_name</i>	Reconfigures, resumes, or restarts the application.
undoprerelease <i>resource_name</i>	Invoked if an error is encountered and the resource is not released. Operations done in the prerelease command should be undone.
checkacquire <i>resource_name</i>	Indicates whether the DLPAR subsystem should proceed with the resource addition. It might be used by a license manager to prevent the addition of a new resource, for example, cpu, until the resource is licensed.
preacquire <i>resource_name</i>	Used to prepare for a resource addition.

Command and parameter	Description
undopreacquire <i>resource_name</i>	Invoked if an error is encountered in the preacquire phase or when the event is acted upon. Operations performed with the preacquire command should be undone.
postacquire <i>resource_name</i>	Reconfigure, resume, or start the application.

The input variables that are provided as environment variables are dependent on the resource that is operated on. For memory add and remove operations, the variables provided in Table 3-6 are provided (one frame is equal to 4 KB).

Table 3-6 Input variables for memory add/remove operations

Input variable	Description
DR_FREE_FRAMES=0xFFFFFFFF	The number of free frames currently in the system, in hexadecimal format.
DR_MEM_SIZE_COMPLETED= <i>n</i>	The number of megabytes that were successfully added or removed, in decimal format.
DR_MEM_SIZE_REQUEST= <i>n</i>	The size of the memory request in megabytes, in decimal format.
DR_PINNABLE_FRAMES=0xFFFFFFFF	The total number of pinnable frames currently in the system, in hexadecimal format. This parameter provides valuable information when removing memory in that it can be used to determine when the system is approaching the limit of pinnable memory, which is the primary cause of failure for memory remove requests.
DR_TOTAL_FRAMES=0xFFFFFFFF	The total number of frames currently in the system, in hexadecimal format.

The environment variables provided in Table 3-7 on page 121 are set for processor add and remove operations.

Table 3-7 Input variables for processor add/remove operations

Input variable	Description
DR_BCPUID=N	The bind CPU ID of the processor that is being added or removed in decimal format. A bindprocessor attachment to this processor does not necessarily mean that the attachment has to be undone. This is only true if it is the Nth processor in the system, because the Nth processor position is the one that is always removed in a CPU remove operation. Bind IDs are consecutive in nature, ranging from 0 to N and are intended to identify only online processors. Use the bindprocessor command to determine the number of online CPUs.
DR_LCPUID=N	The logical CPU ID of the processor that is being added or removed in decimal format.

In the following, an example Korn shell script is given that can be installed. For simplicity and demonstration purposes this script does not take any action. The actions for the process to control would need to be included in the appropriate command section:

```
#!/usr/bin/ksh

if [[ $# -eq 0 ]]
then
    echo "DR_ERROR= Script usage error"
    exit 1
fi

ret_code=0
command=$1
case $command in
    scriptinfo )
        echo "DR_VERSION=1.0"
        echo "DR_DATE=19092002"
        echo "DR_SCRIPTINFO=DLPAR test script"
        echo "DR_VENDOR=IBM";;
    usage )
        echo "DR_USAGE=root_dlpár_test.sh command [parameter]";;
    register )
        echo "DR_RESOURCE=cpu";;
    checkacquire )
        :;;
    :;;

```

```

        preacquire )
            ;;
        undopreacquire )
            ;;
        postacquire )
            ;;
        checkrelease )
            ;;
        prerelease )
            ;;
        undoprerelease )
            ;;
        postrelease )
            ;;
        * )
            ret_code=10;;
    esac

    exit $ret_code

```

In the following section, an introduction to signal API based approach is given.

DLPAR signal API

As previously mentioned, two approaches are provided to make programs DLPAR aware. The script-based approach described in the previous section, and the API-based approach described in this section.

The SIGRECONFIG signal is sent to the applications at the various phases of dynamic logical partitioning. The DLPAR subsystem defines *check*, *pre* and *post* phases for a typical operation. Applications can watch for this signal and use the DLPAR-supported system calls to learn more about the operation in progress and to take any necessary actions.

Note that when using signals, the application might inadvertently block the signal, or the load on the system might prevent the thread from running in a timely fashion. In the case of signals, the system will wait a short period of time, which is a function of the user-specified time-out, and proceed to the next phase. It is not appropriate to wait indefinitely because a non-privileged rogue thread could prevent all DLPAR operations from occurring.

The issue of timely signal delivery can be managed by the application by controlling the signal mask and scheduling priority. The DLPAR-aware code can be directly incorporated into the algorithm. Also, the signal handler can be cascaded across multiple shared libraries so that notification can be incorporated in a more modular way.

To integrate the DLPAR event using APIs, complete the following:

1. Catch the SIGRECONFIG signal by using the sigaction system call. The default action is to ignore the signal.
2. Control the signal mask in at least one of the threads so that the signal can be delivered in real time.
3. Ensure that the scheduling priority for the thread that is to receive the signal is sufficient so that it will run quickly after the signal has been sent.
4. Run the dr_reconfig system call to obtain the type of resource, type of action, and phase of the event, as well as other information that is relevant to the current request.

In the following section an introduction on how to make kernel extensions DLPAR aware is provided.

DLPAR-aware kernel extensions

Like applications, most kernel extensions are DLPAR safe by default. However, some are sensitive to the system configuration and might need to be registered with the DLPAR subsystem. Some kernel extensions partition their data along processor lines, create threads based on the number of online processors, or provide large pinned memory buffer pools. These kernel extensions must be notified when the system topology changes. The mechanism and the actions that need to be taken parallel those of DLPAR-aware applications.

To register and unregister from the kernel to be notified in the case of dynamic reconfiguration events, the following kernel services are available:

- ▶ reconfig_register
- ▶ reconfig_unregister
- ▶ reconfig_complete

In the following sections, programming implications of the dynamic reconfiguration of CPUs and memory are provided.

Programming implications of dynamic CPU reconfiguration

At boot time, CPUs are configured in the kernel. In AIX 5L Version 5.2, a processor is identified by three different identifications, namely:

- ▶ The physical CPU ID, which is derived from the open firmware device tree and used to communicate with RTAS.
- ▶ The logical CPU ID, which is a ppda-based index of online and offline CPUs.
- ▶ The bind CPU ID, which is the index of online CPUs.

The logical and bind CPU IDs are consecutive, and have no holes in the numbering. No guarantee is given across boots that the CPUs will be configured in the same order, or even that the same CPUs will be used in LPAR-enabled environments at all.

Initially, bind CPU IDs coincide with logical CPU IDs; however, DLPAR can remove a processor from the middle of the logical CPU list. The bind CPU IDs remain consecutive since they refer only to online CPUs, so the kernel has to explicitly map these IDs to logical CPU IDs (containing online and offline CPU IDs).

The range of logical CPU IDs is defined to be 0 to M-1, where M is the maximum number of CPUs that can be activated within the partition. M is derived from the Open Firmware device tree. The logical CPU IDs name both online and offline CPUs. The rset APIs are predicated on the use of logical CPU IDs.

Logical CPU numbers can be identified through the `lsrset` command. For example, on a two-way system:

```
# lsrset -a
sys/sys0
sys/node.01.00000
sys/mem.00000
sys/cpu.00000
sys/cpu.00001
```

You can interpret each CPU line as `sys/cpu.logic_cpu_number`.

The following command would list all the online logical CPU IDs:

```
lsrset -vor sys/sys0
```

The range of bind CPU IDs is defined to be 0 to N-1; however, N is the current number of online CPUs. The value of N changes as processors are added and removed from the system by either DLPAR or CPU Guard. In general, new processors are always added to the Nth position. Bind CPU IDs are used by the system call `bindprocessor` and by the kernel service `switch_cpu`.

The number of potential cpus can be determined by:

- ▶ `_system_configuration.max_ncpus`
- ▶ `_system_configuration.original_ncpus`
- ▶ `var.v_ncpus_cfg`
- ▶ `sysconf(_SC_NPROCESSORS_CONF)`

The number of online CPUs can be determined by:

- ▶ `_system_configuration.ncpus`

- ▶ `var.v_ncpus`
- ▶ `sysconf(_SC_NPROCESSORS_ONLN)`.

The number of online CPUs can also be determined from the command line. The following commands are provided by AIX:

- ▶ **`bindprocessor -q`**
- ▶ **`lsrset -a`**

As mentioned earlier, AIX supports two programming models for CPUs. The `bindprocessor` model, which is based on bind CPU IDs, and the `rset` API model, which is based on logical CPU IDs. Whenever a program implements any of these programming models it should be DLPAR aware.

A complete set of new subroutines is provided in AIX 5L Version 5.2 to provide access to the `rset` binding type kernel services. These subroutines are as follows:

- ▶ `krs_numrads`
- ▶ `krs_getrad`
- ▶ `krs_getinfo`
- ▶ `krs_alloc`
- ▶ `krs_free`
- ▶ `krs_op`
- ▶ `kra_creatp`
- ▶ `kra_attachrset`
- ▶ `kra_detachrset`
- ▶ `kra_getrset`
- ▶ `krs_init`
- ▶ `krs_getpartition`
- ▶ `krs_setpartition`
- ▶ `krs_getassociativity`

The following new interfaces (system calls and kernel services) are provided to query bind and logical CPU IDs and the mapping between them:

- ▶ `mycpu()`, returns bind CPU ID of the process.
- ▶ `my_lcpu()`, returns bind logical CPU ID of the process.
- ▶ `b2lcpu()`, returns the bind to logical CPU ID mapping
- ▶ `l2bcpu()`, returns the logical to bind CPU ID mapping

In the following section implications on programming with respect to dynamic memory reconfiguration are described.

Programming dynamic memory reconfiguration

Whenever an application uses plock or pinned shared memory, it should consider being DR aware.

Paging space implications for memory in DLPAR environment

Special attention should be paid to paging space requirements since they are closely related to the size of physical memory. A good rule of thumb is that the system should be preconfigured to handle the worst case.

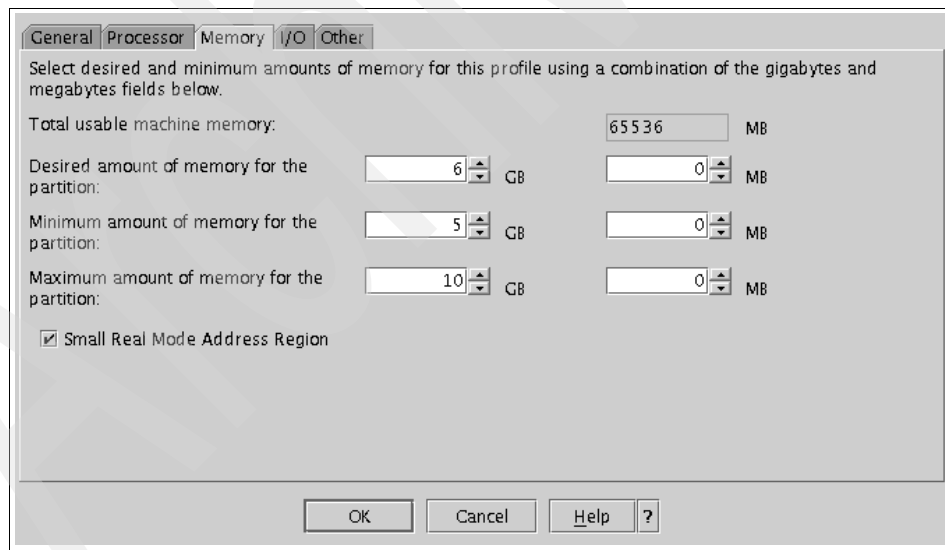
To do so, determine the amount of paging space that is required by applications while under stress with the maximum amount of memory configured as defined in the partition profile. To this number add the amount of paging space that would be needed when reducing the memory down to the minimum as specified in the partition profile. This is the difference between the maximum and the minimum of memory. Summarized in a formula, the paging space should be set to:

(paging space required in worst case) + (memory max) - (memory min)

Partition profile parameters for memory

The setting for memory minimum should be no less than 1/64 of memory maximum in the partition profile, in order to provide AIX with adequate memory. The reason for this limitation is that AIX has to initialize some kernel structures to the maximum that could potentially be available. It will not boot otherwise.

A new option Small Real Mode Address Region is provided in the Memory section of the partition profile on the HMC, as shown in Figure 3-49:



The screenshot shows a dialog box titled "Memory" with tabs for "General", "Processor", "Memory", "I/O", and "Other". The "Memory" tab is selected. The text inside says: "Select desired and minimum amounts of memory for this profile using a combination of the gigabytes and megabytes fields below." There are four rows of input fields:

Field	GB	MB
Total usable machine memory:	65536	
Desired amount of memory for the partition:	6	0
Minimum amount of memory for the partition:	5	0
Maximum amount of memory for the partition:	10	0

Below these fields is a checkbox labeled "Small Real Mode Address Region" which is checked. At the bottom are buttons for "OK", "Cancel", "Help", and a question mark icon.

Figure 3-49 HMC memory profile

For AIX 5L Version 5.2, this option should always be used to give the system greater flexibility when assigning memory. It should not be used in Version 5.1.

Monitoring DLPAR events

There are many components involved in the successful completion of a DLPAR event, including the Hardware Management Console (HMC), the system firmware, and the partition's operating system. Because of the complexity and the cooperative effort required of all of these components, it is difficult to diagnose and correct problems that cause DLPAR operations to fail. Therefore, several ways are provided to monitor DLPAR operations.

DLPAR operations can be monitored in the following ways:

- ▶ Operating panel LEDs
- ▶ Standard output of commands and scripts
- ▶ The AIX syslog facility
- ▶ An AIX trace
- ▶ The error log

Details of these options to monitor a DLPAR operation is given in the following sections.

Operator panel LEDs

You can watch the operator panel LEDs displayed on the HMC. DLPAR event LEDs are displayed while the operation occurs. The LEDs are provided in Table 3-8.

Table 3-8 LED processor indicator codes

Progress indicator code	Text string	Description
2000	CPUA	Dynamic LPAR CPU addition
2001	CPUR	Dynamic LPAR CPU removal
2002	MEMA	Dynamic LPAR memory addition
2003	MEMR	Dynamic LPAR memory removal

Standard output

Detailed data is written to standard output from components, such as the **drmgr** command or the DLPAR scripts. The output is sent back to the HMC and displayed for analysis purposes.

The syslog facility

The AIX syslog facility can be used to log the progress of a DLPAR event. The **drmgr -S** command can be used to specify a channel ID string for the syslog entries. Note that this string will be appended to every syslog entry made by the DR Manager, which allows for you to easily search and **grep** the log file for only

DLPAR events. The timestamps provided within the syslog help to provide a definitive record of exactly when DLPAR events happened.

These timestamps can also be useful in determining time-out values to be used on future DLPAR operations. The syslog facility is not enabled by default. To configure the syslog facility to capture DLPAR (and other) syslog entries, you can do the following as root:

1. Edit /etc/syslog.conf.
2. Add the following entry to the syslog configuration file to log all messages of the priority debug:

```
*.debug /var/adm/syslog.log rotate size 100k
```

3. Touch the file to be used:

```
touch /var/adm/syslog.log
```

4. Reconfigure the syslog daemon by starting and stopping it:

```
stopsrc -s syslogd  
startsrc -s syslogd
```

AIX trace

The AIX trace facility can be used to monitor DLPAR operations. When a trace is taken, the AIX trace report will contain trace hook entries for CPU or memory additions or removals. These trace hooks are not enabled by default. They can be enabled using a normal AIX trace mechanism (such as **trace** or **trcrpt**). To capture only the DR related traces (DR trace hook ID is 38F) and analyze them, perform the following steps:

1. Start trace:

```
trace -a -j 38f
```
2. Invoke the desired DR operation on the HMC.
3. Stop trace after the operations have ended with the **trcstop** command.
4. Analyze the trace events by invoking the **trcrpt** command.

Error log

The AIX error log will contain error log entries in cases involving kernel, kernel extension, or platform failures. These error log entries can be used for failure analysis. The standard messages will indicate when the AIX error log should be consulted. The DR-related error log entries are described in Table 3-9 on page 129.

Table 3-9 DR-related error log entries

Error log entry	Description
DR_SCRIPT_MSG	Application script error or related messages. Entry includes failing script name and DR phase.
DR_CPU_HANDLER_ERR	Kernel extension reconfiguration handler error for CPU add/removes. Entry includes failing handler's registration name, the kernel extension load module's path name, the DR phase and operation (ADD or REMOVE), and also the logical CPU number.
DR_MEM_HANDLER_ERR	Kernel extension reconfiguration handler error for LMB add/removes. Entry includes failing handler's registration name, the kernel extension load module's path name, the DR phase and operation (ADD or REMOVE) and the memory or LMB address range being removed. In the CHECK-phase, the start memory address will always be zero. The end memory address will be the total size of the memory that was to be removed. In the pre phase, the address range is always the LMB physical address range.
DR_APPS_ERR	DR operation failure because an application aborted it. Currently, this error is logged only when a SIGRECONFIG signal handler of a privileged process (root) calls dr_reconfig() during the check phase passing a flag value of DR_EVENT_FAIL. Entry contains the DR phase (always check, for this case), the DR operation (ADD or REMOVE), abort cause (always 0x01, for this case) and abort data (the process ID of the caller, in this case).
CPU_DEALLOC_ABORTED	<p>The DR CPU remove operation failed because the CPU deallocation was aborted. Entry contains the abort cause (a hex value) and abort data (in hex).</p> <p>Abort Cause and Meaning Abort Data 0x2 Bound User Thread Process ID 0x3 HA handler failed Name of handler 0x4 Last online CPU Logical CPU ID 0x7 Bound kernel Thread Process ID</p>

Error log entry	Description
DR_UNSAFE_PROCESS	A process has been detected that uses a non-DR safe library. This error only occurs when trying to add a CPU to a single-CPU system. Examples of unsafe libraries are older versions of libjava.a or libjvm.a, which are not safe to use in the middle of moving from uniprocessor to multiprocessor mode. They are, however, safe if loaded after the second CPU has been added. Entries include the process ID and the path of the loaded unsafe library.
DR_MEM_UNSAFE_USE	Non-DR aware kernel extension's use of physical memory. Results in the affected memory not being available for DLPAR removal. Entry contains affected logical memory address and an address corresponding to the kernel extension's load module, as well as the kernel extension load module's path name.
DR_DMA_MEM_MIGRATE_FAIL	Memory removal failure due to DMA activity. The affected LMB had active DMA mappings, which could not be migrated by the platform. Entry includes the logical memory address within the LMB, hypervisor migration return code, logical bus number of the slot owning the DMA mapping, and the DMA address.
DR_DMA_MEM_MAPPER_FAIL	Memory removal failure due to a kernel extension responsible for controlling DMA mappings error. Entry includes the DMA mapper handler return code, an address corresponding to the DMA mapper's kernel extension load module, and the DMA mapper's kernel extension load module's path name.

The AIX errlog can be displayed with the **errpt** command.

Corrective actions in failure conditions

When a processor deconfiguration fails, it could be because a process has been bound to the upper processor logical number, with the **bindprocessor** command or the **bindprocessor()** programming interface. To check if some processes are bound to a processor you can use the **ps -lmo THREAD** command and check the BND field of the output command. If the BND field is a dash (-) then the process or thread is not bound to a processor. If the BND field contains a number, then this number is the logical processor number from which the process has been

bounded. Figure 3-50 shows that the script script.bound is bound on processor 3.

```

root@server2:/ #ps -lmo THREAD lpg
USER  PID  PPID  TID ST  CP PRI SC  WCHAN      F      TT  BND  COMMAND
root   1    0    -  A   0  60  1    -    200003    -   -   -   /etc/init
-      -    -    259 S   0  60  1    -    410410    -   -   -
root  3654    1    -  A   0  60  1    1a0a84    40401    -   -   -   /usr/lib/errdemon
-      -    -    11869 S   0  60  1    1a0a84    10400    -   -   -
root  4024 24676 -  A  120 135  0    -    200001 pts/2  3 sh -- ./script.bound
-      -    -    49621 R  120 135  0    -      0    -   -   3 -
root  4324 9960 -  A   0  60  1    -    240001    -   -   -   /usr/sbin/inetd
-      -    -    13841 S   0  60  1    -    18400    -   -   -
root  4818    1    -  A   0  60  1    -    40001    -   -   -   /usr/dt/bin/dtlogin -daemon
-      -    -    6109 S   0  60  1    -    418410    -   -   -
root  4960 7246 -  A   0  60  1    -    240001    -   -   -   /usr/dt/bin/dtsession
-      -    -    8797 S   0  60  1    -    418410    -   -   -
root  6014    1    -  A   0  60 13    *    240001    -   -   -   /usr/sbin/syncd 60
-      -    -    5447 S   0  60  1    -    2400400    -   -   -
-      -    -    7033 S   0  60  1  31cd6b98    410410    -   -   -
-      -    -    8535 S   0  60  1  31cd6598    410410    -   -   -
-      -    -    9031 S   0  60  1  31a970d8    410410    -   -   -
-      -    -    9289 S   0  60  1  31cd9018    410410    -   -   -
-      -    -    9547 S   0  60  1  31cd9c58    410410    -   -   -
-      -    -    9805 S   0  60  1  31cd62d8    410410    -   -   -
-      -    -   10063 S   0  60  1  31a8a798    410410    -   -   -
-      -    -   10321 S   0  60  1  31a97258    410410    -   -   -
-      -    -   10579 S   0  60  1  3111ac18    410410    -   -   -
-      -    -   10837 S   0  60  1  31cd9798    410410    -   -   -
-      -    -   11095 S   0  60  1  31cd6b58    410410    -   -   -
-      -    -   11353 S   0  60  1  31a8a218    410410    -   -   -
root  6284 9960 -  A  16  56  1    -    240001    -   -   -   /usr/sbin/ndpd-host
-      -    -   11637 S  16  56  1    -    18400    -   -   -
root@server2:/ #

```

Figure 3-50 Output of the ps -lmo THREAD

To unbind the process you can use the **bindprocessor -u** command. The following command shows how to unbind the script.bound script:

```
bindprocessor -u 4024
```

3.3 Capacity Upgrade on Demand

Capacity Upgrade on Demand (CUoD) is an existing feature on some IBM @server pSeries and RS/6000 systems that allows for upgrading the capacity of a system with CPU resources that were shipped with the system, but which were part of an upgrade feature, providing reserve hardware capacity when growth requires it. CUoD only enables the number of CPUs that the customer is authorized to use. Additional CPUs can be enabled by invoking the **chcod** CUoD command. This command can only be run by the super user or a user with system group membership.

3.3.1 The **chcod** command (5.1.0)

The following example shows the syntax of the **chcod** command:

```
chcod [-r ResourceType -n NbrResources] [-m MailAddr] [-c CustInfo] [-h]
```

To display the current configuration, type the **chcod** command without any options. The output will appear as:

```
# chcod
Current MailAddr =
Current CustInfo =
Current Model and System ID =
Current number of authorized proc(s) out of 1 installed on system = 1
```

The flag options for the **chcod** command are shown in Table 3-10.

Table 3-10 The **chcod** command flags

Flags	Description
<i>-c customer_information</i>	This string of information will be used in the error log and in the body of an e-mail message sent. It may not contain a white space character. Characters supported are alphanumeric, decimal point (.), comma (,), hyphen (-), open parenthesis ((), and closed parenthesis ()). This flag is optional and has a limit of 255 characters.
<i>-h</i>	The command usage message.
<i>-n number</i>	This value must be 0 or greater and specifies the number of resource types to be authorized. The <i>-r</i> option flag and the <i>-n</i> option flag must be used together.
<i>-r resource type</i>	This flag specifies the resource type. The only supported value for resource type in AIX 5L Version 5.1 is <i>proc</i> , for processor. The <i>-r</i> option flag and the <i>-n</i> option flag must be used together.

3.3.2 Enhancement to the **lsvpd** command (5.2.0)

The **lsvpd** command lists all the VPD data. This command has been modified in AIX 5L Version 5.2 to obtain the processor and memory CUoD capacity card information system parameter from the firmware.

The **lsvpd** command prepends the system-wide keyword string, which is N5 for the processor and N6 for the memory CUoD capacity card information, and displays it along with the other VPD data that is being currently displayed. There is no error checking on the format or contents of the cards' VPD data.

The output from the **lsvpd** command is as follows:

```
*VC 5.0
*TM IBM,7038-6M2
*SE IBM,0110AABDD
*PI 00097493
*N5 703810-AABDD525B10-5555555D3C1C240404040PRM10000000159
*N6 703810-AABDD525B10-5555555D3C1C240404040MSM10000000164
...
```

3.4 Dynamic CPU sparing and CPU Guard (5.2.0)

Dynamic CPU sparing allows you to dynamically replace a CPU resource if a CPU failure is reported by Open Firmware. This CPU replacement happens in such a fashion that it is transparent to the user and to user-mode applications.

In AIX 5L Version 5.2, the CPU Guard implementation has been changed and enhanced to work in the new DLPAR Framework. The actual deallocation of the CPU resource is performed in the DLPAR Framework by the dynamic CPU removal procedure.

The DLPAR mechanism allowing the dynamic processor removal is based on leaving holes in the logical CPU ID's sequence, unlike the former CPU Guard implementation where holes in logical CPU IDs are not tolerated for compatibility reasons. The DR strategy is to abstract the status of the CPUs by having CPU bind IDs, which are a sequence of IDs 0 through N-1 representing only the on-line CPUs. This strategy provides better MCM-level affinity, thus breaking the assumption of uniform memory access from all CPUs by RPDP. With the DR approach, the load from the failing CPU is moved to a CPU that corresponds to the last CPU bind ID. Thus the failing CPU bind ID and the last CPU bind ID are swapped, leaving a hole in the logical CPU ID sequence and making the last on-line CPU the failing processor. Therefore, the `bindprocessor` system call interface, the **bindprocessor** command, the **bindintcpu** command, and the `switch_cpu` kernel service have been changed to work with the CPU bind ID model instead of the logical CPU ID model.

CPU Guard dynamically removes a failing CPU, whereas CPU sparing replaces a CPU with a spare one under the cover. During the reconfiguration no notifications of any kind are sent to the user, kernel extensions, or to user-mode applications that are CPU Guard- or DR-aware.

Dynamic CPU sparing is supported only on systems that are loaded with appropriate CPU Guard and DLPAR-enabled firmware such as IBM @server pSeries 690 and pSeries 670 running in LPAR mode with a

CPU Capacity Card present. Spare CPUs are CUoD CPUs that are not activated with a CUoD activation code.

Since CPU Guard operations are considered DR operations, they are serialized with all other DR operations. In this new environment the second-to-last CPU can be removed, which was a restriction to the prior CPU Guard implementation.

The dynamic CPU sparing process is as follows:

1. Open Firmware reports predictive CPU failure.
2. The event is logged to AIX error log and reported to the kernel.
3. The SIGCPUFAIL signal is sent to the init process.
4. The init process starts the **ha_star** command.
5. The **ha_star** command determines from the ODM whether to perform CPU sparing or CPU removal.
6. The **drmgr** command is called to perform CPU sparing or CPU removal.
7. The end of the CPU sparing procedure is logged into the AIX error log indicating the change in the physical cpuid.

A new ODM attribute, CPU sparing, is introduced, which can be set to enable or disable with SMIT using the fast path **smit chgsys**.

3.4.1 Change CPU Guard default (5.2.0)

The default feature of CPU Guard has been changed from disabled to enabled in AIX 5L Version 5.2. This only applies if the feature is supported by the system. To display the current status of CPU Guard, run the following command:

```
lsattr -El sys0 -a cpuguard
```

To change the value of CPU Guard to disabled, run the following command:

```
chdev -l sys0 -a cpuguard=disable
```

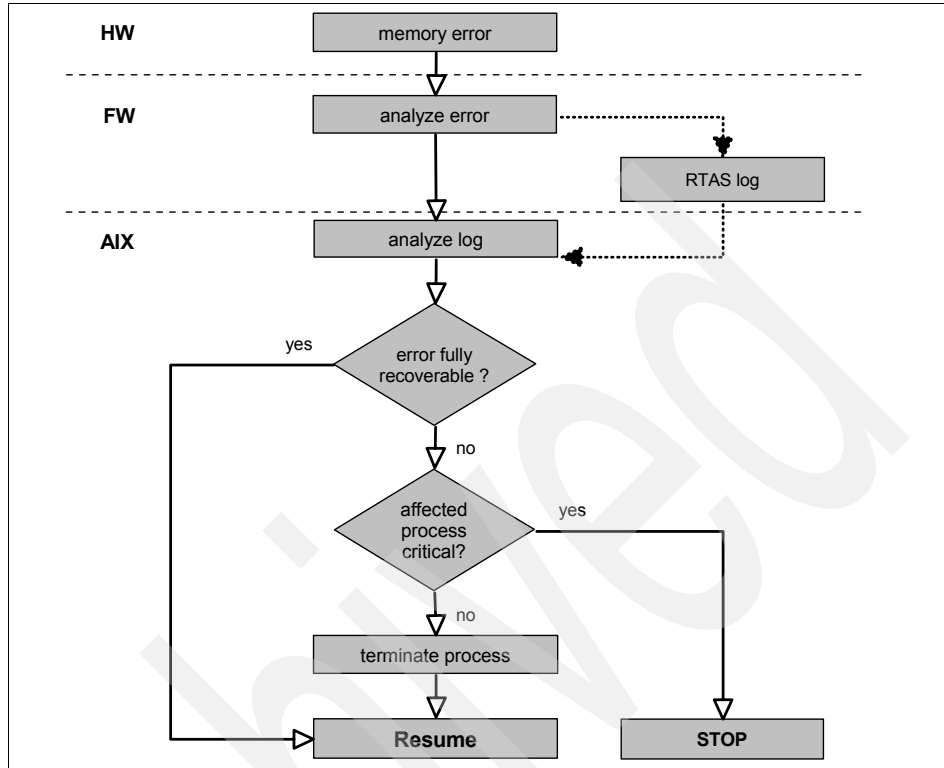


Figure 3-51 UE-Gard logic

A process should be considered critical to the system if, in the case where the process is terminated, the system itself should be terminated. These are all kernel processes or processes being executed in kernel mode.

Furthermore, a process can register itself or another process as being critical to the system. To register or unregister a process, two new system calls are provided that can be called from the process environment:

- ▶ `pid_t ue_proc_register (pid, arg)`
- ▶ `pid_t ue_proc_unregister (pid)`

In some cases an application may want to take action before being terminated, like create its own error log entry. To do so, the process should catch the SIGBUS signal with a SA_SIGINFO type of handler.

A new AIX UE-Gard error log entry is used by the kernel when signalling a process to terminate. This log entry contains the process ID and the signal value that caused the termination. The LABEL and RESOURCE fields in the AIX log indicate an UE-Gard event.

3.5 UE-Gard (5.2.0)

The Uncorrectable Error Gard (UE-Gard) is a Reliability, Availability, and Serviceability (RAS) feature that enables AIX in conjunction with hardware and firmware support to isolate certain errors that would previously have resulted in a condition where the system had to be stopped (checkstop condition). The isolated error is being analyzed to determine if AIX can terminate the process that suffers the hardware data error instead of terminating the entire system.

In the most likely case of intermittent errors, UE-Gard prevents the system from terminating. However, in the unlikely case of a permanent memory error, the system will checkstop eventually if the same memory is reused by a process that cannot be terminated.

The following systems are supported at the time of writing:

- ▶ @server pSeries 690
- ▶ @server pSeries 670
- ▶ @server pSeries 650
- ▶ @server pSeries 630

UE-Gard is not to be confused with (dynamic) CPU Guard. CPU Guard takes a CPU dynamically offline after a threshold of *recoverable* errors is exceeded, to avoid system outages.

The logic for UE-Gard is shown in Figure 3-51 on page 135. On memory errors, the firmware will analyze the severity and record it in a RTAS log. AIX will be called from firmware with a pointer to the log. AIX will analyze the log to determine if the error is recoverable or not. If the error is recoverable then AIX will resume. If the error is not fully recoverable then AIX will determine if the process with the error is critical or not. If the process is not critical, then it will be terminated by issuing a SIGBUS signal with an UE siginfo indicator. In the case where the process is a critical process, then the system will be terminated as a machine check problem.

3.6 Resource set scheduling and affinity services

A resource set is a structure that identifies physical resources. The physical resources supported by the AIX 5L Version 5.2 rsets are CPUs and memory pools (for the moment only one memory pool is supported). A rset parameter is used in many of the AIX resource set APIs or AIX commands to either get information from the system regarding resources or to pass information about requested resources to the system. Applications and job schedulers like Load Leveler may attach a rset to a process. Attaching a rset to a process limits the process to only use the resources contained in the rset. For example, assume a system or partition has 16 CPUs online with IDs of 0–15. Attaching a rset

containing CPUs 4–7 to a process limits that process to running only on CPUs 4–7.

The CPU and memory resources in a resource set are represented by bit maps. In AIX 5L Version 5.2, the primary use of rsets is to perform CPU topology and affinity operations. CPUs are identified in rsets by logical CPU IDs.

A logical CPU ID represents a constant mapping between the ID and a specific CPU in the system topology. This mapping is maintained for the duration of the system boot. A logical CPU ID by itself does not give any information about the CPU's placement in the system topology. For example, a partition with two MCMs of eight processors each may have their 16 logical CPU IDs assigned in any order. Applications cannot assume that logical CPU IDs 0–7 are in one MCM and IDs 8–15 are contained in the other MCM.

The set of logical CPU IDs available in a system may not be contiguous. There may be gaps in logical CPU ID numbers. This can occur when CPUs are dynamically reconfigured out of a partition. AIX 5L Version 5.2 allocates logical CPU IDs for the online CPUs sequentially at boot time. However, this may change in the future if AIX decides to preserve system topology information across system boot. The main system-defined resource sets are the following:

- ▶ **System RSET and sys/sys**

A rset containing the available (online) CPU and memory pool resources in the system or partition. On partitionable machines, this rset contains only the resources that are in the operating system's partition. It does not contain resources that are installed in the machine but not present in the operating system's partition. A dynamic reconfiguration (DR) operation that adds or removes a resource to a partition, adds or removes the resource to the system rset and atomic rset.

- ▶ **Node rsets, sys/node.mm.nnnnn, or sys/node.nnnnn**

These rsets contain resources that are present at various system detail levels (mm) and indexes (nnnnn) in the system. For example, if system detail level 04 represents the level in the system topology that corresponds to a Regatta MCM, then rset sys/node.04.00000 contains the resources in an MCM. Rset sys/node.04.00001 contains the resources in another MCM, and so on. The rset topology functions allow applications to read various levels of the system topology and to determine the hierarchical composition of the system.

Hardware systems that do not provide topology information contain only a single node rset sys/node.00000.

- Atomic rsets, sys/cpu.nnnnn, or sys/mem.nnnnn

These rsets contain a single resource, either a CPU or memory pool. There are atomic resource sets for every available (online) resource contained in the operating system's partition.

The following is an example of the topology of a partition with two processors and 5 GB of memory, displayed with the **lsrset** command.

```
root@lpar06:/ [912] # lsrset -v -a
T Name                Owner  Group  Mode   CPU  Memory
r sys/sys0            root   system r-r-r- 2    5120
  CPU: 0-1
  MEM: 0

r sys/node.01.00000    root   system r-r-r- 2    5120
  CPU: 0-1
  MEM: 0

r sys/mem.00000        root   system r-r-r- 0    5120
  CPU: <empty>
  MEM: 0

r sys/cpu.00000        root   system r-r-r- 1     0
  CPU: 0
  MEM: <empty>

r sys/cpu.00001        root   system r-r-r- 1     0
  CPU: 1
  MEM: <empty>

a test/cpus0and1       root   system rwr-r- 2     0
  CPU: 0-1
  MEM: <empty>

root@lpar06:/ [913] #
```

There are two types of rset, the partition rset and the effective rset:

- The partition rset can only be attached, modified, or detached by a root user. The AIX Workload Manager (WLM) attaches a partition rset when a process is classified with a work class that contains a rset. There is only one partition rset per process and it is updated by replacement. For example, a process is started with a WLM class that attaches a partition rset that contains CPUs 0–3. Later a root user attaches a rset that contains CPUs 2–7. The partition rset attached by WLM is replaced by the new rset. The process now runs on CPUs 2–7.
- The effective rset, generally used by applications, can be attached by root users and non-root users with a CAP_NUMA_ATTACH. Effective rset limits a

process to run only on the resources (CPUs, memory) contained in the rset. This means that a process's effective rset cannot contain more resources than the process's partition rset. For example, a process may have a partition rset established by the WLM that limits the process to running only on CPUs 0–3. A user can attach an effective rset with CPUs 2–3 and the process is limited to running only on CPUs 2–3. An attempt by the user to attach an effective rset with CPUs 2–7 would be rejected because the user attempted to use resources outside its partition rset.

Before AIX 5L Version 5.2, only partition rset exist. This means that WLM was the only user using partition rset. In the future, some job schedulers like Load Leveler may also use partition rset. With the effective rset, several users or applications can use rset, so WLM has been enhanced to handle this new situation. The following is the WLM behavior in a different kind of situation:

1. A process classified with a WLM work class partition rset may fail if the process uses bindprocessor. This prevents a process from using bindprocessor to consume resources on all CPUs in a system after WLM used the partition rset to limit the job to a subset of the CPUs.
2. In the absence of bindprocessor, a non-WLM partition rset, and effective rset use, the AIX 5L Version 5.2 WLM work class rset support is the same as AIX 5L Version 5.1. WLM continues to set partition rsets on processes classified with work classes containing a rset.
3. In the presence of bindprocessor, a non-WLM set partition rset, or an incompatible effective rset, WLM does not set the partition rset on a process when the process is classified. The explicitly set binding takes precedence over the WLM work class rset. In this situation, WLM classifies the process with the specified work class. However, the process's partition rset is not set to the work class's rset. The process's partition rset is unchanged. When WLM activity is initiated by a command such as `wlmcntrl` or `wlmassign`, a warning message is provided to advise the user that WLM was unable to set a partition rset.
4. WLM does not set the partition rset when classifying a process if the process already has a partition rset established either by a root user or a job scheduler.
5. If WLM is not able to set a partition rset when classifying a process, the WLM class partition rset is set if the reason for the inability to set the partition rset is removed. WLM is unable to set a WLM class partition rset due to bindprocessor, conflicting effective rset, or non-WLM partition rset use in the process. When the conflicting reason is removed, the WLM class partition rset is established.
6. WLM removes a WLM set partition rset when WLM is stopped or when a process is classified to a work class that does not have a rset. WLM does not

remove non-WLM set partition rsets when stopping or assigning to a work class without a rset.

rset commands

The rset commands provide an easy way for system administrators to use system rsets. Commands are provided to make, display, and remove rsets from the system registry. Other commands allow rsets to be attached to running processes or to run a command attached to a rset.

The mkrset command

The **mkrset** command creates and places into the system registry a rset with the specified set of CPUs and/or memory regions.

The user must have root authority or CAP_NUMA_ATTACH capability. The rset name must not exist in the registry. The owner and group IDs of the rset is set to the owner and group IDs of the command issuer.

The rset has read/write owner permissions and read permission for group and other.

The following example shows how to create a rset named test/cpu0and1 with CPU 0 and CPU 1.

```
root@lpar06:/ [949] # mkrset -c 0-1 test/cpu0and1
1480-353 rset test/cpu0and1 created.
root@lpar06:/ [950] #
```

The rmrset command

The **rmrset** command removes a rset from the system registry. The user must have root authority or CAP_NUMA_ATTACH capability and write access permission to specify rset.

The following example shows how to remove the above rset create with the **mkrset** command:

```
root@lpar06:/ [947] # rmrset test/cpus0and1
1480-401 rset 'test/cpus0and1' deleted.
```

The attachrset command

The **attachrset** command attaches a rset to a process. The command causes the specified process to be limited to running only on the processors or memory regions contained in the rset.

An rset name in the system registry can be attached to the process, or a rset containing the specified processors and memory regions can be attached to the

process. The user must have root authority or have CAP_NUMA_ATTACH capability and read access to the specified rset registry name (if the -r option used) and the target process must have the same effective user ID as the command issuer. The user must have root authority to set the partition rset on a process.

The following example shows how to attach the process with PID 266398 to the rset test/cpu0and1:

```
attachrset test/cpu0and1 266398
1480-206 rset test/cpu0and1 attached to pid 266398.
```

The **execrset** command

The **execrset** command executes a command with an attachment to a rset. It causes the specified command to be limited to running only on the processors or memory regions contained in the rset. An rset name in the system registry can be used to specify the processors and/or memory regions the command is allowed to use, or a rset containing the specified processors and memory regions can be attached to the process. The user must have root authority or have CAP_NUMA_ATTACH capability. The user must have root authority to attach a partition rset to the command's process.

The **detachrset** command

The **detachrset** command detaches a rset from a process. Detaching a rset from a process allows the process to use any of the processors or memory regions in the system. The user must have root authority or have CAP_NUMA_ATTACH capability, and the target process must have the same effective user ID as the command issuer. The user must have root authority to remove the partition rset from a process.

The **lsrset** command

The **lsrset** command lists all the rsets that exist in the system. The **lsrset** command already exists in AIX 5L Version 5.1. The syntax has been changed to be consistent with the other rset commands. The -o flag that displays the online resources contained in the rset has been added.

The following will list all the CPUs that are currently *known* to this partition:

```
lsrset -vr sys/sys0
```

See the **man** pages for more details about the different flags of the rset commands. To make a user, named *username*, CAP_NUMA_ATTACH capable, run the following command:

```
chuser capabilities=CAP_NUMA_ATTACH username
```

3.6.1 Memory affinity

IBM POWER4 processor SMP hardware systems consist of multiple multichip modules (MCMs) connected by an interconnect fabric. The system memory is attached to the MCMs. The interconnect fabric allows processors in one MCM to access memory attached to a different MCM. One attribute of this system design and interconnect fabric is that memory attached to the local MCM has faster access and higher bandwidth than memory attached to a remote MCM.

The objective is to offer improved performance to high performance computing applications by backing the application's data in memory that is attached to the MCM where the application is running. The MCM local memory affinity is only available in SMP mode and not in partition mode.

To determine if the hardware topology is available on your system for memory affinity, enter the following command:

```
#lsrset -n sys
```

If the answer of the command has several `sys/node` such as `sys/node.01.00000`, `sys/node.02.00001`, then your system has the hardware topology for the memory affinity. If the answer of the `lsrset` command just contains one system/node, such as `sys/node.01.00000`, then your system does not have the hardware topology to benefit from the memory affinity. In order to support MCM local allocation for the memory affinity, the VMM creates multiple memory vmpools. This decision is made at system boot time. If memory affinity is turned on, a vmpool is created for each affinity domain reported by the firmware. Otherwise a single vmpool is used to manage all of system memory.

In AIX 5L Version 5.1 ML 5100-02, the MCM memory affinity support had a global all or nothing `vmtune` parameter to turn on or turn off the MCM local memory affinity. If enabled, all process and kernel space memory allocations use MCM local memory affinity allocation. In Version 5.2, a new shell environment variable `MEMORY_AFFINITY=MCM` is provided to request MCM local memory affinity allocation for selected applications. The `vmo` (or `vmtune`) commands continue to be used to enable MCM local memory affinity allocation. However, using this command *only* enables the ability for a process to request MCM local memory allocation. The MCM local memory allocation is used only when the `MEMORY_AFFINITY=MCM` environment variable is specified.

Enabling the memory affinity on a AIX 5L Version 5.2 is made in two steps, as follows:

1. You need to make your system able to use the memory affinity. For that, run the following sequence:
 - a. **`vmo -p -o memory_affinity=1`**

- b. Answer yes to the question `Run bosboot now?`.
 - c. Reboot the system.
2. Upon reboot, set the `MEMORY_AFFINITY=MCM` variable to the environment of each process that uses the memory affinity. Putting this environment variable in the `/etc/environment` file enable the memory affinity for all the processes of the system.

For removing the memory affinity of a process, it is just necessary to unset the `MEMORY_AFFINITY` variable. A reboot with `vmo` (or `vmtune`) changes is no longer needed.

To benefit from the memory affinity, it is preferable that the processes running are binded to the processors (it is possible to use `wlm` for that). With memory affinity, the performance can be improved for applications that have processes or threads that initialize a memory array. In this case, for a 32-processor machine, for example, you could have 32 threads bound uniquely to the thirty-two processors and each thread operates on a unique, contiguous part of its own array.

3.6.2 Large page support

Large page support can improve performance of applications for several reasons. For example, some applications that have a large amount of sequential memory access, such as scientific applications, need to have the highest memory bandwidth possible. Those applications are using memory prefetching to minimize memory latencies. The prefetching starts every time a new page is accessed and grows as the page continues to be sequentially accessed. However, prefetching must be restarted at page boundaries. This kind of application often accesses user data sequentially, and accesses span 4-KB page boundaries. These applications can realize a significant performance improvement if larger pages are used for their data because this minimizes the number of prefetch startups. The large page performance improvements are also attributable to reduced translation lookaside buffer (TLB) misses due to the TLB being able to map a larger virtual memory range.

AIX supports large page by both 32- and 64-bit applications and both the 32- and 64-bit versions of the AIX kernel support large pages.

The large pages are hardware dependant. On a p690, it is possible to define a memory area of 16 MB pages. The size of the 16 MB pool is fixed at boot time and cannot be changed without rebooting the system. Large pages are only used for applications that explicitly request them. There is no need for a large page memory pool if your applications do not request them. AIX treats large pages as pinned memory and does not provide paging support for them.

To define 100 pages of 16 MB each, use the following command:

```
# vmo -p -olpgpg_regions=100 -olpgpg_size=16777216
Setting lpgpg_size to 16777216 in nextboot file
Warning: bosboot must be called and the system rebooted for the lpgpg_size
change to take effect
Setting lpgpg_regions to 100 in nextboot file
Warning: bosboot must be called and the system rebooted for the lpgpg_regions
change to take effect
Run bosboot now? [y/n] y
```

```
bosboot: Boot image is 17172 512 byte blocks.
#
```

Then reboot the system.

It is also possible to use the large page for the shared memory. To do that with a permanent change to the system tuning parameters, run the following command:

```
# vmo -pov_pinshm=1
Setting v_pinshm to 1 in nextboot file
Setting v_pinshm to 1
```

AIX provides a security mechanism to control use of large page physical memory by non-root users. The security mechanism prevents unauthorized users from using the large page pool and thus preventing its use by the intended users or applications. Non-root user IDs must have a `CAP_BYPASS_RAC_VMM` capability in order to use large pages. A system administrator can grant this capability to a user ID by the `chuser` command. The following command grants the ability to use large pages to user ID `lpguserid`.

```
chuser capabilities=CAP_BYPASS_RAC_VMM,CAP_PROPAGATE lpguserid
```

Both large page data and large page shared memory segments are controlled by this capability.

The applications can run into two different modes:

- ▶ In *advisory mode*, an application may have some of its heap segments backed by large pages and some of them backed by 4-KB pages. 4-KB pages are used to back segments when there are not enough large pages available to back the segment. Executable programs marked to use large pages use large pages in advisory mode.
- ▶ In *mandatory mode*, an application is terminated if it requests a heap segment and there are not enough large pages to satisfy the request. Customers who use the mandatory mode must monitor the size of the large page pool and ensure it does not run out of large pages. Otherwise, their mandatory large page mode applications fail.

There are two ways to request an application's data segments to be backed by large pages:

1. The executable file can be marked to request large pages. The XCOFF header in an executable file contains a new flag to indicate that the program wants to use large pages to back its data and heap segments. This flag can be set when the application is linked by specifying the `-blpdata` option on the `ld` command. The flag can also be set or cleared using the `ldedit` command. The `ldedit -blpdata filename` command sets the large page data flag in the specified file. The `ldedit -bnolpdata filename` clears the large page flag.
2. An environment variable can be set to request large pages. An environment variable is provided to allow users to indicate that they want an application to use large pages for data and heap segments. The environment variable takes precedence over the executable large page flag. Large page usage is provided as the `LDR_CNTRL` environment variable.
 - `LDR_CNTRL=LARGE_PAGE_DATA=Y`
Specifies that the program uses large pages for its data and heap segments. This is the same as marking the executable to use large pages.
 - `LDR_CNTRL=LARGE_PAGE_DATA=N`
Specifies that the program does not use large pages for its data and heap segments. This overrides the setting in a executable marked to use large pages.
 - `LDR_CNTRL=LARGE_PAGE_DATA=M`
Specifies that the program uses large pages in a mandatory mode for its data and heap segments.

Important: Only some specific applications take advantage of the memory affinity or large pages. For other applications, enabling the memory affinity or large pages support can degrade the system performance.

3.7 Resource Monitoring and Control

In AIX 5L, a new Resource Monitoring and Control (RMC) subsystem is available that originated as the Reliable Scalable Cluster Technology (RSCT) on the IBM SP platform. The use of RSCT is growing and, therefore, it is now shipped with AIX. RMC is a major component of RSCT and is automatically installed and configured when AIX is installed.

This subsystem allows you to associate predefined responses with predefined conditions for monitoring system resources. An example is to broadcast a

message when the /tmp file system becomes 90 percent full to summon the attention of a system administrator.

3.7.1 Packaging and installation

The RMC subsystem is installed by default and is delivered in one bundle named rsct.core containing nine different filesets with the following names:

```
# lsipp -L "*rsct*"
Filesset
```

Filesset	Level	State	Description
rsct.core.auditrm	2.2.0.0	C	RSCT Audit Log Resource Manager
rsct.core.errm	2.2.0.0	C	RSCT Event Response Resource Manager
rsct.core.fsrn	2.2.0.0	C	RSCT File System Resource Manager
rsct.core.gui	2.2.0.0	C	RSCT Graphical User Interface
rsct.core.hostrn	2.2.0.0	C	RSCT Host Resource Manager
rsct.core.rmc	2.2.0.0	C	RSCT Resource Monitoring and Control
rsct.core.sec	2.2.0.0	C	RSCT Security
rsct.core.sr	2.2.0.0	C	RSCT Registry
rsct.core.utils	2.2.0.0	C	RSCT Utilities

All executables and related items are installed into the /usr/sbin/rsct directory, while the log files and other temporary data is located in /var/ct. The following entry is located in /etc/inittab:

```
ctrmc:2:once:/usr/bin/startsrc -s ctrmc > /dev/console 2>&1
```

Due to this entry, the RMC subsystem is also automatically started. This subsystem can be controlled using the SRC commands, but it also has its own control command (/usr/sbin/rsct/bin/rmcctl), which is the preferred way to stop and start it. Due to the number of available options on this subsystem, it can only be controlled through the Web-based System Manager. A SMIT interface is not available at the time of this publication.

3.7.2 Concepts of RMC

The basic function of RMC is based on two concepts: Conditions and responses. To provide you a ready-to-use system, 84 conditions and eight responses are predefined for you. You can use them as they are, customize them, or use them as templates to define your own conditions and responses. To monitor a condition, simply associate one or more responses with the condition.

A condition monitors a specific property, such as total percentage used, in a specific resource class, such as JFS. You can monitor the condition for one or

more, or all the resources within the monitored property, such as /tmp, or /tmp and /var, or all the file systems. Each condition contains an event expression to define an event and an optional rearm expression to define a rearm event. The event expression is a combination of the monitored property, mathematical operators, and some numbers, such as PercentTotUsed > 90 in the case of a file system. The rearm expression is a similar entity, for example, PercentTotUsed < 85.

The following figures provide an example of a condition property dialog with two tabs: General (Figure 3-52) and Monitored Resources (Figure 3-53 on page 148).

General Monitored Resources

Name: /tmp space used

Resource class: Journaled File System (JFS or Jfs or jfs) ▼

Monitored property: PercentTotUsed ▼ [Details...](#) [Use defaults](#)

Event expression: PercentTotUsed > 90

Event description: An event will be generated when more than 90 percent of the total space in the /tmp directory is in use.

Rearm expression: PercentTotUsed < 85

Rearm description: The event will be rearmed when the percent of the space used in the /tmp directory falls below 85 percent.

Severity: Informational ▼

[Responses to the condition...](#)

[OK](#) [Cancel](#) [Help](#)

Figure 3-52 Condition Properties dialog - General tab

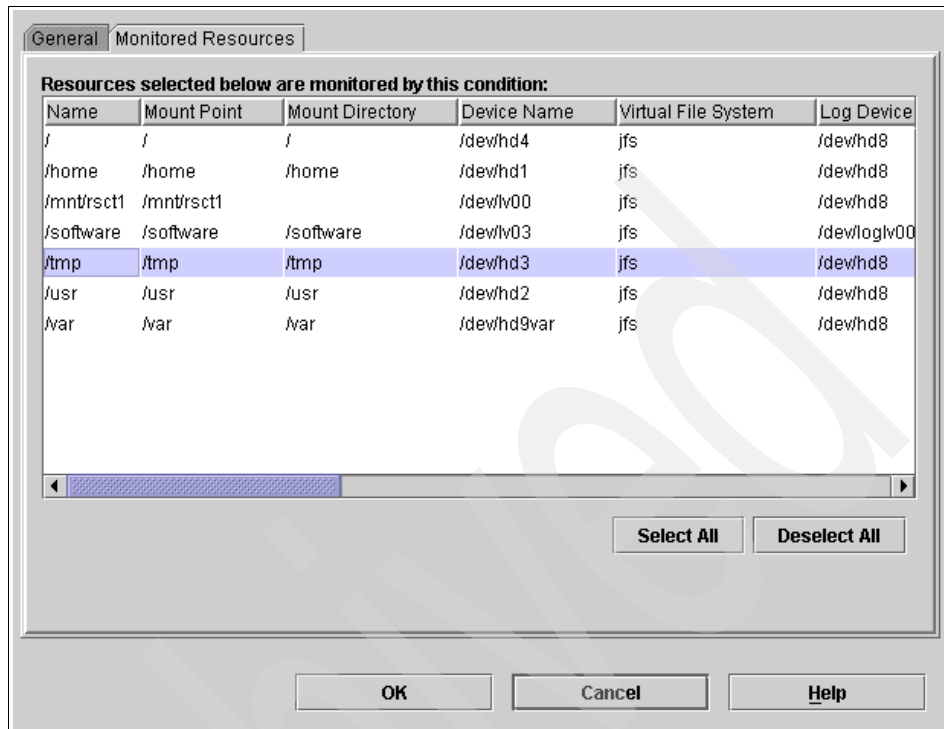


Figure 3-53 Condition Properties dialog - Monitored Resources tab

Each response can consist of one or more actions. Figure 3-54 on page 149 provides an example of a Response Properties dialog.

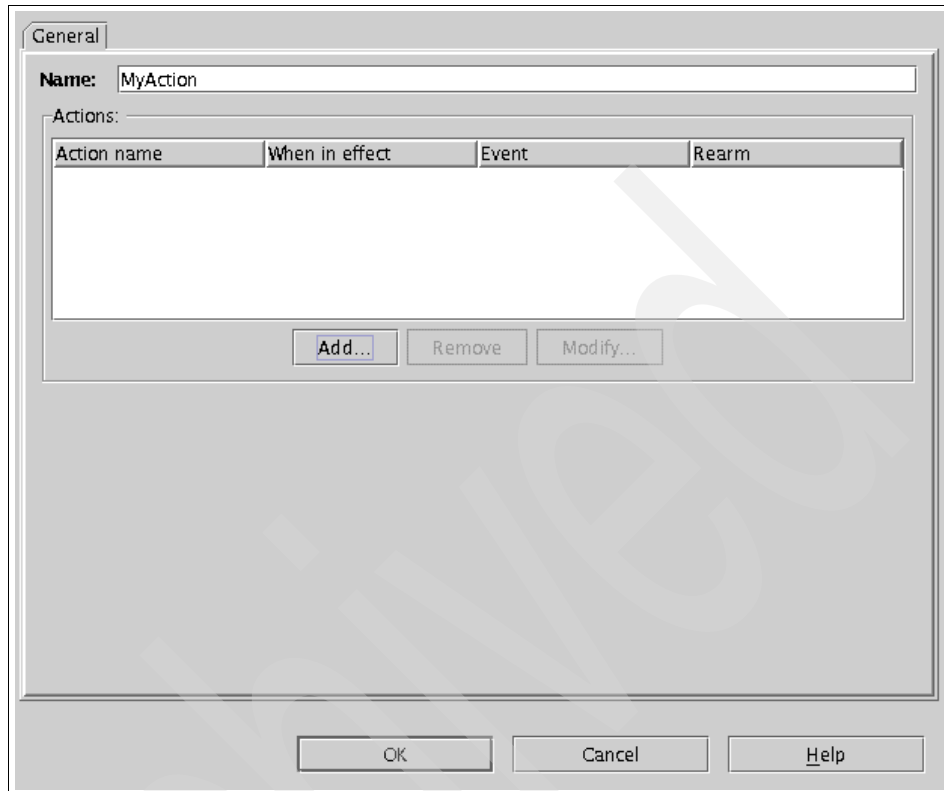
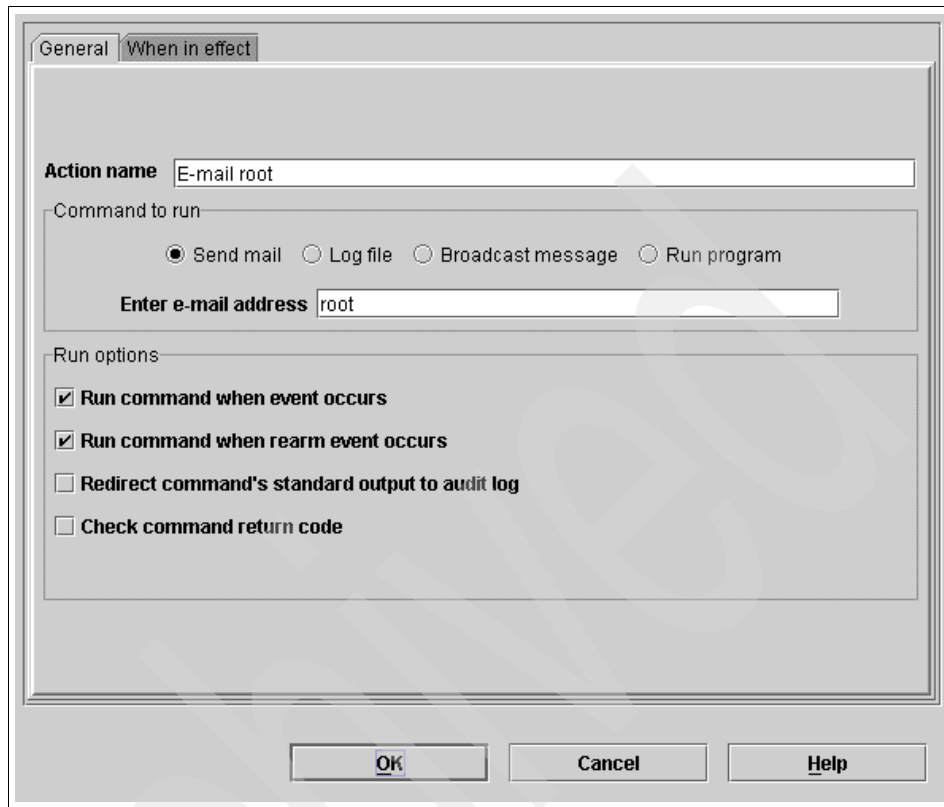


Figure 3-54 Response Properties dialog - General tab

The Add and Modify buttons launch an Action Properties dialog.

To define an action, you can choose one of the three predefined commands, Send mail, Log an entry to a file, or Broadcast a message, or you can specify an arbitrary program or a script of your own by using the Run program option. The action can be active for an event only, for a rearm event only, or for both. You can also specify a time window in which the action is active, such as always, or only during on-shift on weekdays.

The following figures provide an example of an Action Properties dialog with two tabs: General (Figure 3-55 on page 150) and When in effect (Figure 3-56 on page 151).



The image shows a screenshot of the 'Action Properties' dialog box, specifically the 'General' tab. The dialog has two tabs: 'General' and 'When in effect'. The 'General' tab is active. It contains the following fields and options:

- Action name:** A text field containing 'E-mail root'.
- Command to run:** A section with four radio buttons: 'Send mail' (selected), 'Log file', 'Broadcast message', and 'Run program'.
- Enter e-mail address:** A text field containing 'root'.
- Run options:** A section with four checkboxes:
 - ☒ Run command when event occurs
 - ☒ Run command when rearm event occurs
 - ☐ Redirect command's standard output to audit log
 - ☐ Check command return code

At the bottom of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

Figure 3-55 Action Properties dialog - General tab

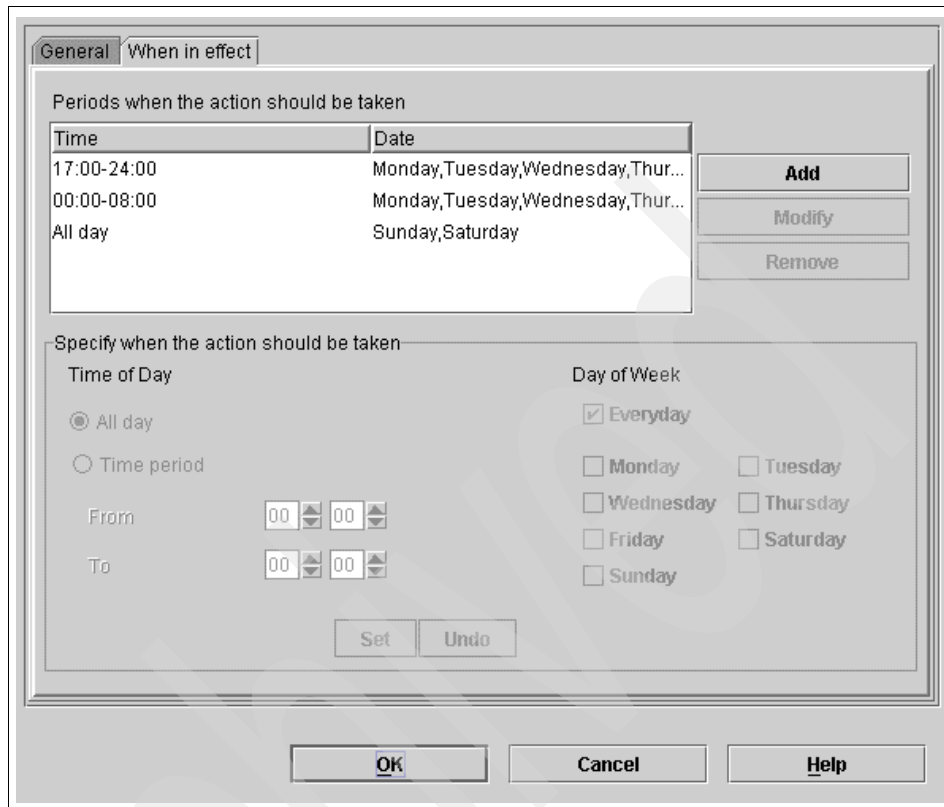


Figure 3-56 Action Properties dialog - When in effect tab

The previously mentioned predefined commands are using the `notifyevent`, `wallevent`, and `logevent` scripts, respectively, in the `/usr/sbin/rsct/bin` subdirectory. These command scripts capture events through the Event Response resource manager (ERRM) environment variables and notify you of the events through e-mails, logs, and broadcast messages. Do not modify these predefined command scripts. However, you can copy these predefined commands as templates to create your own scripts and use them for the Run program option.

Note that because the `logevent` script uses the `alog` command to log events to the files you designate, the content of these files can be listed with the `alog` command.

If the event expression of a condition is evaluated to be true, an event occurs and the ERRM checks all responses associated with the condition and executes the event actions defined in these responses. Only after the rearm expression becomes true and the ERRM has executed the corresponding rearm event

actions defined in the responses can the event and the event actions be generated again.

For each of the event and rearm events, the actions taken in response to them and the success or failure of any commands running in these actions are logged by the Audit Log resource manager (AuditRM) to the audit log. The standard error of a run command, if any, is always logged to the audit log. The standard output of a run command is logged to the audit log only if the “Redirect command’s standard output to audit log” option is selected for the command in the Action Properties dialog. The audit log records can be listed with the **lsaudrec** command or removed from the log file with the **rmaudrec** command.

3.7.3 How to set up an efficient monitoring system

The following steps are provided to assist you with setting up an efficient monitoring system:

1. Review the predefined conditions of your interests. Use them as they are, customize them to fit your configurations, or use them as templates to create your own.
2. Review the predefined responses. Customize them to suit your environment and your working schedule. For example, the response Critical notifications is predefined with three actions:
 - a. Log events to /tmp/criticalEvents.
 - b. E-mail to root.
 - c. Broadcast message to all logged-in users any time when an event or a rearm event occurs.

You may modify the response, such as to log events to a different file any time when events occur, e-mail you during non-working hours, and add a new action to page you only during working hours. With such a setup, different notification mechanisms can be automatically switched, based on your working schedule.

3. Reuse the responses for conditions. For example, you can customize the three severity responses (Critical notifications, Warning notifications, and Informational notifications) to take actions in response to events of different severities, and associate the responses to the conditions of respective severities. With only three notification responses, you can be notified of all the events with respective notification mechanisms based on their urgencies.

4. Once the monitoring is set up, your system continues being monitored whether your Web-based System Manager session is running or not. To know the system status, you may bring up a Web-based System Manager session and view the Events plug-in, or simply use the **lsaudrec** command from the command line interface to view the audit log.

3.7.4 Web-based System Manager enhancements (5.1.0)

The single system monitoring application for Web-based System Manager that was shipped with AIX 5L Version 5.0 has been enhanced with some new monitoring plug-ins.

Enhancements in AIX 5L Version 5.1 include:

- ▶ Host Overview plug-in enhancements
- ▶ Audit log dialog enhancements
- ▶ Conditions plug-in and dialog enhancements

Host Overview plug-in enhancements

As shown in Figure 3-57 on page 154, the Host Overview plug-in provides a convenient summary of a minimal set of vital signs of a system, which are:

- ▶ Operating system level
- ▶ IP address
- ▶ Machine type
- ▶ Serial number
- ▶ Number of processors
- ▶ CPU cycles
- ▶ Memory
- ▶ Paging space
- ▶ File system utilization

The Host Overview plug-in is packaged as part of Web-based System Manager base code. The dynamic status area on the Host Overview plug-in will be shown only if RSCT is installed.

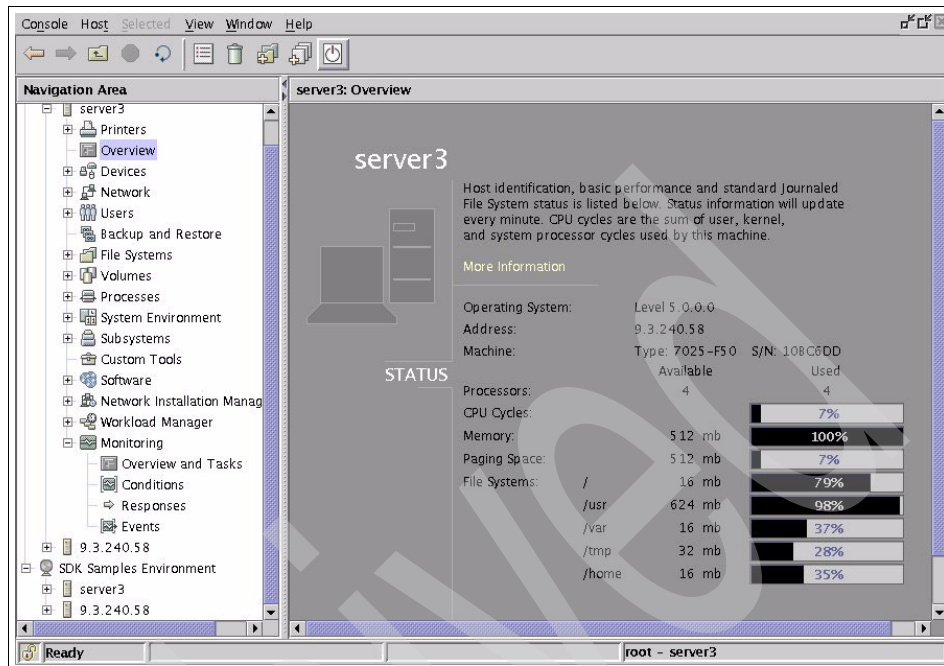


Figure 3-57 Web-based System Manager, Host Overview plug-in

The Host menu, shown in Figure 3-58 on page 155, from the menu bar provides an easy way to perform critical tasks, such as the following:

- ▶ List Top 10 Processes
- ▶ Delete a Process
- ▶ Expand a Journaled File System
- ▶ Increase Paging Space
- ▶ Shutdown
- ▶ Reconnect to RMC System

The menu choice Reconnect to RMC System is shown only if RSCT is installed. It is enabled only when the Host Overview plug-in is disconnected from the RMC monitoring subsystem. Use this menu choice to reconnect the session to the RMC.

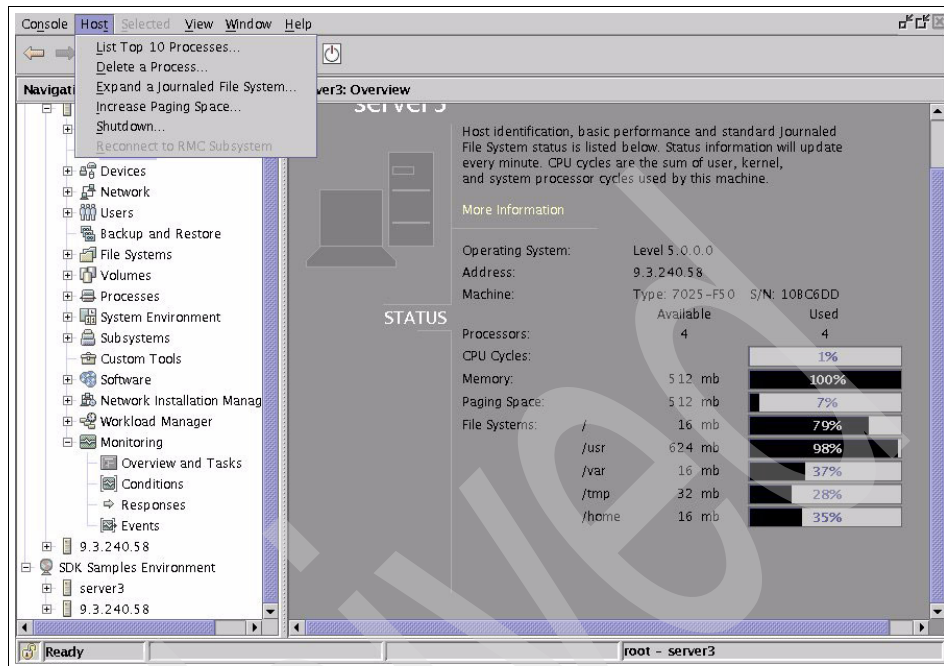


Figure 3-58 Web-based System Manager, Host menu of the Overview plug-in

Events

The Events plug-in shows all the events, rearm events, and errors that occur during the current Web-based System Manager session.

Audit log dialog enhancements

A new audit log plug-in, as shown in Figure 3-59 on page 156, has been added to the Events plug-in. The audit log dialog can be launched from the Events menu on the menu bar. The audit log records events, rearm events, and errors that have occurred on the system once the monitoring function is started, whether a Web-based System Manager session is running or not. In addition, it also records the actions that take place in response to the events or the rearm events, and it records errors on the underlying monitoring subsystems. It can be a useful and informative tool for system administrators. You can also look at the audit log at the command line by issuing the **lsaudrec** command, or remove unwanted audit log entries using the audit log dialog or at the command line by using the **rmaudrec** command.

Specify a filter option below and press the "Refresh" button to apply the filter:

☒ View log records from March 29 2001 to March 29 2001

☐ View all log records [Total records: 3]

Time	Category	Description
Mar 29, 2...	Informatio...	Monitoring of Condition /tmp space used is stopped successfully.
Mar 29, 2...	Informatio...	Monitoring of Condition /tmp space used is started successfully.

Refresh Delete Details... Find: Find Next

Close Help

Figure 3-59 Web-based System Manager, audit log panel

Conditions

The Conditions plug-in displays a rich set of predefined conditions (Figure 3-61 on page 157) for you to monitor your system, such as the memory, paging space, adapters, file systems, physical volume, running programs, and so forth. You can use the conditions as they are or customize them.

Conditions plug-in and dialog enhancements

Several changes have been made to the Conditions plug-in. The enhancements are:

- ▶ In the Condition property dialog (shown in Figure 3-60 on page 157), a new Monitored property field shows you if the condition is currently being monitored or not.
- ▶ In the Conditions plug-in:
 - A new column, Monitored, shows the details view of the Conditions plug-in. Yes indicates that the condition is currently being monitored. Click the column heading to sort the conditions into their monitored states.
 - Additional icons are provided for the condition objects to indicate whether a condition is being monitored.
 - New icons and menu choices have been added so you can start and stop monitoring right from the Conditions plug-in without going through the monitoring dialog.

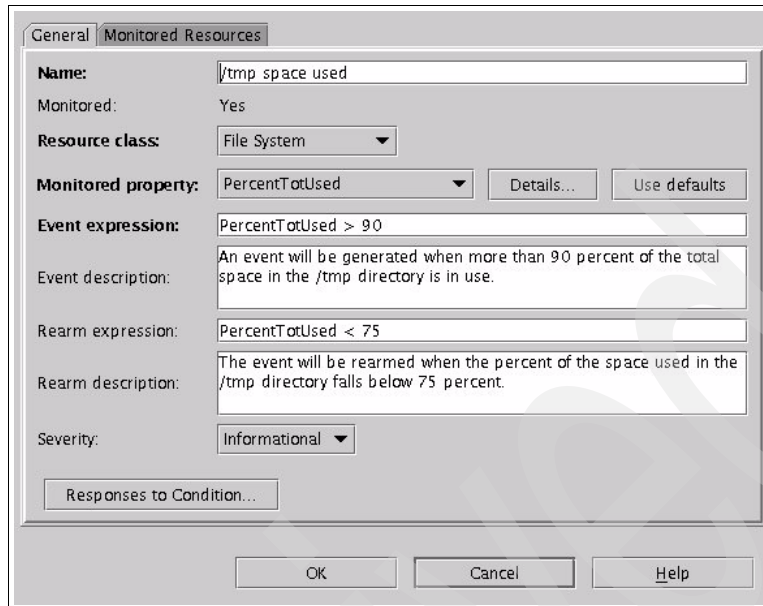


Figure 3-60 Web-based System Manager, condition property panel

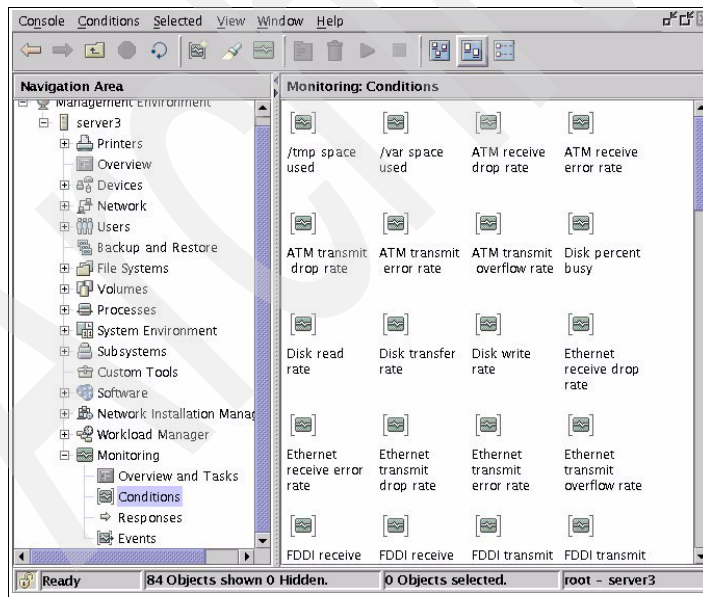


Figure 3-61 Web-based System Manager, conditions panel

3.7.5 Resources

The resources that can be monitored are managed by two resource managers: The File System Resource Manager (FSRM), and the Host Resource Manager (HostRM).

The FSRM monitors all local JFSs on a machine and checks for the status (offline, online), the total percentage used, and the percentage of inodes used in the file system.

The HostRM supports nine different resource classes. The network adapter resource classes (Ethernet Device, Token Ring Device, ATM Device and FDDI Device) each monitor five different properties, such as receive error rates and others. There is one resource class (physical volume) supporting the monitoring of the hard disk. It checks for four different properties, for example, percentage of time the device was busy between two consecutive observations. The percentage of free paging space is currently the only supported property of the resource class Paging Device. The processor resource class monitors processor utilization by checking, for example, for the idle time property and others.

The host resource class supports 46 different properties that represent all different areas, in order to get a system-wide status of your machine. This includes, among others, properties such as the size of the system run queue, sizes and change in size of various memory buffer pools in the kernel, and overall utilization of all processors in the system.

The last resource class (program) checks if a specific program is running or if the number of processes for a specific program is changing. The predefined condition in this resource class checks to see if the sendmail daemon is running. You can restrict this condition by specifying a filter expression, which can use the various fields supported by the **ps** command. This allows, for example, monitoring of only programs running with a specific user ID.

All resource classes support, in addition to their specific properties, a general configuration change property. With this property, you can send a mail to root or any other specified user whenever the configuration of a device changes. The JFS, PagingDevice, and processor resource classes support the operational state property.

The RMC subsystem is comprised of several multithreaded daemons, as shown in the following output:

```
# ps -mo THREAD -p 5948,20388,21942,23792,25348
USER  PID  PPID   TID ST  CP PRI SC   WCHAN      F    TT BND COMMAND
   root 5948  6456   -  A   0  60  3 e6004020 340001    -  -
/usr/sbin/rsct/bin/rmcd -c
      -    -    - 7497 S   0  60  1      - 418410    -  - -
```

```

- - - 29165 S 0 60 1 - 2400400 - - -
- - - 32771 S 0 60 1 e6004020 8c10410 - - -
root 20388 6456 - A 0 60 13 * 240001 - -
/usr/sbin/rsct/bin/IBM.ERRmd
- - - 29441 S 0 60 1 e60039a0 8410410 - - -
- - - 30481 S 0 60 1 7006686c 410410 - - -
- - - 31741 S 0 60 1 7005c06c 400410 - - -
- - - 31761 S 0 60 1 - 418410 - - -
- - - 32037 S 0 60 1 - 2400400 - - -
- - - 32513 S 0 60 1 7038ca6c 410410 - - -
- - - 33033 S 0 60 1 e60040a0 8410410 - - -
- - - 37155 S 0 60 1 e60048a0 8410410 - - -
- - - 37413 S 0 60 1 e6004920 8410410 - - -
- - - 37671 Z 0 61 1 - c00001 - - -
- - - 41837 S 0 60 1 e60051a0 8c10410 - - -
- - - 50319 Z 0 60 1 - c00001 - - -
- - - 51191 Z 0 60 1 - c00001 - - -
root 21942 6456 - A 0 60 9 * 240001 - -
/usr/sbin/rsct/bin/IBM.AuditRMd
- - - 33809 S 0 60 1 70062e6c 410410 - - -
- - - 34073 S 0 60 1 - 418410 - - -
- - - 34595 S 0 60 1 - 2400400 - - -
- - - 34833 S 0 60 1 70179e6c 400410 - - -
- - - 35091 S 0 60 1 e60044a0 8410410 - - -
- - - 36125 S 0 60 1 e60046a0 8410410 - - -
- - - 36381 S 0 60 1 e6004720 8c10410 - - -
- - - 36639 S 0 60 1 e60047a0 8c10410 - - -
- - - 36897 S 0 60 1 e6004820 8c10410 - - -
root 23792 6456 - A 0 60 8 * 240001 - -
/usr/sbin/rsct/bin/IBM.FSrmmd
- - - 41677 S 0 60 1 e6005120 8410410 - - -
- - - 43371 S 0 60 1 70126c6c 410410 - - -
- - - 43641 S 0 60 1 - 2400400 - - -
- - - 44409 S 0 60 1 70317c6c 400410 - - -
- - - 47101 S 0 60 1 e6005ba0 8410410 - - -
- - - 50589 S 0 60 1 - 418410 - - -
- - - 52393 S 0 60 1 e6006620 8c10410 - - -
- - - 52659 S 0 60 1 e60066a0 8c10410 - - -
root 25348 6456 - A 0 60 7 * 240001 - -
/usr/sbin/rsct/bin/IBM.HostRMd
- - - 42359 S 0 60 1 e60052a0 8410410 - - -
- - - 43031 S 0 60 1 e6005420 8c10410 - - -
- - - 48793 S 0 60 1 - 418410 - - -
- - - 50879 S 0 60 1 - 2400400 - - -
- - - 57321 S 0 60 1 e6006fa0 8430410 - - -
- - - 57831 S 0 60 1 7022786c 410410 - - -
- - - 58583 S 0 60 1 70391a6c 400410 - - -

```

The main control daemon (rmcd), the event response daemon (IBM.ERrmd), and the audit daemon (IBM.AuditRMd) run as soon as the RMC subsystem is activated. The file system IBM.FSrmd and host daemon IBM.HostRMd are only active if a file system or host condition, respectively, is monitored.

3.7.6 Command line interface (5.1.0)

This section describes the new Resource Monitoring and Control (RMC) and Event Response Resource Manager (ERRM) command line interfaces (CLI).

The RMC CLI allows system administrators the ability to manage resources and resource classes. A resource class defines a particular software or hardware entity. For example, the IBM.Host resource class defines the system. A resource is an instance of a resource class. The RMC CLI consists of the commands shown in Table 3-11.

Table 3-11 RMC commands

Commands	Description
mkrsrc	Defines a new resource
rmrsrc	Removes a defined resource
lsrsrc	Lists (displays) resources or a resource class
lsrsrcde	Lists a resource or resource class definition
chrsrc	Changes the persistent attribute values of a resource or resource class
refrsrc	Refreshes the resources within the specified resource class
lsactdef	Lists (displays) action definitions of a resource or resource class

The ERRM CLI provides system administrators with a command line alternative to the Web-based System Manager tool to control monitoring on your system. These commands allow you to affect monitoring by creating conditions, responses, and associations between them. The ERRM CLI consists of the commands shown in Table 3-12.

Table 3-12 ERRM commands

Commands	Description
mkcondition	Creates a new condition definition that can be monitored
rmcondition	Removes a condition
chcondition	Changes any of the attributes of a defined condition

Commands	Description
lscondition	Lists information about one or more conditions
mkresponse	Creates a new response definition with one action
rmresponse	Removes a response
chresponse	Adds or deletes the actions of a response or renames a response
lsresponse	Lists information about one or more responses
rmcondresp	Deletes a link between a condition and one or more responses
mkcondresp	Creates a link between a condition and one or more responses
stopcondresp	Stops monitoring a condition that has one or more linked responses
lscondresp	Lists information about a condition and its linked responses, if any

The following example is an output generated from some of the ERRM commands:

```
# startcondresp "/tmp space used" "Critical notifications" "E-mail root
anytime"
```

```
# lscondition | more
Displaying condition information:
Name                               MonitorStatus
"Processes in swap queue"          "Not monitored"
"Processes in run queue"           "Not monitored"
"/var space used"                  "Not monitored"
"/tmp space used"                  "Monitored"
"File system space used"           "Not monitored"
```

```
# lscondresp "/tmp space used"
Displaying condition with response information:
```

```
condition-response link 1:
  Condition = "/tmp space used"
  Response  = "E-mail root anytime"
  State     = "Active"

condition-response link 2:
  Condition = "/tmp space used"
  Response  = "Critical notifications"
  State     = "Active"
```

For additional information, see *Reliable Scalable Cluster Technology Version 2 Release 1 Resource Monitoring and Control Guide and Reference*, SC23-4345.

3.7.7 RSCT NLS enablement (5.2.0)

As was the case with Version 5.1, the rsct.basic.* filesets are shipped with installation media but are not installed as default. The install of applications including HACMP/ES and GPFS for AIX clusters results in the basic* filesets being installed.

The key NLS enhancement is to topology and group services, which are now NLS enabled. This means that debugging information from these services can be displayed in all the current AIX-supported languages.

3.8 Cluster System Management

Cluster System Management in Version 5.2 provides the ability to manage a loose cluster of AIX and Linux servers through a single point, called the cluster manager. Source code is common to both AIX and Linux.

3.8.1 Overview

This section discusses cluster systems management (CSM) for AIX only. CSM has been developed to provide equivalent functionality for Linux although this is beyond the scope of this publication. CSM provides many functions and these are discussed in the following section.

Domain management

The distributed management server resource manager resides on the cluster manager node and contains the following resource classes:

- ▶ **Managed Node**
Contains persistent and dynamic attributes for each node
- ▶ **Node Group**
Contains node group definitions and provides events describing node group changes
- ▶ **Node Authenticate**
Provides a mechanism for nodes to request to be added to the CSM domain
- ▶ **Node Hardware Control and Hardware Control Point**
Maintains attributes and actions needed for hardware control in the cluster

EERM

Enables the administrator to define conditions to watch for in the cluster and appropriate response scripts to invoke for these events. RMC is used to communicate with resource classes to all the nodes. The nodes register for the events and when the event occurs EERM runs the appropriate response script, as defined by the administrator. Logging is made to the audit log.

Hardware control

The **rpower** command talks to the hardware control resource class to query information and perform actions. The hardware control resource class communicates with the service processor on each of the machines using the hardware control point (HMC for AIX p690). The resource class can perform operations on the client nodes.

Remote console

The **rconsole** communicates with the console server to open a console session on a node. AIX p690 uses the HMC.

Distributed shell

The **dsh** command uses either **rsh** (default) or **ssh** (user configured) to run commands on specified nodes. **dsh** calls **lsmnode** and **nodegroup** to get node information as required.

Probe manager

The diagnostic probes component constitutes a probe manager and a set of probes. The probe manager is responsible for running the probes and returning the result. The probes are run on each node to check for software problems.

CFM

Configuration file manger (CFM) can be used to place files in /cfmroot on the management server. CFM used **rdist** to distribute the files to the managed nodes. **rdist** uses **rsh** or **ssh** (if configured). The command runs whenever /cfmroot is updated and also periodically. CFM places failed nodes in a group by using the DMS RM and EERM.

Installation

For information on Cluster System Management installation, see:

<http://rs6000.pok.ibm.com/afs/aix.kingston.ibm.com/project/csm/www/home.html>

Centralized logging

EERM conditions are configured to watch for log entries from each node using the Log Watcher resource class. This is transferred to EERM on the cluster manager using the RMC event response. EERM logs the events in the Audit log.

CSM database

The CSM database is an ODBC-compliant database that is used to store information referring to the CSM cluster.

3.8.2 Hardware control and integration

CSM provides additional support capabilities of the Hardware Management Console (HMC) for pSeries systems and Netfinity systems.

CSM Version 1.3, running on AIX 5L Version 5.2, provides the following capabilities to HMC-attached systems:

- ▶ Multiple read consoles in addition to the previous implementation of a single write console.
- ▶ **ping** test, using getadapters network discovery. This function returns the MAC address, speed, and duplex information of the first or, if specified, all network adapters that respond to the **ping**.
- ▶ Support for hardware control point event notification of power status changes, where the HMC provides event notification when power status changes.
- ▶ Management server CIMOM client is now able to use the SSL protocol for communications with the HMC if **ssh** is configured over **rsh**.
- ▶ Remote network boot of CSM client machines using the HMC. This is particularly useful for NIM installations and general system administration.

3.8.3 AIX consumability

Consumability concerns the ability to feed information from the CSM to an Enterprise Management System such as Tivoli, as well as the ability to send alerts to administrators. Simple Network Management Protocol (SNMP) is the chosen enablement mechanism.

SNMP overview

SNMP is used by networked hosts to exchange information in the management of these devices. Each host runs a SNMP daemon called SNMPd, which maintains the management information base (MIB) for that host. The MIB is a database containing all the information pertinent to a system.

Use of SNMP

A manager is a client application that requests MIB information and processes the responses. The management application may send a request to modify MIB information and also process the raw MIB data into a user-friendly output.

Version 5.2 also ships the SNMPv3 with enhanced security.

SNMP traps are event reports or notifications of a system event, generated as they happen. A trap can be generated by an event to the manager. The manager can then respond by calling a program, which may report to a management tool such as Tivoli, page support, or e-mail the administrator.

To allow enterprise management systems to react to defined events, ERRM generates SNMP traps.

3.8.4 Interoperability between AIX and Linux

Interoperability refers to the ability to support both AIX and Linux in the same CSM cluster. AIX 5L Version 5.2 supports this configuration, with one caveat, that the cluster management server is installed with AIX. If the cluster management server runs on Linux it is only possible to have Linux client nodes in the cluster.

In a mixed cluster it will be possible to perform a CSM-only install on both AIX and Linux nodes. However, it will only be possible to do a full installation, including operating system and CSM installation, on AIX nodes.

It is possible to perform the following administrative functions with a combination of AIX and Linux managed nodes, with an AIX cluster manager server:

- ▶ Distribute commands to nodes in the cluster.
- ▶ Use configuration file manager to synchronize files.
- ▶ Monitor conditions across nodes in the cluster and action responses.
- ▶ Remotely power on and off nodes in the cluster.
- ▶ Perform CSM install to all nodes in the cluster.
- ▶ Software diagnostics to all nodes in the cluster.
- ▶ Predefine responses to generated SNMP events.
- ▶ Common set of RMC, ERRM, and RSCT.

Storage management

AIX 5L introduces several new features for the current and emerging storage requirements. These enhancements include Multipath I/O, improved disk handling by the LVM, JFS2, NFS enhancements, and Veritas support. There is also automatic mounting of CD-ROM material, and a new storage management API.

4.1 Multipath I/O (5.2.0)

AIX 5L Version 5.2 provides a new feature called Multipath I/O (MPIO) that allows for a single device (disk, lun) to have multiple paths through different adapters. These paths must reside within a single machine or logical partition of a machine. Multiple machines connected to the same device are considered as clustering and not as MPIO.

There are three reasons for MPIO:

- ▶ Performance improvement
- ▶ Improved reliability and availability
- ▶ Easier administration

MPIO, part of the base kernel, and is described in the following.

4.1.1 MPIO device driver overview

The device driver and device methods have been modified to support detection, configuration, and management of the device on these paths. The path management functions consist of two modules, a kernel extension (PCMKE), and a run-time loadable configuration module (PCMRTL). The PCMKE I supplies path control management capabilities to a device driver that has been modified to support a defined set of interfaces. The runtime loadable configuration module will provide additional abilities to the device methods to access ODM attributes that the PCMKE needs for initialization

In a multipath I/O subsystem, any device may have one or more paths to it. PCMKE routing depends on device configuration to detect paths and communicate that information to the device drivers. Each MPIO-capable device driver adds the paths to a device from its immediate parent(s). When an I/O request is sent to a device, the device driver must decide which path should be used for that request. The maintenance and scheduling of I/O across different paths is provided by the PCMKE and is transparent to the MPIO-capable device driver. The PCMKE module provides routing algorithms that are user selectable. The PCMKE facilitates the collection of information useful for determining the best path for any I/O request to be sent as well as actual selection of that path. The PCMKE may select the best path based on a variety of criteria including load balancing, connection speed, and connection failure, to name a few.

In general, it is the device driver's responsibility to manage the paths and to select the path on which to queue commands. The design for MPIO support allows for any device that can be uniquely identified to be an MPIO device.

However, the initial release of MPIO only supports SCSI scsd. Additional devices may be added in the future.

While it is the device driver's responsibility to perform *path management*, the MPIO design allows for this functionality to be split from the driver such that it is performed by the PCMKE. The device driver must be written in such a manner as to allow for this separation. If the device driver is not written in this manner, it must perform the path control management functionality internally.

The ability to have a separate PCMKE is being done to make it easier for third party disk vendors, such as EMC or IBM Storage Group, to adopt the AIX MPIO solution. These vendors make use of the AIX SCSI and Fibre Channel disk device drivers, but have their own implementations for path management. The respective AIX disk drivers are being modified to off load path management into a separate PCMKE.

4.1.2 MPIO concepts

It is already possible, without MPIO, to have access to a single device through different adapters using vendor modules. For example, the subsystem device driver (SDD) for ESS (IBM storage). In this case an *hdisk* is created for each path and SDD is in charge of handling the path management.

The disadvantages of the way that those subsystem drivers work are:

- ▶ They are sometimes firmware dependant.
- ▶ Each subsystem has to be administrated differently.
- ▶ Each path generates a logical device entry in the ODM.
- ▶ A dedicated command must be used to create and manipulate volume group.

Figure 4-1 on page 170 represents the behavior of AIX with a non-MPIO single disk accessed by three adapters.

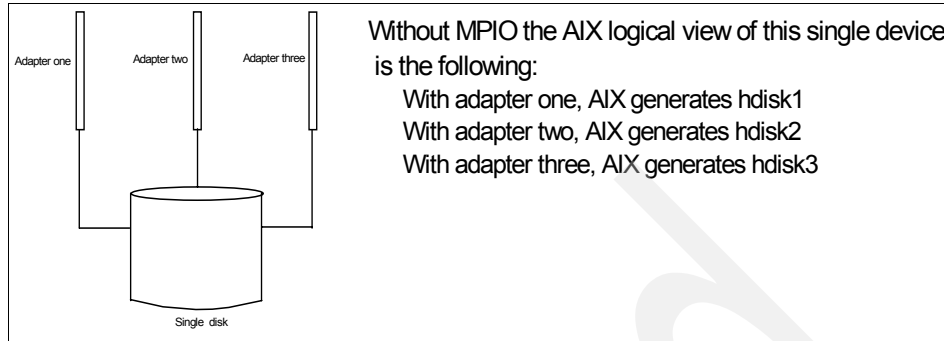


Figure 4-1 Three adapters connected to a single device without MPIO facility

The main difference with MPIO is that one MPIO device or *hdisk* can have multiple paths to its parents (adapter) with a single entry in the ODM. It is also possible to use all the common AIX commands to administrate the volume group, including MPIO devices. But to handle this new feature, changes have been made in the AIX device driver.

Figure 4-2 represents the behavior of AIX with a MPIO single disk accessed by three adapters.

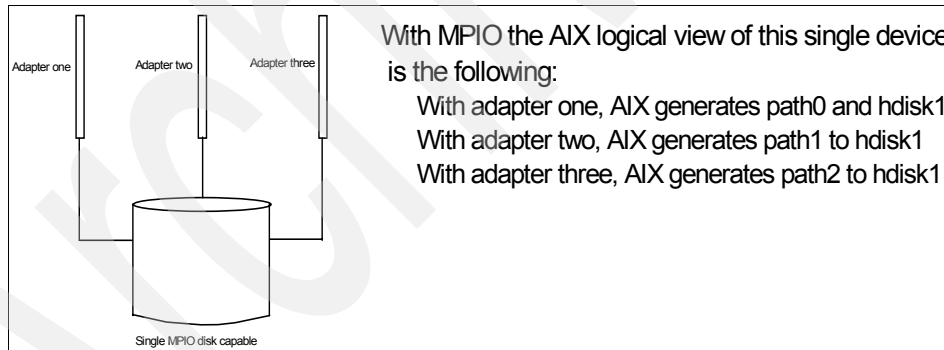


Figure 4-2 Three adapters connected to a single MPIO device

To understand clearly how MPIO works, we need to better understand the following concepts, which are the topic of the following sections.

- The unique device identifier (UDID)
- The reservation policy of MPIO

Unique device identifier

Every MPIO-capable device must provide a unique identifier that allows the device to be distinguished from any other device in the system. This identifier is called the unique device ID, or UDID for short. The UDID value for a particular device is stored as an attribute of the device in the device configuration database. A UDID is viewed by the system as a string of characters that have no implicit meaning other than the one UDID can be compared against another.

UDIDs have different formats depending upon the device from which the UDID was obtained.

When the **cfgmgr** command or when a parent device's configure method is running, it requests the UDID for the child. The UDID is compared with the UDIDs stored in ODM to determine the action to take:

- ▶ A newly discovered device needs to be defined into the system
- ▶ The device already exists and only a new path needs to be defined

For the first release of MPIO, only devices with a Subclass of `scsi` may be supported for MPIO. Each of these device Subclasses has a different UDID format.

Device reservation policy

For a single device, MPIO is able to handle four types of reservation policy.

NO_RESERVE

In this mode the path algorithm of MPIO can support I/O on a single path (fail_over mode) or I/O distributed across multiple paths (load balance mode). This setting would best be used in an HACMP concurrent mode cluster or a third-party product with similar locking capabilities. This mode should not be used in a cluster without a clustering software product, there is no reservation protection of the target device provided, and in a multinode environment there is a high potential for data corruption.

SINGLE_PATH

In this mode the path algorithm of MPIO can support I/O on a single path (fail_over mode). This setting would best be used in a cluster where the device is owned by only one node in the cluster and would fail over to an alternate node should the owning node fail. MPIO in this case will provide improved reliability in the case of an adapter or connectivity failure within the node owning the device. MPIO will not provide any performance improvement.

PR_EXCLUSIVE

In this mode the path algorithm of MPIO can support I/O on a single path (fail_over mode) or I/O distributed across

multiple paths (load balance mode). This setting would best be used in a cluster where the device is owned by only one node in the cluster and would fail over to an alternate node should the owning node fail. If the MPIO path algorithm is set to a load balance mode, the I/O will be spread across multiple paths, which may provide higher performance. If the MPIO path algorithm is set to fail_over mode then the I/O will only be processed down a single path and the system will perform the same as the previous case with reserve_policy.

PR_SHARED

In this mode the path algorithm of MPIO can support I/O on a single path (fail_over mode) or I/O distributed across multiple paths (load balance mode). This setting would best be used in a cluster where the device is owned by one or more nodes in the cluster. If the MPIO path algorithm is set to a load balance mode the I/O from the host will be spread across multiple paths, which may provide higher performance. If the MPIO path algorithm is set to fail_over mode then the I/O will only be processed down a single path and the system will perform the same as the previous case with reserve_policy.

4.1.3 Detecting an MPIO-capable device

In order for a SCSI device to be detected as an MPIO-capable device, additional PdAt ODM attributes are added to a device's predefines. The UDID ODM attribute is required by all MPIO-capable devices (see "Unique device identifier" on page 171).

In addition, a PCM ODM attribute needs to be added to the device PdAt ODM predefines. The PCM ODM attribute points to a ODM *friend*, which will define the PCMKE module that will provide the path control management capabilities for the MPIO-capable device driver. The PCM ODM attribute may contain the name of a vendor-provided PCMKE or the AIX-provided PCMKE.

An example of a SCSI disk that has only one path but it is considered as MPIO capable is as follows:

```
# lsattr -El hdisk5
pvid          0001810fd3838c5e0000000000000000    Physical volume identifier
False
queue_depth    3                                  Queue DEPTH
False
size_in_mb     9100                               Size in Megabytes
False
```

max_transfer	0x40000	Maximum TRANSFER Size
True		
unique_id	2308ZD1GY3950CDPSS-309170M03IBMscsi	Unique device identifier
False		
PR_key_value	none	Size in Megabytes
True		
reserve_policy	single_path	Size in Megabytes
True		
PCM	pcm/aixdisk/scsd	Target NAME
True		
dvc_support		Device Support
False		
algorithm	fail_over	Algorithm
True		
#		

You can see in the above example the unique_id and the PCM field that point to the /pcm/aixdisk/scsd AIX driver.

Important: Note that you can have multiple paths between one adapter and one device. This will be the case, in the future, if a SAN switch is put between the Fibre Channel adapter and a disk subsystem.

4.1.4 ODM changes for MPIO device

New ODM entries are needed to use MPIO. They are discussed in the following sections.

New CuPath class

The CuPath ODM class is being added to hold definitions of paths. This class is roughly analogous to the CuDv class; the CuPath class identifies paths while the CuDv class identifies devices. The CuPath object contains all the information needed to uniquely identify a path. This information includes the name of the target (child) device, the name of the parent device, and the connection point on the parent.

New PdPathAt class

The PdPathAt ODM class is being added to hold predefined attributes that apply to paths. If a device or the friend of a device has attributes that pertain to paths, there must be a PdPathAt attribute to define the attribute. A path-specific attribute cannot be created if there is not a PdPathAt definition for the attribute. This PdPathAt class is roughly analogous to the PdAt class.

New CuPathAt class

The CuPathAt ODM class is being added to hold attributes that apply to specific paths. This class is roughly analogous to the CuAt class. An object cannot be created in the CuPathAt class if there is not a path object in the CuPath class to which the CuPathAt attribute applies, just like the relationship between CuAt objects and CuDv objects. Furthermore, a PdPathAt object must also exist to provide default and other information about the attribute just like the relationship between the CuAt and the PdAt attributes.

PdAt class change

There are three new PdAt attributes that will need to be added to all SCSI and Fibre Channel devices that will be supported as MPIO-capable devices. These attributes are `unique_id`, `PCM`, and `reserve_policy`. The PCM ODM attribute will be a reference to the ODM friend, which contains the path to the PCMKE module. It is expected that device vendors such as EMC, HDS, or LSI supply modified ODM predefines as they convert their disk subsystems to be MPIO capable.

4.1.5 Path management

Four new AIX commands have been added to AIX 5L Version 5.2 to manage the device path, as discussed in the following sections.

The mkpath command

When using the **mkpath** command to define a new path that does not exist, all components of the path must be supplied: The target device, the parent device, and the connection on the parent. Note that any device that cannot be manually defined using the **mkdev** command will not be able to have paths manually defined to using the **mkpath** command. These limitations are both due to the way that path information is stored for these devices. Fibre Channel devices fall into this category.

When using **mkpath** to configure already defined paths, all of the components of a path are not required. Since the paths already exist, some of the components of the path can be left out.

The syntax of the **mkpath** command is as following:

```
mkpath [ -l Name ] [ -p Parent ] [ -w Connection ] [ -d ]
```

or

```
mkpath -h
```

The commonly used **mkpath** command flags are provided in Table 4-1 on page 175.

Table 4-1 The **mkpath** command flags

Flags	Description
-l	The name of the device
-p	The name of the parent adapter
-w	The connection information associated with the path to be added
-d	Defines a new path to the device by adding a path definition to the system
-h	Indicates the mkpath command syntax

In the following example the status of an existing path is changed from disabled to enabled.

```
# > lspath -l hdisk9
Enabled hdisk9 scsi1
Defined hdisk9 scsi2
[root@kenmore] /
# > mkpath -l hdisk9 -pscsi2
paths Available
[root@kenmore] /
# > lspath -l hdisk9
Enabled hdisk9 scsi1
Enabled hdisk9 scsi2
```

4.1.6 The **rmpath** command

The **rmpath** command unconfigures, undefines, or both unconfigures and undefines one or more paths to a specific target device. Only the target device is required by the **rmpath** command. Similar to the **mkpath** command, this capability allows the **rmpath** command to operate on multiple paths in a single invocation. For example, to unconfigure all paths between a specific target device and a specific parent device, only the target device and the parent device need be specified. It is *not* possible to attempt to unconfigure (undefine) the last path to a target device using the **rmpath** command. The only way to unconfigure the last path to a device is to unconfigure the device itself (for example, use the **rmdev** command).

The syntax of the **rmpath** command is as follows

```
rmpath [ -l Name ] [ -p Parent ] [ -w Connection ] [ -d ] [ -p ]
```

or

`rmpath -h`

The commonly used **rmpath** command flags are provided in Table 4-2.

Table 4-2 The *rmpath* command flags

Flags	Description
-l	Is the name of the device.
-p	The name of the parent adapter.
-w	Is the connection information associated with the path to be added.
-d	Delete the path from ODM.
-h	Indicate the mkpath command syntax.

The following example shows the two paths of `hdisk9` from its `scsi1` and `scsi2` parents.

```
#lspath -l hdisk9
Enabled hdisk9 scsi1
Enabled hdisk9 scsi2
#
```

To delete the path from the `scsi2` parent from the ODM, enter the following command.

```
#rmpath -l hdisk9 -p scsi2 -d
path deleted
#
```

Enter the **lspath** command again to show that now `hdisk9` has only one path:

```
#lspath -l hdisk9
Enabled hdisk9 scsi1
#
```

To recreate the path from the adapter `scsi2` to `hdsik3`, enter the following command.

```
#cfgmgr -l scsi2
```

The device `hdisk9` has recover its second path from the `scsi2` parent, as in the following.

```
#lspath -l hdisk9
Enabled hdisk9 scsi1
Enabled hdisk9 scsi2
```

4.1.7 The **lspath** command

The **lspath** command displays one of two types of information about paths to an MPIO capable device. It either displays the operational status for one or more paths to a single device, or it displays one or more attributes for a single path to a single MPIO capable device. The **lspath** command syntax is the following:

```
lspath [ -F Format ] [ -H ] [ -l Name ] [ -p Parent ] [ -s Status ]  
[ -w Connection ]
```

or

```
lspath -A -l Name -p Parent [ -w Connection ] { -D [ -O ] | -E [ -O ] | -F  
Format } [ -a Attribute ] ... [ -f File ] [ -h ] [ -H ]  
lspath -A -l Name -p Parent [ -w Connection ] { -D [ -O ] | -F Format }  
[ -a Attribute ] ... [ -f File ] [ -h ] [ -H ]  
lspath -A -l Name -p Parent [ -w Connection ] -R -a Attribute [ -f File ]  
[ -h ] [ -H ]
```

or

```
lspath -h
```

The commonly used **lspath** command flags are provided in Table 4-3 on page 178.

Table 4-3 The `lspath` command flags

Flags	Description
-a	Identifies the specific attribute to list
-A	Lists the attributes for a specific path
-D	Lists the default values, descriptions, and attribute names of a path
-E	Lists current values, descriptions, and attribute names of a path
-F	Displays the output of a path attribute in a user-specified format, where the format parameter is a quoted list of column names such as <i>parent connection path_id</i> separated by non-alphanumeric characters or white space.
-H	Displays headers above the column output
-O	Displays all attribute names separated by colons and, on the second line, displays all the corresponding attribute values separated by colons.
-R	Displays the legal values for an attribute name
-f	Reads the flags from File parameters
-l	The name of the device
-p	The name of the parent adapter
-w	The connection information associated with the path to be added

With the `lspath` command you can display the status of a path. This status can take different values:

- enabled** Indicates that the path is configured and operational. The path is selectable for I/O.
- disabled** Indicates that the path is configured, but not currently operational. It has been manually disabled and is not selectable for I/O.
- failed** Indicates that the path is configured, but an I/O failure occurs and the path is no longer usable for I/O operations.
- defined** Indicates that the path is configured into the device driver.
- missing** Indicates that the path was defined in a previous boot, but it was not detected in the most recent boot of the system.

detected Indicates that the path was detected in the most recent boot of the system, but for some reason it was not configured. A path should only have this status during boot and so this status should never appear as a result of the **lspath** command.

An example of how to list all the paths defined on the system is as follows:

```
#lspath
```

An example to display all the defined paths is as follows:

```
# > lspath -s defined
Defined hdisk8 scsi2
Defined hdisk9 scsi2
```

An example to display the priority of a device's path, for example, hdisk9 with scsi2 parent, is as follows:

```
# > lspath -AEH -l hdisk9 -p scsi2
attribute value description user_settable
priority 1 Priority True
```

An example to display the name of the device, the parent, the path_id, the connection, and the status of a device path, is as follows:

```
# > lspath -l hdisk9 -H -p scsi2 -F "device parent path_id connection status"
device parent path_id connection status

hdisk9 scsi2 1 14,0 Enabled
```

An example of how to display the allowed value of a path attribute, in this case the priority, is as follows:

```
# > lspath -A -l hdisk9 -p scsi2 -R -a priority
1...255 (+1)
```

4.1.8 The **chpath** command

The **chpath** command is used to perform two different change operations on a specific path. It is used to change the operational status of a path and to change tunable attributes associated with a path. The **chpath** command cannot perform both types of operations in a single invocation.

The operational status of a path is basically a flag indicating whether the path should be used when selecting a path for I/O. If the path is disabled, it is not used in path selection. If it is enabled, it is used for path selection. A path is automatically enabled when it is configured.

When changing path-specific tunable attributes, the **chpath** command is very similar to the **chdev** command.

The syntax is:

```
chpath -l Name -s OpStatus [ -p Parent ] [ -w Connection ]
```

or

```
chpath -l Name -p Parent [ -w Connection ] [ -P ] -a attribute=Value [ -a attribute=Value ... ]
```

or

```
chpath -h
```

The commonly used **chpath** command flags are provided in Table 4-4.

Table 4-4 The *chpath* command flags

Flags	Description
-h	Indicates the chpath command syntax
-l	The name of the device
-p	The name of the parent adapter
-w	The connection information associated with the path to be added
-s	The status of the path
-a	The attribute of the path

With the **chpath** command, you can enable or disable a path when this path is already defined to the system.

You can also change the attribute of a path such as priority from 1 to 255. By default, when the path is created the priority is set to 1, which is the highest priority. If a device has several paths, the priority determines the way that the system initiates the I/O to the device.

Consider two scenarios. In each scenario you have one device with three paths. The path1 with priority 1, the path2 with priority 100, and the path3 with priority 10.

- ▶ If the device is set with the *fail_over* algorithm, then I/O will be done through path1 because it has the highest priority. If this path fails then the I/O will be initiated to path3 because path3 has a higher priority than path2, and so on.
- ▶ The device with *round_robin* algorithm: The sum of the I/O of path1 will be 10 times the sum of the I/O of path3 and 100 times of path2. If one path fails, the system will compare the priority between the second-to-last one.

The following example shows how to change the priority of a path:

```
chpath -l hdisk9 -p SCSI2 -a priority=10
```

4.1.9 Device management

MPIO-capable devices can be managed with two main attributes:

- ▶ The multipath I/O algorithm
- ▶ The reserve policy (see “Device reservation policy” on page 171).

The multipath algorithm handles how the I/O is directed to the paths of a device. The *fail_over* algorithm directs I/O down a single path until the path fails, then an alternate single path is selected for all I/O (see the priority **chpath** command in 4.1.8, “The chpath command” on page 179). The *round_robin* algorithm directs all I/O down all paths depending on the priority of the path (see the priority of the **chpath** command in 4.1.8, “The chpath command” on page 179). To list the device attributes use the **lsattr** command.

The following example shows the attribute of **hdisk9** with a *round_robin* algorithm and a *no_reserve* policy:

```
# > lsattr -El hdisk9
pvid          none                Physical volume identifier
False
queue_depth    3                  Queue DEPTH
False
size_in_mb     9100               Size in Megabytes
False
max_transfer   0x40000            Maximum TRANSFER Size
True
unique_id      23084DYET6800CDDYS-T09170M03IBM scsi Unique device identifier
False
PR_key_value   none              Size in Megabytes
True
reserve_policy no_reserve        Size in Megabytes
True
PCM            pcm/aixdisk/scsd   Target NAME
True
dvc_support    False             Device Support
False
algorithm      round_robin        Algorithm      True
# >
```

The following command shows how to set the algorithm to *fail_over* for **hdisk9**:

```
chdev -l hdisk9 -a algorithm=fail_over
```

A SMIT panel has been added to handle the MPIO devices, as shown in Figure 4-3.

```
Aix SCSI/FCP Disk PCM Change Device Characteristics

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

Device Name      [Entry Fields]
Device Type      hdisk9
Path Control Module scsd
Algorithm        aixdisk
Reservation Policy fail_over      +
                  no_reserve     +

F1=Help          F2=Refresh      F3=Cancel        F4=List
F5=Reset          F6=Command      F7=Edit          F8=Image
F9=Shell          F10=Exit        Enter=Do
```

Figure 4-3 This panel shows the device of `hdisk9`

To list all the devices under a parent, first select the parent **scsi2** in the panel (shown in Figure 4-4 on page 183).

List MPIO Devices under a Parent

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

* Parent Name

[Entry Fields]

[scsi2]

+

F1=Help

F2=Refresh

F3=Cancel

F4=List

F5=Reset

F6=Command

F7=Edit

F8=Image

F9=Shell

F10=Exit

Enter=Do

Figure 4-4 Selection of a parent

Then display the devices under the parent, as shown in Figure 4-5.

COMMAND STATUS

Command: OK stdout: yes stderr: no

Before command completion, additional instructions may appear below.

hdisk1 Available 10-70-00-1,0 16 Bit LVD SCSI Disk Drive

hdisk2 Available 10-71-00-2,0 16 Bit LVD SCSI Disk Drive

hdisk4 Available 10-71-00-4,0 16 Bit LVD SCSI Disk Drive

hdisk5 Available 10-71-00-5,0 16 Bit LVD SCSI Disk Drive

hdisk6 Available 10-71-00-11,0 16 Bit LVD SCSI Disk Drive

hdisk7 Available 10-71-00-12,0 16 Bit LVD SCSI Disk Drive

hdisk8 Available 10-71-00-13,0 16 Bit LVD SCSI Disk Drive

hdisk9 Available 10-71-00-14,0 16 Bit LVD SCSI Disk Drive

F1=Help

F2=Refresh

F3=Cancel

F6=Command

F8=Image

F9=Shell

F10=Exit

/=Find

n=Find Next

Figure 4-5 List the all the devices under a parent

To list all the parents of an MPIO device, first select the device **hdisk9**, as shown in Figure 4-6.

```

List Parents for an MPIO Device

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

* Device Name [Entry Fields] [hdisk9] +

F1=Help      F2=Refresh    F3=Cancel    F4=List
F5=Reset     F6=Command    F7=Edit      F8=Image
F9=Shell     F10=Exit      Enter=Do

```

Figure 4-6 Selection of a device, *hdisk9* in this example

Then list the parent of a device, as shown in Figure 4-7.

```

COMMAND STATUS

Command: OK      stdout: yes      stderr: no

Before command completion, additional instructions may appear below.

scsi1 Available 10-70 Wide/Ultra-2 SCSI I/O Controller
scsi2 Available 10-71 Wide/Ultra-2 SCSI I/O Controller

F1=Help      F2=Refresh    F3=Cancel    F6=Command
F8=Image     F9=Shell     F10=Exit     /=Find
n=Find Next

```

Figure 4-7 Displays the parent of *hdisk9*

Changes have also been made to existing AIX commands to support the MPIO devices. For examples, the **mkdev**, **rmdev**, and **bootlist** commands have been enhanced:

- ▶ When the **mkdev** command is configuring an MPIO-capable device, it requests the associated device driver to configure all known paths to the device. If all the paths are available, the output of the command is the same as before, `hdisk9 Available`. But if all device paths cannot be configured, the output of the command is `hdisk9 Available; some paths are not available`.
- ▶ When using the **rmdev -R** command to recursively unconfigure an MPIO device and other configured paths to the device that exist from another parent device, the **rmdev** command will only unconfigure or undefine the path between the device and the parent through which the recursion has occurred. The entire device will not be unconfigured or undefined. In this case, the output of the command is `hdisk9 Available; some paths are not available`.
- ▶ The **bootlist** command allows AIX to save, in NVRAM, information about what devices firmware should be use to boot the system. This information typically includes firmware path information on how to get to the device, starting from the system bus. This command is modified to have MPIO-capable devices listed multiple times (for example, several `hdisk0`) in the boot list area of NVRAM, once for each path to the device that is configured and available when the **bootlist** command is run. The order taken to update the bootlist is the order of the ODM entries.

Note: As the **bootlist** command is not run automatically, if a new path is added to a *boot device*, the system administrator must run the **bootlist** command to have the new path added to the boot list.

The error log entries and the maintenance packages have been enhanced to manage MPIO device problems.

4.1.10 The **iostat** command enhancements

The **iostat** command is enhanced with new parameters that provide a better presentation of the generated reports.

The **-s** flag adds a new line to the header of each statistic's data that reports the sum of all activity on the system.

```
# iostat -s 1 3
System: server1.itsc.austin.ibm.com
           Kbps      tps    Kb_read  Kb_wrtn
9405.3      2351.3    28216      0
```

Disks:	% tm_act	Kbps	tps	Kb_read	Kb_wrtn
hdisk0	46.7	4693.3	1173.3	14080	0
hdisk1	24.0	2356.0	588.7	7068	0
hdisk2	0.0	0.0	0.0	0	0
hdisk3	24.3	2356.0	589.3	7068	0
hdisk4	0.0	0.0	0.0	0	0
cd0	0.0	0.0	0.0	0	0

The -a flag produces an output similar to the -s flag output, with the difference that it provides an adapter basis sum of activities. After displaying the adapter activity, it provides a per-disk basis set of statistics.

```
# iostat -a 1 3
```

tty:	tin	tout	avg-cpu:	% user	% sys	% idle	% iowait
	0.0	923.7		13.2	41.6	30.9	14.2

Adapter:	Kbps	tps	Kb_read	Kb_wrtn
scsi0	7030.4	1757.6	7048	0

Disks:	% tm_act	Kbps	tps	Kb_read	Kb_wrtn
hdisk0	43.9	4684.3	1171.1	4696	0
hdisk1	24.9	2346.1	586.5	2352	0
hdisk2	0.0	0.0	0.0	0	0
cd0	0.0	0.0	0.0	0	0

Adapter:	Kbps	tps	Kb_read	Kb_wrtn
scsi1	2346.1	585.5	2352	0

Disks:	% tm_act	Kbps	tps	Kb_read	Kb_wrtn
hdisk3	19.0	2346.1	585.5	2352	0
hdisk4	0.0	0.0	0.0	0	0

The iostat enhancement for MPIO

The -m option displays statistics about the path activities with the hdisk associated to the path.

For hdisk1 in fail_over mode:

Disks:	% tm_act	Kbps	tps	Kb_read	Kb_wrtn
hdisk1	0.4	3.7	0.5	212080	2041650
Paths:	% tm_act	Kbps	tps	Kb_read	Kb_wrtn
Path0	0.4	3.7	0.5	212080	2041650
Path1	0.0	0.0	0.0	0	0

For hdisk1 in round_robin mode:

Disks:	% tm_act	Kbps	tps	Kb_read	Kb_wrtn
hdisk1	0.4	3.7	0.5	202080	2041650
Paths:	% tm_act	Kbps	tps	Kb_read	Kb_wrtn

Path0	0.4	3.7	0.5	101040	1020825
Path1	0.4	3.7	0.5	101040	1020825

Note: Due to the migration of ESS machines from vpaths to MPIO, the -m flag displays both the MPIO as well as vpath statistics.

The following example shows the **iostat** command output for vpath:

Disks:		Kbps	tps	Kb_read	Kb_wrtn
vpath0		0.6	0.1	10405	35956
Paths:	% tm_act	Kbps	tps	Kb_read	Kb_wrtn
hdisk0	0.0	0.6	0.1	10405	35956
hdisk1	0.0	0.0	0.0	0	0

4.2 LVM enhancements

The following sections contain the enhancements pertaining to the LVM on AIX.

4.2.1 The **redefinevg** command

The **redefinevg** command is rewritten in C to improve performance.

4.2.2 Read-only **varyonvg**

The **varyonvg** command now supports an -r flag that allows a volume group to be varied-on in read-only mode.

4.2.3 LVM hot spare disk in a volume group

The **chpv** and the **chvg** commands are enhanced with a new -h flag that allows you to designate disks as hot spare disks in a volume group and to specify a policy to be used in the case of failing disks. These commands are not replacements for the sparing support available with SSA disks; they complement it. You can also use them with SSA disks when you add one to your volume group.

Note: These new options have an effect only if the volume group has mirrored logical volumes.

There is a new -s flag for the **chvg** command that is used to specify synchronization characteristics.

The following command marks hdisk1 as a hot spare disk:

```
# chpv -hy hdisk1
```

This is only successful if there are not already allocated logical partitions on this disk. Using n instead of y would remove the hot spare disk marker. If you add a physical volume to a volume group (to mark it as a hot spare disk), the disk has to have, at least, the same capacity as the smallest disk already in the volume group.

After you have marked one or more disks as hot spare disks, you have to decide which policy to use in case a disk is starting to fail. There are four different policies you can specify with the -h flag, shown using the following syntax:

```
# chvg -hhotsparepolicy -ssyncpolicy VolumeGroup
```

The following four values are valid for the hotsparepolicy argument:

- y** This policy automatically migrates partitions from one failing disk to one spare disk. From the pool of hot spare disks, the smallest one that is big enough to substitute for the failing disk will be used.
- Y** This policy automatically migrates partitions from a failing disk, but might use the complete pool of hot spare disks.
- n** No automatic migration will take place. This is the default value for a volume group.
- r** This value removes all disks from the pool of hot spare disks for this volume group.

The syncpolicy argument can only use the values y and n.

- y** This will automatically try to synchronize stale partitions.
- n** This will not automatically try to synchronize stale partitions.

The latter argument is also the default for a volume group.

After setting this up, Volume Group Status Area (VGSA) write failures and Mirror Write Consistency (MWC) write failures will mark a physical volume missing and start the migration of data to the hot spare disk.

Web-based System Manager allows for easy configuration of Hot Spare Disk support as discussed in the following sections.

Enabling hot spare disk support in an existing volume group

Properties can be changed on the fly for an existing volume group in order to turn on hot spare disk support for that volume group by enabling the appropriate check box on the Volume Group Properties Dialog panel (Figure 4-8 on page 189).

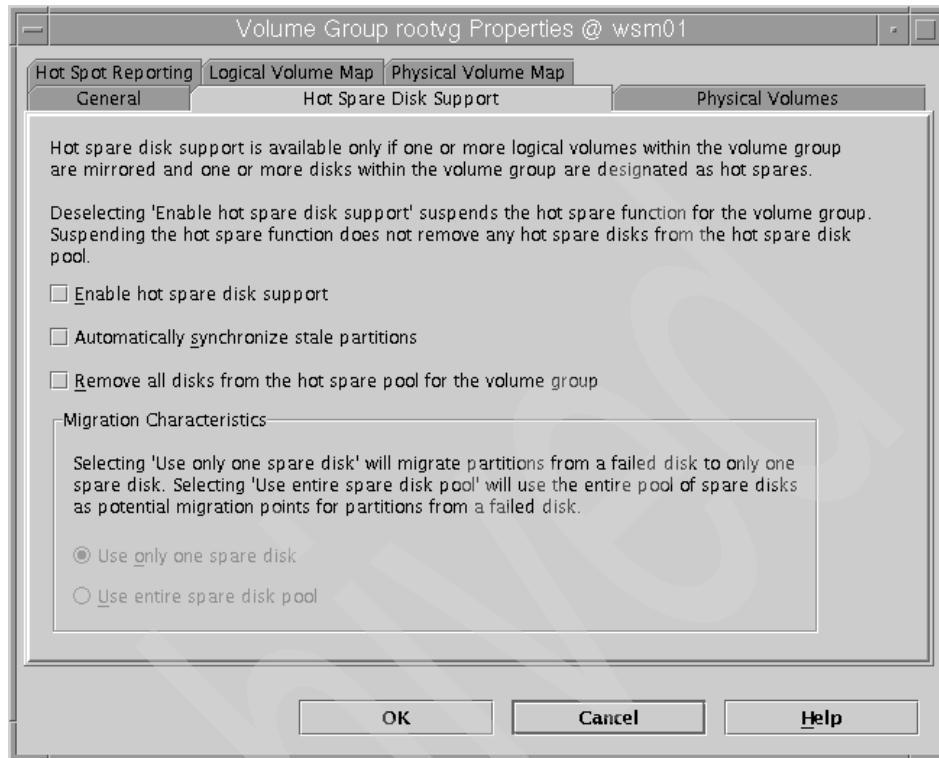


Figure 4-8 Volume Group Properties dialog

After enabling hot spare disk support for a volume group, the Physical Volumes notebook tab of the Volume Group Properties dialog (Figure 4-9 on page 190) allows you to add available physical volumes to the volume group as hot spare disks.

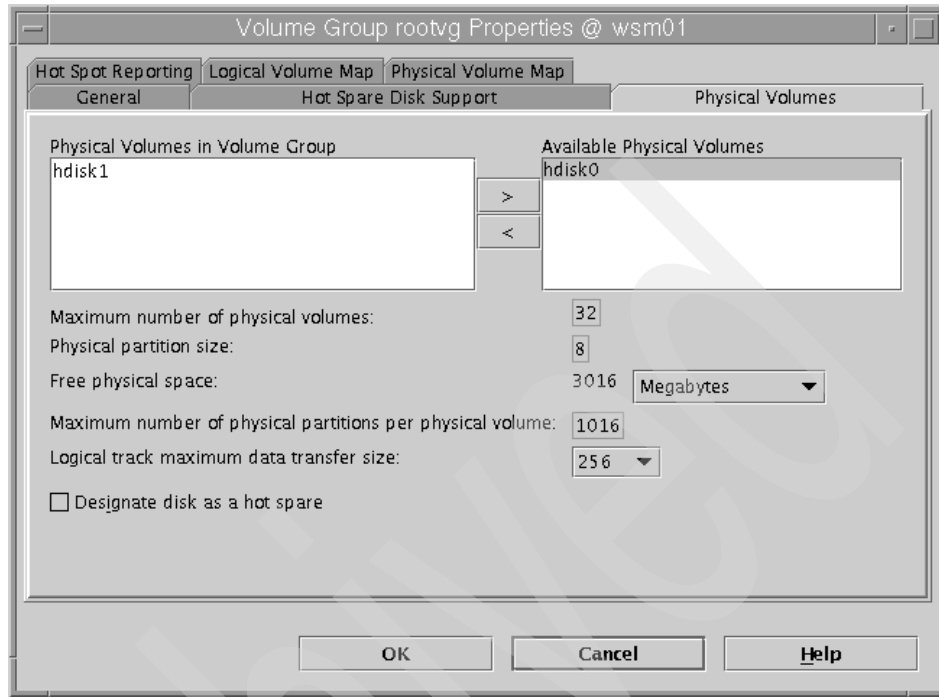


Figure 4-9 Physical Volumes notebook tab

Enabling hot spare during creation of a new volume group

When creating a new volume group in the Web-based System Manager application, the Advanced Method of volume group creation allows you to specify hot spare disk support options (Figure 4-10 on page 191).

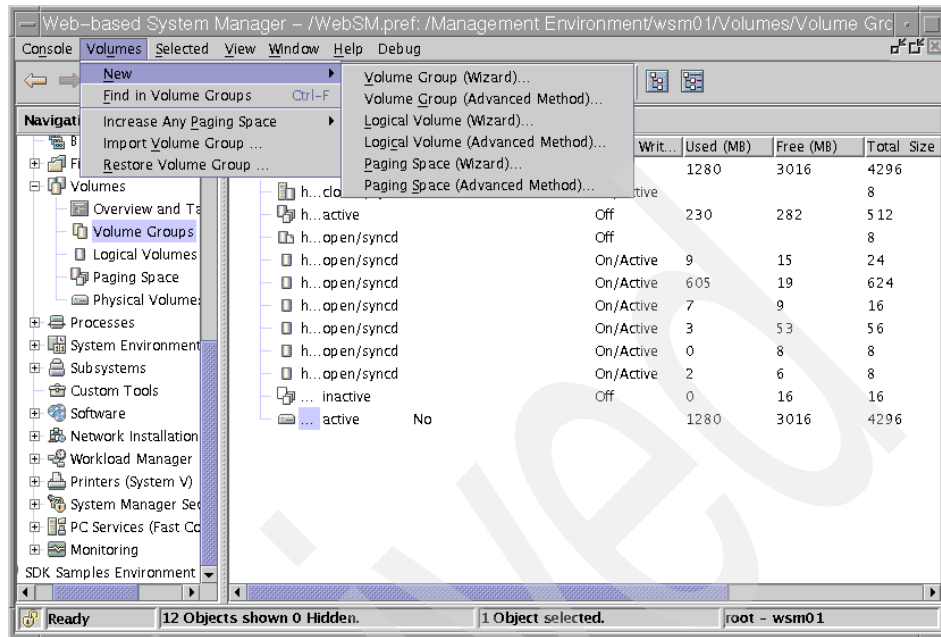


Figure 4-10 Advanced Method of volume group creation

As in previous releases of Web-based System Manager, you assign physical volumes to a volume group, along with a volume group name and any other attributes, such as logical track maximum data transfer size (Figure 4-11 on page 192).

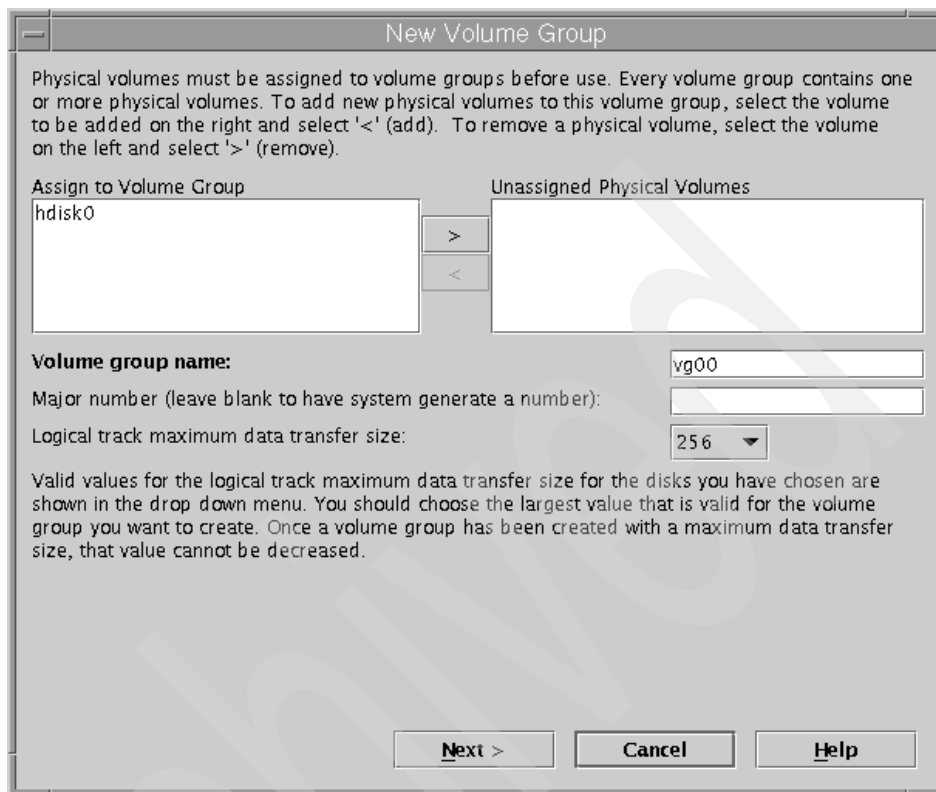


Figure 4-11 New Volume Group dialog

Subsequent panels in the sequential dialog allow configuration of large volume groups (those volume groups as great as 128 physical disks) and allow for support of *big* disks (those with more than 1016 partitions per physical disk), as shown in Figure 4-12 on page 193.

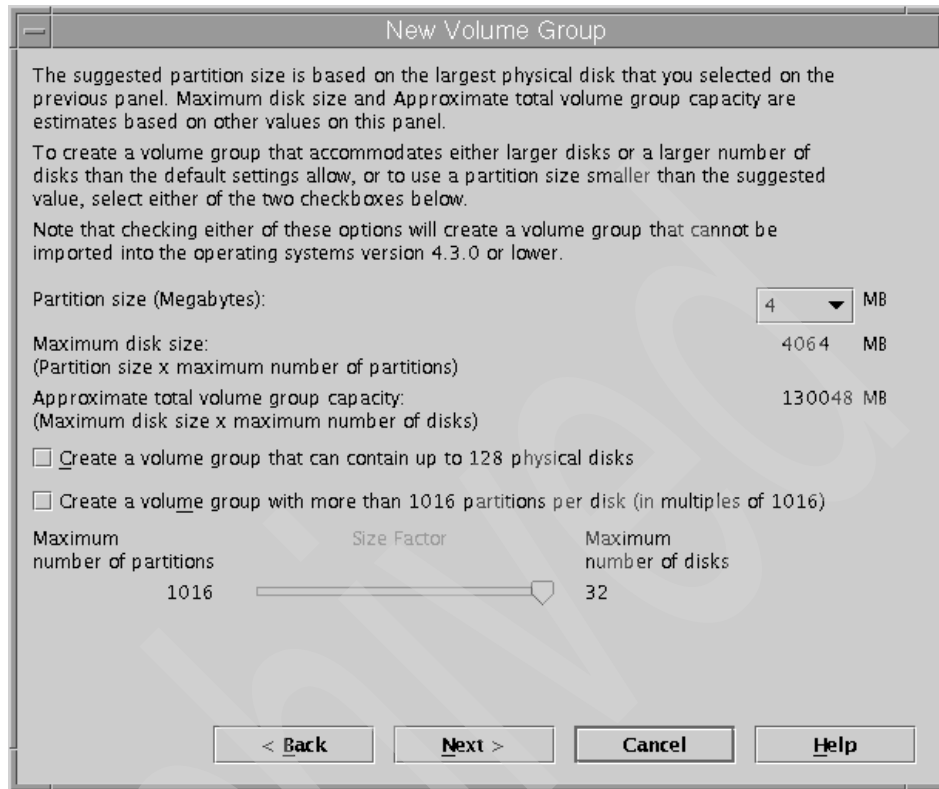


Figure 4-12 New Volume Group, second panel in dialog

The third panel in the new volume group sequence allows you to enable the support for hot spare disks (Figure 4-13 on page 194).

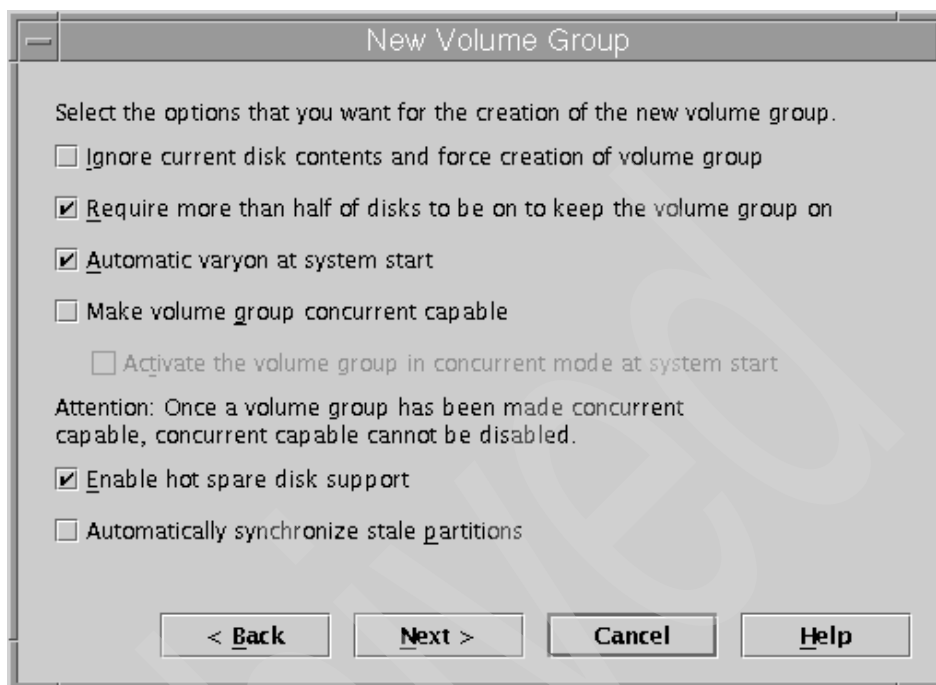


Figure 4-13 New Volume Group, third panel in dialog

The fourth panel allows you to select any unused physical volumes that you may have in your system and assign them to the volume group being created as hot spares (Figure 4-14 on page 195).

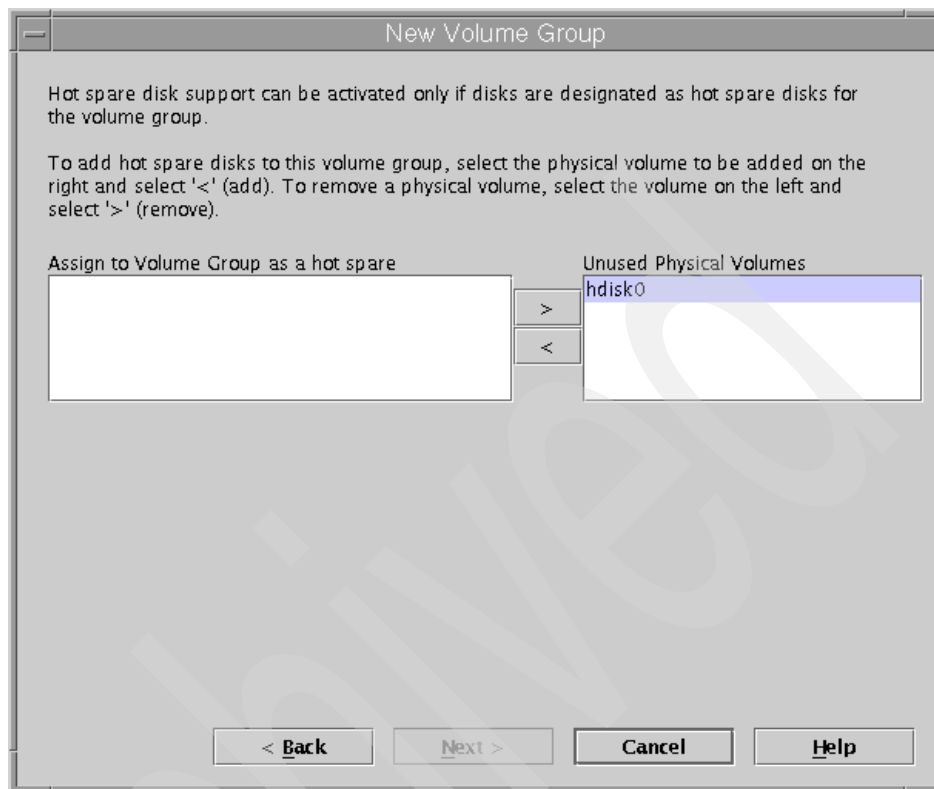


Figure 4-14 New Volume Group, fourth panel in dialog

The fifth panel allows you to set the migration characteristics for the failover from a bad disk to those assigned as hot spares in the hot spare disk pool (Figure 4-15 on page 196).

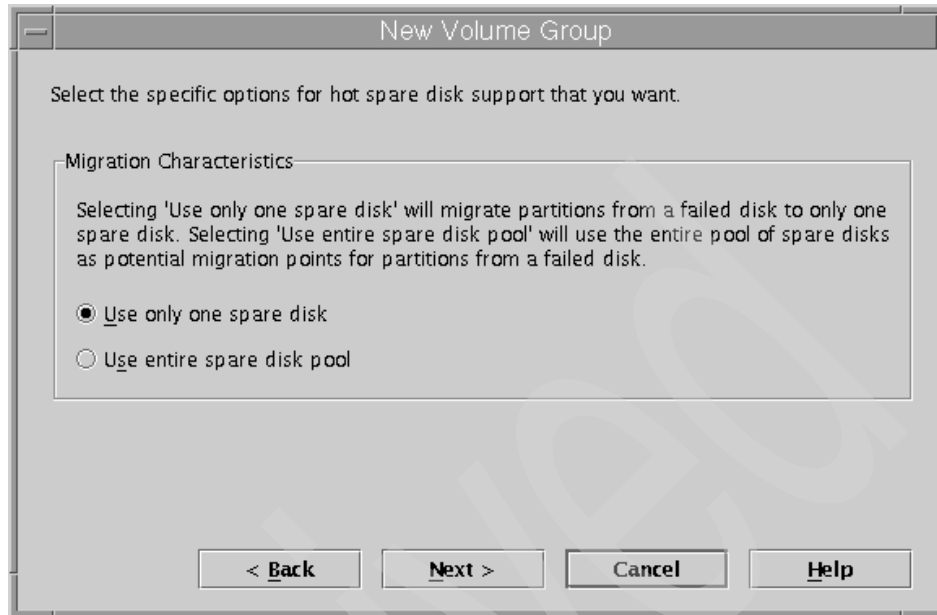


Figure 4-15 New Volume Group, fifth panel in dialog

4.2.4 Support for different logical track group sizes

AIX 5L now supports different logical track group (LTG) sizes. In previous versions of AIX, the only supported LTG size was 128 KB. This is still the default for the creation of new volume groups, even under AIX 5L. You can change this value when you create a new volume group with the **mkvg** command, or later for an existing volume group with the **chvg** command.

The LTG corresponds to the maximum allowed transfer size for disk I/O (many disks today support sizes larger than 128 KB). To take advantage of these larger transfer sizes and get a better disk I/O performance, AIX 5L now accepts values of 128 KB, 256 KB, 512 KB, and 1024 KB for the LTG size, and possibly even larger values in the future. The maximum allowed value is the smallest maximum transfer size supported by all disks in a volume group. The **mkvg** SMIT screen shows all four values in the selection dialog for the LTG. The **chvg** SMIT screen shows only the values for the LTG supported by the disks. The supported sizes are discovered using an `ioctl(IOCINFO)` call.

Since there may be several physical volumes existing in one volume group, and LTG is an attribute of a volume group, you should specify minimum LTG size among physical volumes, if they consist of different types of disk drives.

The following command shows how to change the LTG size for testvg from the default of 128 KB to 256 KB.

```
# chvg -L256 testvg
```

To ensure the integrity of the volume group, this command varies off the volume group during the change. The **mkvg** command supports the same new -L flag.

To find out what the maximum supported LTG size of your hard disk is, you can use the **lquerypv** command with the -M flag. The output gives the maximum LTG size in KB, as can be seen from the following lines:

```
# /usr/sbin/lquerypv -M hdisk0
256
```

You can list the values for all the new options (LTG size, AUTO SYNC, and HOT SPARE) with the **lsvg** command. Note that the volume group identifier has been widened from 16 to 32 characters.

```
# lsvg rootvg
VOLUME GROUP:  rootvg                      VG IDENTIFIER:
000bc6fd00004c00000000e10fdd7f52
VG STATE:      active                      PP SIZE:      16 megabyte(s)
VG PERMISSION: read/write                  TOTAL PPs:    1084 (17344 megabytes)
MAX LVs:       256                        FREE PPs:     1032 (16512 megabytes)
LVs:           11                        USED PPs:     52 (832 megabytes)
OPEN LVs:      10                        QUORUM:       2
TOTAL PVs:     2                        VG DESCRIPTORS: 3
STALE PVs:     0                        STALE PPs:    0
ACTIVE PVs:    2                        AUTO ON:      yes
MAX PPs per PV: 1016                    MAX PVs:      32
LTG size:    128 kilobyte(s)           AUTO SYNC:   yes
HOT SPARE:   yes (one to one)
```

Logical track group size can be selected at volume group creation time or changed from the Physical Volumes tab in the Volume Group Properties Notebook. Web-based System Manager, in the Logical track maximum data transfer size drop-down list, shows all data transfer sizes. Those that are not valid for the selected volume group are grayed out and not selectable (Figure 4-16 on page 198).

Note: Because the physical volume and volume group identifiers have been changed from 16 characters to 32 characters, you can only access a volume group created on AIX 5L from an AIX Version 4.3.3 system after you have applied the appropriate fixes from the Fall 2000 AIX Version 4.3.3 Update CD. You can access a volume group created on AIX Version 4.3.3 on an AIX 5L system, but using any of the new features, like setting a different logical track group size, will change some of the volume group identification internal data structures in a way so that the volume group becomes unusable on AIX Version 4.3.3 or a previous release.

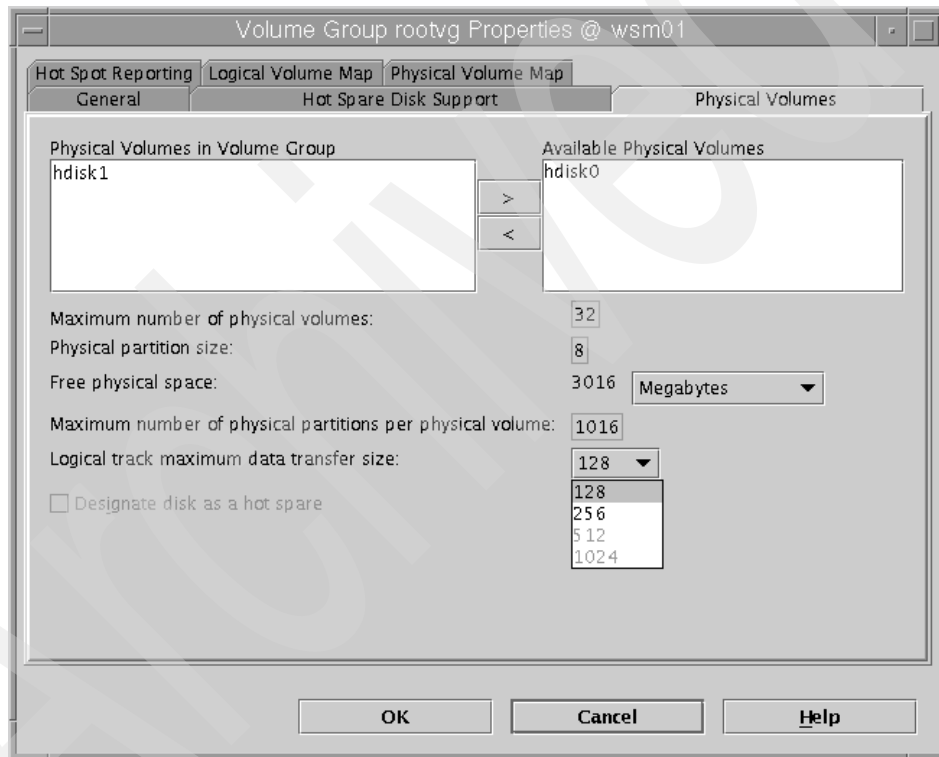


Figure 4-16 Volume Group Properties dialog

4.2.5 LVM hot-spot management

Two new commands, **lvmstat** and **migratelv**, help you to identify and remedy hot-spot problems within your logical volumes. You have a hot-spot problem if some of the logical partitions on your disk have so much disk I/O that your system performance noticeably suffers. By default, no statistics for the logical

volumes are gathered. The gathering of statistics has to be enabled first with the **lvostat** command for either a logical volume or an entire volume group.

The complete command syntax for **lvostat** is as follows:

```
lvostat { -l | -v } Name [ -e | -d ] [ -F ] [ -C ] [ -c Count ] [ -s ]
[ Interval [ Iterations ] ]
```

The meanings of the flags are provided in Table 4-5.

Table 4-5 The **lvostat** command flags

Flag	Description
-e	Enables the gathering of statistics about the logical volume.
-d	Disables the gathering of statistics.
-l	Specifies the name of a logical volume to work on.
-v	Specifies the name of a volume group to work on. You can also enable, in the first step, a volume group and selectively disable afterwards some logical volumes you are not working with.
-F	Separates the output of the statistics by colons (to make it easier for parsing by other scripts).
-c	Specifies how many lines from the top you want to have listed.
-C	Clears the counter for the specified logical volume or volume group.
-s	Suppresses the header lines for subsequent outputs if you are using the interval and iteration arguments. In the case of interval and iteration, only values for logical volumes for which there was a change in the last interval will be listed. If there was no change at all, only a period (.) will be printed to the console.

The first use of **lvostat**, after enabling, displays the counter values since system reboot. Each usage thereafter displays the difference from the last call.

The following example is a session where data was copied from /unix to /tmp:

```
# lvostat -v rootvg -e
# lvostat -v rootvg -C
# lvostat -v rootvg
```

Logical Volume	iocnt	Kb_read	Kb_wrtn	Kbps
hd8	4	0	16	0.00
paging01	0	0	0	0.00
lv01	0	0	0	0.00
hd1	0	0	0	0.00

hd3	0	0	0	0.00
hd9var	0	0	0	0.00
hd2	0	0	0	0.00
hd4	0	0	0	0.00
hd6	0	0	0	0.00
hd5	0	0	0	0.00

The previous output shows that, basically, all counters have been reset to zero. Before the following example, data was copied from /unix to /tmp:

```
# cp -p /unix /tmp
# lvmstat -v rootvg
```

Logical Volume	iocnt	Kb_read	Kb_wrtn	Kbps
hd3	296	0	6916	0.04
hd8	47	0	188	0.00
hd4	29	0	128	0.00
hd2	16	0	72	0.00
paging01	0	0	0	0.00
lv01	0	0	0	0.00
hd1	0	0	0	0.00
hd9var	0	0	0	0.00
hd6	0	0	0	0.00
hd5	0	0	0	0.00

As shown, there is activity on the hd3 logical volume, which is mounted on /tmp; on hd8, which is the jfslog logical volume; on hd4, which is / (root); on hd2, which is /usr; and on hd9var, which is /var. The following output provides details on hd3 and hd2:

```
# lvmstat -l hd3
```

Log_part	mirror#	iocnt	Kb_read	Kb_wrtn	Kbps
1	1	299	0	6896	0.04
3	1	4	0	52	0.00
2	1	0	0	0	0.00
4	1	0	0	0	0.00

```
# lvmstat -l hd2
```

Log_part	mirror#	iocnt	Kb_read	Kb_wrtn	Kbps
2	1	9	0	52	0.00
3	1	9	0	36	0.00
7	1	9	0	36	0.00
4	1	4	0	16	0.00
9	1	1	0	4	0.00
14	1	1	0	4	0.00
1	1	0	0	0	0.00

The output for a volume group provides a summary for all the I/O activity of a logical volume. It is separated into the number of I/O requests (iocnt), the kilobytes read and written (Kb_read and Kb_wrtn, respectively), and the transferred data in KB/s (Kbps). If you request the information for a logical volume, you receive the same information, but for each logical partition separately. If you have mirrored logical volumes, you receive statistics for each of the mirror volumes. In the previous sample output, several lines for logical partitions without any activity were omitted. The output is always sorted in decreasing order in the iocnt column.

Web-based System Manager allows for easy configuration of hot spot management.

Enabling hot spot reporting at the volume group level, from the Hot Spot Reporting tab of the Volume Group Properties tab (Figure 4-17), turns on the reporting feature for all logical volumes within the volume group.

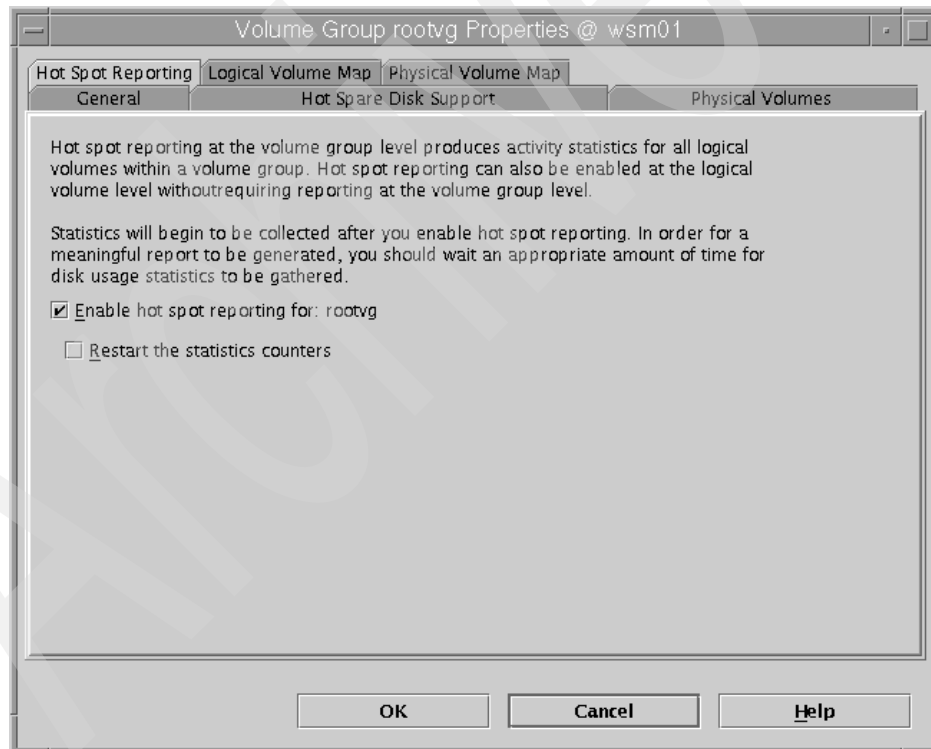


Figure 4-17 Volume Group Properties Hot Spot Reporting tab

Hot spot reporting can also be enabled from the Hot Spot Reporting tab of the Logical Volumes Property notebook (Figure 4-18) without having to enable the feature for the entire volume group.

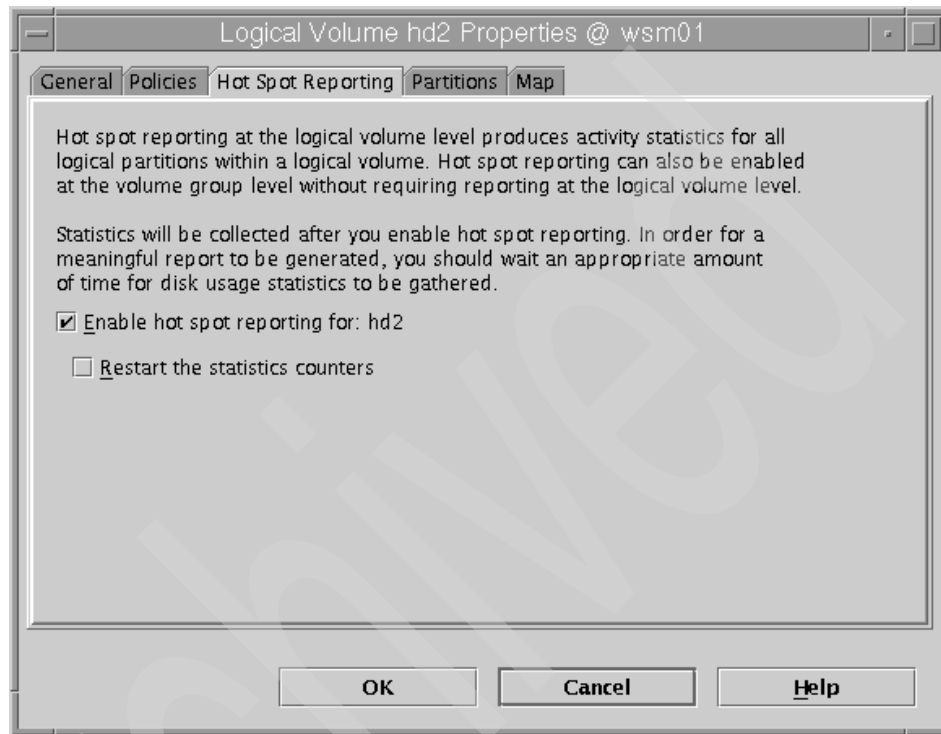


Figure 4-18 Logical Volumes Properties notebook

Once the hot spot feature is enabled, either for a logical volume or a volume group, you can select either entity and use the pull-down or pop-up menu to access the Manage Hot Spots... Sequential dialog (Figure 4-19 on page 203).

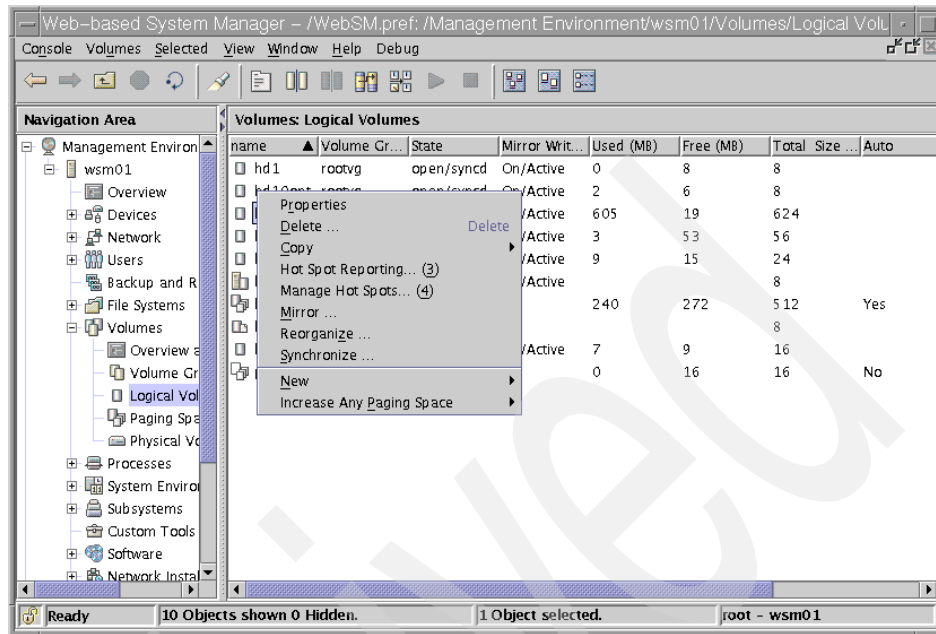


Figure 4-19 Manage Hot Spots sequential dialog

The first dialog in the series, once Manage Hot Spots... has been selected (Figure 4-20 on page 204). It allows you to define your reporting and display statistics.

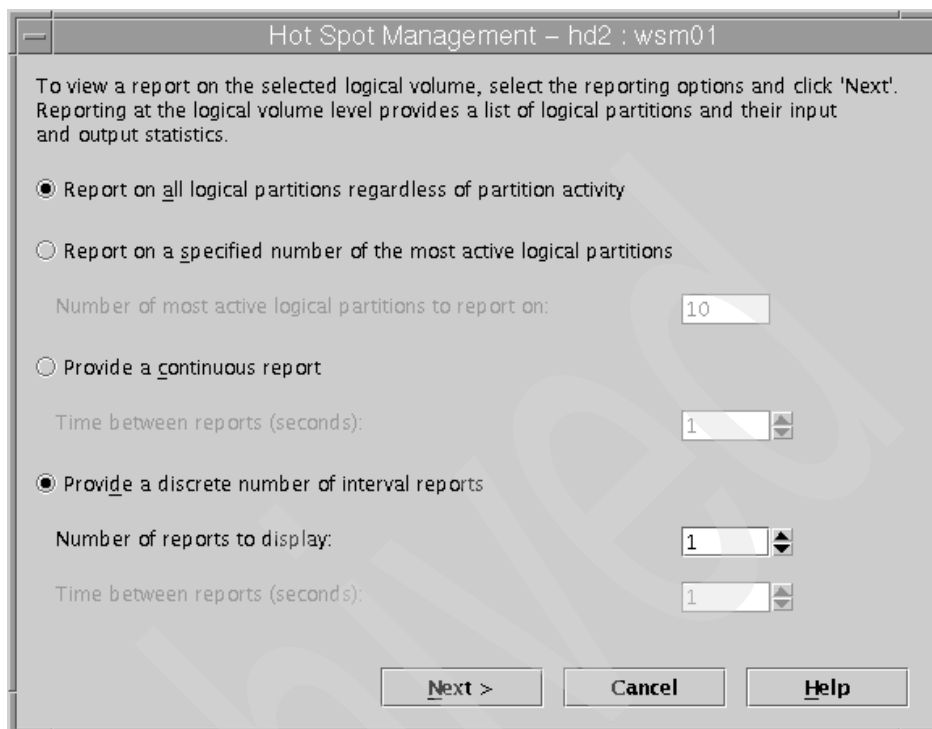


Figure 4-20 Hot Spot Management dialog

The second dialog displays information the user specified in the previous panel. This includes logical partition number, number of mirrors, I/O count, KB read and written, and data transfer rate (Figure 4-21 on page 205).

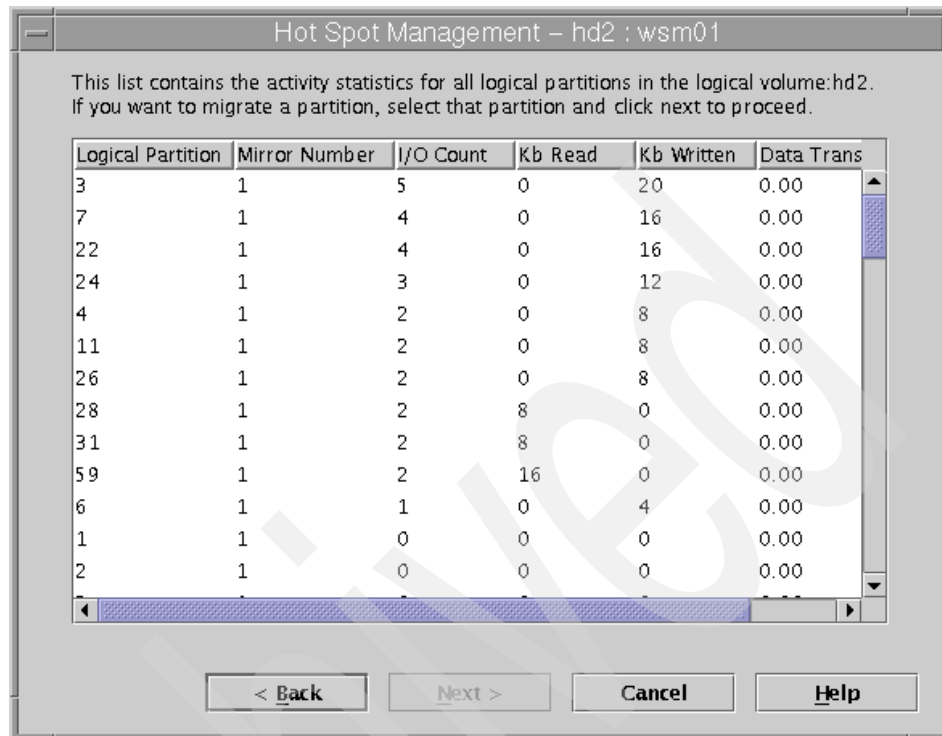


Figure 4-21 Hot Spot Management statistics

This list not only displays information, but also allows you to select (Figure 4-22 on page 206) the logical partition that the user may want to migrate to a disk with less I/O activity. This feature allows the user to manage potential disk I/O bottlenecks.

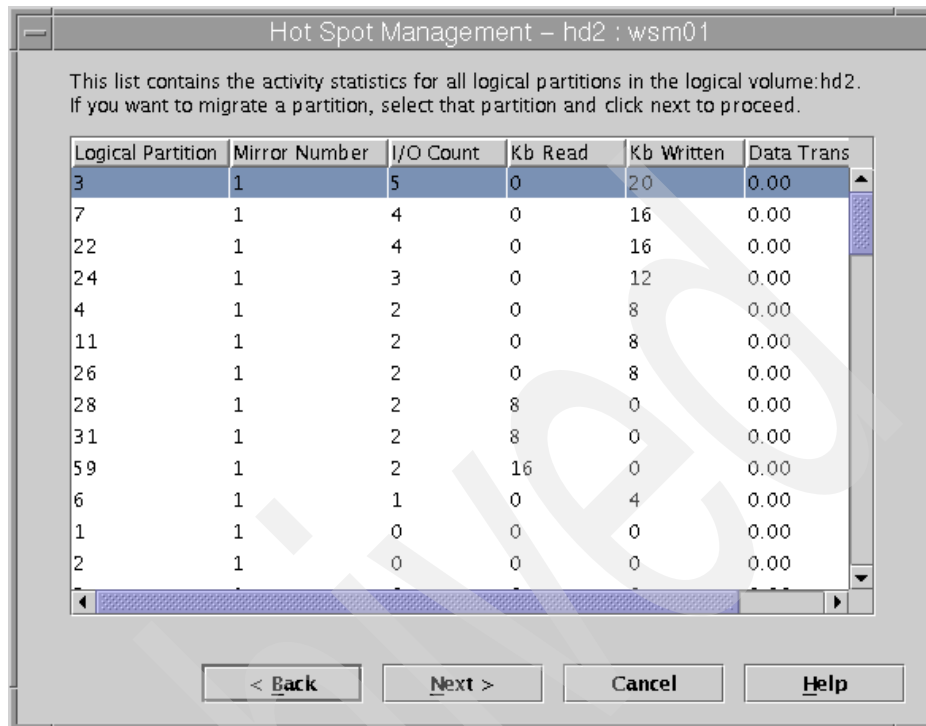


Figure 4-22 Hot Spot selection

The final dialog panel (Figure 4-23 on page 207) in the sequence allows the user to specify the destination physical partition and check the information before committing any changes to the system.

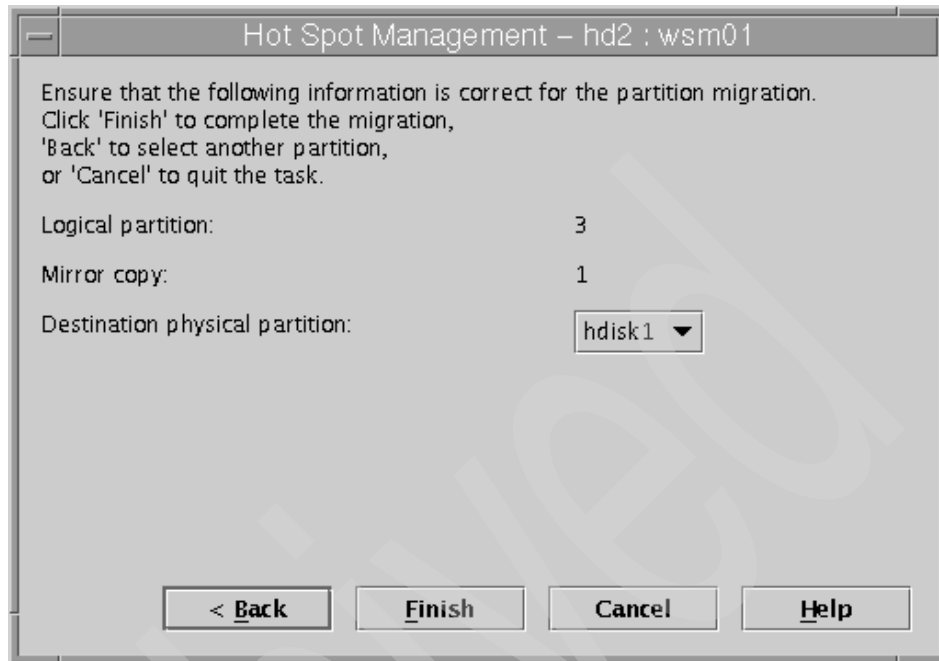


Figure 4-23 Physical destination partition

4.2.6 The **migratelp** command

With the output of the **lvmsstat** command described in the previous section, it is easy to identify the logical partitions with the heaviest traffic. If you have several logical partitions with heavy usage on one physical disk and want to balance these across the available disks, you can use the new **migratelp** command to move these logical partitions to other physical disks.

Note: The **migratelp** command will not work with partitions of striped logical volumes.

The **migratelp** command uses the following syntax:

```
migratelp lvname/lpartnum[/copynum] destpv[/ppartnum]
```

This command uses, as parameters, the name of the logical volume, the number of the logical partition (as it is displayed in the **lvmsstat** output), and an optional number for a specific mirror copy. If information is omitted, the first mirror copy is used. You have to specify the target physical volume for the move; in addition,

you can specify a target physical partition number. If successful, the output will appear similar to the following:

```
# migrate lp hd3/1 hdisk1/109
migrate lp: Mirror copy 1 of logical partition 1 of logical volume
          hd3 migrated to physical partition 109 of hdisk1.
```

4.2.7 The **recreatevg** command

The **recreatevg** command is used when you have a disk-to-disk copy to perform but you want to create a unique volume and not an exact mirror. A direct **dd** copy would create a problem because all the information, such as VGDA's and LV's, in one disk is copied to the other. Duplicate volume group, logical volume, and file system mount points are prevented by using the **recreatevg** command. Command options allow you to specify a logical volume name (a prefix label to uniquely define the VG). Automatic name generation is the default.

The **recreatevg** command is also supported in AIX Version 4.3.3 maintenance level 8 with APAR IY10456. To utilize this command, you have to issue the following command sequence after the real duplication of the physical volume contents using ESS's FlashCopy function or another resembled function. These operations are mandatory to avoid potential collisions of LVM component names (PVID, volume group name, logical volume name, file system name).

```
# chdev -l hdiskX -a pv=clear
# recreatevg -y newvg_name -L /newfs -Y newlv -hdiskX
```

In the previous example, *hdiskX* is the duplicated target physical volume name, *newvg_name* is the newly assigned volume group name, and */newfs* and *newlv* are used for prefixes of the newly assigned file systems and logical volumes contained in this volume group.

4.2.8 The **mkvg** command (5.1.0)

In AIX 5L Version 5.1, the **mkvg** command has been enhanced to automatically determine the correct PP size when creating a new volume group. If no PP size is specified (-s flag), the **mkvg** command attempts to figure out the correct PP size based on the disks you are trying to put into a volume group. The following examples show how to use the new enhancements.

In the first example, a 2.2 GB disk is used to create a new volume group named *ds9vg*. The PP size for the new volume group should be at least 4 MB.

```
# mkvg -y ds9vg hdisk2
ds9vg
```

The output of the **lsvg** command shows that the volume group was created with a PP size of 4 MB:

```
# lsvg ds9vg
VOLUME GROUP:   ds9vg           VG IDENTIFIER:
000bc6fd00004c00000000e524747a95
VG STATE:       active          PP SIZE:         4 megabyte(s)
VG PERMISSION:  read/write      TOTAL PPs:       537 (2148 megabytes)
MAX LVs:        256             FREE PPs:        537 (2148 megabytes)
LVs:            0               USED PPs:        0 (0 megabytes)
OPEN LVs:       0               QUORUM:          2
TOTAL PVs:      1               VG DESCRIPTORS:  2
STALE PVs:      0               STALE PPs:       0
ACTIVE PVs:     1               AUTO ON:         yes
MAX PPs per PV: 1016            MAX PVs:         32
LTG size:       128 kilobyte(s) AUTO SYNC:          no
HOT SPARE:      no
```

For the second example, two 8 GB disks and one 2.2 GB disk are used to create a new volume group. Here, the PP size must be 16 MB or greater:

```
# mkvg -y bigvg hdisk3 hdisk4 hdisk5
bigvg
```

To verify the size chosen, use the **lsvg** command and have a look at the PP size field:

```
# lsvg bigvg
VOLUME GROUP:   bigvg           VG IDENTIFIER:
000bc6fd00004c00000000e524858625
VG STATE:       active          PP SIZE:         16 megabyte(s)
VG PERMISSION:  read/write      TOTAL PPs:       1218 (19488 megabytes)
MAX LVs:        256             FREE PPs:        1218 (19488 megabytes)
LVs:            0               USED PPs:        0 (0 megabytes)
OPEN LVs:       0               QUORUM:          2
TOTAL PVs:      3               VG DESCRIPTORS:  3
STALE PVs:      0               STALE PPs:       0
ACTIVE PVs:     3               AUTO ON:         yes
MAX PPs per PV: 1016            MAX PVs:         32
LTG size:       128 kilobyte(s) AUTO SYNC:          no
HOT SPARE:      no
```

4.2.9 Passive mirror write consistency check

AIX 5L introduces a new passive mirror write consistency check (MWCC) algorithm for mirrored logical volumes. This option only applies to big volume groups.

Previous versions of AIX used a single MWCC algorithm, which is now called the active MWCC algorithm to distinguish it from the new algorithm. With active MWCC, records of the last 62 distinct logical transfer groups (LTG) written to disk are kept in memory and also written to a separate checkpoint area on disk. Because only new writes are tracked, if new MWCC tracking tables have to be written out to the disk checkpoint area, the disk performance can degrade if there are a lot of random write requests issued. The purpose of the MWCC is to guarantee the consistency of the mirrored logical volumes in case of a crash. After a system crash, the logical volume manager will use the LTG tables in the MWCC copies on disk to make sure that all mirror copies are consistent.

The new passive MWCC algorithm does not use an LTG tracking table, but sets a dirty bit for the mirrored logical volume as soon as the volume is opened for writes. This bit gets cleared only if the volume is successfully synced and is closed. In the case of a system crash, the entire mirrored logical volume will undergo a background resynchronization spawned during varyon of the volume group, because the dirty bit has not been cleared. Once the background resynchronization completes, the dirty bit is cleared, but can be reset at any time if the mirrored logical volume is opened. It should be noted that the mirrored logical volume can be used immediately after system reboot, even though it is undergoing background resynchronization.

The trade-off for the new passive MWCC algorithm compared to the default active MWCC algorithm is better performance during normal system operations. However, there is additional I/O that may slow system performance during the automatic background resynchronization that occurs during recovery after a crash.

The **lslv** and **chlv** commands have been changed accordingly. Instead of outputting just an off or on in the MIRROR WRITE CONSISTENCY field, the value now reads on/ACTIVE or on/PASSIVE, as shown in the following example:

```
# lslv lv00
LOGICAL VOLUME:    lv00                      VOLUME GROUP:    software
LV IDENTIFIER:     000bc6fd00004c00000000e1b374aba8.2 PERMISSION:
read/write
VG STATE:          active/complete          LV STATE:         opened/syncd
TYPE:              jfs                      WRITE VERIFY:     off
MAX LPs:           512                      PP SIZE:         8 megabyte(s)
COPIES:            1                        SCHED POLICY:    parallel
LPs:               62                      PPs:             62
STALE PPs:         0                       BB POLICY:       relocatable
INTER-POLICY:      minimum                  RELOCATABLE:     yes
INTRA-POLICY:      middle                   UPPER BOUND:     32
MOUNT POINT:       /software                LABEL:           /software
MIRROR WRITE CONSISTENCY: on/ACTIVE
EACH LP COPY ON A SEPARATE PV ?: yes
```


The `-w` flag for the **chlv** command now accepts either an `a` or `y` option to turn on active mirror write consistency checking, or a `p` option to use the new passive MWCC algorithm. The `n` option turns off mirror write consistency checking.

The passive MWCC function is supported on big VG format volume groups only.

4.2.10 Thread-safe liblvm.a

In AIX 5L, the libraries implementing query functions of the logical volume manager (LVM) functions (**liblvm.a**) are now thread-safe. Because LVM commands must be able to run even when the system is booting or being installed, the LVM library cannot rely on the availability of the pthread support library. Therefore, the internal architecture of the **liblvm.a** library ensures that the library is thread safe.

The following libraries are now thread safe:

- ▶ **lvm_querylv**
- ▶ **lvm_querypv**
- ▶ **lvm_queryvg**
- ▶ **lvm_queryvgs**

4.2.11 Advanced RAID support (5.2.0)

Today's storage subsystems have the ability to increase the size of a Logical Unit (LUN) or a RAID array, and therefore the size of the corresponding physical volume (PV) that AIX uses grows. With AIX 5L Version 5.2, this space can be used by dynamically adding physical partitions (PP) to that hdisk.

To accommodate this new capability, a new flag is added to the **chvg** command:

```
chvg -g vgname
```

The **chvg -g** command will examine all the disks in the volume group to see if they have grown in size. If any disks have grown in size it attempts to add additional PPs to the PVs. If necessary, the proper t-factor is applied or the volume group (VG) is converted to a big VG.

Typically, before a disk device is aware that it has grown in size it needs to be opened and closed. This is done by a varyoff then varyon cycle. Note that all file systems in the affected volume group need to be unmounted before the volume group can be varied off.

For example, to increase the size of a LUN in a FASSt 500 storage subsystem and make AIX aware of this change, the following steps need to be performed:

1. Change the size of the LUN in the FASSt 500 storage subsystem.
2. Unmount all file systems in the affected volume group for every file system using the following command:

```
umount /filesystem
```
3. Vary off the volume group, using the following command:

```
varyoffvg vgroupname
```
4. Vary on the volume group, using the following command:

```
varyonvg vgroupname
```
5. Mount all the file systems unmounted in step 2, using the following command:

```
mount /filesystem
```
6. Add the new PPs to the volume group using the following command:

```
chvg -g vgroupname
```

The growing of disks in the rootvg and in activated concurrent VGs is not supported. The change of the **chvg** command is reflected in Web-based System Manager.

4.2.12 Bad block configuration

Another feature in AIX 5L Version 5.2 is the new **-b** flag that is added to the **chvg** command. It allows you to turn bad block relocation on or off. If enabled, the logical volume manager (LVM) will relocate a block when it receives notification from the device that the block is bad. Bad block relocation is enabled by default. The syntax of the **chvg** command with the **-b** flag is as follows:

```
chvg -b {y/n} vgroupname
```

The **chvg -b y** command will turn on the bad block relocation policy of a volume group.

The **chvg -b n** command turns off the bad block relocation policy of a volume group.

Bad block relocation policy should be turned off for RAID devices and storage subsystems unless the manufacturer tells you otherwise.

4.2.13 Snapshot support for mirrored VGs (5.2.0)

Snapshot support for a mirrored volume group is provided to split a mirrored copy of a fully mirrored volume group into a snapshot volume group. To split a volume group, all logical volumes in the volume group must have a mirror copy and the mirror must exist on a disk or set of disks that contains only this set of mirrors. The original volume group will stop using the disks that are now part of the snapshot volume group. New logical volumes and mount points will be created in the snapshot VG.

Both volume groups will keep track of changes in physical partitions (PPs) within the volume group so that when the snapshot volume group is rejoined with the original volume group, consistent data is maintained across the rejoined mirror copies.

Consistency is maintained in the following way: When a write is issued to a PP in the original VG, the corresponding PP in the snapshot VG is marked stale. And when a write is issued to a PP in the snapshot VG, this PP is marked stale in the snapshot VG also. The rejoin process will merge the split stale PP lists into the volume group. The stale partitions will then be resynchronized by a background process. Therefore, the user will see the same data in the rejoined VG as was in the original VG before the rejoin.

To split a mirrored VG, the following restrictions apply:

- ▶ There is no support with classic concurrent mode.
- ▶ There is support under enhanced concurrent mode, but the snapshot volume group will not be made enhanced concurrent mode capable.
- ▶ The snapshot volume cannot be made concurrent capable or enhanced concurrent capable.
- ▶ The only allowable **chvg** options on the snapshot volume group are **chvg -a -R -S -u**.
- ▶ The only allowable **chvg** options on the original volume group are **chvg -a -R -S -u -h**.
- ▶ Partition allocation changes will not be allowed on the snapshot VG.
- ▶ A volume group cannot be split if a disk is already missing.
- ▶ A volume group cannot be split if the last non-stale partition would be on the snapshot volume group.

The command syntax to split a mirrored volume group into a snapshot volume group is the following and the most commonly used flags are provided in Table 4-6 on page 214:

```
splitvg [ -y SnapVGname ] [ -c Copy ] [ -f ] [ -i ] VGname
```

Table 4-6 The *splitvg* command flags

Flag	Description
-y <i>SnapVGname</i>	Specifies the name of the snapshot volume group to use instead of system-generated name.
-c <i>Copy</i>	Specifies which mirror to split. Valid values are 1, 2, or 3. The default is the second copy.
-f	Will force the split even if the mirror copy specified to create the snapshot volume group has stale partitions.
-i	Will split the mirror copy of a volume group into a independent volume group that cannot be rejoined into the original.

The command syntax to rejoin the snapshot volume group with the original volume group is the following:

```
joinvg [ -f ] VGname
```

Specify the -f flag to force the join when disks in the snapshot volume group are not active. The mirror copy on the inactive disks will be removed from the original volume group.

In the following example, the file system /data is a file system in the volume group datavg mirrored from hdisk2 to hdisk3. To split the mirror in the snapshot volume group, run the **snapvg** command and take an online backup of the data, then run the following command sequence:

1. **splitvg -y snapvg datavg**

The VG datavg is split and the VG snapvg is created. Furthermore, the mount point /fs/data is created.

2. **backup -f /dev/rmt0 /fs/data**

An inode based backup of the unmounted file system /fs/data is created on tape.

3. **joinvg datavg**

The snapshot VG snapvg is rejoined with the original VG datavg and synced in the background.

4.2.14 Performance improvement of LVM commands (5.2.0)

The execution time of **mkvg**, **extendvg**, **mk1v**, and **extend1v** have been improved for all volume group types. The execution time of some common **ls1v** and **lsvg** options have been improved for all volume group types. The improvements are

more significant for volume groups created with the -B (Big volume group) option of **mkvg**.

4.2.15 Unaligned I/O support in LVM (5.2.0)

In AIX 5L Version 5.2, file systems and kernel extensions have no LVM restrictions to contend with for size and alignment of I/O requests from the LVM strategy routine. A file system or kernel extension can now issue a single large I/O to the LVM strategy layer instead of breaking this I/O up into many individual smaller I/Os. This now allows LVM to issue a single `iodone` to the layer above LVM when the I/O is complete. The enhanced journal file system (JFS2) and AIO I/O requests currently take advantage of this feature.

4.2.16 Logical Volume serialization (5.2.0)

The serialization feature for logical volumes (LVs) serializes parallel I/Os to the same block of an application. Since this behavior is very rare for an application and activated serialization may degrade performance, this feature should generally be disabled.

If an application specifically requires logical volume serialization, it can be activated on closed LVs in one of the following ways:

- ▶ Using the **chlv -o y lvname** command.
- ▶ Using the SMIT fast path **smit chlv** command.
- ▶ Using the Logical Volume properties panel of the Web-based System Manager (see Figure 4-24 on page 216).
- ▶ Changing the attribute `SERIALIZE_IO` in the LV stanza `image.data` or `vgname.data` file would only take effect when restoring from a backup of a volume group containing the changed `image.data` or `vgname.data` file. It would not affect the logical volume on the current volume group.

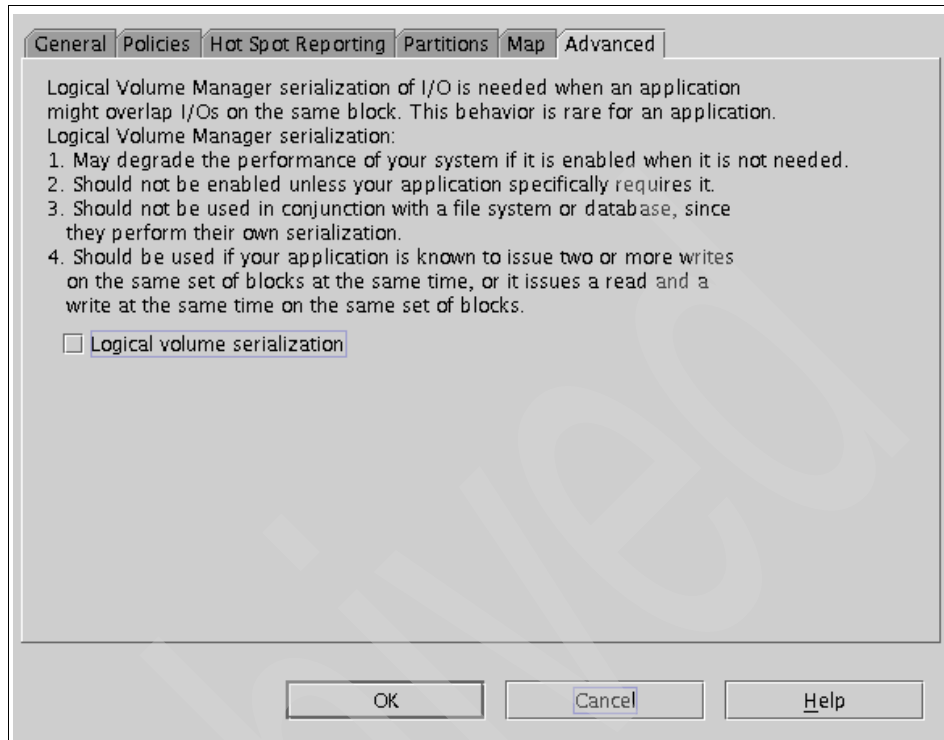


Figure 4-24 Logical volume serialization

4.2.17 The **mklv** and **extendlv** commands (5.1.0)

In AIX 5L Version 5.1, to create or extend a logical volume, you can now specify blocks, KB, MB, and GB, rather than number of partitions. The **mklv** and **extendlv** commands automatically determine the minimum number of partitions needed to fill the request.

Size units that can be used are as follows:

b,B	For blocks (512 byte)
k,K	For KB
m,M	For MB
g,G	For GB

In the following example, a logical volume that contains at least one block (512 byte) is created. Since the PP size of the bigvg volume group is 16 MB, the size of the new logical volume will be 16 MB.

```
# mklv -y block_lv bigvg 1b
block_lv
```

```
# lslv block_lv
LOGICAL VOLUME:      block_lv          VOLUME GROUP:  bigvg
LV IDENTIFIER:      000bc6fd00004c00000000e524858625.1 PERMISSION:
read/write
VG STATE:           active/complete    LV STATE:       closed/syncd
TYPE:               jfs                 WRITE VERIFY:   off
MAX LPs:            512                 PP SIZE:        16 megabyte(s)
COPIES:             1                   SCHED POLICY:   parallel
LPs:                1                   PPs:           1
STALE PPs:          0                   BB POLICY:      relocatable
INTER-POLICY:        minimum             RELOCATABLE:    yes
INTRA-POLICY:        middle              UPPER BOUND:    32
MOUNT POINT:        N/A                 LABEL:          None
MIRROR WRITE CONSISTENCY: on/ACTIVE
EACH LP COPY ON A SEPARATE PV ?: yes
```

The next example shows how to create a logical volume that is at least 20000 KB in size:

```
# mklv -y kb_lv bigvg 20000k
kb_lv
```

```
# lslv kb_lv
LOGICAL VOLUME:      kb_lv              VOLUME GROUP:  bigvg
LV IDENTIFIER:      000bc6fd00004c00000000e524858625.3 PERMISSION:
read/write
VG STATE:           active/complete    LV STATE:       closed/syncd
TYPE:               jfs                 WRITE VERIFY:   off
MAX LPs:            512                 PP SIZE:        16 megabyte(s)
COPIES:             1                   SCHED POLICY:   parallel
LPs:                2                   PPs:           2
STALE PPs:          0                   BB POLICY:      relocatable
INTER-POLICY:        minimum             RELOCATABLE:    yes
INTRA-POLICY:        middle              UPPER BOUND:    32
MOUNT POINT:        N/A                 LABEL:          None
MIRROR WRITE CONSISTENCY: on/ACTIVE
EACH LP COPY ON A SEPARATE PV ?: yes
```

In the following example, an existing logical volume is extended by 50 MB:

```
# lslv mb_lv
LOGICAL VOLUME:      mb_lv              VOLUME GROUP:  bigvg
LV IDENTIFIER:      000bc6fd00004c00000000e524858625.4 PERMISSION:
read/write
VG STATE:           active/complete    LV STATE:       closed/syncd
TYPE:               jfs                 WRITE VERIFY:   off
MAX LPs:            512                 PP SIZE:        16 megabyte(s)
COPIES:             1                   SCHED POLICY:   parallel
LPs:                309                 PPs:           309
```

```

STALE PPs:          0                      BB POLICY:      relocatable
INTER-POLICY:      minimum                  RELOCATABLE:   yes
INTRA-POLICY:      middle                   UPPER BOUND:   32
MOUNT POINT:       N/A                     LABEL:         None
MIRROR WRITE CONSISTENCY: on/ACTIVE
EACH LP COPY ON A SEPARATE PV ?: yes

```

```

# lsvg -l bigvg
bigvg:
LV NAME          TYPE          LPs    PPs    PVs    LV STATE      MOUNT POINT
mb_lv            jfs             309    309    1      closed/syncd  N/A

```

The mb_lv logical volume in the next example is extended by 50 MB. Since a PP has 16 MB in size, the extended LV should at least have four more PPs.

```

# extendlv mb_lv 50M

# lsvg -l bigvg
bigvg:
LV NAME          TYPE          LPs    PPs    PVs    LV STATE      MOUNT POINT
block_lv         jfs             1       1       1      closed/syncd  N/A
k_lv             jfs             1       1       1      closed/syncd  N/A
kb_lv            jfs             2       2       1      closed/syncd  N/A
mb_lv            jfs            313    313    1      closed/syncd  N/A
#

```

4.3 JFS enhancements

The following are enhancements that affect the JFS.

4.3.1 The root file system ownership (5.1.0)

In previous versions of AIX, the root file system (/) was owned by bin.bin. In AIX 5L Version 5.1, that ownership has changed to root.system to avoid the root user's dead letter from writing to the root file system.

4.3.2 Directory name lookup cache (5.2.0)

Version 5.2 unifies the directory name lookup cache for LFS, JFS, and JFS2. This cache will now support long file names.

Directory name lookup cache (DNLC) looks up the inode of a file when given the parent directory pointer, its file system, and file name. Version 5.2 replaces the multiple implementations of DNLC for LFS, JFS, and JFS2 with one implementation. Support for long file names, up to 255 characters, is also

provided. Version 5.2 still makes the old LFS cache available as it is an exported interface.

The long file name pointer references to memory have changed. There are a number of reasons why the memory allocation has changed:

- ▶ To enable the memory requirement to change dynamically.
- ▶ File systems not using long file names will not take up any extra space.
- ▶ To avoid memory fragmentation.
- ▶ The file system will not continually grow as long as space is freed when short file names are used.

4.3.3 The .indirect for JFS (5.1.0)

When a file is opened, an in-core inode is created by the operating system. The in-core inode contains a copy of all the fields defined in the disk inode, plus additional fields for tracking the in-core inode.

The JFS caches in-core inodes very aggressively. Once an in-core inode has been bound to a virtual memory object, the indirect pages required to access all of the file's indirect blocks are allocated. These indirect pages are not freed up until the inode is pushed out of cache, the file system is unmounted, or the file is deleted or truncated.

Failures due to .indirect exhaustion are increasing. The typical scenario is that the customer is copying a large number of large files to a large file system. Because the JFS caches the inode for each new target file, .indirect can fill up fairly quickly and writes will start failing with the errno of ENOMEM.

In the previous versions of AIX, the default behavior of the .indirect is to use a single segment, and the segment is used by the JFS to map in .indirect blocks. For AIX 5L Version 5.1, the default behavior is to use multiple segments. In all cases, the user is able to specify, using a mount option, whether or not multiple segments are used, thus having the ability to override the default.

Additional file system-specific options for the mount command are as follows:

- o Options mind Specifies the use of multiple segment default for AIX
- nomind Specifies the use of single segment

Note: This enhancement is for JFS only. JFS2 has a different design.

4.3.4 Complex inode lock (5.1.0)

In AIX 5L Version 5.1, a complex inode lock has been added to allow multiple simultaneous readers and exclusive writers. The inode locks have been changed to reduce contention on multiuser workloads. The inode lock macros are shown below:

► IWRITE_LOCK()

The INODE_LOCK() macro from previous versions of AIX has been renamed IWRITE_LOCK() in AIX 5L Version 5.1 and its function has changed to acquire the complex lock i_rwlock in write mode.

► IREAD_LOCK()

This is the new macro added to acquire the complex lock i_rwlock in read mode.

► INODE_UNLOCK()

The INODE_UNLOCK() macro of previous versions of AIX has been changed to release the complex lock i_rwlock.

► ISIMPLE_LOCK()

A new inode lock macro called ISIMPLE_LOCK() has been added and its function is to acquire the simple lock i_nodelock.

► ISIMPLE_UNLOCK()

A new inode unlock macro called ISIMPLE_UNLOCK().

4.3.5 The defragfs command enhancement (5.2.0)

A new -s flag has been added to the **defragfs** command. This flag provides a short report of a given file system.

An example on a JFS file system is as follows:

```
defragfs -s /tmp
/tmp filesystem is 40 percent fragmented
Total number of fragments                : 1000
Number of fragments that may be migrated : 400
```

An example on a JFS2 file system is as follows:

```
$ defragfs -s /tmp
/tmp filesystem is 40 percent fragmented
Total number of blocks                   : 1000
Number of blocks that may be migrated    : 400
```

The Web-based System Manager has been updated for this new feature.

4.3.6 du and df command enhancements (5.2.0)

This enhancement of the **du** and **df** commands provides two new flags, **-m** and **-g**, to report the output in MB blocks and GB blocks. The following example shows the output of the **df** and **du** command using these flags.

```
df -m /usr
Filesystem      MB blocks      Free %Used      Iused %Iused Mounted on
/dev/hd2        1248.00      46.89  97%      31494    10% /usr
# df -g /usr
Filesystem      GB blocks      Free %Used      Iused %Iused Mounted on
/dev/hd2         1.22        0.05  97%      31494    10% /usr
# du -sm /usr
1149.79 /usr
# du -sg /usr
1.12 /usr
```

4.3.7 rmfs command enhancement (5.2.0)

A new flag **-i** is introduced for the **rmfs** command that provides a warning message and prompts for confirmation from the user before removing the file system. This is shown in the following example:

```
rmfs -i /tartest
rmfs: Warning, all data contained on /tartest will be destroyed.
rmfs: Remove filesystem: /tartest? y(es) n(o)? y
rmlv: Logical volume lv02 is removed
```

4.3.8 Increased file descriptor limit (5.2.0)

AIX 5L Version 5.2 increased the maximum number of open file descriptors per process from 32767 to 65534. This limit is defined as **OPEN_MAX** in the include file **/usr/include/sys/limits.h**.

```
#define OPEN_MAX      65534 /* max num of files per process */
```

4.3.9 importvg command enhancement (5.2.0)

The **importvg** command is enhanced to accept a PVID as a command line argument, as shown in the following example:

```
lspv
hdisk0      0001810ff004704d      rootvg      active
hdisk1      0001810f70cd4dee      rootvg      active
hdisk2      0001810fce3bf383      stuffvg      active
hdisk3      0001810fce3bf4ed      stuffvg      active
hdisk4      0001810fd3838ada      None
hdisk5      0001810fd3838c5e      None
```

```
# importvg -y myvg 0001810fd3838c5e
myvg
# lspv
hdisk0      0001810ff004704d      rootvg      active
hdisk1      0001810f70cd4dee      rootvg      active
hdisk2      0001810fce3bf383      stuffvg      active
hdisk3      0001810fce3bf4ed      stuffvg      active
hdisk4      0001810fd3838ada      myvg        active
hdisk5      0001810fd3838c5e      myvg        active
```

4.3.10 RAM disk enhancement (5.2.0)

The purpose of the **mkramdisk** command is to create file systems directly in memory. This is useful for an application that makes many temporary files.

AIX 5L Version 5.2 removes the 2 GB limitation per RAM disk.

An example to create a ramdisk of 4 MB is as follows:

```
#mkramdisk 4m
/dev/rramdisk0
# mkfs -V jfs /dev/ramdisk0
mkfs: destroy /dev/ramdisk0 (yes)? y
Device /dev/ramdisk0:
Standard empty filesystem
Size:      8192 512-byte (UBSIZE) blocks
Initial Inodes: 1024
# mount -V jfs -o nointegrity /dev/ramdisk0 /ramdisk
# df -k
Filesystem      1024-blocks      Free %Used      Iused %Iused Mounted on
/dev/hd4         16384          5812   65%         1463   18% /
/dev/hd2        753664          1836  100%        23751   13% /usr
/dev/hd9var      16384          9976   40%          456   12% /var
/dev/hd3         32768         28280   14%          264    4% /tmp
/dev/hd1         16384         15820    4%           18    1% /home
/proc            -              -    -            -    - /proc
/dev/hd10opt     32768         25164   24%          278    4% /opt
/dev/cd0         636190           0  100%       318095  100% /cdrom/cd0
/dev/ramdisk0    4096           3924    5%           17    2% /ramdisk
```

Important: Use ramdisk only for data that can be lost. After each reboot the ramdisk file system is destroyed and must be rebuilt.

4.3.11 Megabyte and Gigabyte file systems (5.2.0)

The **mkfs -s** flag, the **chfs -a** flag, and the **crfs -a** flag now support a size using M (for Megabyte) or G (for Gigabyte).

An example of changing the size of the /tmp files system to 55 MB using M:

```
chfs -a size=55M /tmp
```

The SMIT and Web-based System Manager panels have been modified, as shown in Figure 4-25.

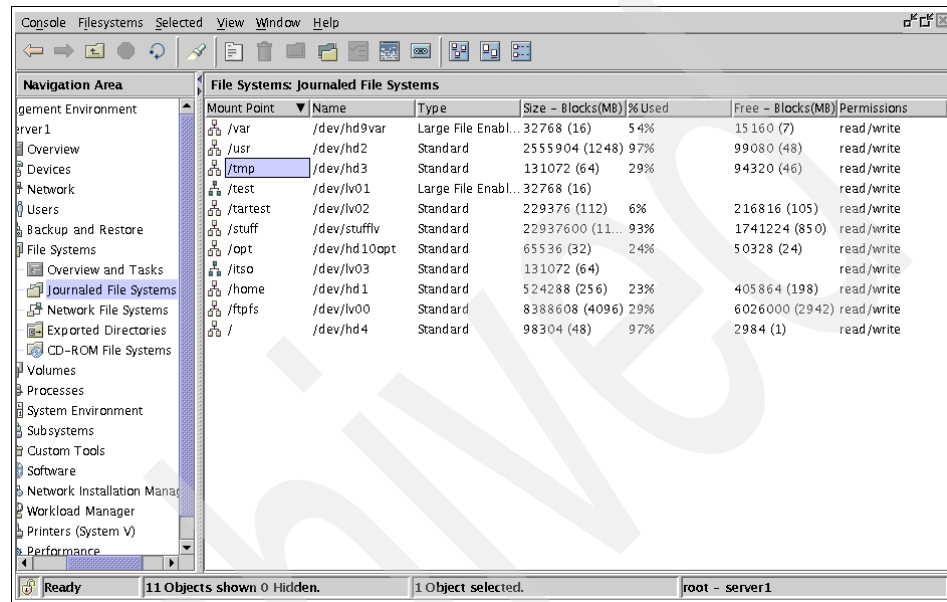


Figure 4-25 File system list panel

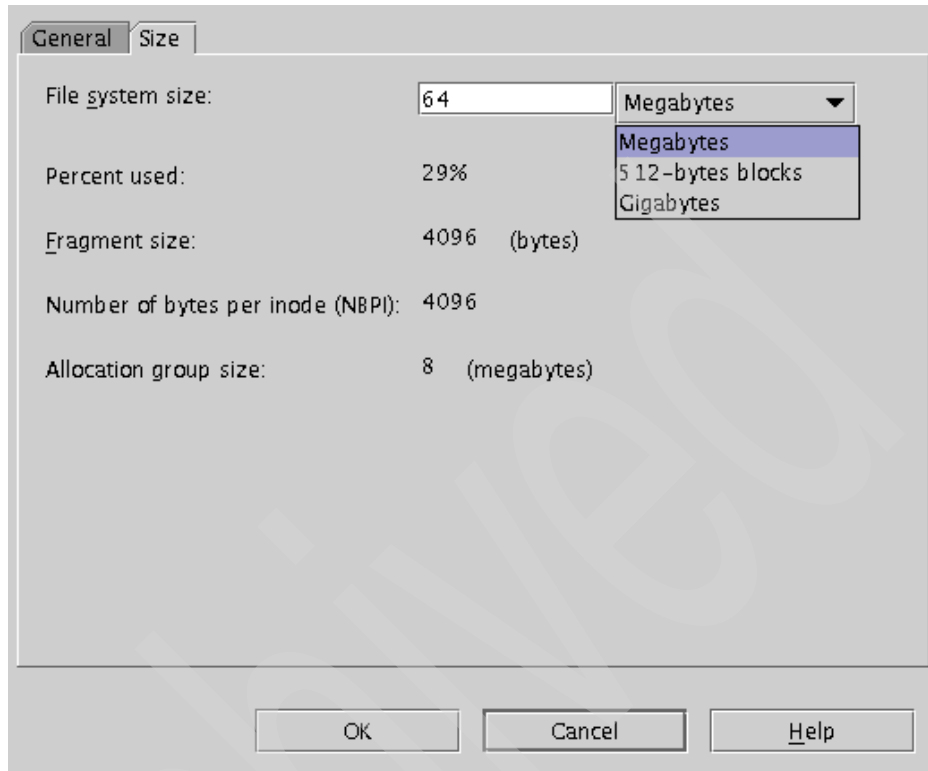


Figure 4-26 File system Size panel

We can see in Figure 4-26 that the default for this file system size is Megabyte.

4.4 The enhanced Journaled File System

The Journaled File System 2 (JFS2) is an enhanced and updated version of the JFS on AIX Version 4.3 and previous releases. The journaled file system JFS and JFS2 are native to the AIX operating system. The file system links the file and directory data to the structure used by storage and retrieval mechanisms.

JFS2 has new features that include extent-based allocation, sorted directories, and dynamic space allocation for file system objects.

4.4.1 New in JFS2

Table 4-7 on page 225 provides a comparison chart between the JFS2 and the standard JFS.

Table 4-7 *Journalized file system specifications*

Function	JFS2	JFS
Fragments/Block Size	512–4096 Block Sizes	512–4096 Fragments
Architectural Maximum File	1 PB ¹	64 GB
Architectural Maximum File System Size	4 PB	1 TB (64-bit kernel) 1 TB (32-bit kernel) ²
Maximum File Size Tested	16 TB (64-bit kernel) 1 TB (32-bit kernel)	64 GB
Maximum File System Size	16 TB (64-bit kernel) 1 TB (32-bit kernel)	1 TB
Number of Inodes	Dynamic, limited by disk space	Fixed, set at file system creation
Directory Organization	B-tree	Linear
Online Defragmentation	Yes	Yes
Compression	No	Yes
Default Ownership at Creation	root.system	sys.sys
SGID of Default File Mode	SGID=off	SGID=on
Quotas	No	Yes
Extended ACL	Yes	Yes
¹ PB stands for PetaBytes, which is equal to 1,048,576 GigaBytes. ² TB stands for TeraBytes, which is equal to 1,024 GigaBytes.		

Extent-based addressing structures

JFS2 uses extent-based addressing structures, along with aggressive block allocation policies, to produce compact, efficient, and scalable structures for mapping logical offsets within files to physical addresses on disk.

An extent is a sequence of contiguous blocks allocated to a file as a unit and is described by a triple, consisting of *logical offset*, *length*, *physical address*. The addressing structure is a B+-tree populated with extent descriptors (the triples above), rooted in the inode, and keyed by logical offset within the file.

Variable block size

JFS2 supports block sizes of 512, 1024, 2048, and 4096 bytes on a per file system basis, allowing users to optimize space utilization based upon their

application environment. Smaller block sizes reduce the amount of internal fragmentation within files and directories and are more space efficient. However, small blocks can increase path length, since block allocation activities will occur more often than if a larger block size were used. The default block size is 4096 bytes, since performance, rather than space utilization, is generally the primary consideration for server systems.

Dynamic disk inode allocation

JFS2 dynamically allocates space for disk inodes as required, freeing the space when it is no longer required. This support avoids the traditional approach of reserving a fixed amount of space for disk inodes at file system creation time, thus eliminating the need for customers to estimate the maximum number of files and directories that a file system will contain.

Directory organization

Two different directory organizations are provided. The first organization is used for small directories and stores the directory contents within the directory's inode. This eliminates the need for separate directory block I/O as well as the need for separate storage allocation. Up to eight entries may be stored inline within the inode, excluding the self (.) and parent (..) directory entries, which are stored in a separate area of the inode.

The second organization is used for larger directories and represents each directory as a B+-tree keyed on name. The intent is to provide faster directory lookup, insertion, and deletion capabilities when compared to traditional unsorted directory organizations.

On-line file system free space defragmentation

JFS2 supports the defragmentation of free space in a mounted and actively accessed file system. Once a file system's free space has become fragmented, defragmenting the file system allows JFS2 to provide more I/O-efficient disk allocations and to avoid some out of space conditions.

Defragmentation support is provided in two pieces. The first piece is a user space JFS2 utility, which examines the file system's metadata to determine the extent of free space fragmentation and to identify the file system reorganization activities required to reduce or eliminate the fragmentation. The second piece is integrated into the JFS2 kernel extension and is called by the user space utility. This second piece actually performs the reorganization activities, under the protection of journaling and with appropriate serialization to maintain file system consistency.

4.4.2 Compatibility

In this section how the JFS2 interacts with the JFS environment is described.

Mixed volumes compatibility

In some cases there will be many servers coexisting with different levels of AIX in a data center. From the JFS point of view, you can only import volume groups and mount file systems from AIX 4.X to AIX 5L servers. It is not possible to mount the JFS2 file system on AIX 4.X machines.

AIX 5L servers importing volume groups with JFS file systems

Figure 4-27 shows an example of an AIX Version 4.X machine exporting a volume group, and an AIX 5L machine importing this volume group and mounting a file system.

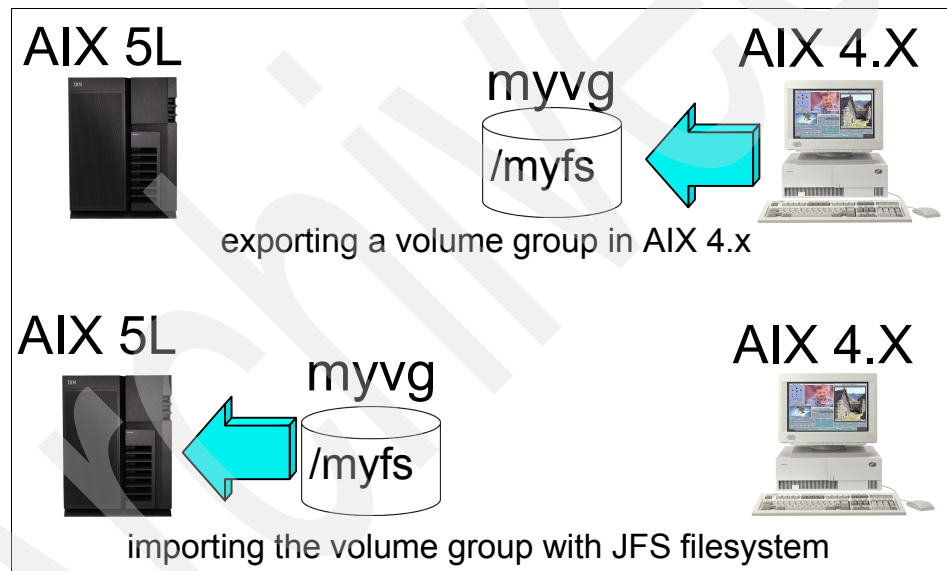


Figure 4-27 Example of a server importing and mounting JFS volumes

Tip: In a case of JFS-type migration (for example, for performance or security reasons), a backup/restore approach is required. There is no LVM or JFS command that migrates JFS volumes automatically.

It is possible to migrate JFS volumes in two different ways:

1. Backing up the file system, removing it, and recreating it in the JFS2 type, then restoring the backup above the new file system.
2. If there is enough disk space available in the volume group, it is possible to create a new JFS2 file system structure with the same attributes, and just copy all the files from one file system to another.

NFS mounting compatibility

There are two possible scenarios when mounting NFS file systems across different versions of JFS:

1. An AIX 5L JFS2 machine NFS mounting a remote JFS file system, as shown in Figure 4-28.

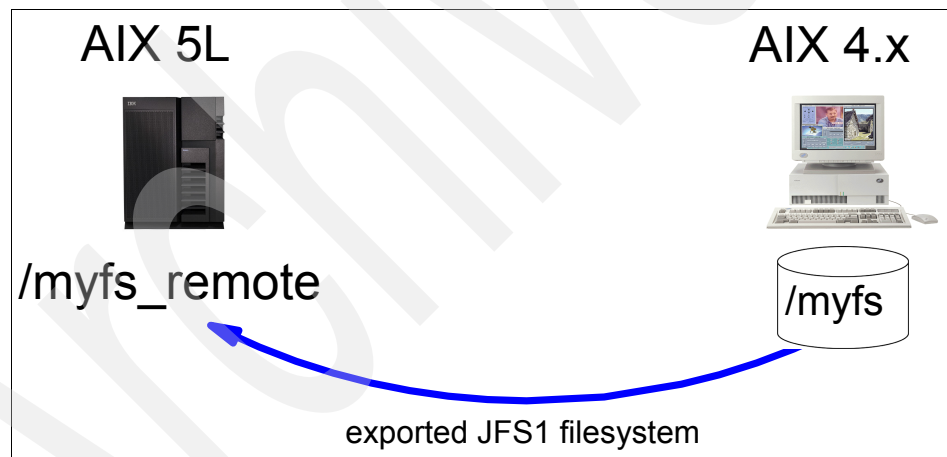


Figure 4-28 AIX 5L JFS2 machine NFS mounting a JFS file system

2. An AIX 4.X JFS machine NFS mounting a remote JFS2 file system, as shown in Figure 4-29.

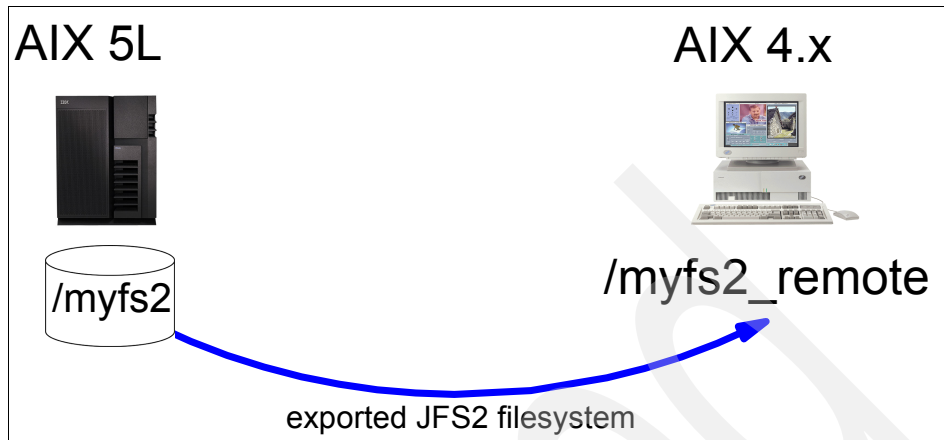


Figure 4-29 AIX 4.X JFS machine NFS mounting a JFS2 file system

Both scenarios have no compatibility issues.

4.4.3 Commands and utilities changes

There is a set of new commands included in AIX for JFS2 management, and a set of JFS commands that are updated to handle JFS2 file systems.

In this section a brief explanation about these JFS commands is provided.

Creating a JFS2 file system

The easiest way to create a JFS2 file system is through SMIT. Using the SMIT jfs2 fast path will show a JFS2 management menu, as seen in Figure 4-30.

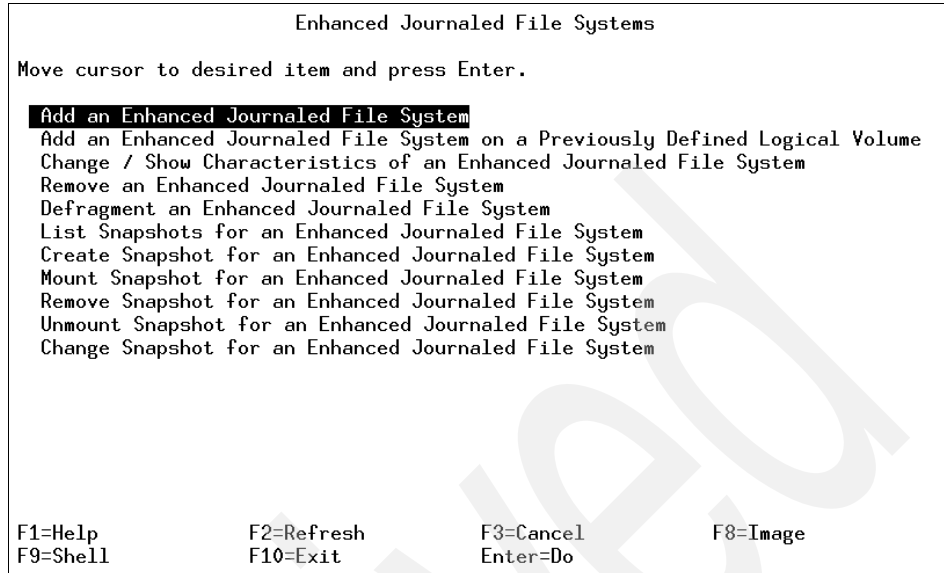


Figure 4-30 SMIT panel for JFS2 management

Using the SMIT menu, the first option, Add an Enhanced Journaled File System, creates the JFS2 file system, and the second option, Add an Enhanced File System on a Previously Defined Logical Volume, creates a JFS2 file system on a previously created logical volume, which may be needed for organization or by the application.

In the following sections, the add options from Figure 4-30 are discussed.

Add an enhanced file system

This option in the SMIT JFS2 menu allows the creation of a JFS2 file system with a size of 512-byte blocks and the mount point, as shown in Figure 4-31 on page 231.

Add an Enhanced Journaled File System

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

	[Entry Fields]	
Volume group name	rootvg	
SIZE of file system		
Unit Size	512bytes	+
* Number of units	[512000]	#
* MOUNT POINT	[/jfs2]	
Mount AUTOMATICALLY at system restart?	no	+
PERMISSIONS	read/write	+
Mount OPTIONS	[]	+
Block Size (bytes)	4096	+
Inline Log?	no	+
Inline Log size (MBytes)	[]	#

F1=Help
F5=Reset
F9=Shell

F2=Refresh
F6=Command
F10=Exit

F3=Cancel
F7=Edit
Enter=Do

F4=List
F8=Image

Figure 4-31 SMIT panel for adding a JFS2 file system

Add on a previously defined logical volume

If a non-default logical volume is needed for the JFS2 file system creation, this logical volume must be defined prior to the file system creation.

The logical volume type must be assigned as JFS2; otherwise, it will not appear as a selectable logical volume in the file system creation, as shown in Figure 4-32 on page 232.

Add a Logical Volume			
Type or select values in entry fields. Press Enter AFTER making all desired changes.			
[TOP]	[Entry Fields]		
Logical volume NAME	[jfs2lv]		
* VOLUME GROUP name	rootvg		
* Number of LOGICAL PARTITIONS	[100]	#	
PHYSICAL VOLUME names	[hdisk0]	+	
Logical volume TYPE	[jfs2]		
POSITION on physical volume	middle	+	
RANGE of physical volumes	minimum	+	
MAXIMUM NUMBER of PHYSICAL VOLUMES	[]	#	
to use for allocation			
Number of COPIES of each logical partition	1	+	
Mirror Write Consistency?	active	+	
Allocate each logical partition copy	yes	+	
[MORE...11]			
F1=Help	F2=Refresh	F3=Cancel	F4=List
F5=Reset	F6=Command	F7=Edit	F8=Image
F9=Shell	F10=Exit	Enter=Do	

Figure 4-32 SMIT panel for adding a logical volume and assigning as JFS2

After creating the logical volume, you must associate this logical volume with the file system to be created. Go to the SMIT jfs2 panel and choose the second option.

If the logical volume was created correctly, it must appear as a selectable logical volume, as shown in Figure 4-33 on page 233.

Add an Enhanced Journaled File System

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

	[Entry Fields]	
* LOGICAL VOLUME name		+
* MOUNT POINT	[]	+
Mount AUTOMATICALLY at system restart?	no	+
PERMISSIONS	read/write	+
Mount OPTIONS	[]	+
Block Size (bytes)	4096	+
Inline Log?	no	+
		#

LOGICAL VOLUME name

Move cursor to desired item and press Enter.

jfs2lv

F1=Help
F5=Image
F9=/=Find

F2=Refresh
F10=Exit
n=Find Next

F3=Cancel
Enter=Do

Figure 4-33 SMIT panel for showing the logical volume selection

After selecting the correct logical volume, you have to complete the relevant SMIT fields.

Command line interface

It is also possible to create the JFS2 file system using the command line interface (CLI). An additional VFS type was added to the **crfs** command.

When using CLI operations, the **crfs** command requires a **-v jfs2** flag in order to create a JFS2-type file system.

```
# crfs -v jfs2 -g rootvg -a size=1 -m /jfs2 -A yes -p rw -a agblksize=4096
mkfs completed successfully.
16176 kilobytes total disk space.
New File System size is 32768.
```

The output above illustrates a **crfs** command used to create a **/jfs2** file system using JFS2.

Web-based System Manager

You can manage JFS2 file systems from the Web-based System Manager interface. It is possible to create, enlarge, remove, and monitor JFS2 file systems from this management tool, as shown in Figure 4-34 on page 234.

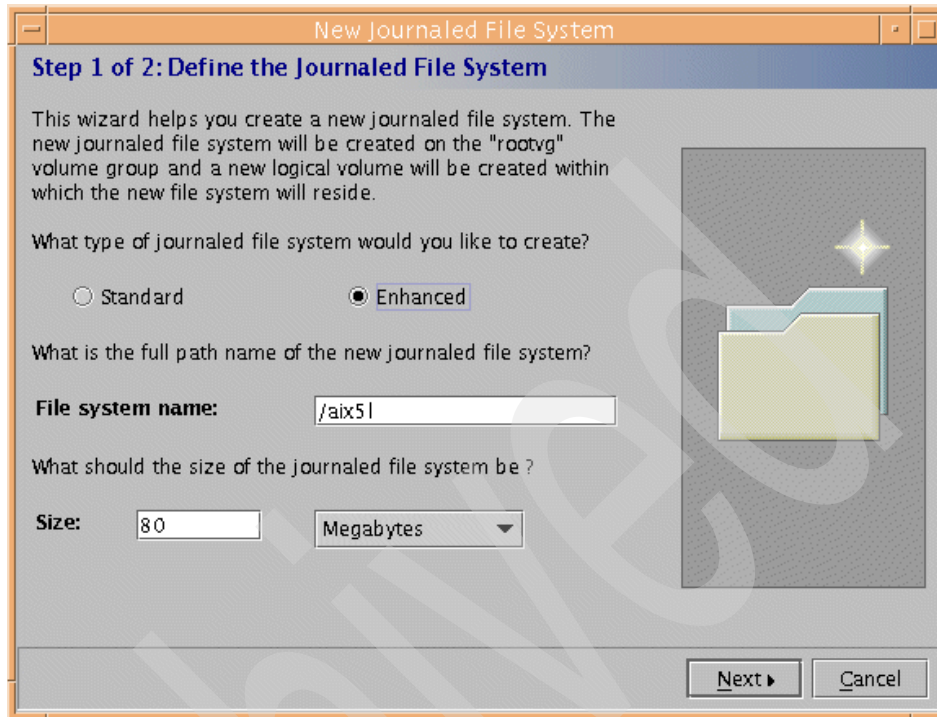


Figure 4-34 Web-based System Manager panel for file system creation

Check and recover file system

The **fsck** utility was enhanced to also handle JFS2-type file systems. This utility checks the file system for consistency and repairs problems found.

```
# fsck -V jfs2 /myfs
*****
The current volume is: /dev/lv01
File system is clean.
All observed inconsistencies have been repaired.
```

If the **-V** flag is not specified, **fsck** will figure out the JFS type by the VFS type specified for this file system and work in the assumed way:

```
# fsck /myfs
*****
The current volume is: /dev/lv01
File system is clean.
All observed inconsistencies have been repaired.
```


Creating a JFS2 log device

If you need to create a separate log device for a JFS2 file system, you must specify JFS2LOG as the logical volume type, as shown in Figure 4-35.

Add a Logical Volume

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]

Logical volume NAME

* VOLUME GROUP name

* Number of LOGICAL PARTITIONS

PHYSICAL VOLUME names

Logical volume TYPE

POSITION on physical volume

RANGE of physical volumes

MAXIMUM NUMBER of PHYSICAL VOLUMES

to use for allocation

Number of COPIES of each logical

partition

Mirror Write Consistency?

Allocate each logical partition copy

[MORE...11]

[Entry Fields]

[newlog]

rootvg

[1] #

[hdisk0] +

[jfs2log] +

middle +

minimum +

[] #

1 +

active +

yes +

F1=Help

F2=Refresh

F3=Cancel

F4=List

F5=Reset

F6=Command

F7=Edit

F8=Image

F9=Shell

F10=Exit

Enter=Do

Figure 4-35 SMIT panel for adding a logical volume as a jfs2log device

Otherwise, you will not be able to format the log device and use it as a log for a JFS2 file system.

Format a JFS2 log device

If you need to format a separate log device for a JFS2 file system, keep in mind that the **logform** command is set to **-V jfs2** flag in order to create a correct type of log device. For example:

```
# logform -V jfs2 /dev/jfs2log
logform: destroy /dev/jfs2log (y)?y
```

If the **-V** flag is not specified, the **logform** command will try to determine what kind of log device will be created through the VFS information encountered in the logical volume.

To verify the VFS type of a logical volume, you must check the output of the following command:

```
# lslv newlog | grep TYPE
TYPE:                jfs2log                WRITE VERIFY:    off
```

Inline log

A new type of log can be created for JFS2 type file systems. An inline log is a feature specific to JFS2 file systems that allows you to create the log within the same data logical volume.

With an inline log, each JFS2 file system can have its own log device without having to share this device. For a scenario with multiples of hot swap disk devices and large number of file systems, this feature can be used to improve RAS if a system loses a single disk that contains the log device for multiple file systems. See Figure 4-31 on page 231 for the SMIT panel with inline log enablement.

In the following example, the output for the **mount** command shows the logical volume and log device as the same device:

# mount node	mounted	mounted over	vfs	date	options
-----	-----	-----	-----	-----	-----
/dev/hd4	/		jfs	Sep 01 11:32	rw,log=/dev/hd8
/dev/hd2	/usr		jfs	Sep 01 11:32	rw,log=/dev/hd8
/dev/hd9var	/var		jfs	Sep 01 11:32	rw,log=/dev/hd8
/dev/hd3	/tmp		jfs	Sep 01 11:32	rw,log=/dev/hd8
/dev/hd1	/home		jfs	Sep 01 11:33	rw,log=/dev/hd8
/proc	/proc		procfs	Sep 01 11:33	rw
/dev/1v02	/jfs22		jfs2	Sep 05 10:00	rw,log=/dev/1v02

4.4.4 JFS2 rootvg support for 64-bit systems (5.1.0)

AIX 5L Version 5.1 introduced a feature to set all file systems in the rootvg as JFS2-type file systems.

While installing a system with the complete overwrite option, you can enable the 64-bit kernel and JFS2, as shown in Figure 4-36 on page 237. If this option is enabled, the installation task will create JFS2 file systems in the rootvg.

Advanced Options

Either type 0 and press Enter to install with current settings, or type the number of the setting you want to change and press Enter.

- 1 Installation Package Set..... Default
- 2 Enable Trusted Computing Base..... no
- 3 Enable 64-bit Kernel and JFS2..... yes

Figure 4-36 Advanced Options installation menu

If the system is not 64-bit enabled, the third menu item, regarding 64-bit kernel and JFS2, will not be displayed. If you do a migration install, the third menu item is also available, but it will not convert the existing file systems to JFS2. The installation task will install the 64-bit kernel only.

Complete overwrite installation

After an new and complete overwrite installation, all file systems in the rootvg are of the type JFS2, as shown in the following example:

```
# lsvg -l rootvg
rootvg:
LV NAME      TYPE      LPs   PPs   PVs   LV STATE      MOUNT POINT
hd5          boot      1     1     1     closed/syncd  N/A
hd6          paging    48    48    1     open/syncd    N/A
hd8          jfs2log   1     1     1     open/syncd    N/A
hd4          jfs2      1     1     1     open/syncd    /
hd2          jfs2      15    15    1     open/syncd    /usr
hd9var       jfs2      1     1     1     open/syncd    /var
hd3          jfs2      1     1     1     open/syncd    /tmp
hd1          jfs2      1     1     1     open/syncd    /home
hd10opt      jfs2      1     1     1     open/syncd    /opt
```

Migration installation

A migration BOS install does not convert the existing file systems to JFS2. But, of course, you can create JFS2 file systems later on. The following example shows rootvg file systems as JFS:

```
# lsvg -l rootvg
rootvg:
LV NAME      TYPE      LPs   PPs   PVs   LV STATE      MOUNT POINT
hd5          boot      1     1     1     closed/syncd  N/A
hd6          paging    48    48    1     open/syncd    N/A
```

hd8	jfslog	1	1	1	open/syncd	N/A
hd4	jfs	1	1	1	open/syncd	/
hd2	jfs	15	15	1	open/syncd	/usr
hd9var	jfs	1	1	1	open/syncd	/var
hd3	jfs	10	10	1	open/syncd	/tmp
hd1	jfs	1	1	1	open/syncd	/home

JFS2 support for NIM installations

For NIM installations, you have to customize the `bosinst.data` file if you want JFS2 for the root file systems. You need to enable the 64-bit kernel and JFS2 file systems option from the BOS install. In order to do that, the `INSTALL_64BIT_KERNEL` field needs to be set to `yes`.

Extract from the `bosinst.data` file:

```
control_flow:
  CONSOLE = /dev/tty0
  INSTALL_METHOD = overwrite
  PROMPT = no
  EXISTING_SYSTEM_OVERWRITE = yes
  INSTALL_X_IF_ADAPTER = yes
  RUN_STARTUP = yes
  RM_INST_ROOTS = no
  ERROR_EXIT =
  CUSTOMIZATION_FILE =
  TCB = no
  INSTALL_TYPE =
  BUNDLES =
  SWITCH_TO_PRODUCT_TAPE =
  RECOVER_DEVICES = yes
  BOSINST_DEBUG = no
  ACCEPT_LICENSES = no
  INSTALL_64BIT_KERNEL = yes
  INSTALL_CONFIGURATION = Default
```

Note: Only 64-bit enabled systems support NIM installations of the 64-bit kernel and JFS2 support for root file systems.

4.4.5 JFS2 performance enhancements (5.1.0)

To enhance the performance on a JFS2 file system, a vnode cache has been added and the inode generation numbers have changed.

vnode cache

The problem is that on each access of a file (vnode) by NFS, the vnode and its accompanying inode must be reactivated. Use of a vnode cache keeps these objects in an active state and it becomes much simpler to find and use them. The

vnode cache has been adapted from the existing JFS design and implemented in JFS2.

- ▶ The existing interfaces have been renamed.
- ▶ Old interface names versus new interface names is provided in Table 4-8.
- ▶ The vnc_remove interface has changed to handle the JFS2 requisites.
- ▶ The inode numbers are increased in size to 64 bits.
- ▶ The size of the cache had been tied to the size of the JFS inode cache. The default number is 50 cache entries per megabyte of real memory.

Table 4-8 Old JFS names versus new JFS2 interface names

Existing interface name	New interface name
jfs_vnc_init	vnc_init
jfs_vnc_lookup	vnc_lookup
jfs_vnc_enter	vnc_enter
jfs_vnc_remove	vnc_remove
jfs_vnc_purge	vnc_purge

File system changes

To improve the hash key distribution, the inode generation number has changed. In AIX 5L Version 5.0, the inode generation number started at zero when a file system was mounted, and new inodes got ever-increasing values. In AIX 5L Version 5.1, the inode generation number starts at a number derived from the current time. This results in more non-zero bits and more variation.

4.4.6 JFS2 support for filemon and fileplace (5.2.0)

Support for JFS2 has been added to the **fileplace** command in AIX 5L Version 5.2. A new flag has been added to the **fileplace** command to display the logical-to-physical mapping for a logical volume. The syntax is as follows:

```
fileplace [-m] lvname
```

The **filemon** command has been enhanced so that the description field in the Most Active Logical Volumes section contains the details of the JFS2 logical volume getting accessed.

4.4.7 JFS2 large file system (5.2.0)

In Version 5.2, JFS2 can have a 1 TB file system on a 32-bit machine and 16 TB on a 64-bit machine running the 64-bit kernel.

4.4.8 JFS and JFS2 file system sizes (5.2.0)

Version 5.2 introduces several 64-bit version commands. This enables the use of a very large JFS2 file system, up to 16 TB, on a 64-bit machine running the 64-bit kernel. The 32-bit version of these commands still coexist and are always called first. If a 64-bit kernel is the currently running kernel, then a new child process is forked to call these commands' 64-bit version.

The `mk1v` command has been changed to support the creation of a logical volume up to 1 TB when using the 32-bit kernel. When using the 64-bit kernel it is possible to create a logical volume up to 128 TB in size.

4.4.9 JFS2 log sizes (5.2.0)

In previous versions of AIX the outline log had a maximum size of 1 GB and the inline log had a maximum size of 32 MB unless otherwise specified by the user. These logs sizes were insufficient for file systems up to 16 TB, and therefore these maximum log sizes have been changed.

The inline log size can be from 256 KB up to 2 GB (a warning will be generated if this size is exceeded) depending on the size of file system. A new algorithm has been created to calculate the appropriate size of the log. The outline log is dynamic in nature, as many file systems of varying sizes may use the same outline log. For 32-bit kernel, the outline log can be up to 1 GB and for 64-bit kernel the outline log can be up to 64 GB. For more information on log sizes, see:

http://publib16.boulder.ibm.com/pseries/en_US/infocenter/base

4.4.10 JFS2 performance enhancements (5.2.0)

The reserved but not allocated heuristic has been added to JFS2 on Version 5.2. The introduction of the reserved but not allocated heuristic essentially delays the writing of smaller files to disks. In the case of temporary files, these file types may never be written to disk at all before they are removed from memory, thus removing the overhead of a disk write. This also aids contiguous allocation of disk space by batching up the reservation of small incremental writes and allocating them as a single contiguous extent.

What affects the allocation

JFS2 delays the allocation of the last 32 4-KB pages of a file to disk space by holding them in memory for as long as is reasonably possibly, while guaranteeing that the space for the eventual write is available. A small temporary file, defined as a file that is equal to or less than 32 4-KB pages (128 KB), will probably never be written to disk. Larger files, greater than 128 KB, will also benefit as the contiguity of the disk is enhanced by this feature.

There are a number of factors that control when files are written to disk. They include the `minfree` parameter, `syncd`, the `sync` command, and random-write-behind threshold. These parameters are tunable, and can be tuned to further enhance the way in which small file allocation is delayed as long as possible. Caution should be exercised when changing any of these parameters as each one can drastically change the systems operation.

In Version 5.2, the `vm tune` command is being phased out and simply calls three new commands. They are: `vmo` (vmm parameters), `ioo` (I/O parameters), and `vmstat`. The changes are referenced in the online documentation. The factors controlling when files are written to disk are discussed in more detail below:

- ▶ `minfree`

This parameter refers to the minimum number of memory frames on the free list, once this threshold is reached the VMM page stealer starts to free pages. This causes allocations (writes) of files to disk. This parameter is tunable with the `vmo` command.

- ▶ `syncd`

The sync daemon, `syncd`, by default will start allocations every sixty seconds. This attribute is tunable by altering the startup value as specified in `/sbin/rc.boot`.

- ▶ `sync`

If the `sync` command is manually called, all files will be written to disk. This would not normally occur.

- ▶ Write-behind (`j2_maxRandomWrite`)

This is the asynchronous write of dirty pages in memory to disk rather than relying on `syncd`. In JFS2, the write-behind parameter and also other parameters that control writes are tunable with the `ioo` command, using the variable names:

- `j2_maxRandom Write`

The number of files in RAM before pages are allocated to disk.

- j2_nPagesPerWriteBehindCluster

The number of pages per cluster (16 KB partition consisting of 4 KB pages) processed by JFS2 write-behind algorithm.

- j2_nRandomCluster

Specifies the distance a cluster must be apart to be considered random by the write-behind algorithm.

4.4.11 JFS2 snapshot image (5.2.0)

Version 5.2 introduces the JFS2 snapshot image. The JFS2 snapshot image gives a consistent block level image of a file system at a given point in time. The snapshot will stay stable even if the file system that the snapshot was taken from, referred to hereafter as the snappedFS, continues to change.

The snapshot can then be used to create a backup of the file system at the given point in time that the snapshot was taken. The snapshot also provides the capability to access files or directories as they were at the time of the snapshot.

Version 5.2 provides the following functionality for a snapshot image:

- ▶ Snapshot creation on a separate logical volume from the snappedFS.
- ▶ Read-only access to a snapshot through a mounted file system.
- ▶ Read-only access to a snappedFS while snapshot is created.
- ▶ Snapshot information listing.
- ▶ Snapshot removal.
- ▶ Capability of multiple snapshots for a file system.
- ▶ Snapshots are persistent when snappedFS is mounted or unmounted. Not persistent if system crash occurs.
- ▶ Backup support for backbyname and backbynode.

Overview of JFS2 snapshot

During creation of a snapshot the file system being snapped, the snappedFS, will be quiesced and all writes are blocked. This ensures that the snapshot really is a consistent view of the file system at the time of snapshot. When a snapshot is initially created, only structure information is included. When a write or delete occurs then the affected blocks are copied into the snapshot file system.

Write operations on a snapshot have a performance impact caused by the additional overhead of making sure there is consistency between file systems during write operations and the overhead of moving the prior version of an updated block.

Read operations on the snappedFS remain unaffected, although every read of the snapshot will require a lookup to determine whether the block needed should be read from the snapshot or from the snappedFS. For instance, the block will be read from the snapshot file system if the block has been changed since the snapshot took place. If the block is unchanged since the snapshot, it will be read from the snappedFS. A snapshot, once completed, can be used to make a backup of the file system and is able to guarantee the consistency of the backup image.

This operation makes use of the snapshot map, whose location is stored in the snapshot superblock. The snapshot map logically tracks the state of the blocks in the snappedFS and contains the following details:

- ▶ Block address of blocks that were in use in the snappedFS at the time the snapshot was taken.
- ▶ Block address of blocks in the snappedFS that were in use and have subsequently been modified or deleted after the snapshot was created.
- ▶ Block address of newly allocated blocks in the snapshot that contain the before image of blocks that have been deleted or written to.

Typically, a snapshot will need two to six percent of the space needed for the snappedFS. In the case of a highly active snappedFS. This estimate could rise to 15 percent, although this is really file system dependent. This space is needed if either a block in the snappedFS is either written to or deleted. If this happens the block is copied to the snapshot. Therefore, in highly active file systems the space in a snapshot file system can be used quite rapidly. Any blocks associated with new files written after the snapshot was taken will not be copied to the snapshot, as they were not current at the time of the snapshot and therefore not relevant.

If the snapshot runs out of space, the snapshot will be discarded as would any other snapshots associated with the snappedFS. Two possible entries could be created in the AIX error log. They have either of the following labels: J2_SNAP_FULL or J2_SNAP_EIO. If a snapshot file system fills up before a backup is taken, the backup is not complete and will have to be re run from a new snapshot, with possibly a larger size, to allow for changes in the snappedFS.

JFS2 file systems from previous versions of AIX are fully supported for snapshot images. Snapshot information is stored in a region of the superblock. It is not possible to mount snapshots on a system running AIX at a version prior to Version 5.2.

Creation of a JFS2 snapshot

JFS2 snapshots can be created through the command line, SMIT, or the Web-based System Manager. The following example uses the last of these tools to illustrate the process.

Select **File Systems** in the left menu and from there **Journalled File Systems** and the JFS2 (or Enhanced, as referred to in Web-based System Manager) that is to be snapped. Now either right-click the file system or to go to the selected drop-down menu. (This is shown in Figure 4-37.)

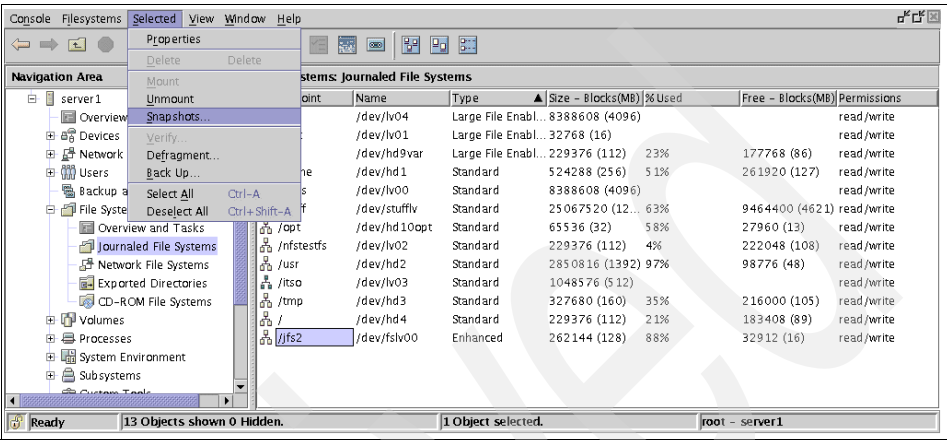


Figure 4-37 Selecting snapshot in the Journalled File Systems submenu

This leads to a screen where the **Create** button on the right-hand side should be selected (this is shown in Figure 4-38 on page 245).

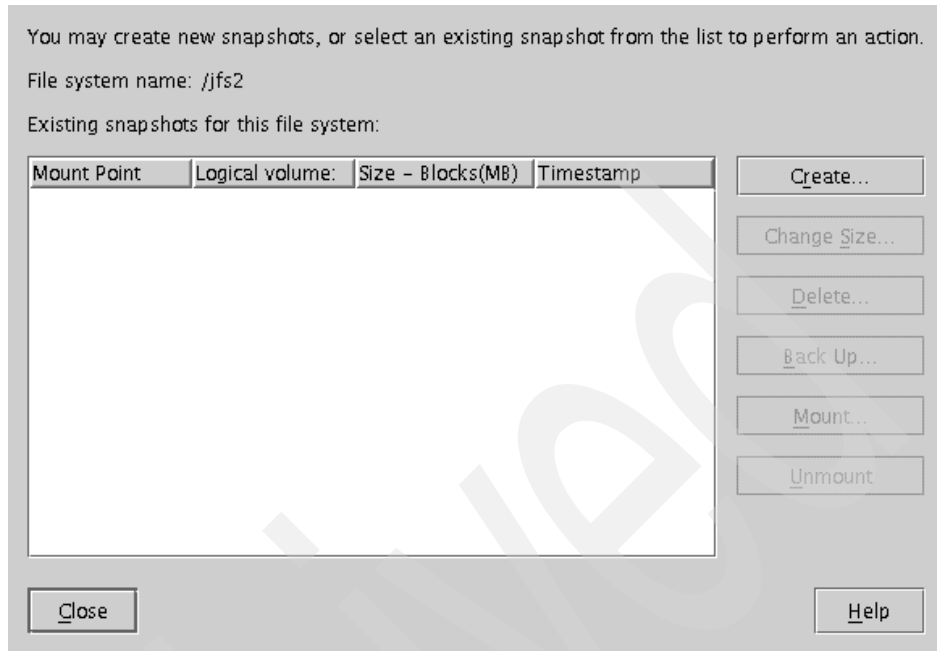


Figure 4-38 Snapshot creation screen, click Create

The Create button takes the user to the following screen, where it is possible to input the snapshot file system size and the mount point, back up the snapshot to removable media, and mount the snapshot after creation (default is yes). These options are selected in Figure 4-39 on page 246.

The screenshot shows a graphical user interface for creating a snapshot. At the top, there is a dropdown menu for 'Logical volume:' set to 'system default'. Below it, 'Snapshot size:' is set to '10' in a text box, with a dropdown menu for 'Megabytes' to its right. A section titled 'Backup options' contains several settings: a checked checkbox for 'Remove snapshot after backup completes', a 'Remove choices:' dropdown set to 'snapshot only', a 'Backup device:' dropdown set to 'No backup performed now', and two unchecked checkboxes for 'Pack files on backup' and 'Display verbose output'. Below this section, there is a checked checkbox for 'Mount the snapshot after it is created' and a 'Mount point:' text box containing '/jfs2_snap'. At the bottom, there are three buttons: 'OK', 'Cancel', and 'Help'.

Figure 4-39 Snapshot creation screen with options configured

Once **OK** is clicked this will go back to the initial snapshot screen but will show the snapshot file system created. If this file system is created, it is possible to change its size, unmount it, or back it up, as shown in Figure 4-40 on page 247.

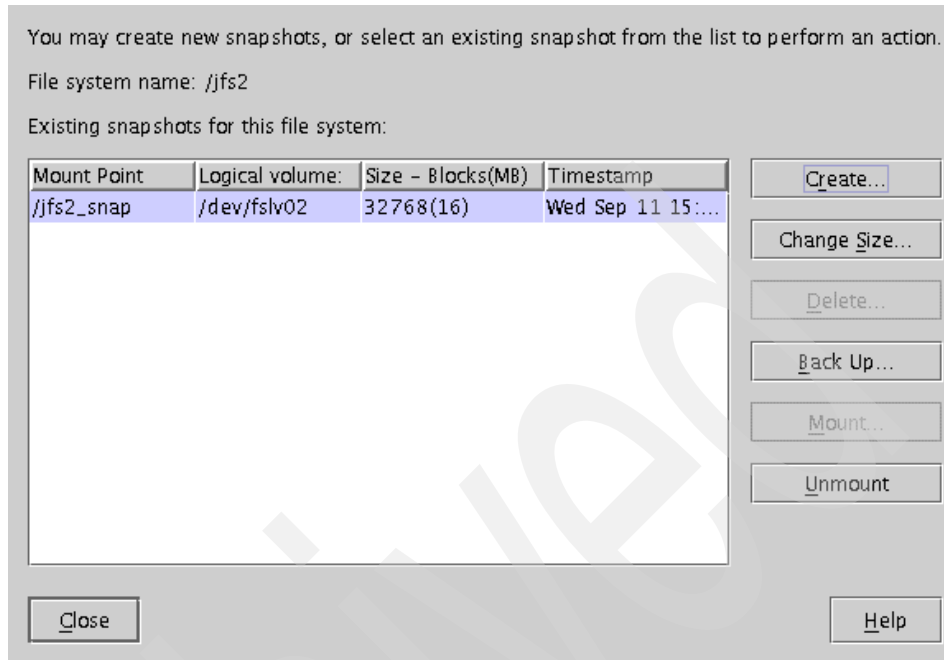


Figure 4-40 It is possible to changes its size, back it up, or unmount it

If the snapshot is unmounted, different options are possible, such as the Delete option. It is only possible to delete a snapshot when it is unmounted, as shown in Figure 4-41 on page 248.

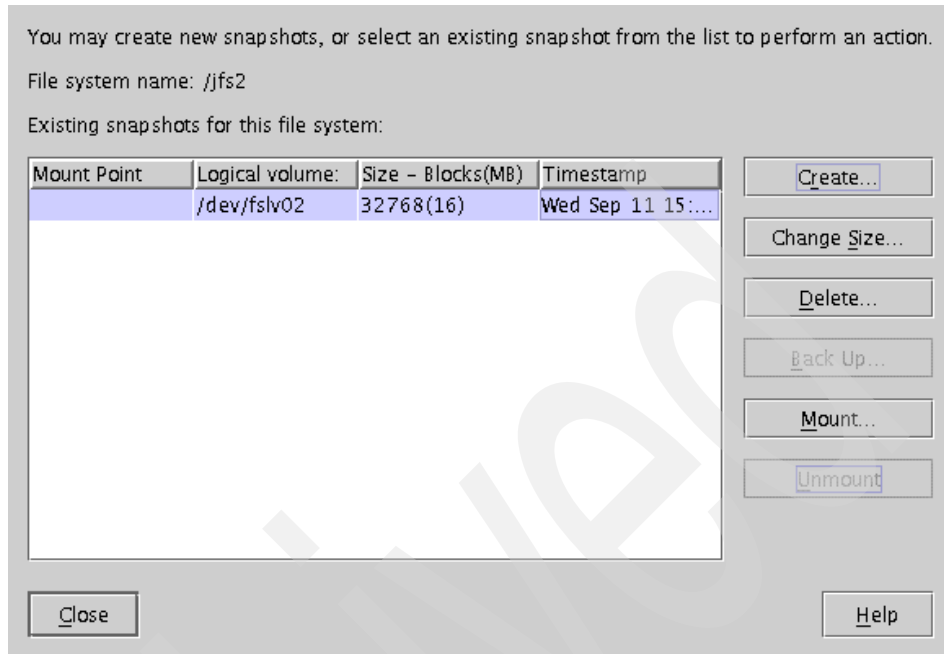


Figure 4-41 Possible to delete unmounted snapshots

Once mounted again, it is possible to go and back the snapshot up to removable media (this is shown in Figure 4-42).

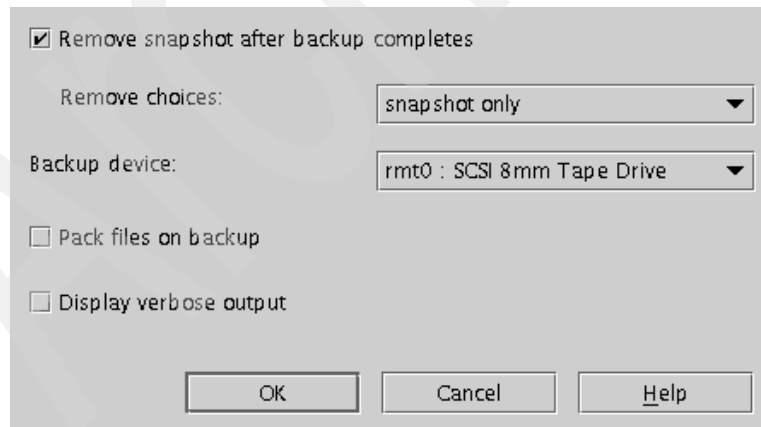


Figure 4-42 Snapshot image screen

At the AIX command line, the two file systems appear as shown in the following:

```
# df -k |grep jfs2
```

/dev/fs1v00	131072	16456	88%	64	2%	/jfs2
/dev/fs1v02	16384	16000	3%	-	-	/jfs2_snap

Snapshot hints

There are a few snapshot-specific concepts worth noting:

- ▶ Deleting a snapshot is only possible with SMIT, Web-based system manager, and the command line (`snapshot -d`) once the snapshot file system is unmounted.
- ▶ If the `chfs` command is run on a snappedFS it will have no effect on the snapshot file system. This is because the snapshot will not need to know about any new blocks (or new files created after the snapshot was taken).
- ▶ Backing up a snapshot is possible as long as a snapshot file system is not full and hence invalidated. It is possible to back up the snapshot using the following methods: `tar`, `cpio`, `backbyname`, and `backbyinode`. The `backbyinode` command does not require the snapshot to be mounted.

New commands or commands with new function

To support JFS2 snapshot images there are a number of new commands included in Version 5.2. Full documentation is provided by the online documentation and man pages. The syntax is provided here for information only:

- ▶ **snapshot** - Creates, deletes, and queries a snapshot.
`snapshot { -o snapfrom=snappedFS -o size=Size | {-d [-s] |
-q [-c fieldSeparator] | -o snapfrom=snappedFS | -o size=Size} Object}`
- ▶ **backsnap** - Creates and backs up a snapshot.
`backsnap [-R] -m MountPoint -s size=Size [BackupOptions] file system`
- ▶ **fsdb** - Examines and modifies snapshot superblock, snapshot map, block xtree copy, and segment headers.
`fsdb file system [-]`
- ▶ **mount** - Caters for snapshots.
 - **mount -o snapshot** - Specifies device is a snapshot
 - **mount -o snapto=snapshot** - When mounting a JFS2 file system, start a snapshot to it to the specified device
- ▶ **umount** - Caters for snapshots. Mounted snapshots must be unmounted before the snappedFS can be unmounted.
- ▶ **dumpfs** - This command can be run against a snapshot and will display information on the superblock, snapshot map, and block map xtree copy.

Commands to exercise caution with

There are three commands whose impact of running should be understood before their execution. They are as follows:

- ▶ **defragfs**

All data that moved would have to be copied into the snapshot area. This could be a large amount of data that could fill the snapshot. Therefore we recommend deleting any snapshots on the snappedFSs, run the command, and recreate the snapshots. The command will run, and data is not lost, but the results will not be what you expected.

- ▶ **fsck**

The **fsck** command modifies the snappedFS. Any associated snapshots cannot guarantee that they contain all the before images of the snappedFS. **fsck**, therefore, deletes snapshots of snappedFSs that it is run against.

- ▶ **logredo**

The snapshots cannot guarantee that they contain all the before images of the snappedFS. **logredo** will delete snapshots associated with the snappedFS.

Packaging

The **snaphost** and **backsnap** commands are packaged as follows:

- ▶ The **snaphost** command is packaged in the bos.rte.file fileset and `/usr/sbin/snapshot` is a symbolic link to `/sbin/helpers/jfs2/snapshot`.
- ▶ The **backsnap** command is packaged in the bos.rte.file fileset and `/usr/sbin/backsnap` is a symbolic link to `/sbin/helpers/jfs2/backsnap`.

4.5 VERITAS Foundation Suite for AIX (5.1.0)

VERITAS Foundation Suite for AIX has recently been announced for the IBM AIX 5L Version 5.1 operating system. VERITAS NetBackup has been available for some years on IBM's AIX platform, but since May 2002, VERITAS Foundation Suite has been available.

VERITAS Foundation Suite for AIX is comprised of two base products: VERITAS Volume Manager (VxVM) and VERITAS File System (VxFS), plus VERITAS Enterprise Administrator (VEA) graphical user interface (GUI). VVR and VCS are separate products that require separate licenses. VERITAS FlashSnap is an advanced feature of VERITAS Foundation Suite for AIX that requires a separate license key. Note that VxVM and VxFS are not available as separate products on the AIX 5L Version 5.1 platform.

VERITAS Volume Manager is a simple to use, yet powerful disk and storage management system for enterprise computing. It supports online disk management, thus affording continuous data availability. Disk configuration can be done online without impacting users. VxVM also supports disk striping and disk mirroring. For data redundancy and protection against disk and hardware failures, VxVM supports RAID levels RAID 0 (disk striping), RAID 1 (disk mirroring), RAID 5, RAID 0+1, and RAID 1+0.

VERITAS File System is a reliable, scalable, fast-recovery journaling file system with increased data availability and data integrity features. Data availability is at the level necessary for mission-critical systems, where file system data is available within seconds of a system crash and reboot. Data integrity is maintained through the journaling file system that records changes in an intent log and then recovers from a crash using that log. Online management features are available with VxFS, such as file system backup, defragmentation, and growing and shrinking file systems.

The VERITAS Enterprise Administrator (VEA) GUI is provided with VERITAS Foundation Suite for AIX and supports both VxVM and VxFS. VEA enables easy online volume management and file system management. This is available not only for managing a set of AIX machines, but in a heterogeneous environment with many platforms. VEA can be used to do disk management across all the platforms simultaneously. From just one VEA console, multiple hosts and operating systems can be managed.

4.5.1 VERITAS Foundation Suite on the AIX Bonus Pack

An evaluation version of VERITAS Foundation Suite for AIX and Foundation Suite/HA for AIX are both available on the AIX 5L Version 5.1 July 2002 Bonus Pack. The Foundation Suite/HA is the high-availability version of the Foundation Suite, and includes VERITAS Cluster Server. Both VERITAS Foundation Suite for AIX and Foundation Suite/HA for AIX are full-featured versions of the software. Once you have installed the software, you need to request a demo license directly from VERITAS. The demo license is valid for 60 days.

4.5.2 Why use VERITAS Foundation Suite on AIX

Although the IBM AIX operating system has its own native Logical Volume Manager (LVM) and journaled file system (JFS) that provide similar functionality to the VERITAS Foundation Suite components, there are compelling business reasons to use VERITAS Foundation Suite for AIX. The key differentiator is the common cross-platform management and integration.

For organizations that already have the required skill base on VERITAS Foundation Suite on other platforms, such as SUN Solaris, HP-UX, or others,

there is an easy migration from those platforms to AIX. No additional storage management software training is required to support the AIX platform. The functionality of VERITAS Foundation Suite on other supported platforms is the same as that on AIX. The GUI interface provided with VERITAS Foundation Suite is common across Solaris, AIX 5L, Windows, HP-UX, and Linux. Additionally, users can take advantage of the comprehensive features of VERITAS Foundation Suite for AIX, which are described in the following chapters.

One of the most important reasons for using VERITAS Foundation Suite on AIX is the ease of use in a heterogeneous environment with servers from IBM, SUN, HP, and others. In a heterogeneous environment, being able to use one common storage management software system makes the administrator's job much simpler. Common storage management lowers overall administrative costs and gives better total cost of ownership by reduced training costs. By using VERITAS Foundation Suite on AIX, the power of VERITAS software is available on the wide range of IBM @server pSeries servers, providing world-class solutions for organizations.

4.5.3 Support for LVM and JFS for AIX

IBM continues to support the native Logical Volume Manager and journaled file system for IBM AIX 5L Version 5.1. LVM and JFS are strategic products for IBM, and continue to be developed and enhanced.

It is possible for LVM and JFS/JFS2 to easily coexist with VERITAS Foundation Suite on the same AIX machine. It is possible to have the LVM and JFS/JFS2 used for one physical volume and VERITAS Volume Manager and VERITAS File System used on another physical volume on the same machine.

4.6 AIX iSCSI Initiator Version 1.0 (5.2.0)

AIX iSCSI Initiator Version 1.0 allows AIX to send and receive SCSI commands and responses over TCP/IP. Because TCP/IP and Ethernets are widely deployed, using iSCSI is very attractive for storage access. iSCSI is particularly attractive in server farms where large numbers of servers are deployed. It enables storage access without requiring Fibre Channel adapters and associated storage area network (SAN) infrastructure by making use of network adapters and LANs. Compared to Fibre Channel SANs, performance of iSCSI is lower because of the overhead associated with TCP/IP. Thus, it is not recommended for storage-intensive applications such as database servers. iSCSI Protocol draft Version 0.8 is supported on AIX 5L Version 5.2. The AIX iSCSI Initiator is available in the AIX Bonus Pack.

4.7 NFS enhancements

The following are the enhancements that have been made to NFS.

4.7.1 NFS statd multithreading

In AIX 5L, the NFS statd daemon is multithreaded. In AIX Version 4.3, when the statd daemon is detecting whether the clients are up or not, it hangs and waits for a time out when a client cannot be found. If there are a large number of clients that are offline, it can take a long time to time out all of them sequentially. In AIX 5L, rpc.statd is now running as a daemon user, not as root user.

With a multithreading design, stat requests run in parallel to solve the time-out problem. The server statd monitors clients and the client's statd monitors the server if a client has multiple mounts. Connections are dropped if the remote partner cannot be detected without affecting other stat operations. The following example is an output from the **ps -mo THREAD** command that shows three different threads for rpc.statd daemon:

```
# ps -mo THREAD -p 17570
  USER  PID  PPID    TID ST  CP PRI SC   WCHAN      F   TT  BND  COMMAND
  daemon 17570  6456      -  A   0  60  3      - 240001      -   -   -
/usr/sbin
/rpc.statd
  -      -      - 20409 S   0  60  1      - 418400      -   -   -
  -      -      - 26065 Z   0  60  1      - c00001      -   -   -
  -      -      - 26579 Z   0  60  1      - c00001      -   -   -
```

4.7.2 Multithreaded AutoFS

In AIX 5L, the automountd daemon implementing the AutoFS function is now multithreaded, as can be seen from the following output of the **ps** command:

```
# ps -fmo THREAD -p 19134
  USER  PID  PPID    TID ST  CP PRI SC   WCHAN      F   TT  BND  COMMAND
  root 19134  6456      -  A   0  60  2 e60056a0 240001      -   -   -
/usr/sbin
/automountd
  -      -      - 35747 S   0  60  1      - 418400      -   -   -
  -      -      - 44443 S   0  60  1 e60056a0 8410400      -   -   -
```

With this new feature, the AutoFS mounter daemon remains responsive, even if one of the servers from which it tries to mount file systems becomes unavailable. As a single-threaded application, it would not be possible for the kernel to switch to the corresponding process if that process waits for a network connection to an unresponsive server.

4.7.3 Cache file system enhancements

In AIX 5L, the cache file system (cachefs) allows 64-bit operations. In both 32- and 64-bit environments, cachefs now handles files larger than 2 GB. In AIX Version 4.3.3 and earlier releases, cachefs only runs on a 32-bit system and all files must be 2 GB (at a maximum).

When making the transition from a 32-bit POWER kernel to a 64-bit POWER kernel, there is no need to recreate the cache directory.

4.7.4 The cachefslog command (5.1.0)

A new command is available in AIX 5L Version 5.1 named **cachefslog**. To use the **cachefslog** command, you must be logged in as the superuser. The following example shows the setup of a cache file system (CacheFS) and the use of the **cachefslog** command to set up cache file system logging. In the example, the NFS mount point and exported file systems have already been set up, but are not mounted through the use of the standard **mount** command. The /home file system of server3 is to be mounted locally on the /mnt directory using the following command:

```
# mkcfsmnt -d /mnt -t nfs -h server3 -p /home -c /my_cachefs -N
```

If the **df -k** command is invoked, the mount point is displayed in the following manner:

Filesystem	1024-blocks	Free	%Used	Iused	%Iused	Mounted on
server3:/home	16384	15800	4%	25	1%	
/my_cachefs/.cfs_mnt_points/_home						
server3:/home	16384	15800	4%	25	1%	/mnt

The purpose of the **cachefslog** command is to display and set up where CacheFS statistics are logged. The cachefslog file is used to log CacheFS statistics, such as populating and removing files, and so forth. At this point in the example there is no log file for CacheFS. This is evident after running the following command:

```
# cachefslog /mnt
not logged: /mnt
```

To set up the file /my_cachefs/cachelog to log the statistics for CacheFS, the following command should be used:

```
#cachefslog -f /my_cachefs/cachelog /mnt
/my_cachefs/cachelog: /mnt
```

To verify that this file is being used as the `cachefslog`, the following command should be used:

```
# cachefslog /mnt
/my_cachefs/cachelog: /mnt
```

Logging for a directory such as `/mnt` can be stopped as follows:

```
#server1:/>cachefslog -h /mnt
not logged: /mnt
```

The information that is logged in the file, specified by the `cachefslog` command, can be displayed with the following command:

```
# cachefswsize -a /my_cachefs/cachelog
```

The resulting output from the command will appear similar to that displayed in the following example and is used for debugging purposes only:

```
3/19 14:25 0 Mount 3098fa44 211 65536 256 /mnt (_ftpctest:_mnt)
3/19 14:33 0 Filldir 3098fa44 <fid> 2 4096
3/19 14:33 0 Rfdir 3098fa44 <fid> 2 0
3/19 14:33 0 Rfdir 3098fa44 <fid> 2 0
3/19 14:33 22 Rfdir 3098fa44 <fid> 2 0
3/19 14:33 22 Rfdir 3098fa44 <fid> 2 0
3/19 14:33 22 Rfdir 3098fa44 <fid> 2 0
3/19 14:33 22 Rfdir 3098fa44 <fid> 2 0
3/19 14:33 0 Rfdir 3098fa44 <fid> 2 0
3/19 14:34 0 Mdcreate 3098fa44 <fid> 24576 1
3/19 14:34 0 Filldir 3098fa44 <fid> 24576 4096
3/19 14:34 0 Rfdir 3098fa44 <fid> 24576 0
```

4.7.5 NFS cache enhancement

NFS is now able to cache file names longer than 31 characters.

4.7.6 Netgroups for NFS export (5.1.0)

A netgroup file can be created on an NFS server to list a group of systems that can access a network file system. In the following example, the host name of the NFS server is `itsos7a`. Using netgroups makes system administration of NFS mounts easier. The following example shows the format of the `/etc/netgroup` file:

```
root_group_name (server1,,)
(server2,,)
(server3,,)
```

The group has a label name of `root_group_name`. Any label name can be used. The three fields within parentheses are known as a triple. The first field of the

triple is the name of a server, the second field is the user name, and the third field is the domain name. In the preceding example, the second and third fields are not required. The names server1, server2, and server3 are the names of systems that are required to access network file systems on the NFS server itsos7a.

The /etc/netgroup file is searched before /etc/hosts, if it exists. Therefore, the netgroup name is always searched before the host name.

The /etc/exports file must be edited to include an entry for the exported file system, as in the following example:

```
/home -access=root_group_name
```

The implication from the preceding examples of the /etc/netgroup and /etc/exports files is that the systems named server1, server2, and server3 will be able to mount and access the data on the /home file system of the NFS server itsos7a. To mount the /home file system of the NFS server itsos7a from the client system server1, enter the following command:

```
# mount itsos7a:/home /mnt
```

Additional groups can be added to the /etc/netgroup file as shown below, and additional exports can be added to the /etc/exports file:

```
root_group_name (server1,,)
(server2,,)
(server3,,)
my_group (swift,,)
(concorde,,)
```

4.7.7 unmount command enhancement (5.2.0)

A new -f flag has been added to force the unmount of NFS file systems.

This function adds support in the automount subsystem to shut down the automounter including unmounting all file systems, regardless if there is activity on those file systems or not. This includes changes in the NFS file system code to handle forceful unmounting of NFS file systems. Note that all the data in the cache is discarded. For example, the following command shows a forced unmount:

```
# lsof |grep nfsfs
ksh      52412    root  cwd   VDIR      NFS,28 1020759107436544    3 /nfsfs
(9.3.4.98:/nfsfs)
vi       56646    root  cwd   VDIR      NFS,28 1020759107436544    3 /nfsfs
(9.3.4.98:/nfsfs)
# unmount -f /nfsfs
```

forced unmount of /nfsfs

In the previous example, the **lsof** command shows two open files that belong to the /nfsfs NFS file system. Despite those open files, the file system is unmounted using the -f flag.

The **lsof** command is part of the RPM can be downloaded from the Web to the following URL:

<ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/lsof/lsof-4.61-2.aix5.1.ppc.rpm>

Or installed as follows:

```
# rpm -i
ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/lsof/lsof-4.61-2.aix5.1.ppc.rpm
# rpm -q lsof
lsof-4.61-2
```

4.8 CD-ROM/DVD-RAM automount facility (5.2.0)

You can now automatically mount a CD-ROM/DVD-RAM file system when a media is inserted in a drive. User commands to mount, unmount the file system, and eject the media from the drive are also available.

The CD-ROM/DVD-RAM automount facility is contained in the bos.cdmount fileset, which is installed by default.

4.8.1 The cdromd daemon

This automount capability for CD-ROM/DVD-RAM file systems is implemented in the cdromd daemon. The cdromd daemon is controlled by the system resource controller. To start the cdromd daemon, issue the following command:

```
startsrc -s cdromd
```

To have the cdromd daemon started at system startup, include the cdromd daemon in the /etc/inittab by issuing the following command:

```
mkinitab "cdromd:23456789:wait:/usr/bin/startsrc -s cdromd"
```

When started, the cdromd daemon reads the /etc/cdromd.conf configuration file to get the list of devices to manage and their mount point, and the list of supported file systems and their mount options. By default (no entry in cdromd.conf), all the available CD-ROM devices in CuDv are used and the default mount point is defined as /cdrom/cdX; the supported file system types are

cdrfs and udfs, and the mount options are -V cdrfs -o ro and -V udfs -o ro, respectively. For a description of the syntax of the cdromd.conf file refer to the file itself.

For each device to be managed, cdromd allocates and initializes a device structure, and issues an open on the corresponding device driver. The openx() (extended open) is used with the SC_DIAGNOSTIC flag for SCSI devices, and SC_SINGLE for IDE devices. With these flags, the open will succeed even if no media is present, and will reserve the access to the device. Any application attempting to open one of these devices will get an EACCES error code. If an application is using the device when cdromd is started, this open will fail, indicating that the device is busy, and the openx() will be attempted later.

The cdromd daemon then creates a UNIX socket that will be used by the user commands to issue requests to the cdromd daemon.

After initialization completes, the cdromd daemon loops and periodically checks if media is present in one of the drives (for devices that are not already mounted), or if a message is available on the socket.

4.8.2 User commands for the automount facility

User commands are available to unmount and eject the specified device. In addition, further functions to control and check the cdromd are provided. The list of functions is as follows:

- ▶ Unmount the file system and eject the media.
- ▶ Only unmount the file system.
- ▶ Re-mount the file system.
- ▶ Check if a media is present in the device.
- ▶ Check if a media is mounted.
- ▶ Check if a device is managed by cdromd daemon.
- ▶ Suspend the management of a device by cdromd daemon.
- ▶ Resume the management of a device by cdromd daemon.

The commands to execute these functions are **cdutil**, **cdeject**, **cdumount**, **cdmount**, and **cdcheck**. The latter four are links to **cdutil**. An overview of these commands is in the following:

- ▶ **cdcheck**

```
cdcheck {-a|-e|-m|-u} [-q] [-h|-?] device_name|mount_point
```


The **cdcheck** command asks cdromd daemon information about a device. To check if a media is mounted on device cd0, issue the following command:

```
cdcheck -m cd0
```

► **cdeject**

```
cdeject [-q] [-h|-?] device_name|mount_point
```

The **cdeject** command ejects a media from a CD drive managed by the cdromd daemon. To eject a media from drive cd0, issue the following command:

```
cdeject cd0
```

► **cdmount**

```
cdmount [-q] [-h|-?] device_name|mount_point
```

The **cdmount** command takes a file system available for use on a device managed by the cdromd daemon. To mount a file system on device cd0, issue the following command:

```
cdmount cd0
```

► **cdumount**

```
cdumount [-q] [-h|-?] device_name|mount_point
```

The **cdumount** command unmounts a previously mounted file system on a device managed by cdromd daemon. To unmount a file system on device cd0 issue the following command:

```
cdumount cd0
```

► **cdutil**

```
cdutil {-l|-r|-s [-k]} [-q] [-h|-?] device_name|mount_point
```

The **cdutil** command tells the cdromd daemon to load a media or to suspend or resume management of a device. To suspend device management of cd0 by cdromd daemon without ejecting the media, issue the following command:

```
cdutil -sk cd0
```

Table 4-9 provides a description of the most important flags of the commands described previously.

Table 4-9 CD-ROM/DVD-RAM automount flags

Flag	Description
-a	Checks if a device is managed by cdromd.
-e	Checks if a media has been ejected from a device.
-l	Loads the media, if one is present in the drive.

Flag	Description
-m	Checks if a media is mounted on a device.
-q	Specifies silent mode: Does not print any information or error message. This is useful when called in shell-scripts.
-r	Resumes device management by cdromd.
-s	Suspends device management by cdromd and eject media.
-sk	Suspends device management by cdromd and does not eject media.
-u	Checks if a media is not mounted on a device.

4.9 Uppercase mapping for ISO CD-ROM (5.1.0)

For some case-sensitive applications, such as SAP, there is a requirement that the content of the CD-ROM be translated into uppercase where, in fact, this content is recorded on the medium in lower or mixed case. An option has been added to the **mount** command in AIX 5L Version 5.1 to accommodate this. Note that this feature is for ISO-formatted CD-ROMs.

```
# mount -v'cdrfs' -p -r -o upcase /dev/cd0 /cdrom
# ls /cdrom
CDLABEL.ASC  DATA          LABEL.ASC      OS390          VERSION.EBC
CDLABEL.EBC  DOCU           LABEL.EBC      UNIX
CRCFILE.DAT  GROUP.ASC      NT             VERSION.ASC
```

Using the standard method of mounting a CD-ROM is still supported and the content remains in lowercase.

```
# mount -v'cdrfs' -p -r /dev/cd0 /cdrom
# ls /cdrom
cdlabel.asc  data          label.asc      os390          version.ebc
cdlabel.ebc  docu          label.ebc      unix
crcfile.dat  group.asc     nt             version.asc
```

The **nocase** option of the **mount** command, at the time of writing, is still under development and will probably be released at a later date. This option will preserve the case as it is on the CD-ROM.

```
# mount -v'cdrfs' -p -r -o nocase /dev/cd0 /cdrom
# ls /cdrom
CDLABEL.ASC  DATA          LABEL.ASC      OS390          VERSION.EBC
CDLABEL.EBC  DOCU           LABEL.EBC      UNIX
CRCFILE.DAT  GROUP.ASC      NT             VERSION.ASC
```

The upcase and nocase mount options are *not* available in the SMIT mount panels or other system administration tools.

4.10 Common HBA API support (5.2.0)

Upper-level software applications that operate or use Fibre Channel (FC) Host Bus Adapters (HBAs) require FC information (for example, WWN, attached LUNs) for Storage Area Network (SAN) management or other reasons. The FC information is not available from HBAs in a consistent manner across operating systems, vendors, and platforms, and in some cases not at all. Implementations to obtain such information are HBA vendor specific, for example, specific drivers or OS-specific calls have to be utilized to get to this information. This results in long qualification times, difficult integration across platforms, and inconsistency between HBA vendors, making implementation of SAN applications tedious to develop for upper-level software applications.

The Common HBA API, which is an industry standard programming interface for accessing management information in FC HBAs, provides a consistent low-level standard interface that can be implemented across vendors. Developed through the Storage Networking Industry Association (SNIA), the HBA API has been overwhelmingly adopted by SAN vendors to help, manage, monitor, and deploy storage area networks in an interoperable way. With AIX 5L Version 5.1 ML 5100-03 and AIX 5L Version 5.2, support for the Common HBA API Version 1.92 has been added with the exception of the HBA_GetEventBuffer().

The Common HBA API is implemented as a set of C programming language library functions, which allow access to low level, FC HBA information, and the OS mappings.

Reliability, availability, and serviceability

In this chapter, descriptions of the enhancements for AIX 5L can be found on the following topics:

- ▶ Error logs
- ▶ Trace facilities
- ▶ Dump facilities
- ▶ System hang detection
- ▶ PCI fault isolation
- ▶ Debuggers
- ▶ Tools to assist you in gathering system information for problem determination

5.1 Error log enhancements

AIX 5L provides three enhancements in the area of error logging. First, you can specify a time threshold that treats identical errors arriving closer than this threshold as duplicates and count them only once. Second, with the **errpt** command, you can now request an intermediate format that removes seldom needed data from the detailed error report format. A third enhancement, the diagnostic tool, will now put additional information into the error log entry.

5.1.1 Elimination of duplicate errors

The **errdemon** command was enhanced in AIX 5L to support four additional flags. The flags -D and -d specify if duplicate error log entries are to be removed or not. The default is the -D flag, which instructs the command to remove the duplicates. With the -t and -m flags you can control what is considered a duplicate error log entry. A value in the range 1 to $2^{31} - 1$ specifies the time in milliseconds within which an error identical to the previous one is considered a duplicate. The default value for this flag is 100 or 0.1 seconds. The -m flag sets a count, after which the next error is no longer considered a duplicate of the previous one. The range for this value is 1 to $2^{31} - 1$ with a default of 1000.

The following command increases the time threshold to one second and the number of duplicates after which the same error would again be counted as a new one to 100000:

```
# /usr/lib/errdemon -m 100000 -t 1000
```

The **errpt** command also has a new -D flag, which consolidates duplicate errors. In conjunction with the -a flag, only the number of duplicate errors and the timestamps for the first and last occurrence are reported. This is complemented by a new -P flag, which displays only the duplicate errors logged by the new mechanisms of errdemon mentioned previously.

5.1.2 The errpt command enhancements

In addition to the two new flags (-D and -P) mentioned in the previous section, **errpt** now supports an intermediate output format using the -A flag, in addition to the summary and the details already provided. Only the values for LABEL, Date/Time, Type, Resource Name, Description, and Detail Data are displayed.

The following lines show the output of the **errpt** command for one specific error using the summary, intermediate, and detailed options, respectively:

```
# errpt -j 9DBCDFDEE
IDENTIFIER TIMESTAMP T C RESOURCE_NAME DESCRIPTION
9DBCDFDEE 0919101600 T 0 errdemon ERROR LOGGING TURNED ON
# errpt -A -j 9DBCDFDEE
```

```
-----
LABEL:          ERRLOG_ON
Date/Time:      Tue Sep 19 10:16:41 CDT
Type:          TEMP
Resource Name:  errdemon
Description
ERROR LOGGING TURNED ON
# errpt -a -j 9DBCDFDEE
```

```
-----
LABEL:          ERRLOG_ON
IDENTIFIER:     9DBCDFDEE

Date/Time:      Tue Sep 19 10:16:41 CDT
Sequence Number: 1
Machine Id:     000BC6FD4C00
Node Id:        localhost
Class:          0
Type:           TEMP
Resource Name:  errdemon
```

```
Description
ERROR LOGGING TURNED ON
```

```
Probable Causes
ERRDEMON STARTED AUTOMATICALLY
```

```
User Causes
/USR/LIB/ERRDEMON COMMAND
```

```
Recommended Actions
NONE
```

5.1.3 Link between error log and diagnostics

When the diagnostic tool runs, it automatically tries to diagnose hardware errors it finds in the error log. Starting with AIX 5L, the information generated by the **diag** command is put back into the error log entry so that it is easy to make the connection between the error event and, for example, the FRU number required to repair failing hardware.

The following lines show an example of this process; first the header of the error log entry is shown, and then the information added by the diagnostic tool:

LABEL: EPOW_SUS_CHRP
IDENTIFIER: BE0A03E5

Date/Time: Wed Sep 20 13:47:27 CDT
Sequence Number: 14
Machine Id: 000BC6DD4C00
Node Id: server3
Class: H
Type: PERM
Resource Name: sysplanar0
Resource Class: planar
Resource Type: sysplanar_rspc
Location: 00-00
...
Diagnostic Analysis
Diagnostic Log sequence number: 8
Resource tested: sysplanar0
Resource Description: System Planar
Location: P1
SRN: 651-812
Description: System shutdown due to: 1) Loss of AC power, 2)
Power button was pushed without proper
system shutdown, 3) Power supply failure.

5.1.4 Error log enhancements (5.2.0)

AIX 5L Version 5.2 provides the following enhancements in the area of error logging.

- ▶ You can specify a time threshold that treats identical errors arriving closer than this threshold as duplicates and count them only once.
- ▶ With the **errpt** command, you can now request an intermediate format that removes seldom needed data from the detailed error report format.
- ▶ A new enhancement, the diagnostic tool, will now put additional information into the error log entry.
- ▶ A new kernel service, **errresume**, checks whether the error logging subsystem is active and was stopped using **errsave**.

The **errresume** service

This API allows other kernel code to continue error logging after having called **errsave** (which ends error logging).

Consider the example of a power failure. Basically a power failure results in the system going over to battery backup (if one is provided). At this time, AIX power monitoring interface kernel code calls `errsave` to log the serious nature of a power failure problem. This results in the error to be retained in the NVRAM (because of no more error logging, this NVRAM entry will not be overwritten) for after boot access. But in this situation if the power returns before the machine has completely powered off, the system returns to its normal operation. However, no more error logging is possible since `errsave` was called earlier.

The `errresume` service ensures that AIX can return back to normal error logging even after calling `errsave` for situations described previously. To do this, `errresume` checks whether the error logging subsystem is active and was stopped using `errsave`. If so, it reverts back the flags and performs the signalling necessary to wake up the `errdaemon` read thread.

5.2 Trace facility (5.1.0)

AIX 5L Version 5.1 introduces several new features for the trace facility. These include a new command, `trcevggrp`, and additional flags for the `trace` and `trcrpt` commands.

5.2.1 The trace command enhancements

The `trace` command has been enhanced in AIX 5L Version 5.1 with the addition of a new flag and enhancement to other flags.

The -f flag enhancement

In single mode, the collection of trace events stops when the in-memory trace buffer fills up. The maximum in-memory buffer has been increased to extend the trace.

The `-f` flag has been modified to allow a maximum trace buffer size of 268435184×2 or 536870368 bytes. The `maxbuffer` size for other options is unchanged.

The `-f` option actually uses two buffers, which behave as a single buffer. The two buffers are now used for the single-buffer trace. Thus, the term single-buffer refers to the function. In order to keep the function the same as before, I/O is held until all the tracing has been done. If I/O is started from buffer A while tracing to B, then the tracing in buffer B would reflect the I/O for buffer A. This would represent a function change from the previous action of trace `-f`.

The -T Size flag overrides the default trace buffer size of 128 KB with the value stated. You must be root to request more than 1 MB of buffer space. The maximum possible size is 268435184 bytes, unless -f is used, in which case it is 536870368 bytes. In the circular and the alternate modes, the trace buffer size must be one-half or less the size of the trace log file. In the single mode, the trace log file must be at least the size of the buffer. See the -L flag for information on controlling the trace log file size. Also note that trace buffers use pinned memory, in other words, they are not pageable. Therefore, the larger the trace buffers, the less physical memory is available to applications. Unless the -b or -B flags are specified, the system attempts to allocate the buffer space from the kernel heap. If this request cannot be satisfied, the system then attempts to allocate the buffers as separate segments.

The -J and -K flag enhancement

The **trace** command has been enhanced to specify the event groups to be included (-J) or excluded (-K). Event groups are described in 5.2.3, “Trace event groups” on page 269. The -J and -K flags work like -j and -k, except with event groups instead of individual hook IDs. All four flags (-j, -J, -k, and -K) may be specified. The -J has been available in previous versions of AIX, but not universally documented.

SMIT panels have also been updated, with the addition of event groups to EXCLUDE from trace, as shown in Figure 5-1.

START Trace

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

EVENT GROUPS to trace

ADDITIONAL event IDs to trace

Event Groups to EXCLUDE from trace

Event IDs to EXCLUDE from trace

Trace MODE

STOP when log file full?

LOG FILE

SAVE PREVIOUS log file?

Omit PS/NM/LOCK HEADER to log file?

Omit DATE-SYSTEM HEADER to log file?

Run in INTERACTIVE mode?

Trace BUFFER SIZE in bytes

LOG FILE SIZE in bytes

Buffer Allocation

[Entry Fields]

[]

[]

[]

[]

[alternate]

[no]

[/var/adm/ras/trcfile]

[no]

[yes]

[no]

[no]

[131072]

[1310720]

[automatic]

F1=Help

F2=Refresh

F3=Cancel

F4=List

F5=Reset

F6=Command

F7=Edit

F8=Image

F9=Shell

F10=Exit

Enter=Do

Figure 5-1 SMIT panel for START Trace

5.2.2 The trcrpt command enhancements

Previous versions of **trcrpt** only allow the **-d** and **-k** flags to specify a list of hooks to include and exclude. **trcrpt** has been enhanced to allow hook groups (5.2.3, “Trace event groups” on page 269) to be included/excluded; the **-D** flag includes and the **-K** flag excludes.

The new -D and -K flags

The **-D** flag limits the report to hook IDs in the event groups list, plus any hook IDs specified with the **-d** flag.

The **-K** flag excludes from the report hook IDs in the event groups list, plus any hook IDs specified with the **-k** flag.

The trace report SMIT screen has also been updated, with the additional line Event Groups to INCLUDE in report (**-D** flag) and Event Groups to EXCLUDE from report (**-K** flag), as shown in Figure 5-2.

Generate a Trace Report

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

	[Entry Fields]	
Show exec PATHNAMES for each event?	[yes]	+
Show PROCESS IDs for each event?	[no]	+
Show THREAD IDs for each event?	[no]	+
Show CURRENT SYSTEM CALL for each event?	[yes]	+
Time CALCULATIONS for report	[elapsed only]	+
Event Groups to INCLUDE in report	[]	+
IDs of events to INCLUDE in report	[]	+X
Event Groups to EXCLUDE from report	[]	+
ID's of events to EXCLUDE from report	[]	+X
STARTING time	[]	
ENDING time	[]	
LOG FILE to create report from	[/var/adm/ras/trcfile]	
FILE NAME for trace report (default is stdout)	[]	

F1=Help
F5=Reset
F9=Shell

F2=Refresh
F6=Command
F10=Exit

F3=Cancel
F7=Edit
Enter=Do

F4=List
F8=Image

Figure 5-2 SMIT panel for Trace Report

5.2.3 Trace event groups

Trace event groups combine multiple trace hook IDs into a trace group; this allows hooks to be turned on or off at once when starting a trace.

The **trcevggrp** command provides a facility for you maintain the trace event groups. The Event groups are hook IDs grouped together. You must be in the system group to add, delete, or change trace event groups. You may not modify or delete event groups whose type is *reserved*. Figure 5-3 shows the SMIT panel for Manage Event Groups (fast path smit grpmenu).

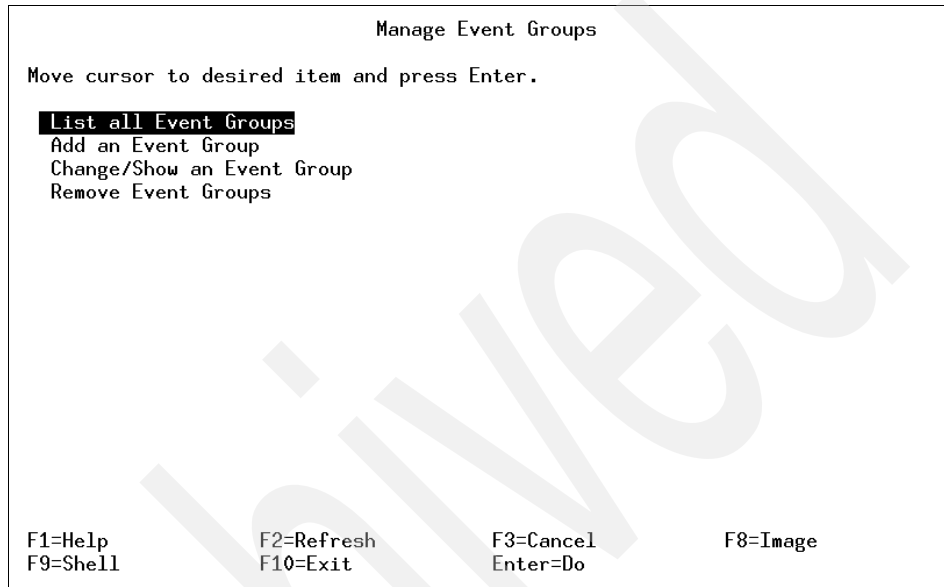


Figure 5-3 SMIT panel for Manage Event Groups

The following are descriptions of the fields for the Manage Event Groups:

List all Event Groups

This will use **trcengrp -l** to get the list of event groups.

Add an Event Group

This allows you to add a new event group based on an existing event group or create your own event group. It uses **trcevggrp -a** to add the event group. The Add function (as shown in Figure 5-4 on page 271 and Figure 5-5 on page 272) allows you to add a new event group from a template. Figure 5-4 on page 271 shows the first screen for adding an Event Group.

Change/Show an Event Group

This allows you to retrieve and modify an event group. The **trcevggrp -l** is used to retrieve the information. **trcevggrp -u** is used to update the existing record.

Remove Event Group	This allows you to remove user-created event groups. trcevgrp -r is used to remove the event groups.
The following descriptions are of additional sub-panels of those selected by choosing the previous options:	
Event Group ID (optional)	This allows the user to select a template from a list of existing event groups.
Event Group ID	This is the name of the new event group.
Event Group Description	A brief description of the new event group.
Event Group Hook IDs	The hook IDs you wish to trace. The hook IDs should be separated with a comma and no spaces.

Note: Groups that are *reserved* may not be modified or removed; for example, **tidhk - Hooks needed to display thread name (reserved)**.

Select a template Event Group

Type or select a value for the entry field.
Press Enter AFTER making all desired changes.

Event Group ID (optional)

If none, no template group is used.

[Entry Fields]

+

F1=Help

F2=Refresh

F3=Cancel

F4=List

F5=Reset

F6=Command

F7=Edit

F8=Image

F9=Shell

F10=Exit

Enter=Do

Figure 5-4 SMIT panel for creating a new event group

Add an Event Group

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

- * Event Group ID
- * Event Group Description
- * Event Group Hook IDs

[Entry Fields]

+

F1=Help
F5=Reset
F9=Shell

F2=Refresh
F6=Command
F10=Exit

F3=Cancel
F7=Edit
Enter=Do

F4=List
F8=Image

Figure 5-5 SMIT panel for creating a new event group

To get a listing of all event groups, enter the following command:

```
# trcevgrp -l
```

To add a new group, enter the command:

```
# trcevgrp -a -d "description of this group" -h "500 501 502" mygrp
```

This will add the group named *mygrp* and give it the description description of this group, and define it to have hooks of 500, 501, and 502.

To add another hook to the group above, enter the following command:

```
# trcevgrp -u -d "description of this group" -h "500 501 502 503" mygrp
```

Note that it is necessary to specify all the hook IDs.

To remove a group, enter:

```
# trcevgrp -r test
```

5.3 Trace Report GUI (5.2.0)

The Trace Report GUI (graphical user interface) viewer is a graphical tool to analyze raw trace data. It is not meant to replace **trcrpt** but offers an easy-to-use alternative. It reduces the complexity of managing traces because it

avoids the need of having to save large files of filtered output and having to maintain complex scripts.

Trace Report GUI is provided as a sample and therefore should be used as is. It is included in the `bos.sysmgt.trcgui_samp` fileset.

To run Trace Report GUI, complete the following steps:

1. Install the `bos.sysmgt.trcgui_samp` fileset.
2. Include `/usr/samples/trcgui` to your path by issuing the following command:

```
export PATH=$PATH:/usr/samples/trcgui
```
3. Run the **tgvt -client** command.

The main window will appear once these steps have been completed. To open a trace file on the local host, click **File -> Native Open**. In the file open dialog specify the trace and format file. Defaults are `/var/adm/ras/trcfile` and `/etc/trcfmt`, respectively.

Alternatively, a remote file may be opened by using the **File -> Remote Open** menu. On the server where you want to open the file, the Trace Report GUI server must be running. It is started using the **tgvt -server** command.

The initial trace view will open next (see Figure 5-6 on page 274) and the first entries of subtrace 0 are displayed. The trace is divided into subtraces and only the first entries are shown for performance reasons. It is possible to view the first entries and work with them, while the others are still being loaded in the background. This is especially important for large traces.

To move to a different subtrace, click the according entry in the left frame of the view and press Enter on the keyboard. To go back to subtrace 0 select **Action -> First**.

To view further entries in the subtrace click the **Page Down** button. Do not select **Action -> Next**; this is for debugging purposes only and does not work as you might assume.

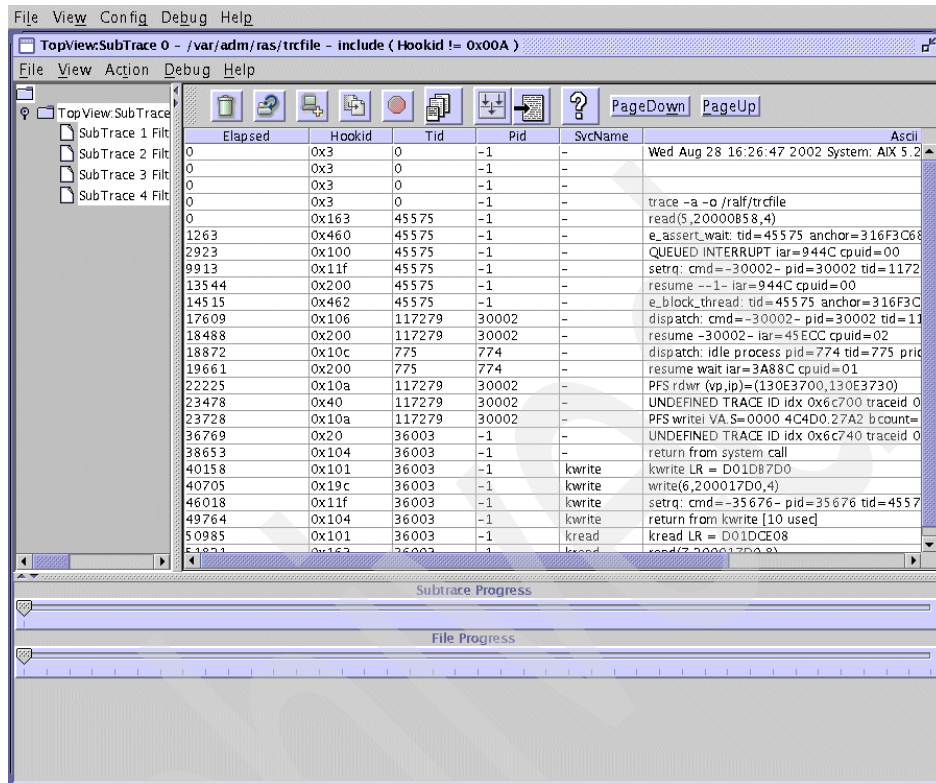


Figure 5-6 tgv view window

The most important feature is to use filters on the trace file. To open the filter dialog as shown in Figure 5-7 on page 275 select **Action -> Edit Filter**. Check **Include Filtering** and **Use this filter** before you specify the criteria that must match the entries you want to see in your view. In the example only entries with the hookid=0x104 will be shown after pressing the **OK** button. You can specify several parameters for one filter and a maximum of four filters at a time.

To find an entry quickly that has been visited earlier, bookmarks can be used. To add a bookmark, right-click the entry and add a description for the bookmark. You can jump to the bookmark from anywhere in the trace by selecting the **Action -> Seek to entry** menu item and selecting the previously added description.

Figure 5-7 tgv filter window

5.4 Loader trace hooks (5.2.0)

Trace hooks have been added to the loader that allows developers to gather data about the activity of the loader using the **trace** command. Tracing of loader activity is available based on the trace hooks activated.

The trace hooks are placed in critical entry/exit sections of the loader as well as error handling routines. The trace hook IDs listed in Table 5-1 are stored in the `/usr/include/sys/trchkid.h` file.

Table 5-1 Loader trace hooks

Trace hook ID	Trace hook	Description
5A0	HKWD_LDR	This event is recorded by the system loader's module load/unload related routines.

Trace hook ID	Trace hook	Description
5A1	HKWD_LDR_KMOD	This event is recorded by the system loader's kernel extensions load/unload related routines.
5A2	HKWD_LDR_PROC	This event is recorded by the system loader's ld_execload routine.
5A3	HKWD_LDR_ERR	This event is recorded by the system loader's routines whenever errors occur.
5A4	HKWD_LDR_CHKPT	This event is recorded by the system loader's check point restart related routines.

5.5 System dump enhancements

AIX 5L provides the following enhancements in the area of system dumps:

- ▶ A new command, **dumpcheck**, that checks to see if the dump device and the copy directory for the dump are large enough to actually accept a system dump
- ▶ The creation of a core file for a process without terminating the process
- ▶ Minor enhancements to the **snaps** command
- ▶ Dedicated dump device

5.5.1 The dumpcheck command

The new **dumpcheck** command has the following syntax:

```
/usr/lib/ras/dumpcheck [ [ -l ] [ -p ] [ -t Time ] [ -P ] ] | [ -r ]
```

By default, **dumpcheck** is started by a crontab entry each afternoon at 3:00 p.m. local time. The output of the command will be logged in the system error log. With the **-p** flag, you can request a **dumpcheck** at any time and the result is printed to stdout. The output would look similar to the following example:

```
# /usr/lib/ras/dumpcheck -p
There is not enough free space in the file system containing the copy directory
to accommodate the dump.
File system name          /var/adm/ras
Current free space in kb   14360
Current estimated dump size in kb      25600
```

The **-l** flag logs the command output into the system error log and is the default parameter if no other parameter is specified. With the **-t** flag, you can specify (with a time value in crontab format enclosed in single or double quotation marks) at what time this check will be run by the cron facility. The **-P** flag updates the crontab entry to reflect whatever parameters are specified with it. The cron facility mails the standard output of a command to the user who runs this command (in this case, root). If you use the **-p** flag in the crontab entry, root will be sent a mail with the standard output of the **dumpcheck** command.

Note: Currently, the command output redirection (**> /dev/null 2>&1**) will not automatically be removed, which prevents the cron facility from sending the mail. You have to remove this redirection manually.

The **-r** flag removes the corresponding crontab entry. This flag cannot be used together with any other flag.

5.5.2 The **coredump()** system call

An application can now create a core file by using the new **coredump()** system call. This call takes, as a single parameter, a pointer to a **coredumpinfo** structure that sets the path and file name for the core file to be generated.

To use **coredump()**, you must compile your source with the **-bM:UR** options. The **-b** flag is for **1d**, **M** is to specify a module type, and **UR** saves the user registers on system calls.

5.5.3 The **snap** command enhancements

The **snap** command in AIX 5L uses the **pax** command instead of the **tar** command to create the **snap** file. This is necessary to manage the ever-increasing sizes of the dump files, as file sizes larger than 2 GB are only supported by the **pax** command. The **snap** command also links the dump file to the directory structure it creates instead of copying it into the structure, which wastes disk space. The data needed most for analyzing the situation (that is, what caused the dump) is written out first, so that it has a good chance to be part of the archive file created by **snap** even if the dump is only partially successful. For other enhancements to **pax**, see 5.17, “The **pax** command enhancements” on page 313.

5.5.4 Dedicated dump device (5.1.0)

In AIX Version 4.3.3 and earlier, the paging space is used as the default dump device created at installation time. AIX 5L Version 5.1 servers with a real memory

size larger than 4 GB will, at installation time, have a dedicated dump device created. This dump device is automatically created and no user intervention is required. The default name of the dump device is lg_dumplv. This name and the size of the dump device can be changed by using the bosinst.data file on a diskette at boot time. A new stanza has been added to the bosinst.data file called large_dumplv, which contains two fields. The first field is DUMPDEVICE, which is the name of the dump device and has a maximum size of 15 characters. In the case of an alternate installation disk, the DUMPDEVICE field is limited to 11 characters. The second field is SIZE_GB, which denotes the size of the dump device in GB. SIZE_GB is a maximum of three characters long and it must be a whole number. The stanza will appear similar to that shown in the following example.

```
large_dump:
    DUMPDEVICE = /dev/lg_dumplv
    SIZE_GB = 1
```

Once the operating system installation has completed, the following command can be used to display the dump device:

```
# sysdumpdev -l
primary          /dev/lg_dumplv
secondary        /dev/sysdumpnull
copy directory   /var/adm/ras
forced copy flag TRUE
always allow dump FALSE
dump compression OFF
```

Information pertaining to the dump device can be displayed, as shown in the following examples:

```
# lspv -l hdisk0
hdisk0:
LV NAME      LPPs DISTRIBUTION MOUNT POINT
hd5          1 1 01..00..00..00..00N/A
hd6          4 4 00..52..00..00..00N/A
lg_dumplv    64 64 00..64..00..00..00N/A
hd8          1 1 00..00..01..00..00N/A
hd4          1 1 00..00..01..00..00/
hd2          22 22 00..01..22..00..00/usr
hd9var1      1 00..00..01..00..00/var
hd3          2 2 00..00..02..00..00/tmp
hd1          1 1 00..00..01..00..00/home
hd10opt1     1 00..00..01..00..00/opt

# lsvg -l rootvg
rootvg:
LV NAME      TYPE LPPs PPs PVs LV STATE MOUNT POINT
hd5boot      1 1 1 closed/syncd N/A
```

```
hd6      paging52521 open/syncdN/A
hd8      jfslog11 1 open/syncdN/A
hd4      jfs1 1 1 open/syncd/
hd2      jfs37 37 1 open/syncd/usr
hd9var   jfs1 1 1 open/syncd/var
hd3      jfs53 53 1 open/sync/tmp
hd1      jfs1 1 1 open/syncd/home
hd10opt  jfs83 83 1 open/syncd/opt
lg_dump1v sysdump64641 open/syncdN/A
```

The dedicated dump device size is determined by the amount of memory. In Table 5-2, the memory size to dump device size ratio is shown.

Table 5-2 System memory to dump device size ratios

System memory size	Dump device size
4 GB to, but not including, 12 GB	1 GB
12 GB to, but not including, 24 GB	2 GB
24 GB to, but not including, 48 GB	3 GB
48 GB and up	4 GB

If there is insufficient disk space for the system to create a dump device at installation time, then the default action is using the paging space /dev/hd6 as the dump device occurs. Systems with less than 4 GB of real memory also use the paging space as the default dump device.

5.5.5 System dump facility enhancements (5.2.0)

The system dump facility has been enhanced to allow greater functionality in component dump routines. There is also support for unlimited dump size to allow a dump routine to return unknown amounts of dump data.

Prior to Version 5.2, individual components would use the dmp_add and dmp_del services to register and unregister data areas to be included in the system dump. The components were each required to allocate and pin their own buffer space during initialization. The master dump table only has a pointer to the component's dump routine and has no visibility to the actual size of the component's dump data. This prevents the system from obtaining an accurate dump size estimate. When a system dump is started and the component's dump routine is called, the component is required to return all the dump data in one array. The maximum number of cdt_entries for the 64-bit dump is approximately 21840. This is problematic when the system has to dump data for 30000 processes.

Version 5.2 introduces a new kernel service, `dmp_ctl`, to allow the component developer to avoid the previous restrictions. The `dump_add` and `dump_del` are still supported for compatibility reasons. With the `dmp_ctl` service, the individual components no longer need to allocate and pin their own buffer area. When a component calls the `dmp_ctl` service to register its dump routine with the dump facility, it will give the amount of buffer space required for its dump data. The dump facility will then allocate the required memory in the global dump buffer. With the new dump facility, the component's dump routine can be sent different operations beyond the normal dump start and dump done. One of the defined operations that component owners may implement is to return a dump size estimate. The component's dump routine must ignore all operations it does not support, which allows for future enhancements without breaking existing components.

The new dump facility also supports an unlimited dump table, where the component dump routine can return the dump data in multiple calls. This is useful when you want to dump an unknown number of data areas without preallocating the maximum array of `cdt_entry` elements as is required by the classic dump table. The dump facility will continue to call the component's dump routine until it returns a null `cdt_u` pointer.

5.6 The `adump` command enhancement (5.2.0)

Automated dump analysis tool `adump` has been enhanced to enable users to run custom scripts from the interactive `adump` prompt. Users' PERL scripts can be invoked using the new `usemaster` command, and a set of default problem conditions could be checked out.

The `adump` command allows you to modify and run predefined objects and macros to run analysis scripts. You are able to add new objects and macros or enhance the predefined objects with new methods. The primary goal of the `adump` command is to build up a script database to help analyze dumps. Adding or modifying existing objects and methods in the `adump` utility requires advanced knowledge of the PERL language.

The `adump` command is currently intended for use by IBM service personnel for diagnosing customer problems.

5.7 System hang detection

The system hang detection mechanism in AIX has been enhanced to detect lost I/O conditions. System hang detection is based on a daemon (`shdaemon`)

monitoring the system at regular intervals. Also the **shconf** command provides control and configuration support for the system hang condition.

In a multi-process environment such as an AIX system, there is a remote possibility of application processes clashing with each other for resources and locks resulting in a application/system hang condition. The priority of an application could also change due to a variety of reasons resulting in a situation where the lower priority processes are not getting any time to operate. In this situation, it is difficult to distinguish a system that really hangs (it is not doing any meaningful work anymore) from a system that is so busy that none of the lower priority tasks, such as user processes, have a chance to run. This condition, also referred to as priority hang condition, results in the system not being utilized for doing any useful work. It becomes necessary to break out of this condition or reboot the system.

Also, in certain situations it is possible that the various layers in the I/O path are made to wait infinitely on I/O completion. The I/O may not be completed due to an error in the I/O path and resulting in an I/O hang condition. It is important to break out of these conditions.

System hang detection provides for the above-mentioned priority and lost I/O hang detection and recovery when possible.

The system hang detection feature uses a shdaemon entry in the /etc/inittab file with an action field that is set to off by default. Using the **shconf** command or SMIT (fast path shd), you can enable this daemon and configure the actions it takes when certain conditions are met. The following flags are allowed with the **shconf** command:

```
shconf [ -d ] [ -R | -D [ -O ] | -E [ -O ] | [ [ -a Attribute ] ... ] -l  
name [ -H ]
```

The name may be either prio or lio.

prio

Means that the system hang daemon will always compare the priorities of all running processes to a set threshold, and will take one of the five supported actions, each of a different priority, when the entire system fails to run a process below the specified priority any time in the time-out period.

lio

Refers to the lost I/O detection mechanism, which provides user options to display a console warning message or reboot the system on a lost I/O detection.

5.7.1 Priority management (5.2.0)

The first existing detection name is prio, which means that the system hang daemon will always compare the priorities of all running processes to a set threshold, and will take one of the five supported actions, each of a different priority, when the entire system fails to run a process below the specified priority any time in the time-out period.

The -d flag displays the current status of the shdaemon. The -R flag restores the system default values. With the -D and -E flags, you can display either the default or the effective values of the configuration parameters. The -H flag adds an optional header to this output. You can request a more concise output by using the -O flag together with either the -D or -E flags (in this case, the -H flag is not allowed). It displays two lines: One with the colon-separated names, and one with the colon-separated values of the configuration parameters. With the -a flag and a name/value pair, you can change the parameter values.

After a new default system installation that has effective values that are identical to the default values occurs, the output of the **shconf** command appears as follows:

```
# shconf -d
sh_pp=disable
# shconf -E -l prio -H
attribute  value      description

sh_pp      disable    Enable Process Priority Problem
pp_errlog  disable    Log Error in the Error Logging
pp_eto     2          Detection Time-out
pp_eprio   60         Process Priority
pp_warning disable    Display a warning message on a console
pp_wto     2          Detection Time-out
pp_wprio   60         Process Priority
pp_wterm   /dev/console Terminal Device
pp_login   enable     Launch a recovering login on a console
pp_lto     2          Detection Time-out
pp_lprio   56         Process Priority
pp_lterm   /dev/tty0  Terminal Device
pp_cmd     disable    Launch a command
pp_cto     2          Detection Time-out
pp_cprio   60         Process Priority
pp_cpath   /          Script
pp_reboot  disable    Automatically REBOOT system
pp_rto     5          Detection Time-out
pp_rprio   39         Process Priority
```


The `ss_pp` parameter determines the availability of the system hang detection feature. Enabling it with the default configuration may generate the following error:

```
# shconf -l prio -a sh_pp=enable
shconf:Enable to configure the emergency login.
shconf: Configuration method error.
```

You have to disable the `pp_login` action, enable the system hang detection, and then configure the desired actions. The output of these commands appears as follows:

```
# shconf -l prio -a sh_pp=disable
shconf: Priority Problem Conf has changed.
# shconf -l prio -a pp_login=disable
shconf: Priority Problem Conf has changed.
# shconf -l prio -a sh_pp=enable
shconf: Priority Problem Conf has changed.
shconf: WARNING: Priority Problem Detection is enabled with all actions
disabled.
```

The last command shown in the previous output toggles the `action` field of the `shdaemon` entry in `/etc/inittab` to `respawn` and starts the `/usr/sbin/shdaemon` program. After enabling (for example, the `errlog` action), the priority of the `shdaemon` process is 0, the highest possible value. This is shown in the following example:

```
# ps lwx 19580
  F S UID  PID  PPID  C PRI NI ADDR  SZ  RSS  WCHAN  TTY  TIME CMD
240001 A   0 19580    1   0  60 20 fa5e 192  236  EVENT   -   0:00
/usr/sbin/shdaemon
# shconf -l prio -a pp_errlog=enable
shconf: Priority Problem Conf has changed.
# ps lwx 19584
  F S UID  PID  PPID  C PRI NI ADDR  SZ  RSS  WCHAN  TTY  TIME CMD
240001 A   0 19584    1   0   0 20 fa5e 33000 33044  EVENT   -   0:00
/usr/sbin/shdaemon
```

This action makes sure that the `shdaemon` is always scheduled and can evaluate the current machine status and take the configured actions when appropriate. The available actions include the following:

- errlog** Generates an entry in the error log.
- warning** Displays a warning message on a console; the default is `/dev/console`.
- login** Enables a login shell with priority 0 on a serial terminal; the default is `/dev/tty0`.
- cmd** Starts a command with priority 0.

reboot Automatically reboots the machine.

5.7.2 Lost I/O management (5.2.0)

The second existing detection name is lio. In this case the system hang daemon checks every 10 minutes (this is the default) if a synchronous I/O does not terminate. The daemon only check synchronous I/O for logical volumes.

If a lost I/O is detected, the shdaemon daemon will systematically log an error in the errorlog file. It is also able to send a message to a console or reboot the system if those options have been chosen by the system administrator.

The SMIT panel (Figure 5-8) shows that lio is enabled and an error message will be sent to the console in case of lost I/O detection, but the system will not reboot.

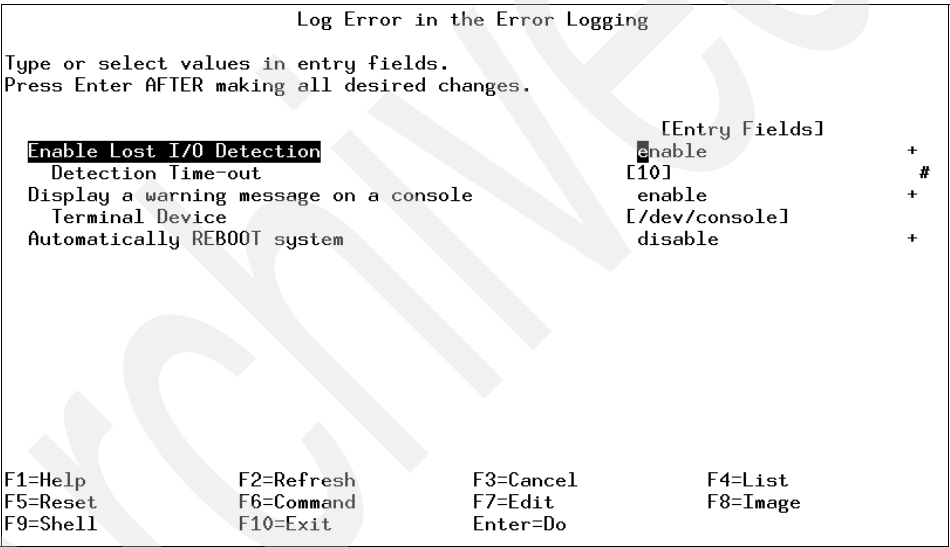


Figure 5-8 SMIT panel for lost I/O management

If only the lost I/O management is enabled, and the priority disabled, the shdaemon does not run with a priority 0.

5.8 Fast device configuration enhancement

AIX 4.3.3 introduced a new device configuration methodology in order to reduce the time needed to detect and configure all the devices attached to the system. The **cfgmgr** command was changed so that it can run device configuration

methods in parallel rather than sequentially (one at a time). This function does not support every device on every bus type.

AIX 5L adds support for parallel configuration of Fiber Channel (FC) adapters and devices, and an expanded list of devices and bus types:

- ▶ Fiber Channel adapters and devices
- ▶ PCI buses on CHRP systems
- ▶ PCI SCSI adapters on CHRP and PReP systems
- ▶ PCI async adapters and their concentrators on CHRP and PReP systems
- ▶ SCSI disks on any POWER platform
- ▶ TTYs on any POWER platform

5.9 Boot LED displays (5.2.0)

AIX 5L Version 5.2 provides enhanced support for the front panel display. The boot scripts now display additional information on the second line of the front display panel to give more information of specific LED values. During bootup, some of the LEDs can be displayed for an extended period of time. An example of this would be the 551 code, which is the **varyonvg rootvg** command. The second line for specific LEDs shows whether the phase is complete or if there is an error. The changes to the boot LEDs for Version 5.2 are shown in Table 5-3.

Table 5-3 Second line of front panel display information

LED display number	Second line display message	Description
510	DEV CONF START <i>phase</i> # STRLOAD	Starting device configuration. In case of tape it does strload before calling cfgmgr .
511	DEV CONF COMP <i>phase</i> #	Device configuration complete.
512	RESTORE FILES	Restoring device configuration files from media.
512	CP FILESOBJREPOS	Copy diagnostic /etc/objrepos to files.
513	RESTORING FILES	Restoring files from diskette.
517	MOUNT /DEV/HD4 MOUNT CDRFS	Mounting client remote file systems during network boot; mounting cdrfs for CD-ROM boot.

LED display number	Second line display message	Description
518	MOUNT USR FAILED MOUNT VAR FAILED	Remote mount of /usr and /var file system during network boot did not complete successfully.
546	SAVEBASE FAILED	IPL cannot continue due to error in customized data base.
548	RESTBASE FAILED	Restbase failed.
549	SRVBOOT FAILED	Console could not be configured for the "Copy a System Dump Menu".
551	IPLVARYON RUN	IPL varyon is running.
552	IPLVARYON ERROR	IPL varyon failed.
553	BOOT 1 COMPLETE	Boot phase 1 is complete.
554	CANT READ BOOT	The boot device could not be opened or a read failed.
555	FSCK FAILED hd4	ODM error when trying to varyon the rootvg.
556	LVM RET ERROR	LVM subroutine error from the ipl_varyon.
557	MOUNT / FAILED	The root file system will not complete the fsck command or mount.
600	NETBOOT START CONFIG NETBOOT	Starting network boot portion of /sbin/rc.boot.
606	IFCONFIG RUNNING	Running /usr/sbin/ifconfig on logical network boot device.
607	IFCONFIG FAILED	/usr/sbin/ifconfig failed.
608	TFTP CLIENTFILES	Attempting to retrieve the client.info with tftp .
609	NIMINFO FAILED	The client.info file does not exist or it is zero length.
610	MOUNT /SPOT/USR	Attempting remote mount of NFS file system.
611	MOUNT FAIL /SPOT	Remote mount of the NFS file system failed.

LED display number	Second line display message	Description
612	CP RCONF FILE	Accessing remote files; unconfiguring network boot device.
613	ROUTE FAILED	Setting route table.
C00	RESTORE OVER	AIX install/maintenance loaded successfully.
C03	WRONG DISKETTE	The wrong diskette is in the diskette drive.
C06	UNKNOWN BOOT	The rc.boot configuration shell script is unable to determine the type of boot.
C07	NEXT DISKETTE	Insert the next diagnostic diskette.
C09	PROCESS DISKETTE	The diskette is reading or writing a diskette.

5.10 Improved PCI FRU isolation (5.2.0)

Version 5.2 introduces the concept of enhanced I/O error handling, a recovery strategy for I/O errors that occur on the PCI bus.

5.10.1 EEH overview

Version 5.2 further enables the enhanced I/O error handling (EEH) error recovery strategy for I/O operations on the PCI bus. EEH is made possible by the EADS chip, by allowing each PCI slot to have its own PCI bus. Each adapter can therefore be isolated in the case of an error. This enables error recovery to occur without affecting any of the other adapters on the system.

Without EEH, pSeries machines would checkstop in the event of a PCI bus error, either caused by the bus or a device on the bus. The EADS chip gives the functionality to freeze an adapter in the event of an I/O error and hence avoids the checkstop. An adapter reset is tried and is allowed to fail three time before the adapter is marked as dead.

EEH on AIX was initially introduced with Version 5.1. Subsequently the functionality of EEH has been enhanced as follows:

- ▶ Version 5.1

Introduced the ability to register and recover from EEH events and established the basic principles for detection and recovery of PCI I/O errors for single function adapters.

- ▶ Version 5.1 RML 5100-02

Built on the ability to register and recover from EEH events for single function adapters established in Version 5.1 for the detection and recovery of PCI I/O errors for multi-function adapters. This incorporated the need to synchronize device drivers by introducing new kernel services.

- ▶ Version 5.2

Introduced PCI FRU isolation, which is a RAS enhancement to unify and expand the AIX error logging of EEH events. Version 5.2 enables the device drivers to use a common EEH AIX error log template rather than writing device driver specific events to the AIX error as was the case in previous versions. AIX error log information now contains EADS-specific information for diagnosis.

5.10.2 Detailed description of EEH

PCI FRU isolation occurs at the adapter slot level, although it is possible to have hardware adapters that have more than one logical device defined to a physical adapter and hence PCI slot adapters can be one of the following types:

- ▶ Single function adapter

Single function adapters include any adapter that for each physical defines only one logical AIX-level device. Most common adapters are of this type, for example, the Type 9-P 10/100 Ethernet TX PCI Adapter (FC 2968).

- ▶ Multi-function adapter

Multi function adapters include an adapter that defines greater than one logical AIX-level device for each physical device. For example, the Type 9-Z 4-port 10/100 Base -TX Ethernet PCI Adapter (FC 4951). Although this adapter will use the same device driver for each logical interface, there will be more than one instance of the driver on the physical slot. For this reason when a slot is marked as frozen, the multiple device driver instances must all report error information and be reset.

- ▶ Adapters with one or more PCI bridge and controller

At the time of writing, there are no existing adapters of this type; however, the adapter would function as a single device under the current EEH function.

The following section provides more detail into how EEH functions on AIX:

- ▶ The device driver registers and enables the slot for EEH prior to the first I/O access.
- ▶ EEH error recovery resources are enabled on the slot for use in the advent of a freeze condition.
- ▶ The device drivers save the PCI configuration registers initiated by the firmware, which may be needed if the device is reset due to a freeze condition.
- ▶ The device driver monitors for freeze conditions in the following locations: Watchdog timer, interrupt handler, and strategy routine (although this last location may be covered by monitoring the watchdog timer).
- ▶ Once a freeze condition exists, EEH recovery begins. Recovery includes gathering and logging error and RAS information to the AIX error log.
- ▶ Once complete, the device driver activates the reset line of the PCI adapter and tests it before resuming normal operation.
- ▶ The adapter reset operation is tried three times before the adapter is marked as permanently unavailable.
- ▶ If the adapter is marked unavailable the device driver will not attempt to reuse the adapter until the next IPL or hot-plug event.

With multi-function adapters, the last device driver to issue a callback for the adapter will be treated as the master. The master device driver has the role of driving error recovery. This includes gathering and logging error and RAS information to the AIX error log. The kernel services will also enable the logging of callback arguments registered by sibling functions on the adapter to enhance problem determination. The device driver then activates the reset line of the PCI adapter, testing, and then resumes normal adapter operation.

5.10.3 EEH-supported adapters

Device Driver support for EEH and, hence, PCI FRU isolation, is limited to the devices operating on AIX 5L Version 5.2 listed in Table 5-4.

Table 5-4 EEH adapter support

Adapter description	Feature code	Support for EEH
PCI SCSI-2 Differential Fast/Wide	2409	Yes
3-port Ultra2 SCSI RAID	2494	Yes
4-port Ultra3 SCSI RAID	2498	Yes

Adapter description	Feature code	Support for EEH
HIPPI	2732	No
Keyboard/mouse attachment card	2737	Yes
FDDI	2741	No
ESCON control unit	2751	Yes
POWER GXT135P Graphics Accelerator	2848	Yes
4/16Mbs token ring	2920	Yes
8 port RS232/RS422 async adapter	2943	Yes
128 port RS232/RS422 async adapter	2944	Yes
622 Mbps PCI ATM	2946	Yes
4-port ARTIC960HX MP	2947	No
4-port ARTIC960HX T1/E1	2948	No
2 port SDLC X.25	2962	Yes
Turboways 155 PCI MMF ATM	2963	Yes
10/100 Ethernet	2968	Yes
10/100/1000 Ethernet Fibre	2969	Yes
10/100/1000 Ethernet UTP	2975	Yes
10Base2 Ethernet	2985	Yes
Turboways 155 PCI UTP ATM	2988	Yes
Quad 10/100 Ethernet	4951	No
PCI Cryptographic Coprocessor	4958	Yes
4/16 token ring	4959	Yes
IBM e-business Cryptographic Coprocessor	4960	Yes
Quad 10/100 Ethernet Universal	4961	Yes
PCI Dual Channel Ultra3 SCSI	6203	Yes
PCI SE Ultra SCSI	6206	Yes

Adapter description	Feature code	Support for EEH
PCI SCSI-2 SE Fast/Wide	6208	Yes
PCI SCSI-2 Differential Fast/Wide	6209	Yes
Advanced SerialRAID adapter	6225	Yes
Gigabit Fibre Channel	6227	Yes
2 Gigabit Fibre Channel	6228	Yes
Advanced SerialRAID adapter	6230	Yes
Advanced SerialRAID adapter	6232	Yes
Digital trunk adapter	6310	No
Digital trunk adapter	6311	No

5.10.4 AIX error logging

EEH events are logged in the AIX error log and are marked as either recovered or permanent. These are referred to as INFO or PERM, respectively. RAS information in the form of sense data is included in the AIX error log entry. For multi function adapters the device-specific data for non-master device drivers is also logged in the AIX error log.

Each EEH AIX error log entry will have both the platform-specific extended log debug data and the device driver-specific extended debug data in the sense data. The former is there for FRU isolation, while the latter can be used to isolate host software problems.

5.10.5 Error log entries

The following section contains an overview of the contents and format of the AIX error log sense data.

Detail Data

PROBLEM DATA

0444 2201 0000 xxxx 8E00 9340 hhmm ss00 yyyy mmdd 2000 bddf dddd vvvv rrss bddf

0444 - version 4, Warning, Fully Recovered, Extended Error Log Present

2201 - Initiator PCI IOA, Target PCI IOA, Type Retry

0000 xxxx - Length of Extended Error Log

This is the start of the error log

8E00 - Log Valid, Predictive Error (recoverable), New Log, Big

Endian

9340 - Power PC Format, Address not valid, I/O Error, Single Error

m,s,y,m,d - Date stamp

```

2000          - Other error
bbdf          - Bus#, Dev#, Func# of signalling
dddd          - Device ID
vvvv          - Vendor ID
rrss          - Revision ID, Slot Identifier
bbdf          - Bus#, Dev#, Func# of sending
dddd          - Device ID
vvvv          - Vendor ID
rrss          - Revision ID, Slot Identifier
00's          - bytes 30-39 are reserved
4942 4D00     - "IBM"
5531 2E31 332D 5031 2D48 3130          - location code "U1.13-P1-I10"
000C 4444 0406 0089 1111 1111      Speedwagon CSR
000C 4444 0406 0040 2222 2222      Speedwagon PLSSR

```

Then there are 12 EADS register reads, all of which are 12 bytes long:

```

Detail Data
PROBLEM DATA

```

```

0444 2201 0000 xxxx 8E00 9340 hhmm ss00 yyyy mmdd 2000 bbdf dddd vvvv rrss bbdf
dddd vvvv rrss 0000 0000 0000 0000 0000 4942 4D00 5531 2E31 332D 5031 2D48 3130
0000 000C 4444 0406 0089 1111 1111 000C 4444 0406 0040 2222 2222 000C 4444 1801
0001 1111 1111 000C 4444 1801 0002 2222 2222 000C 4444 1801 0003 3333 3333 000C
4444 1801 0004 4444 4444 000C 4444 1801 0005 5555 5555 000C 4444 1801 0006 6666
6666 000C 4444 1801 0007 7777 7777 000C 4444 1801 0008 8888 8888 000C 4444 1801
0009 9999 9999 000C 4444 1801 000A AAAA AAAA 000C 4444 1801 000B BBBB BBBB 000C
4444 1801 000C CCCC CCCC dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd
dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd
dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd
dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd
dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd
dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd
dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd
0002

```

All the dd's are the device-specific data concatenated to the log.

The 0002 at the end is to terminate the log.

5.11 DBX enhancements

The print subcommand in DBX is enhanced to provide an easier-to-read display output. In AIX Version 4.3.3 and previous releases, array elements, and structure

or union fields are printed serially, one after the other, on a single line, which sometimes makes it hard to understand.

A sample output of the dbx print output subcommand in AIX Version 4.3 follows:

```
(dbx) print x
(op = 0_CONT, nodetype = (nil), value = union:(sym = 0x20076d88, name
= 0x20076d88, lcon = 0x20076d88, dash = 0x20076d88, llcon = 0x20076d88
00000000, addrcon = 0x20076d8800000000, fcon = 2.1841616996348188e-154
, qcon = (val = (2.1841616996348188e-154, 0.0)), kcon = (real = 2.1841
616996348188e-154, imag = 0.0), qkcon = (real = (val = (2.184161699634
8188e-154, 0.0)), imag = (val = (1.605837571007193e-154, 1.72522746112
82083e-314))), scon = "", fscon = (scon = "", strsize = 0x0), arg = (0
x20076d88, (nil), (nil), (nil), 0x20013980), trace = (exp = 0x20076d88
, place = (nil), cond = (nil), inst = false, event = 0x20013980, actio
ns = (nil)), step = (source = 537357704, skipcalls = false), examine =
(mode = "", beginaddr = (nil), endaddr = (nil), count = 0x0), procret
urn = (proc = 0x20076d88, retLocation = 0x0, caller_fp = 0x20013980000
00000), funcList = 0x20076d88), touch = '^A', refcount = '\0')
```

You can enable the new print subcommand style using the **set \$pretty="on"** command. This mode will use indentation to represent static scope of each value. A sample output is provided below:

```
(dbx) print a
{
    NamedObject::identity = {
        name = "0"
        number = 0x20008528
    }
    id = 0x1
    motion[0] = {
        ColoredObject::color = yellow
        a = 48.0
        b = 1000.0
        c = 0.0
    }
    motion[1] = {
        ColoredObject::color = indigo
        a = 2.0
        b = 100.0
        c = 0.0
    }
    motion[2] = {
        ColoredObject::color = orange
        a = 0.0
        b = 5.0
        c = 0.0
    }
}
```

Another output style can be enabled. The verbose mode will use qualified names instead of indentation to represent the static scope. To enable verbose mode, use the **set \$pretty="verbose"** command. A sample output for verbose mode is provided below:

```
(dbx) print a
NamedObject::identity.name = "0"
NamedObject::identity.number = 0x20008528
id = 0x1
motion[0].ColoredObject::color = yellow
motion[0].a = 48.0
motion[0].b = 1000.0
motion[0].c = 0.0
motion[1].ColoredObject::color = indigo
motion[1].a = 2.0
motion[1].b = 100.0
motion[1].c = 0.0
motion[2].ColoredObject::color = orange
motion[2].a = 0.0
motion[2].b = 5.0
motion[2].c = 0.0
```

These settings can be preserved by adding them to the `.dbxinit` file in your home directory.

5.11.1 The dbx command enhancements (5.2.0)

The **dbx** command has been enhanced to allow greater compatibility with the GNU **gcc** compiler and to assist developers in examining core files when the developer and program runtime environments differ with the `-p` flag.

Prior to Version 5.2, **dbx** only supported debugging applications compiled with **x1c**. Now **dbx** also supports debugging applications compiled with **gcc**. In order to debug your **gcc** applications in **dbx** you must use the `-gxcoff` compiler flag for **gcc**. If you do not use the `-gxcoff` flag, **gcc** will use XCOFF extensions and substrings only supported by the GNU debugger **gdb**. The following example shows how to compile the application `mytest.c` with **gcc** and debug it with **dbx**.

```
$ gcc -gxcoff mytest.c -o mytest
$ dbx mytest
Type 'help' for help.
reading symbolic information ...
(dbx) list
...
```

The new `-p` flag in **dbx** allows you to override the locations of object modules when examining core files. The core file contains an image of the process's state at the time of its termination. The loader information section of the core file

contains a table with all the object modules loaded by the application. All the object modules, except the main executable module, are specified as absolute file names in this table.

When examining core files, **dbx** uses this table to resolve library and shared object references, not the **LIBPATH** environment variable. If **dbx** is used to examine a core file and the modules are unable to be resolved, **dbx** will fail to load. This often happens when the core file is moved to another machine for debugging and the required libraries are either missing or in different locations. You must collect all the required libraries and put them in an expected location or edit the core file directly with the new library paths. The **-p** flag in **dbx** allows you to provide a mapping from the old to new library names, without modifying the core file.

The following example shows a session inspecting a core file generated from the **dhcpsd** process. In this example, **dbx** loaded all the required modules because the current and application runtime environment were the same. Notice that module Entry 3 is specified as the absolute file name **/usr/sbin/db_file.dhcpo**.

```
# ls -l core
-rw-r--r--  1 root    system      8149287 Sep  7 12:02 core
# dbx /usr/sbin/dhcpsd core
Type 'help' for help.
[using memory image in core]
reading symbolic information ...

Quit in _event_sleep at 0xd00555b0 ($t1)
0xd00555b0 (_event_sleep+0xa8) 80410014      lwz    r2,0x14(r1)
(dbx) where
_event_sleep(??, ??, ??, ??, ??) at 0xd00555b0
sigwait(??, ??) at 0xd005a394
main(??, ??) at 0x10000948
(dbx) map

...

Entry 3:
  Object name: /usr/sbin/db_file.dhcpo
  Text origin: 0xd5aff000
  Text length: 0x41e18
  Data origin: 0x20256ec8
  Data length: 0xaf0c
  File descriptor: 0x6

...

(dbx) quit
```

In the following example, the `db_file.dhcpo` library was deliberately renamed to `db_file.dhcpo.newname` to demonstrate the problem with mismatched core files. The **dbx** debugger will fail to start if it is unable to resolve all the modules in the loader information table.

```
# mv /usr/sbin/db_file.dhcpo /usr/sbin/db_file.dhcpo.newname
# dbx /usr/sbin/dhcpsd core
Type 'help' for help.
[using memory image in core]
reading symbolic information ...dbx: fatal error: cannot open
/usr/sbin/db_file.dhcpo
```

This problem can be resolved easily by using the `-p` flag for **dbx**. The `-p` flag can be either a list of colon-separated mappings or a file name. If a file name was given, the file must contain one mapping per line. The following example shows how to use the `-p` flag to map the `/usr/sbin/db_file.dhcpo` library to its new location `/usr/sbin/db_file.dhcpo.newname`.

```
# dbx -p /usr/sbin/db_file.dhcpo=/usr/sbin/db_file.dhcpo.newname
/usr/sbin/dhcpsd
Type 'help' for help.
[using memory image in core]
reading symbolic information ...
```

```
Quit in _event_sleep at 0xd00555b0 ($t1)
0xd00555b0 (_event_sleep+0xa8) 80410014      1wz   r2,0x14(r1)
(dbx) where
_event_sleep(??, ??, ??, ??, ??) at 0xd00555b0
sigwait(??, ??) at 0xd005a394
main(??, ??) at 0x10000948
(dbx) map
```

...

```
Entry 3:
  Object name: /usr/sbin/db_file.dhcpo.newname
  Text origin: 0xd5aff000
  Text length: 0x41e18
  Data origin: 0x20256ec8
  Data length: 0xaf0c
  File descriptor: 0x6
```

...

```
(dbx) quit
```

The following example is similar to the previous one, except that it uses the `-p` flag with a file name.

Create a file called `libmap` that contains the following mapping:

```
/usr/sbin/db_file.dhcpo=/usr/sbin/db_file.dhcpo.newname
```

Run **dbx** specifying the file `libmap` as the parameter for the `-p` flag.

```
# dbx -plibmap /usr/sbin/dhcpsd core
Type 'help' for help.
[using memory image in core]
reading symbolic information ...

Quit in _event_sleep at 0xd00555b0 ($t1)
0xd00555b0 (_event_sleep+0xa8) 80410014      1wz    r2,0x14(r1)
(dbx)
```

5.12 KDB kernel and kdb command enhancements

The KDB kernel debugger and **kdb** command are enhanced, as described in the following sections. For AIX 5L and subsequent releases, the KDB kernel debugger is the standard kernel debugger and is included in the `unix_up`, `unix_mp`, and `unix_64` kernels, which may be found in `/usr/lib/boot`.

5.12.1 Kernel debugger introduction

The KDB kernel debugger must be loaded at boot time. This requires that a boot image is created with the debugger enabled. To enable the KDB kernel debugger in AIX 5L, the **bosboot** command must be invoked with options set to enable KDB. The kernel debugger can be enabled using either the `-I` or `-D` options of **bosboot**.

Examples of **bosboot** commands:

- ▶ **bosboot -a -d /dev/ipldevice**
- ▶ **bosboot -a -d /dev/ipldevice -D**
- ▶ **bosboot -a -d /dev/ipldevice -I**

5.12.2 New functions and enhancements (5.1.0)

New subcommands were added to KDB in AIX 5L Version 5.1 in order to provide some functions already present in the **crash** command.

alias

The **alias** subcommand defines or displays aliases. The **alias** subcommand creates or redefines alias definitions or writes existing alias definitions to standard output. The syntax of the command is:

```
alias [AliasName [=string]]
```

ext

The **ext** subcommand prints the contents of memory in terms of words, in a linked list format. For example, you can print *n* contiguous words and then, on start, print from the word whose address is in the next pointer offset until the terminating address. This performs the same function as the **link** function in the **crash** utility. The syntax of the command is:

```
ext start_addr num_words [next_ptr_offset[end_value]]
```

set scroll

The **set scroll** subcommand is a new toggle introduced to the **kdb** command. Using this command at the **kdb** command prompt, you can toggle the page scrolling during the output of any **kdb** subcommand. For example:

```
set scroll on  
set scroll off
```

dcal and hcal

The **dcal** and **hcal** subcommands are modified to include the additional operators **^**, **%**, and **()**.

conv

The **conv** subcommand performs base conversions. The syntax for this command is:

```
conv [-bdox | -axx] num
```

Where *num* is the value to be converted and the optional flags indicate the base for *num*:

- ▶ -b = binary
- ▶ -d = decimal (default)
- ▶ -o = octal
- ▶ -x = hex
- ▶ -axx = base xx (2 to 36)

The input value is then displayed in binary, octal, decimal, and hex.

dump

The **dump** subcommand performs exactly the same function as the **dump** subcommand in **crash**, to dump the contents of storage.

errpt

The **errpt** subcommand prints all error log entries not picked up by the errdemon and allows the printing of a user-specified number of entries that have been picked up by the errdemon (the default is 3).

inode

The **inode** subcommand has two additional options. A **-c** flag displays the reference count of an inode. The second flag is **-d**. This flag requires that the next three arguments to the subcommand specify the major and minor device numbers and the inode number to be displayed. These changes will be made for both the KDB kernel debugger and the **kdb** command.

lke

Option **-n name** is added to the **lke** subcommand to allow specification of a substring that is required to occur within a loader entry name (for it to be displayed).

mbuf

A new **-n** option allows following the chain for the **m_next** element until the end of the chain. This chain is the collection of **mbufs** for a single packet. The **-a** option allows following the chain of **m_act** entries. This chain is a group of packets linked together. The **-a** and **-n** options can be used together. When both options are used, information for the **mbufs** within each packet is displayed; then the display proceeds to the next packet. These options were added to both the KDB kernel debugger and **kdb** command.

netm

The **netm** subcommand displays the most recent **net_malloc_police** record when invoked without any arguments. It may be invoked with an **-a** option to display all **net_malloc_police** records. It may also be invoked with an address to display records whose address or caller fields match the given address.

proc or p

In AIX 5L Version 5.1, the **proc** subcommand has an additional minus character (**-**) option. This option will list all the contents of the **proc** table. The asterisk (*****) lists a summary of the **proc** table content.

In Version 5.0, the **-s** option was added to the KDB **proc** subcommand. This option will be available for use in conjunction with the asterisk option, which displays a summary of all processes. The **-s** option will limit output to processes that are in the state specified following the **-s** flag.

sock

An additional function is added to the KDB **sock** subcommand. This function is available through the use of the **-p** flag and may be used to limit the output from the socket subcommand to just sockets associated with a specific process.

sr64

A new **-n** option is added to the **sr64** subcommand. This option may be used to indicate the *uadnode* data structure's information to be displayed for the uadnodes associated with the segment information displayed.

status

The **status** subcommand is added to both the KDB kernel debugger and **kdb** command. For each CPU, the CPU number and the thread ID, thread slot, process ID, process slot, and process name for the current thread are displayed.

thread or th

In AIX 5L Version 5.1, the **thread** subcommand has an additional minus character option. This option will display all the contents of the thread table. The asterisk lists a summary of the thread table contents.

In AIX Version 5.0, the **thread** subcommand received the **-r** and **-p** flag. The **-r** flag displays only runnable threads. The **-p** flag requires that a process table entry be specified and will display all threads for the indicated process.

varrm

The **varrm** subcommand is added to both the KDB kernel debugger and **kdb** command, and it allows user-defined variables to be cleared. A variable will be cleared by issuing the **varrm** subcommand and specifying the variable name as a parameter. Clearing a variable deletes the variable from the list of user-defined variables, freeing the slot for use by another user-defined variable.

varlist

The **varlist** subcommand is added to the KDB kernel debugger and **kdb** command, and it lists the names and values for any user-defined variables.

5.12.3 New functions and enhancements (5.2.0)

New subcommands have been added to the kernel debugger and to the **kdb** command. They are described as follows.

The set logfile subcommand

This **set** subcommand allows specification of a log file name or disablement of logging. The following **kdb** command will log the **kdb** command and the output of those commands into the ASCII file `/tmp/kdb.output`:

```
(0)> set logfile /tmp/kdb.output
```

The set loglevel subcommand

This **set** subcommand allows the granularity for the logging to be chosen. Valid choices are:

- ▶ off
- ▶ Log **kdb** commands only
- ▶ Log **kdb** commands and output

The set edit subcommand

This command (available on KDB and the **kdb** command) provides command line editing features similar to those provided by the korn shell, such as **vi**, **emacs**, and **gmacs**. For example, to turn on a **vi** style command line editing the command would be:

```
set edit vi
```

The output redirection facility

The **kdb** command allows now output redirection using the operators `|`, `>`, and `>>`. For example, to pipe to output of the **help** subcommand to the **pg** command, run the following:

```
(2)> help | pg
```

The di subcommand

The **di** subcommand displays the actual instruction, with the opcode and the operands, of the given input hexadecimal instruction.

The **di** subcommand is shown as follows:

```
(0)> di 9fe6212e
      stbu    r31,212E(r6)
(0)>
```

The which subcommand

The **which** subcommand displays the name of the kernel source file containing a specified symbol or address, as in the following.

```
(0)> which 100
      Addr: 24          Symbol: start
      Source filename: low.s
(0)> which start
      Addr: 24          Symbol: start
      Source filename: low.s
(0)>
```

The symptom subcommand

The **symptom** subcommand displays the symptom string from a dump. This command is not valid on a running system. The **-e** flag may be specified to generate an error log entry containing the symptom string.

The ndd subcommand

The **ndd** subcommand displays the network device driver statistics.

The netstat subcommand

The **netstat** subcommand symbolically displays the contents of various network-related data structures for active connections such as the AIX **netstat** command.

The print subcommand

The **print** subcommand is new to AIX 5L Version 5.2 and supports the formatted printing of the C language data structures. The use of the **print** subcommand requires a symbol file, such as `vnode.h`, as shown in the following example:

```
kdb -i /usr/include/sys/vnode.h
```

Then under the **kdb** prompt, run the following command:

```
(0)> vfs
```

	GFS	MNTD	MNTDOVER	VNODES	DATA	TYPE	FLAGS	
1	316F383C	0071E360	13000A80	00000000	14E0E880	316FCAF0	JFS	DEV MOUNT
...	/dev/hd4 mounted over /							
2	316F3870	0071E360	14A3EF80	13C07E00	1503A000	316FCB58	JFS	DEV MOUNT
...	/dev/hd2 mounted over /usr							
3	316F38A4	0071E360	145DFF80	14C6EF00	14F26A80	316FCC90	JFS	DEV MOUNT
...	/dev/hd9var mounted over /var							
4	316F3808	0071E360	146F8000	13E38800	131A0A00	316FCCF8	JFS	DEV MOUNT
...	/dev/hd3 mounted over /tmp							
5	316F390C	0071E360	13463F80	14EA2300	14CF4880	31A6B220	JFS	DEV MOUNT

```

... /dev/hd1 mounted over /home
6 316F3940 0071E420 00D75E48 1357BE80 00D75E48 00000000 PROCFS
... /proc mounted over /proc
7 316F3974 0071E360 13F53280 13693D80 14772100 31A6B2F0 JFS      DEVMOUNT

```

To display the structure for the vnode 14E0E880, run the following command:

```

(0)> print vnode 14E0E880
struct vnode {
    ushort v_flag      = 00x0;
    ulong32int64_t v_count = 000000x1;
    int v_vfsgen       = 000000x0;
    union Simple_lock {
        simple_lock_data _slock = 000000x0;
        struct lock_data_instrumented *_slockp = 000000x0;
    } v_lock;
    struct vfs *v_vfsp = 0x316F383C;
    struct vfs *v_mvfsp = 0x316F3A44;
    struct gnode *v_gnode = 0x14E0E8C0;
    struct vnode *v_next = 000000x0;
    struct vnode *v_vfsnext = 0x1503E500;
    struct vnode *v_vfsprev = 0x13C8FE80;
    union v_data {
        void *_v_socket = 000000x0;
        struct vnode *_v_pfsvnode = 000000x0;
    } _v_data;
    unsigned char *v_audit = 000000x0;
} foo[0];
(0)>

```

kdb routing information subcommands

Version 5.2 introduced three new **kdb** subcommands to display kernel routing information: **route**, **rtentry**, and **rxnode**.

The **route** subcommand displays information about the route structure for a specific address. The following example shows how to use the **route** subcommand.

```

# netstat -Aan | grep EST
7039b1f0 tcp4      0      2  9.3.149.21.23      9.53.150.13.37552  ESTABLISHED
700761f0 tcp4      0      0  9.3.149.21.32768    9.3.149.21.32769  ESTABLISHED
700769f0 tcp4      0      0  9.3.149.21.32769    9.3.149.21.32768  ESTABLISHED
700ed1f0 tcp4      0      0  9.3.149.21.32768    9.3.149.21.32770  ESTABLISHED
700ed5f0 tcp4      0      0  9.3.149.21.32770    9.3.149.21.32768  ESTABLISHED
702a99f0 tcp4      0      0  9.3.149.21.32768    9.3.149.21.32771  ESTABLISHED

```

```

# kdb
...
(0)> tcpcb 7039b1f0

```

```

---- TCPCB ----(@ 7039B1F0)----
seg_next..... 7039B1F0 seg_prev..... 7039B1F0
t_softerror... 00000000 t_state..... 00000004 (ESTABLISHED)
t_timer..... 00000005 (TCPT_REXMT)
t_timer..... 00000000 (TCPT_PERSIST)
t_timer..... 00003840 (TCPT_KEEP)
t_timer..... 00000000 (TCPT_2MSL)
t_rxtshift.... 00000000 t_rxtcur..... 00000005 t_dupacks.... 00000000
t_maxseg..... 000005B4 t_force..... 00000000
t_flags..... 00000000 ()
t_oobflags.... 00000000 ()
t_iobc..... 00000000 t_template.... 7039B218 t_inpcb..... 7039B144
t_timestamp... 900D3801 snd_una..... C2BD2AF0 snd_nxt..... C2BD2AF2
snd_up..... C2BD2AF0 snd_wll..... 89F1E904 snd_wl2..... C2BD2AF0
iss..... C2BC9B5F snd_wnd..... 0000E6A0 rcv_wnd..... 00004470
rcv_nxt..... 89F1E906 rcv_up..... 89F1E8F9 irs..... 89F1E58A
snd_wnd_scale. 00000000 rcv_wnd_scale. 00000000 req_scale_sent 00000000
req_scale_rcvd 00000000 last_ack_sent. 89F1E906 timestamp_rec. 00000000
timestamp_age. 00002433 rcv_adv..... 89F22D76 snd_max..... C2BD2AF2
snd_cwnd..... 0000FFFF snd_ssthresh.. 3FFFC000 t_idle..... 00000000
t_rtt..... 00000001 t_rtseq..... C2BD2AF0 t_srtt..... 00000008
t_rttvar..... 00000004 t_rttmin..... 00000002 max_rcvd..... 00000000

```

(0)> tcb 7039B144

```

----- TCB ----- INPCB INFO ----(@ 7039B144)----
next..... 00000000 prev..... 00000000 head..... 05E24000
iflowinfo... 00000000 faddr_6... @ 7039B158 fport..... 000092B0
fatype..... 00000001 oflowinfo... 00000000 laddr_6... @ 7039B170
lport..... 00000017 latype..... 00000001 socket..... 7039B000
ppcb..... 7039B1F0 route_6... @ 7039B188 ifa..... 00000000
flags..... 00000400 proto..... 00000000 tos..... 00000000
ttl..... 0000003C rcvttl..... 00000000 rcvif..... 3216D270
options..... 00000000 refcnt..... 00000002
lock..... 00000000 rc_lock..... 00000000 moptions.... 00000000
hash.next... 31A82F88 hash.prev... 31A82F88
timewait.nxt 00000000 timewait.prv 00000000
icmp6filter 00000000 cksumoffset FFFFFFFF
.....

```

(0)> route 7039B188

```

Destination.. 9.53.150.13
.....rtentry@ 7018D700.....

```

rt_nodes[0].....

```

rn_mklist @.. 7007B1E0
rm_b..... FFFFFFFF rm_unused.....
rm_flags..... 00000004 rm_mklist..... 00000000

```

```

        rmu_mask..... 70078F80
        mask..... 0.0.0.0
        rm_refs..... 00000000

        rn_p @..... 00000000
        rn_b..... FFFFFFFF rn_bmask..... 0000
        rn_flags..... 00000004 (ACTIVE)
        rn_key..... 0.0.0.0rn_mask..... 0.0.0.0
        rn_dupedkey @ 00000000
rt_nodes[1].....

        rn_mklist @.. 00000000
        rn_p @..... 00000000
        rn_b..... 00000000 rn_bmask..... 0000
        rn_flags..... 00000000 ()
        rn_key.....
        rn_dupedkey @ 00000000
gateway..... 9.3.149.1
rt_redisctime 00000000 rt_refcnt.... 00000003
rt_flags..... 00000003 (UP|GATEWAY)
ifnet @..... 3216D270 ifaddr @..... 70078780
rt_genmask @. 00000000 rt_llinfo @.. 00000000
rt_rmx (rt_metrics):
    locks ... 00000000 mtu ..... 00000000 hopcount. 00000000
    expire .. 3D818D76 recvpipe. 00000000 sendpipe. 00000000
    ssthresh. 00000000 rtt ..... 00000000 rttvar .. 00000000
    pksent... 00000466
rt_gwroute @. 7018D400 rt_idle..... 00000000
ipRouteAge... 00000000 rt_proto @... 00000000
gidstruct @.. 70076400 rt_lock..... 00000000
rt_intr..... 00000003 rt_duplist @. 00000000
rt_lu @..... 00000000 rt_timer..... 00000000
rt_cost_config 00000000

```

The **rxnode** subcommand displays information about the `radix_node` structure for a specific address. The **rtentry** subcommand displays information about the `rtentry` for the specified address. The following example shows how to use the **rxnode** and **rtentry** subcommands.

```
# netstat -rAn | more
```

Routing tables

Address Groups	Destination	Gateway	Flags	Refs	Use	If	PMTU	Exp
----------------	-------------	---------	-------	------	-----	----	------	-----

Route Tree for Protocol Family 2 (Internet):

```

700fe544 (32) 7007da18 : 700fe55c mk = 7007b1e0 {(0), (0) }
7007da18 (33) 701efe18 : 7007da00
701efe18 (36) 700fe52c : 701efe00
700fe52c 7018d700 default          9.3.149.1          UG          3          870  en2

```

```

- -
      mask (0) mk = 7007b1e0 {(0), (0) }
701efe00 9.3.149.21      127.0.0.1      UGHS      6      111 1o0      - -

7007da00 127/8      127.0.0.1      U      5      111 1o0      - -

# kdb
...
(0)> rxnode 701efe00

      rn_mklist @.. 00000000
      rn_p @..... 701EFE18
      rn_b..... FFFFFFFF rn_bmask..... 0000
      rn_flags..... 00000004 (ACTIVE)
      rn_key..... 9.3.149.21
      rn_dupedkey @ 00000000
      Traverse radix_node tree :
      parent - 1      quit - 0
      Enter Choice : 1

      rn_mklist @.. 00000000
      rn_p @..... 7007DA18
      rn_b..... 00000024 rn_bmask..... 0008
      rn_flags..... 00000004 (ACTIVE)
      rn_off..... 00000004
      rn_l @..... 700FE52C rn_r @..... 701EFE00
      Traverse radix_node tree :
      parent - 1      rn_r - 2      rn_l - 3      quit - 0
      Enter Choice :

(0)> rtentry 701EFE00

.....rtentry@ 701EFE00.....

      rt_nodes[0].....

      rn_mklist @.. 00000000
      rn_p @..... 701EFE18
      rn_b..... FFFFFFFF rn_bmask..... 0000
      rn_flags..... 00000004 (ACTIVE)
      rn_key..... 9.3.149.21
      rn_dupedkey @ 00000000
      rt_nodes[1].....

      rn_mklist @.. 00000000
      rn_p @..... 7007DA18
      rn_b..... 00000024 rn_bmask..... 0008
      rn_flags..... 00000004 (ACTIVE)
      rn_off..... 00000004

```



```

rn_l @..... 700FE52C rn_r @..... 701EFE00
gateway..... 127.0.0.1
rt_redisctime 00000000 rt_refcnt.... 00000006
rt_flags..... 00000807      (UP|GATEWAY|HOST|STATIC)
ifnet @..... 00BFF4A8 ifaddr @..... 70078C80
rt_genmask @. 00000000 rt_llinfo @.. 00000000
rt_rmx (rt_metrics):
    locks ... 00000000 mtu ..... 00000000 hopcount. 00000000
    expire .. 3D803859 recvpipe. 00000000 sendpipe. 00000000
    ssthresh. 00000000 rtt ..... 00000000 rttvar .. 00000000
    pksent... 0000006F
rt_gwroute @. 7007DA00 rt_idle..... 00000000
ipRouteAge... 00000000 rt_proto @... 7007B220
gidstruct @.. 00000000 rt_lock..... 00000000
rt_intr..... 0000000B rt_duplist @. 00000000
rt_lu @..... 00000000 rt_timer..... 00000000
rt_cost_config 00000000

```

The **trcstart** and **trcstop** subcommands

KDB is now able to start and stop an in-memory trace facility with **trcstart** and **trcstop** subcommands. The tracing does not cause any I/O. The resulting trace may only be viewed with KDB's **trace** command. However, if a dump is taken, the current trace data is written to the dump. These subcommands are only valid for KDB, not the **kdb** command.

5.13 Lightweight core file support

AIX 5L supports lightweight core files (lwcf) that consist of stack tracebacks from each thread and process. This enhancement assists large parallel jobs that need a way of collecting and displaying the state of all threads and processes when the job is abnormally terminated.

This enhancement provides two new routines, `mt_trce()` and `install_lwcf_handler()`, to be used by programs to generate a lightweight core file. This lightweight core file provides traceback information for each thread in each process of a potentially distributed application for debugging purposes.

Core files can be generated without process termination to increase application availability.

5.14 Core file naming enhancements (5.1.0)

AIX 5L Version 5.1 has changed the way it names the core file used for a core dump. In earlier AIX releases, a core file was always named *core*. If more than one application dumped or the same application dumped more than once, you always lost the earlier core file. Beginning with AIX 5L Version 5.1, each core file can be uniquely named so no core file will be overwritten with a new one. This feature helps debugging and tracing application failures.

5.14.1 File naming

By default, a new core file is named *core*. To enable the new enhancement, set the `CORE_NAMING` environment variable to `yes`.

After setting the `CORE_NAMING` variable, the new core file names are of the format `core.pid.ddhhmmss`, where:

pid	Process ID
dd	Day of the month
hh	Hours
mm	Minutes
ss	Seconds

Note: The expected value of the `CORE_NAMING` variable is `yes`. However, any value will work. So if `CORE_NAMING` variable is set to `no`, it will also generate the new style core file (`core.pid.ddhhmmss`).

The following is an example of core files recorded on a test system:

```
# ls -l
total 1080
-rw-r--r--  1 root    system      389223 Feb 20 17:40 core.20136.20234026
-rw-r--r--  1 root    system      180423 Feb 20 17:40 core.20138.20234059
-rw-r--r--  1 root    system      221923 Feb 10 14:20 core.10138.20202033
```

Note: Be aware that the timestamp in the file name is in GMT time format, so it does not reflect the current time on the system if an offset is used. To have the actual time the application dumped, you have to manually add the time zone offset.

5.14.2 Error log entry (5.2.0)

A program performing an illegal access on the system will result in its termination and a core file will be created containing the program's state. Core file creation also results in an errlog entry being logged to the AIX system error log file. Note that the core file will not be created under a set of circumstances, for example, if program's owner does not have write permission to the directory where the core file is being stored. This entry provides information about the program causing the coredump and stack information of the coredump, when possible.

The PROCESS ID stanza shows the process ID of the coredumping process. The PROGRAM NAME identifies the program causing the core dump. The CORE FILE NAME stanza shows the name of the core file created with its complete path. Note that the name of the core file name is restricted to 256 bytes. If the file name with path exceeds this limit the core file name will be truncated and this will be indicated by placing . . (dot space dot) in the middle of the core file name.

```
# errpt -a
```

```
LABEL:          CORE_DUMP
IDENTIFIER:      C60BB505
Date/Time:       Tue May 1 03:41:44 CDT
Sequence Number: 15
Machine Id:      000BC6FD4C00
Node Id:         server1
Class:           S
Type:            PERM
Resource Name:   SYSPROC
Description
SOFTWARE PROGRAM ABNORMALLY TERMINATED
Probable Causes
SOFTWARE PROGRAM
User Causes
USER GENERATED SIGNAL
```

```
Recommended Actions
CORRECT THEN RETRY
```

```
Failure Causes
SOFTWARE PROGRAM
```

```
Recommended Actions
RERUN THE APPLICATION PROGRAM
IF PROBLEM PERSISTS THEN DO THE FOLLOWING
CONTACT APPROPRIATE SERVICE REPRESENTATIVE
```

```
Detail Data
```

```

SIGNAL NUMBER
    11
USER'S PROCESS ID:
    18048
FILE SYSTEM SERIAL NUMBER
    5
INODE NUMBER
    2050
PROGRAM NAME
vi
ADDITIONAL INFORMATION
oncore 184
??
??
Unable to generate symptom string.

```

5.15 Gathering core files (5.1.0)

This enhancement automates core collection processes and packages them into a single archive. This archive will have all the necessary information to successfully analyze the core on any machine.

5.15.1 Using the snapcore command

The **snapcore** command gathers a core file, program, and libraries used by the program and compresses the information into a pax file. The file can then be downloaded to disk or tape, or transmitted to a remote system. The information gathered with the **snapcore** command allows you to identify and resolve problems within an application.

Collecting information

To collect all the information you might need to debug and analyze the problem. You can use the **snapcore** command, as shown in the following steps:

1. Change to the directory where the core dump file is located:

```

# ls -l
total 84176
-rw-r--r--  1 root    system      2704 Feb 21 09:52
core.18048.01084144
-rw-r--r--  1 root    system    38572032 Feb 20 23:49 gennames.out
-rw-rw-rw-  1 root    system     2260904 Feb 20 23:43 trace.out
-rw-r--r--  1 root    system     2260224 Feb 20 23:43 trace.rpt

```

2. Run the **snapcore** command to collect all needed files:

```

# snapcore -d /tmp/myDir core.18048.01084144

```

The **snapcore** command will gather all information and create a new compressed pax archive in the /tmp/myDir directory. If you do not specify a special directory using the -d flag, the archive will be stored in the /tmp/snapcore directory. The new archive file will be named snapcore_*\$pid*.pax.Z.

```
# s -l /tmp/myDir
total 5504
-rw-r--r--  1 root      system 2815081 Feb 21 09:56 snapcore_20576.pax.Z
```

To check the content of the pax archive, use the following command:

```
# uncompress -c snapcore_20576.pax.Z | pax
core.18048.01084144
README
ls1pp.out
errpt.out
vi
./usr/lib/libc.a
./usr/lib/libcrypt.a
./usr/lib/libcurses.a
./usr/lib/nls/loc/en_US
./usr/lib/libi18n.a
./usr/lib/libiconv.a
```

5.15.2 Using the check_core utility

The check_core utility is used by the **snapcore** command to gather all information about the core dump. This is a small C program and is located in the /usr/lib/ras directory.

Change to the directory where the core dump file is located and run the check_core utility against the core dump file. You will receive a list containing the program that caused the core dump and the libraries used by it.

```
# /usr/lib/ras/check_core core.24214.25124072
/usr/lib/libc.a
/usr/lib/libcrypt.a
/usr/lib/libcurses.a
/usr/lib/nls/loc/en_US
/usr/lib/libi18n.a
/usr/lib/libiconv.a
vi
```

Note: To make the `check_core` utility available for use, you must have the `bos.rte.serv_aid` fileset installed, as shown with the following command:

```
# ls1pp -w /usr/lib/ras/check_core
```

File	Fileset	Type
-----	-----	-----
/usr/lib/ras/check_core	bos.rte.serv_aid	File

5.16 Non-sparseness support for the restore command

In AIX 5L, the **restore** command has a new `-e` flag, which preserves the sparseness or non-sparseness of files created with the **backup** command.

A file is a sequence of indexed blocks of arbitrary size. The indexing is accomplished through the use of direct mapping or indirect index blocks from the files inode. Each index within a file's address range is not required to map to an actual data block.

A file that has one or more indexes that are not mapped to a data block is referred to as being sparsely-allocated or a sparse file. A sparse file will have a size associated with it, but it will not have all of the data blocks allocated to fulfill the size requirements. To identify if a file is sparsely-allocated, use the **fileplace** command. It will indicate all blocks in the file that are not currently allocated.

Such files are commonly used by database applications. The blocks with the NULL values are also often called holes. The default behavior of the **restore** command is to save disk space and therefore to create sparse files (if possible). This is the correct behavior if the original file is also a sparse file, but incorrect if the backup is a non-sparse file.

This enhancement restores the non-sparse files as non-sparse as they were archived by the name format of the **backup** command for both packed and unpacked files. It is necessary to know the sparseness/non-sparseness of the files before archiving the files, because enabling this flag restores the sparse files as non-sparse.

This flag should be enabled only if the files to be restored are non-sparse, consisting of more than 4 KB nulls. If the `-e` flag is specified during restore, it successfully restores all normal files normally and non-sparse database files as non-sparse.

5.17 The pax command enhancements

In AIX 5L, the **pax** command is enhanced to support a 64-bit POSIX-defined data format, which is used by default. The objective of this command is to allow the archiving of large files, such as dumps. The **cpio** and **tar** commands do not support files used as input larger than 2 GB because they are limited by their 32-bit formats. There are no plans to enhance these programs to support this situation in the future.

If you have to archive files larger than 2 GB, the only available option is the **pax** command, provided your file system supports it. Suppose you have several **tar** archives with a size in total exceeding the 2 GB limit. With the following command, you can create an archive for all of them:

```
# pax -x pax -wvf soft.pax ./soft?.tar
```

The default mode for **pax** (without the **-x** option) is to behave as **tar**. The **-x** option will allow **pax** the ability to work with files larger than 2 GB, a behavior **tar** does not have.

This enhancement is also available on AIX Version 4.3.3 service releases.

5.18 The snap command enhancements (5.1.0)

The **snap** command gathers system configuration information and compresses the information into a **pax** file. The information gathered with the **snap** command may be required to identify and resolve system problems.

5.18.1 Flag enhancements

The following sections discuss the new and enhanced flags for the **snap** command.

The -t flag

If in AIX 5L Version 5.0, the **-t** flag is used for the **snap** command, the following information will be collected in the **tcpip.snap** output file:

```
# lssrc -a
# netstat -m
# netstat -in
# netstat -v
# netstat -s
# netstat -an
# netstat -sr
# netstat -nr
```

```
# no -a
# arp -a
# arp -t atm -a
# ifconfig -a
# more /etc/resolv.conf
```

The enhancement to the **snap** command, when used with the **-t** flag, is that in addition to creating the **tcpip.snap** file, **snap** will add the following TCP/IP configuration files to the output device:

```
/etc/aliases
/etc/binld.cnf
/etc/bootptab
/etc/dhcprd.cnf
/etc/dhcpsd.cnf
/etc/dhcpd.ini
/etc/dlpi.conf
/etc/gated.conf
/etc/hostmibd.conf
/etc/hosts
/etc/hosts.equiv
/etc/inetd.conf
/etc/mib.defs
/etc/mrouted.conf
/etc/policyd.conf
/etc/protocols
/etc/pse.conf
/etc/pse_tune.conf
/etc/pxed.cnf
/etc/rc.bsdnet
/etc/rc.net
/etc/rc.net.serial
/etc/rc.qos
/etc/rc.tcpip
/etc/resolv.conf
/etc/rsvpd.conf
/etc/sendmail.cf
/etc/services
/etc/slip.hosts
/etc/snmpd.conf
/etc/snmpd.peers
/etc/syslog.conf
/etc/telnet.conf
/etc/xtiso.conf
```

When **snap** is used with the **-c** flag (to create a compact pax image), these files will be included in the image.

5.18.2 The -T flag

The -T flag gathers all the log files for a multiple-CPU trace. Only the base file, named *trcfile*, is captured with the -g flag.

```
snap [-g] -T trcfile
```

For example, you can gather a multiple-CPU trace file with the **trace** command:

```
# trace -C all
```

The trace can be stopped from collecting with the **trcoff** command. If no alternative log file is specified, **trace** will write to the default log file */var/adm/ras/trcfile*.

To run the **snap** command on the default log file, enter the following command:

```
# snap -g -T /var/adm/ras/trcfile
```

The -w flag

Running the **snap** command with the -w flag will gather all WLM information in the directory */tmp/ibmsupt/wlm*. This information includes the following files:

```
/etc/wlm/current/classes  
/etc/wlm/current/limits  
/etc/wlm/current/rules  
/etc/wlm/current/shares
```

The -x flag

The -x flag has been added to the **snap** command to launch the **adump** command without any parameter. The -x flag is used in conjunction with the -D flag. The result of the **adump** command will go into the */tmp/ibmsupt/dump* directory. The file is called *adump.report*.

```
# snap  
usage: snap -x -D  
# cd /tmp/ibmsupt/dump/  
# ls  
adump.report  dump.Z      dump.snap   unix.Z
```

The **adump** command runs a Perl script that gathers information needed for support professionals to start the dump analysis.

5.19 The tar command enhancements (5.2.0)

The **tar** command has been modified to exit now with an error when trying to extract a file that is not part of the **tar** archive.

The following example shows the **tar** command is new error message:

```
#tar -xvf /dev/rmt0 aaa bbb ccc
File aaa not present in the archive.
File bbb not present in the archive.
File ccc not present in the archive.
#echo $?
3
```

The return code of the **tar** command will be equal to the number of files that were not found in the archive. This is useful for scripts that manage automatic extractions.

System management

AIX 5L provides many enhancements in the area of system management and utilities. This chapter discusses these enhancements. Topics include:

- ▶ Installation and migration
- ▶ Web-based System Manager
- ▶ System backup tools and utilities
- ▶ Obtaining useful system information
- ▶ System access
- ▶ Mail

6.1 Installation and migration

The following discussion covers the enhancements to AIX 5L that assist you with installing and migrating AIX.

6.1.1 Alternate disk install enhancement (5.2.0)

Alternate disk install migration for network installation management (NIM) is now configurable through both the command line and SMIT. It is also possible to install the software from the BOS installation menus at system install time.

Alternate disk install at BOS installation time

There are two ways to install the software. It is now possible to install alternate disk installation at BOS install time, and the usual way with the **installp** command.

When installing a new system from the AIX CDs it is possible to install the software necessary to use alternate disk installation once the system is fully operational. The menu required is located under the More Options, option 3 screen, and from here the Install More Software, option 5. The screen where alternate disk install is selected is shown in Figure 6-1.

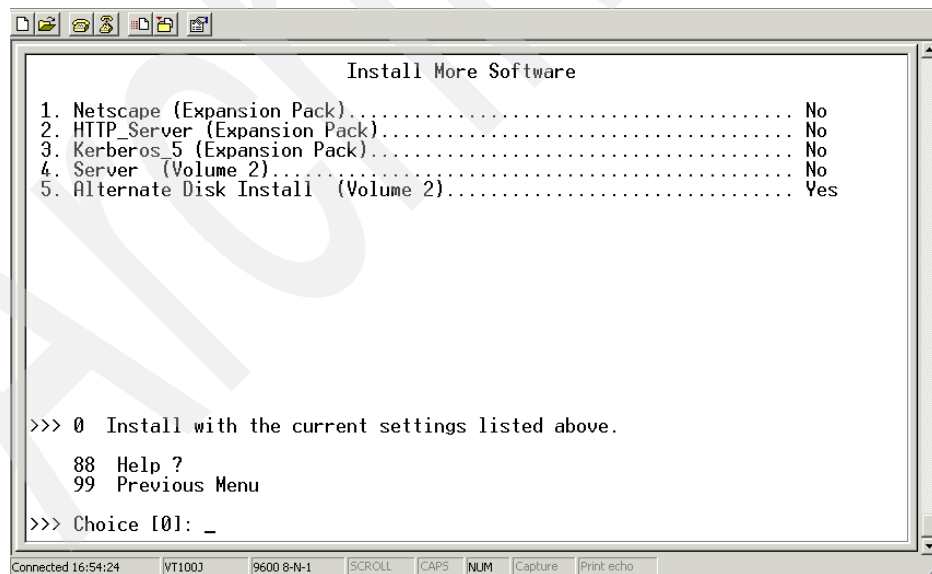


Figure 6-1 Selecting alternate disk install from the Install More Software screen

The following filesets are required to install the software necessary to enable alternate disk install:

- ▶ `bos.alt_disk_install.rte`
- ▶ `bos.alt_disk_install.boot_images`

These filesets can be installed during the BOS install (by selecting from the menus), or later.

Enabling NIM alternate disk migration

With AIX 5L Version 5.2 is a NIM alternate disk migration option available through the `nimadm` command and a SMIT `nimadm` fast path.

The `nimadm` command (network install manager alternate disk migration) is a utility that allows the system administrator to create a copy of rootvg to a free disk (or disks) and simultaneously migrate it to a new version or release level of AIX. `nimadm` uses NIM resources to perform this function.

There are several advantages to using `nimadm` over a conventional migration:

- ▶ Reduced downtime.

The migration is performed while the system is up and functioning normally. There is no requirement to boot from install media, and the majority of processing occurs on the NIM master.

- ▶ `nimadm` facilitates quick recovery in the event of migration failure.

Since `nimadm` uses `alt_disk_install` to create a copy of rootvg, all changes are performed to the copy (`altinst_rootvg`). In the event of serious migration installation failure, the failed migration is cleaned up and there is no need for the administrator to take further action. In the event of a problem with the new (migrated) level of AIX, the system can be quickly returned to the pre-migration operating system by booting from the original disk.

- ▶ `nimadm` allows a high degree of flexibility and customization in the migration process.

This is done with the use of optional NIM customization resources: `image_data`, `bosinst_data`, `exclude_files`, pre-migration script, `installp_bundle`, and post-migration script.

Access to this function is also available through SMIT from the Alternate Disk Installation menu (as shown in Figure 6-2 on page 320).

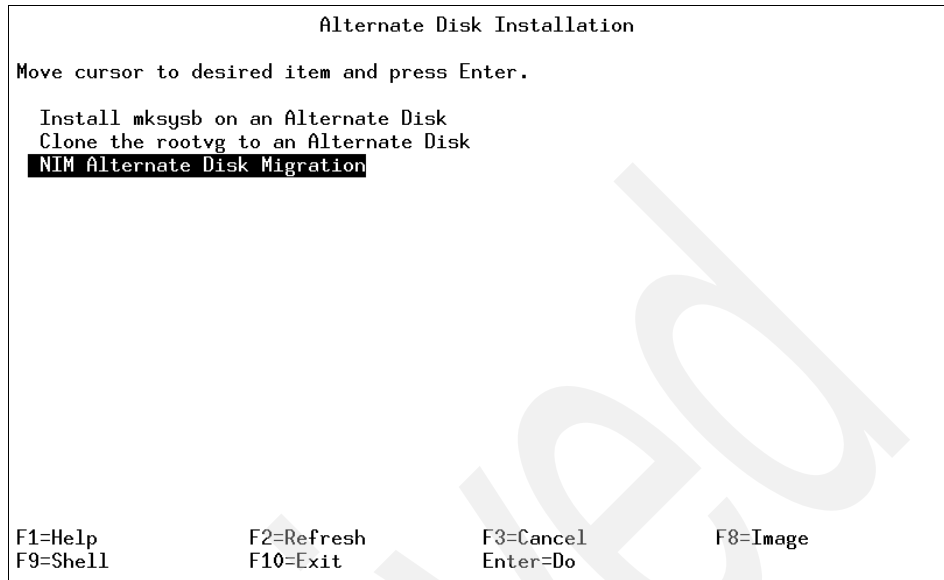


Figure 6-2 SMIT Alternate Disk Installation panel

From this menu screen (Figure 6-2) select **NIM Alternate Disk Migration** (**smitty nimadm**). This fast path shown in Figure 6-3.

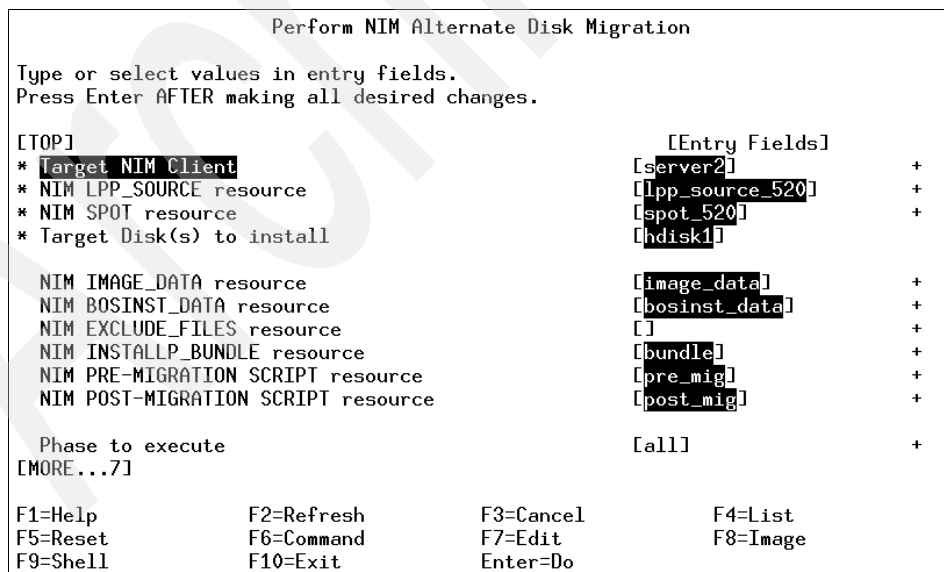


Figure 6-3 NIM Alternate Disk Migration screen

Alternate disk migration can only be selected on NIM Clients and so they should be set up from the NIM master.

6.1.2 NIM enhancement (5.2.0)

Before AIX 5L Version 5.2 it was possible to copy packages into a `lpp_source` directory or remove packages from a `lpp_source` directory and run `nim -o check` to update the `lpp_source` attributes. With AIX 5L Version 5.2, the `update` function is added to the `nim` command to provide a new enhancement to easily update `lpp_source` resources by adding and removing packages. The syntax of this new function is the following:

```
nim -o update -a packages=<all | list of packages with levels optional>
[-a gencopy_flags=flags] [-a installp_bundle=bundle_file]
[-a smit_bundle=bundle_file] [-a rm_images=<yes>]
[-a source=<dir | device | object>] lpp_source_object
```

The following example shows how to remove the `bos.games` package from the `lpp_source` `lppsource234`:

```
#nim -o update -a packages="bos.games" -a rm_images=yes lppsource234:
#
```

The following example shows how to add the `bos.games` package from the source directory `/stuff/0232A_520` to the `lppsource` resource `lppsource234`:

```
nim -o update -a packages="bos.games" -a source=/stuff/0232A_520 lppsource234
```

The `nim` command is also enhanced to display the `simage` warning only in two cases:

- ▶ When creating a `lppsource` with a default option that does not contain all the minimum filesets for a `simage`
- ▶ When a `nim -o check` command is run on a non-system image `lppsource`

The `simage` warning is not displayed if the `packages` option of the `nim` command is used, even if the `lppsource` does not contain all of the minimum filesets. The following examples show the creation of a non-`simage` `lppsource` resource with default options that display the `simage` warning:

```
nim -o define -t lpp_source -a server=master -a location=/lpp_source/perl
lppsource_perl
warning: 0042-267 c_mk_lpp_source: The defined lpp_source does not have the
"simages" attribute because one or more of the following
packages are missing:
    bos
    bos.net
    bos.diag
    bos.sysmgmt
```

```

bos.terminfo
bos.terminfo.all.data
devices.graphics
devices.scsi
devices.tty
x1C.rte
bos.up
bos.mp
devices.common
bos.64bit

```

The same `lppsource` resource is created, but now with the `packages` option and exits without warning, as shown in the following example:

```

nim -o define -t lpp_source -a packages="perl.rte perl.man.en_US" -a
server=master -a location=/lpp_source/perl lppsource_perl

```

The `nim` command also includes the `lppmgr` option to manage the `lpp_source` resource by cleaning the undesirable software, like duplicate filesets, or extra language and locale. See the `lppmgr` command for more information in 6.6, “The `bffcreate` and `lppmgr` enhancement (5.2.0)” on page 365. The Figure 6-4 SMIT panel shows how to eliminate the unnecessary software image in a `lpp_source` resource.

Eliminate Unnecessary Software Images in an `lpp_source`

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

	[Entry Fields]	
TARGET <code>lpp_source</code>	<code>lppsource234</code>	
PREVIEW only?	<input checked="" type="checkbox"/> yes	+
REMOVE DUPLICATE software	<input type="checkbox"/> yes	+
REMOVE SUPERSEDED updates	<input type="checkbox"/> yes	+
REMOVE LANGUAGE software	<input type="checkbox"/> yes	+
PRESERVE language	<code>[en_US]</code>	
REMOVE NON-SIMAGES software	<input type="checkbox"/> no	+
SAVE removed files	<input type="checkbox"/> no	+
DIRECTORY for storing saved files	<code>[]</code>	
EXTEND filesystems if space needed?	<input type="checkbox"/> yes	+

F1=Help

F5=Reset

F9=Shell

F2=Refresh

F6=Command

F10=Exit

F3=Cancel

F7=Edit

Enter=Do

F4=List

F8=Image

Figure 6-4 SMIT `nim_lppmgr` panel for the `lppsource lppsource234`

6.1.3 Version 5.2 AIX migration (5.2.0)

Version 5.2 migration is possible from Version 4.2.X onwards and on PCI architecture machines only.

Prerequisites for Version 5.2 migration

One of the features of Version 5.2 is the removal of MCA and PReP support. For this reason, migration from releases prior to Version 4.2 (for example, Version 3.2 and Version 4.1) is not supported, as these versions did not support CHRP hardware. CHRP hardware is the only hardware platform supported in Version 5.2 (not to be confused with PCI architecture).

Points to consider before starting

The following points should be checked before an AIX 5L Version 5.2 migration is undertaken:

- ▶ AIX 5L Version 5.2 release notes have been fully read and all appropriate actions taken prior to start of the migration.
- ▶ Full recoverable backup of system is available.
- ▶ System is fully documented should a recovery be needed or for further configuration after the migration.
- ▶ All licensed applications that run on the system are able to run at the new level and there are no licensing issues.
- ▶ The system has the following minimum hardware configuration:
 - Platform is chrp based. This is the only supported platform at Version 5.2 (**bootinfo -p**).
 - 128 MB RAM.
 - 512 MB paging space.
 - 2.2 GB hard drive for base operating system (although this may depend on the number of packages installed and any further upgrades that need to be done post-migration).
- ▶ Check that firmware on CD-ROM is up to date, so system can be booted from CD.
- ▶ If migrating from AIX Version 4.2.1, the system must be updated to the September 1999 or later update CD. bos.rte.install should be at 4.2.1.17 or later.
- ▶ If migrating from AIX Version 4.2x or AIX Version 4.3x, xlc.rte should be at level 5.0.2.x; otherwise, install APAR IY17981.
- ▶ If pmtoolkit is at Version 1.3.1.6, it must be uninstalled prior to the migration and the machine rebooted.

- ▶ Only systems with a 64-bit kernel will be able to run the 64-bit kernel and therefore use the JFS2 enhancement. These systems will also be able to run the 32-bit kernel.
- ▶ When migrating from versions of AIX prior to AIX 5L using mirrored root volume groups, note that the two additional file systems, /proc and /opt, will need to be manually mirrored. File systems in rootvg that already exist will remain mirrored assuming they were prior to migration.
- ▶ Version 5.2 uses Java Version 1.3.1, and previous versions should be removed unless required by applications that will still reside on the system after the upgrade. It is only possible to remove Java Version 1.8 from the installation screens; other versions will need to be removed manually. It might be required to ensure that the PATH variable for users that need Java should include the following: /usr/java131/bin:/usr/java131/jre/bin. References to previous versions should be removed unless they are needed.
- ▶ We recommend that you reinstall performance toolbox to Version 3 and reinstall the AIX toolkit for Linux applications. The LIBPATH for the AIX-rpm must be checked so that it is used over the Linux-rpm. The path should be /usr/lib:/usr/local/lib.

Features of migration

As Version 5.2 only supports PCI architecture machines, part of the migration is to remove now obsolete filesets from the BOS. The migration to Version 5.2 has the following steps:

- ▶ Configuration files are saved in /tmp/bos.
- ▶ Prepare for the removal of old files.
- ▶ Restore new files to the bos image.
- ▶ Remove obsolete filesets.
- ▶ Migrate configuration data where possible.
- ▶ Update vital product database (VPD) with migration information, including filesets that are removed.
- ▶ Update additional filesets.

Steps to migrate to Version 5.2

The following example was taken from a Version 5.1 system. The steps are the same from Version 4.2 up to Version 5.1.

Ensure that full bootable system backups are available in the form of a **mksysb** or in-line with the tested system recovery procedures in place for the service environment. Do not proceed with a migration unless the system is recoverable. It is also advisable to fully document the system setup. This is possible by using

the **snap -a** command and copying the contents of /tmp/ibmsupt to offline media or another machine.

The machine needs to be booted into the system maintenance screen. Ensure that Version 5.2, CD1 is in the drive and either the bootlist is set to read this device before either a disk or network boot, or the boot process is interrupted with the 5 or F5 key sequence.

Select the terminal as the system console and press Enter, then select the language of your choice for the install. The default is English.

This will go into the Installation and Maintenance menu, where the Change/Show option should be selected. This is shown in Figure 6-5.

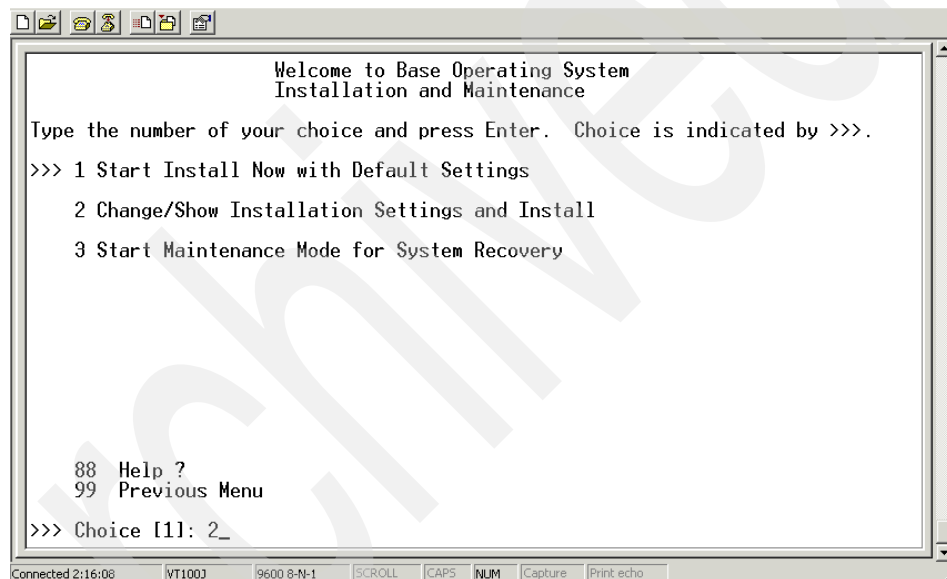


Figure 6-5 BOS Installation and Maintenance menu

Choose option 2 to go into installation and settings. Ensure that the install option is set to migration by selecting option 1 to change it if necessary. This is shown in Figure 6-6 on page 326.

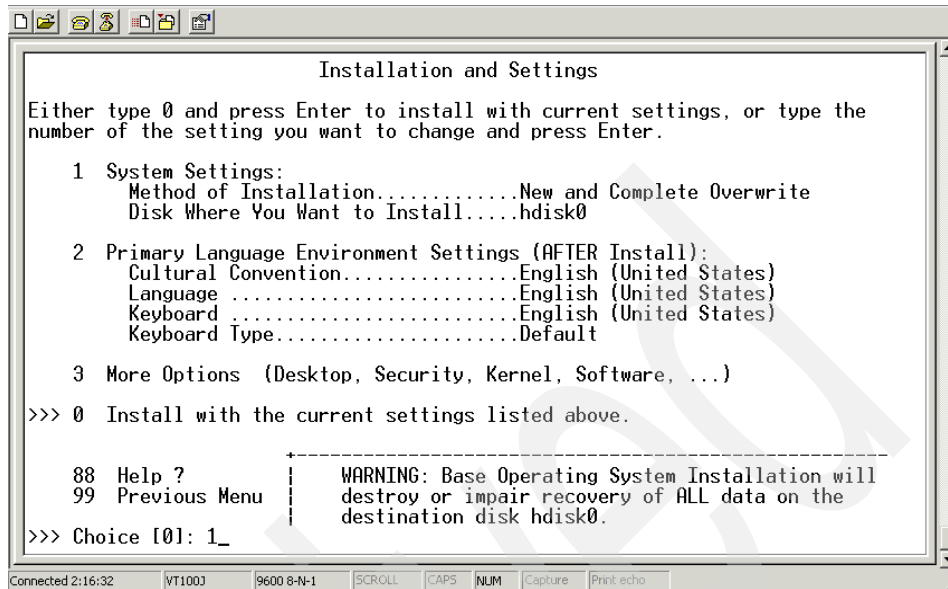


Figure 6-6 Installation and Settings screen

Option 1 moves the user to the installation method screen, as shown in Figure 6-7. Select option 3 at this point.

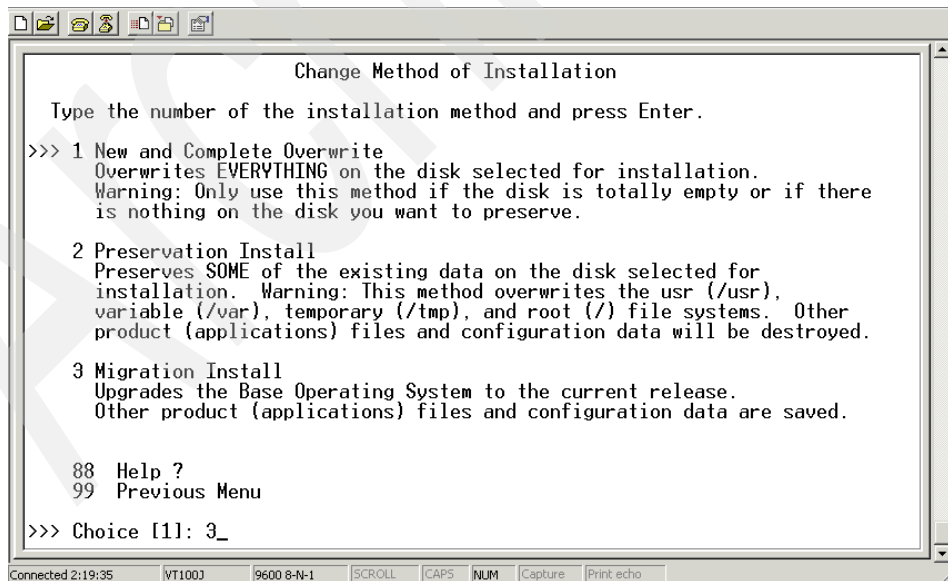


Figure 6-7 Method of Installation screen

Disks that are already assigned to rootvg will be automatically selected (signified by >>> on the left-hand side of the screen). Ensure that this is the case and accept the selected disks. In this example, choose option 1 to accept hdisk0, as shown in Figure 6-8. This returns you to the Installation and Settings menu. Here select option 3, More Options. Notice that the installation method is now set to migration. This is shown in Figure 6-9 on page 328.

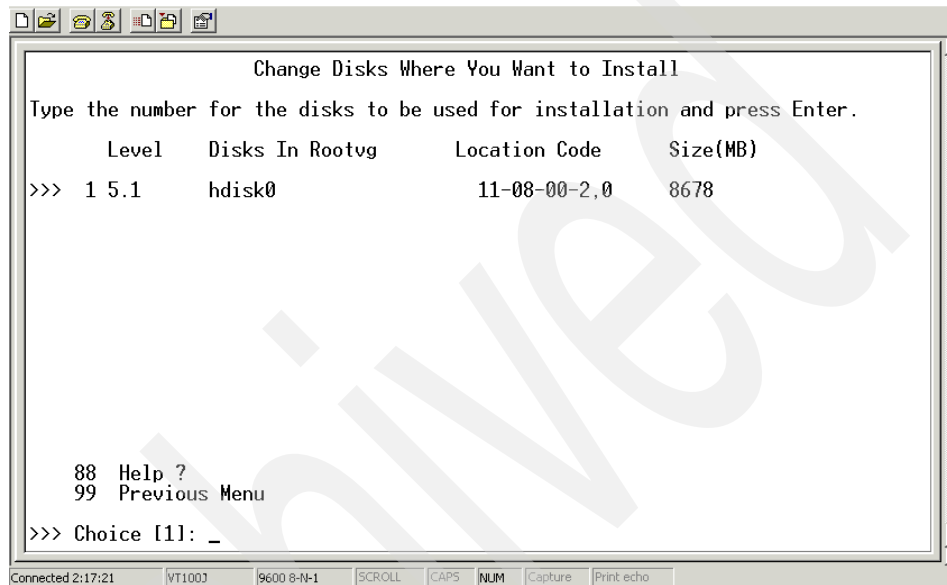


Figure 6-8 Disks to install screen

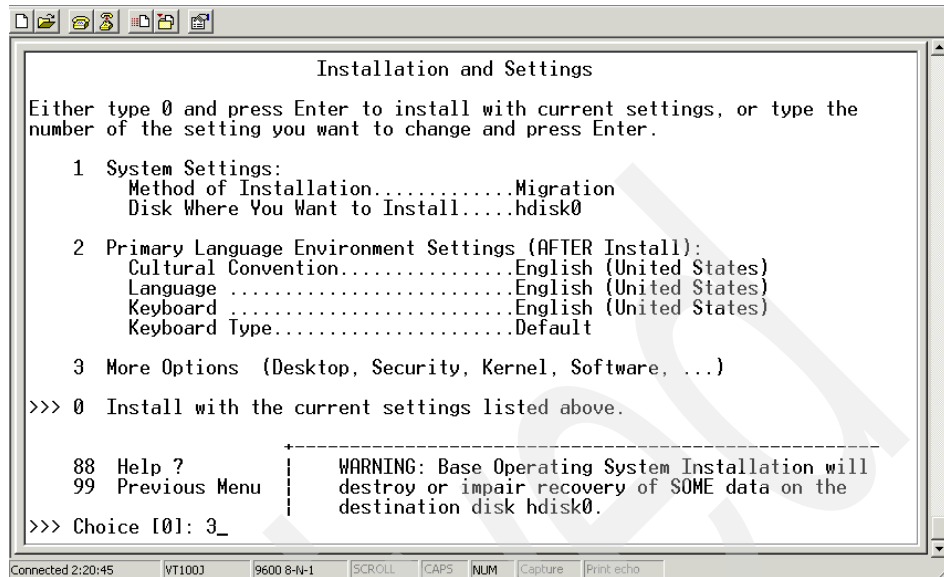


Figure 6-9 Installation and Settings screen, install method set to migrate

Selecting option 3, More Options, takes the user into the menu shown in Figure 6-10. Select options as required.

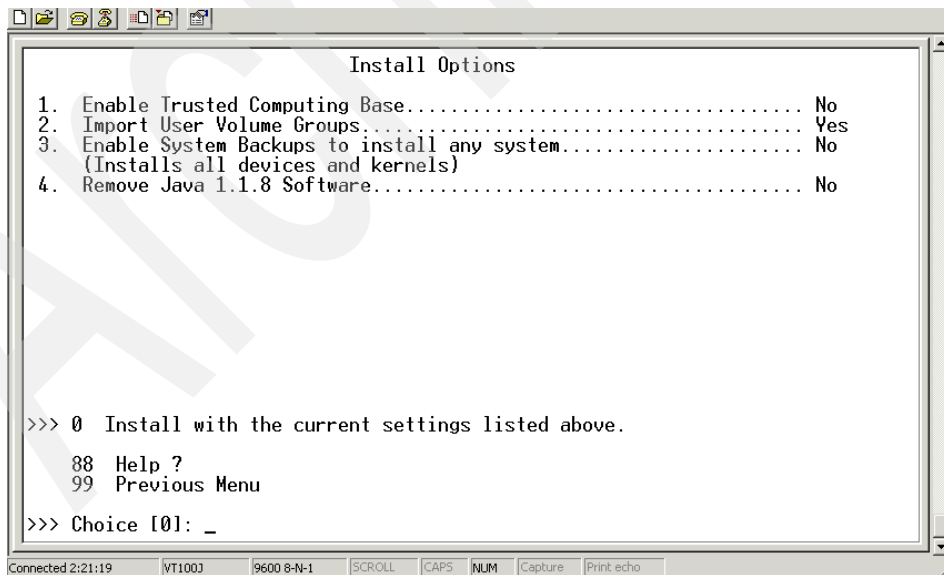


Figure 6-10 Install Options for migration install

The options are worthy of note, especially the TCB and system backups options. These are discussed below:

- ▶ Enable Trusted Computing Base

The TCB option should only be used if TCB was initially installed on the system.

- ▶ Import User Volume Groups

If volume groups other than rootvg are detected, this option is automatically set to yes. If no is selected, volume groups other than rootvg are not imported and remain unaffected by the install process, this is useful in the case of shared volume groups.

- ▶ Enable System Backups to install any system

This install kernels and device drivers not necessarily needed for the current system, but that might be needed should a system backup of this system be used to clone the image onto different hardware (PCI only).

Once installs options have been selected, choose option 0 to continue with the install. This will present the user with an install summary screen and a chance to go back and change all settings. If option 1 is selected on the migration installation summary screen the install will start.

6.2 Web-based System Manager

The Web-based System Manager is enhanced in AIX 5L. This section provides an in-depth look at what has changed from previous versions.

Keep in mind that the discussion of AIX Version 4.3.3 in this section is only for historical reference.

Note: For more information about AIX System Management or the Web-based System Manager architecture and previous releases features, refer to *AIX Version 4.3 Differences Guide*, SG24-2014.

It is also possible to press F1 during a Web-based System Manager session to display the main help panel.

6.2.1 Web-based System Manager architecture

The Web-based System Manager enables a system administrator to manage AIX machines either locally from a graphics terminal or remotely from a PC, Linux, or AIX client. Information is entered through the GUI components on the

client side. The information is then sent over the network to the Web-based System Manager server, which runs the necessary commands to perform the required action.

The Web-based System Manager is implemented using the Java programming language. The implementation of Web-based System Manager in Java provides:

- ▶ Cross-platform portability: Any client platform with a Java 1.3-enabled Web browser is able to run a Web-based System Manager client object.
- ▶ Distributed processing: A Web-based System Manager client is able to issue commands to AIX machines remotely through the network.
- ▶ Multiple launch points: The Web-based System Manager can be launched either in a Java application mode locally within the machine to manage both a local and remote system, in a Java Applet mode through a system with a Web browser with Java 1.3, and in Windows PC Client mode, where client code is downloaded from an AIX host.

User interface

The user interface has improved noticeably; the console provides a convenient and familiar interface for managing multiple AIX hosts. The console panel is divided into two panes: A Navigation Area on the left for displaying the hierarchy of host computers and management applications, and a Contents Area, on the right for displaying the contents of each level in the navigation hierarchy, as shown with the optional SDK Samples Environment seen installed in Figure 6-11 on page 331.

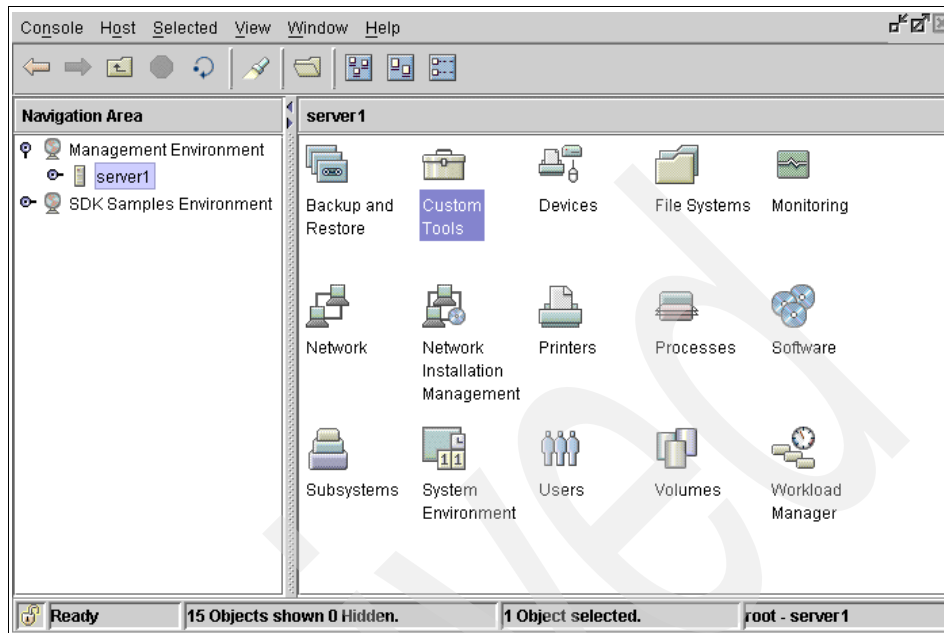


Figure 6-11 Web-based System Manager user interface

Plug-in architecture

As shown in Figure 6-11, the Navigation Area, on the left, has the host names of the servers to be administered, and each server contains a list of items that the Web-based System Manager can handle.

Each item contains a name and an icon. Each icon in this area is a *plug-in*. When the user selects a plug-in icon in the Navigation Area, the plug-in displays its contents in the Contents Area, updates the menu bar and tool bar with its actions, and updates the Tips Area with links for help on relevant tasks. Plug-ins are somewhat analogous to applications; they encapsulate a collection of management functions in the form of managed objects, collections of managed objects, tasks, and actions. A plug-in can consist of:

- ▶ An overview panel
- ▶ One or more sub-plug-ins
- ▶ An overview and one or more sub-plug-ins
- ▶ A collection of managed objects
- ▶ A panel for launching management interfaces in a panel external to the console

The Web-based System Manager plug-in architecture is designed to provide a high degree of flexibility in the design of client applications. Both object and task-oriented plug-in models are provided, as well as the ability to integrate applications developed outside of the Web-based System Manager framework. The object-oriented design of the framework supports consistency across plug-ins while enabling the flexibility to extend and customize plug-in classes. The Web-based System Manager supports the classes of plug-ins discussed in the following sections.

Container

Container plug-ins are the most common type of plug-in used in the Web-based System Manager user interface. Container plug-ins are somewhat analogous to directories in a file system (or *folders* in a graphical file system manager). They contain other plug-ins, managed objects, or combinations of plug-ins and managed objects. Figure 6-12 shows a Container plug-in example.

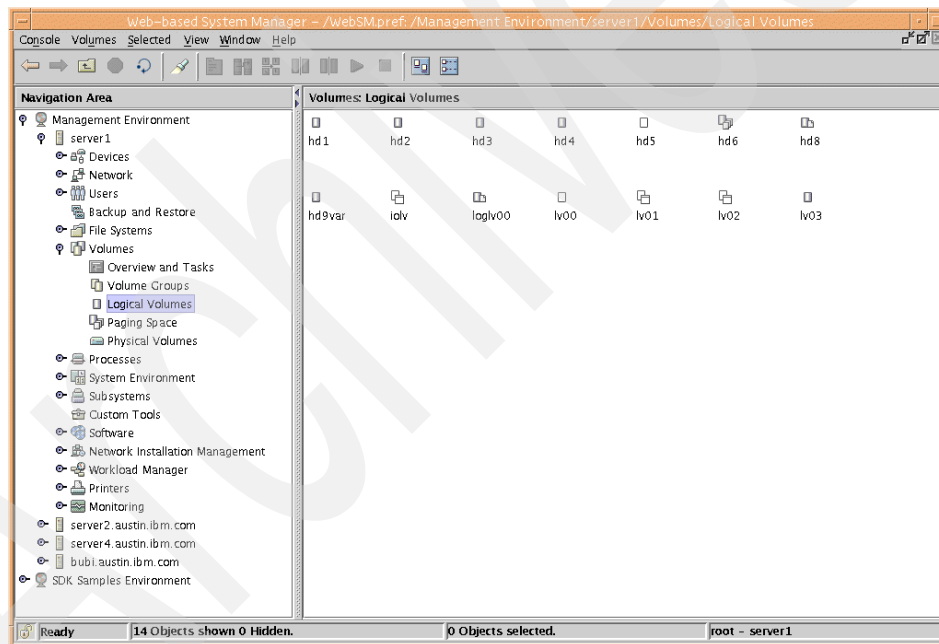


Figure 6-12 Container plug-in example

Containers present objects in views. The Web-based System Manager supports the typical object views (Large Icon, Small Icon, and Details), as well as two hierarchical views (Tree and Tree-Details). Figure 6-12 shows an example of a Container plug-in used in the Large Icon view; Figure 6-13 on page 333 illustrates the detail view.

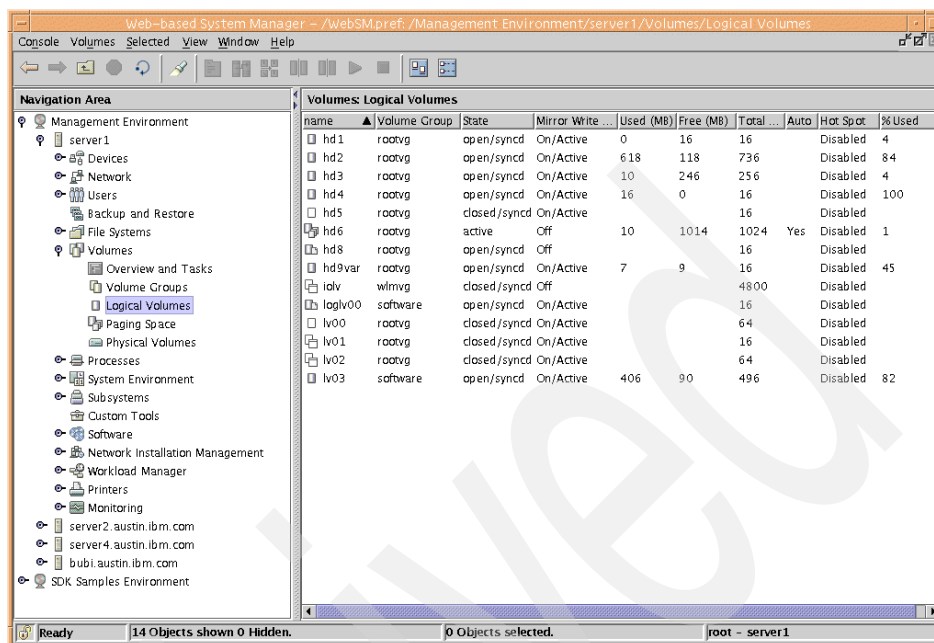


Figure 6-13 Example of logical volumes container in detail view

Overview

Overview plug-ins are panel interfaces that appear in the contents area of a console child panel. The primary functions of overviews are to:

- Explain the function provided by an application plug-in.
- Provide a launch point for routine or *getting started* tasks.
- Summarize the status of one or more management functions.

In addition, because overviews are task-based rather than object-based, they can be used to provide quicker and easier access to some functions than container views. In cases where a management function does not lend itself to an object-oriented design (for example, backup and restore), the entire application can be implemented using one or more Overview plug-ins.

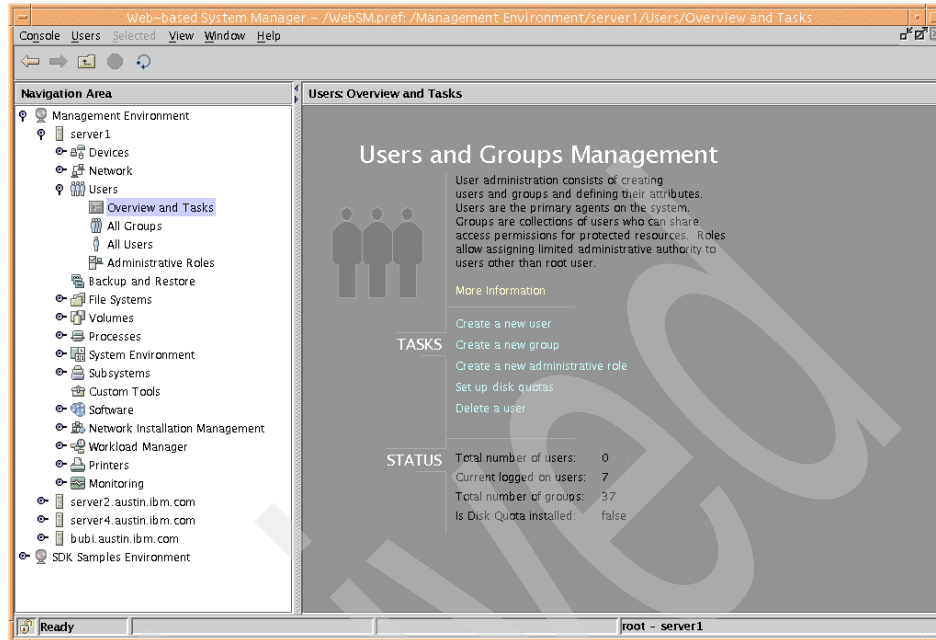


Figure 6-14 Overview plug-in example, users and groups overview

Launch

Launch plug-ins serve as a mechanism for launching applications that were implemented outside of the Web-based System Manager framework. By using a launch plug-in, these *external* applications may be integrated into the Web-based System Manager console. The launch plug-in provides an overview-like panel with title, description area, a link to browser-based help, and a task link for launching the external application.

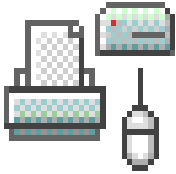
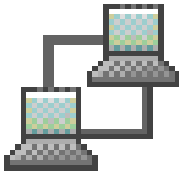
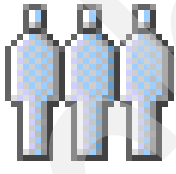
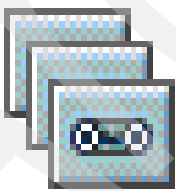
Standard plug-ins for Web-based System Manager

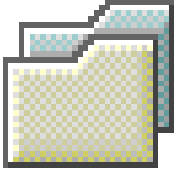
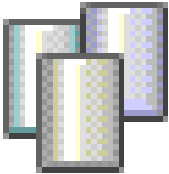
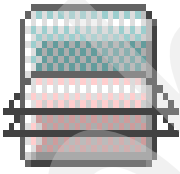
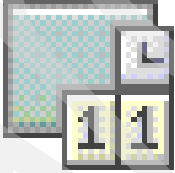
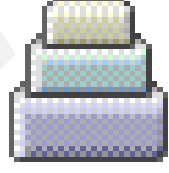
When you first run Web-based System Manager using the new graphical interface, keep in mind that all navigation is performed on the left side of the user interface.

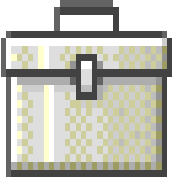
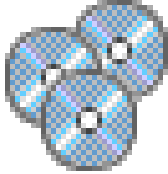
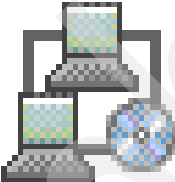

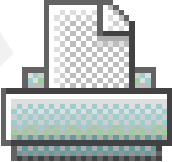
Even if you have more than one server registered, each server will have standard plug-ins, as shown in Table 6-1 on page 335.


A Security plug-in, not available with a default install, will be made available once you install the Expansion Pack. It is part of the base system, however.

Table 6-1 List of standard plug-ins in Web-based System Manager

Plug-In	Containers	Action
<p>Devices</p> 	<p>Overview and Tasks</p> <p>All Devices</p> <p>Communication</p> <p>Storage Devices</p> <p>Printers, Display</p> <p>Input Devices</p> <p>Multimedia</p> <p>System Devices</p>	<p>All hardware device-related actions like add, remove, change and show</p>
<p>Network</p> 	<p>Network Overview</p> <p>TCP/IP (IPv4 or IPv6)</p> <p>Point-to-Point (PPP)</p> <p>NIS</p> <p>NIS+</p> <p>SNMP: Included in AIX 5L.</p> <p>Virtual Private Networks</p>	<p>All network-related actions such as TCP/IP network, basic configuration, remove network interface, and NIS</p>
<p>Users</p> 	<p>Overview and Tasks</p> <p>All Groups</p> <p>All Users</p> <p>Administrative Roles</p>	<p>User- and group-related actions, as well as administrative roles for user authorization</p>
<p>Backup and Restore</p> 	<p>No containers, all options are located in the overview panel</p>	<p>Performs actions related to backup, such as image backup, incremental backup, and restore</p>

Plug-In	Containers	Action
File Systems 	Overview and Tasks Journaled File Systems Network File Systems Exported Directories CD-ROM File Systems Cache File Systems	All file system-related tasks, such as add and remove a file system
Volumes 	Overview and Tasks Volume Groups Logical Volumes Paging Space Physical Volumes	All logical volume manager-related actions, including volume groups and physical volumes
Processes 	Overview and Tasks All Processes	Process-related action, such as changing priority, killing a process, and listing all processes
System Environment 	Overview and Tasks Settings	System environment will handle operations, such as shut down and broadcast messages, as well as licenses and Kerberos settings. License manager container is a new option
Subsystems 	Overview and Tasks All Subsystems	All subsystem-related tasks can be done through this option, such as list, start, or kill a subsystem

Plug-In	Containers	Action
Custom Tools 	No containers, just a Custom Tools helps icon; Additional icons will be added for each custom tool created	Custom tools allows you to integrate any command or Web application into Web-based System Manager
Software 	Overview and Tasks Installed Software	All software-related tasks, such as list and install new software
NIM 	Overview and Tasks	Network Installation Manager (NIM) can be set up from this option, as well as NIM administration
Workload Manager 	Overview and Tasks Configurations/Classes Resources	All Workload Manager-related tasks, such as create class assignment rules, update, and stop Workload Manager; incorporates all new enhancements for AIX 5L
Printers 	Overview and Tasks All Printers	All printing-related tasks, such as add a printer, remove a printer queue, and list all printers; includes System V printing subsystem

Plug-In	Containers	Action
Monitoring 	Overview and Tasks Conditions Responses Events	All monitoring-related tasks, such as create new conditions, list responses and events; it is a new option in Web-based System Manager

Modes of operation

As in previous releases, the Web-based System Manager can be launched from a variety of launch points. For example:

- ▶ Java application mode through the `wsm` command in the AIX command line on the system being managed.
- ▶ Java application mode, where the console is running on one AIX system, but managing remote systems. Called client-server mode.
- ▶ Management Console icon on CDE.
- ▶ Java applet mode through Java 1.3-enabled Web browser.
- ▶ Windows PC client mode.

The Windows PC client code is downloaded from an AIX host, then installed permanently on the PC. Because all the Java code is native on the PC, startup time and performance are exceptionally good compared to applet mode.

The user can start Web-based System Manager PC client in several ways:

- Double-click the Web-based System Manager icon that was installed on the system desktop.
- Select the Web-based System Manager entry in the Programs menu.
- Locate the `wsm.exe` executable in Windows Explorer by changing to the install directory and double-clicking.
- Change to the install directory within an MS-DOS panel and type `wsm.exe`.

This flexibility allows you to perform administrative tasks across multiple servers regardless of where you perform them. From a mode of operation point of view, the Web-based System Manager can be managed from three different ways, as discussed in the following sections.

Local

AIX systems with a graphical user interface (GUI) can use this mode to perform local tasks. This mode is enabled by default.

Figure 6-15 shows the Management Console icon that starts the Web-based System Manager on CDE.



Figure 6-15 Web-based System Manager icon on CDE user interface

Client-server mode

The administrator can add hosts, represented by icons, to additional Internet-attached hosts in the Navigation Area of the console. The list of hosts and user interface preferences are stored in a console preferences file. The console preferences file can be stored on a specific host that will serve as the contact host or in a distributed file system (to allow it to be accessed directly from multiple hosts). When multiple hosts are set up to be managed from a single console, the Web-based System Manager operates in client-server mode. The first machine contacted by the client acts as the managing host while the other hosts in the navigation area are managed hosts.

Applet mode

In applet or browser mode, the administrator can manage one or more AIX hosts remotely from the client platform's Web-browsers with Java 1.3. To access the console in this manner, an AIX host need only be configured with a Web-server (provided on the AIX Bonus or Expansion Pack CDs). Once the Web-server is installed and configured, the host can serve the console to the client. The administrator simply enters a URL, `hostname/wsm.html`, into the browser. A Web page is then served to the browser that prompts the user for a user name and

password. Once authenticated to the server, the console launches into a separate panel frame. In Web-based System Manager applet mode, the browser is used only for logging in and launching the console. Once running, the console is relatively independent of the browser.

6.2.2 Web-based System Manager enhancements for AIX 5L

Table 6-2 provides a comparison list of new enhancements on the Web-based System Manager presented with AIX 5L.

Table 6-2 Comparison chart with the new enhancements

AIX Version 4.3	AIX 5L Version
Launch pad and multiple panels	Management Console
Single host management	Point-to-Point multiple host management
Java 1.1	Java 1.3
Back end shell script execution	Shell script and API execution interface
Stateless user interface	Dynamic user interface
Session UI customization	Persistent UI preferences
SSL security option	SSL security option
	Kerberos Version 5 integration in AIX
	Monitoring, notification, and control

Monitoring

Refer to 3.7, “Resource Monitoring and Control” on page 145, for monitoring details.

Session log

A new feature introduced in Web-based System Manager for AIX 5L is the Session Log. This log is located on the Console menu, and will log the following events:

- ▶ All actions performed in any managed host
- ▶ Success or failure messages
- ▶ Security level messages

Figure 6-16 on page 341 shows a sample output from a session log.

	Date/Time	User	Hostname	Message
i	1:13 PM ...		server1	Environment Error: Machine types file does not exist.: RealmMachineTypes.db
i	1:38 PM ...		server1	Run command: /usr/sbin/cfgmgr
i	1:40 PM ...		server1	Create new device, run command: /usr/sbin/mkdev -c printer -t opp -s parallel -p ppa0 -w p
i	1:41 PM ...		server1	Delete lp0
i	1:41 PM ...		server1	Create new device, run command: /usr/sbin/mkdev -c printer -t opp -s parallel -p ppa0 -w p
i	1:42 PM ...		server1	Delete lp0
i	3:03 PM ...		server1	Start monitoring: /tmp space used ...
i	3:03 PM ...		server1	java.lang.NullPointerException ...
i	3:03 PM ...		server1	Start monitoring: /var space used ...

Find: Find Next

Save Save As Clear Close

Figure 6-16 An example of output from a session log

When this log is opened, you will discover the following controls:

- Find** Searches for a particular string or sentence among the messages already logged
- Save** Saves any new entry in the log table, and will append to the log file specified in the Save as option
- Save as** Saves all entries in the log table, and will store them in a new file, or will create the default file in /tmp/websm.log
- Clear** Removes all entries in the log table
- Close** Closes the Session Log panel

If you double-click any entry in the log table, a new panel will pop up with detailed information on that specific entry. An example is shown in Figure 6-17 on page 342.

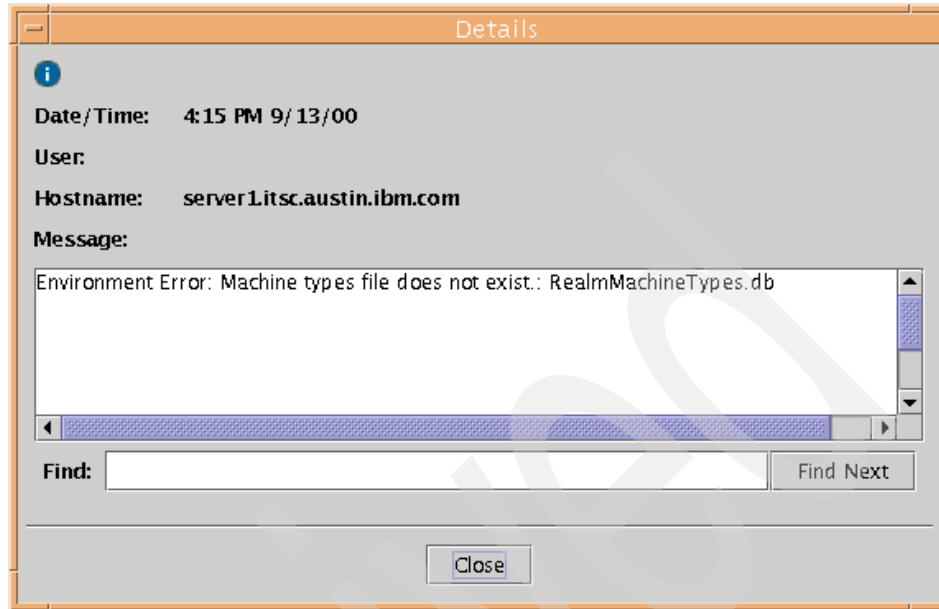


Figure 6-17 An example of session log detailed entry

Custom tools

It is possible to integrate other administration applications into Web-based System Manager. Custom tools extends the capabilities of the registered applications tool in previous releases. As before, URL-based applications can be added, but in addition, a new command tool option allows any tool that can be invoked through the command line to be integrated into Web-based System Manager.

There are two different types of custom tools:

- ▶ Web tools, which are the URL-based applications to be integrated
- ▶ Command tools, which are the shell executable-based applications to be integrated

The Web tool acts exactly the same way as in the previous Web-based System Manager release.

Figure 6-18 on page 343 shows the command tool creation.

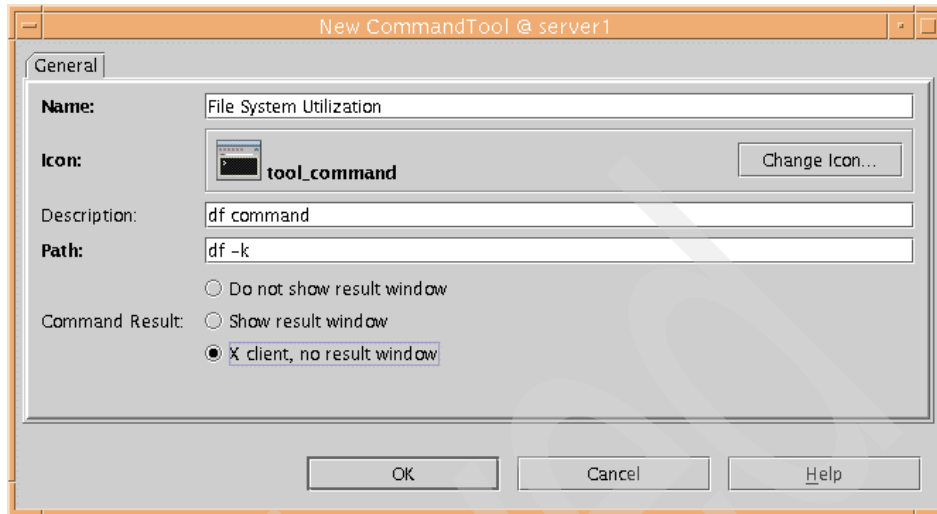


Figure 6-18 Command tool creation dialog

The command tool is a new option that allows you to integrate virtually any command line executable into the Web-based System Manager. To create a command tool, you need to specify the name of the tool (a default icon is provided, but you can specify an alternate icon in GIF format), an optional description of the tool, the complete path to the command, and a chosen result type. The result type can be one of the following:

- | | |
|-------------------------------------|---|
| Do not show the result panel | Executes the command, but will not display the results of this command. |
| Show result panel | Opens a new panel with output generated by the specified command. |
| X client, no result panel | The tool is an X client application. It will display its own GUI interface as the result panel. |

Figure 6-19 on page 344 shows the sample output of a command tool that chose show result panel as the result type.

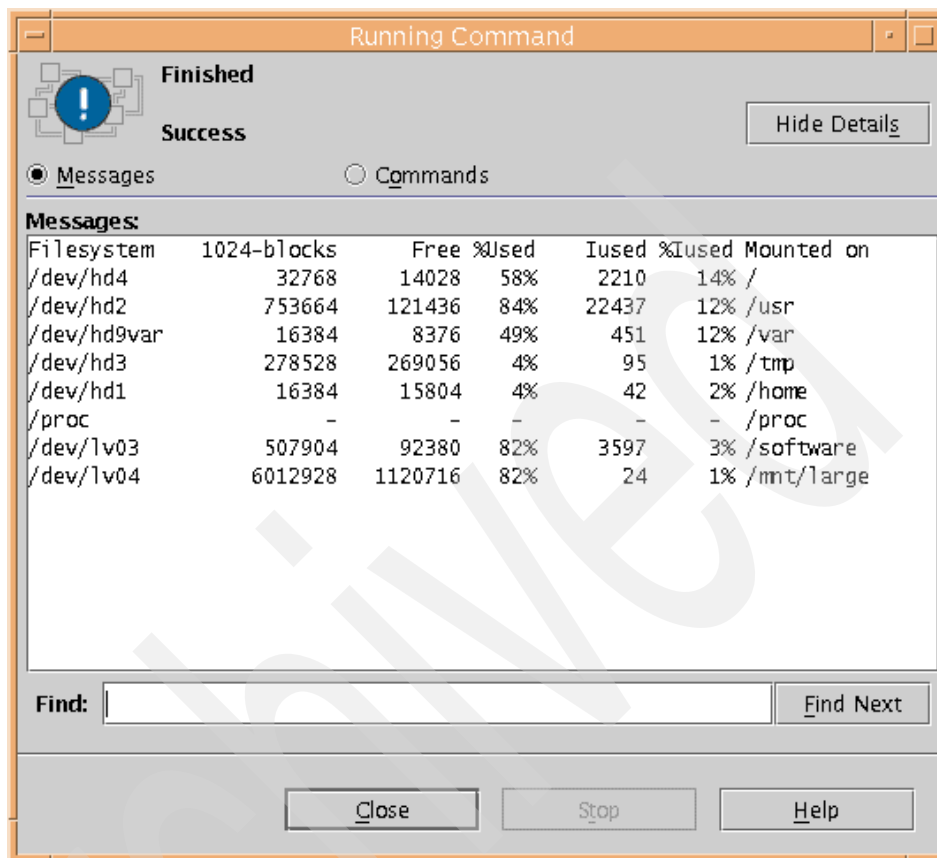


Figure 6-19 Example of result type Show result panel

Tips area

Any container that you select on the Navigation Area will bring you tips on the related topic if Show Tips Bar is enabled. To enable it, you need to select **View** in the menu bar and then **Show**, and **Enable Tips Bar**.

Figure 6-20 shows an example of a tip.

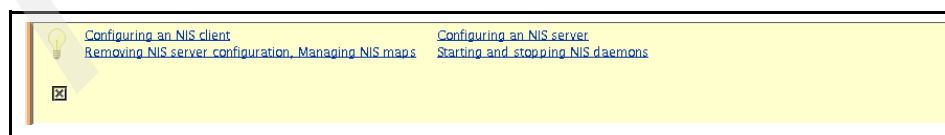


Figure 6-20 Tips bar example

Preferences

In the AIX 5L release of the Web-based System Manager, it is possible to have a customized environment for any user in any machine for the Web-based System Manager. This can be done through the new control for preferences.

When the Web-based System Manager is started, the session uses the stored preferences. This includes such preferences as the console panel format and the machines being managed. By default, the preference file is saved to \$HOME/WebSM.pref, which is the user's home directory on the managing machine.

To save the state of the console without closing a session, use the menu option Console, and then Save. A user is always prompted to save the console state when closing Web-based System Manager.

Table 6-3 shows which components are saved in the preferences file.

Table 6-3 Components that are saved in the preferences file

Component	Status saved in preferences file?
Navigation Area	No
Tool bar	Yes
Tips bar	Yes
Description bar	Yes
Status bar	Yes

SNMP integration

AIX 5L provides the SNMP interface for the Web-based System Manager framework for use by applications that need to do monitoring; it also provides overview query enhancements to Network applications.

Figure 6-21 on page 346 shows the panel for the SNMP monitor configuration.

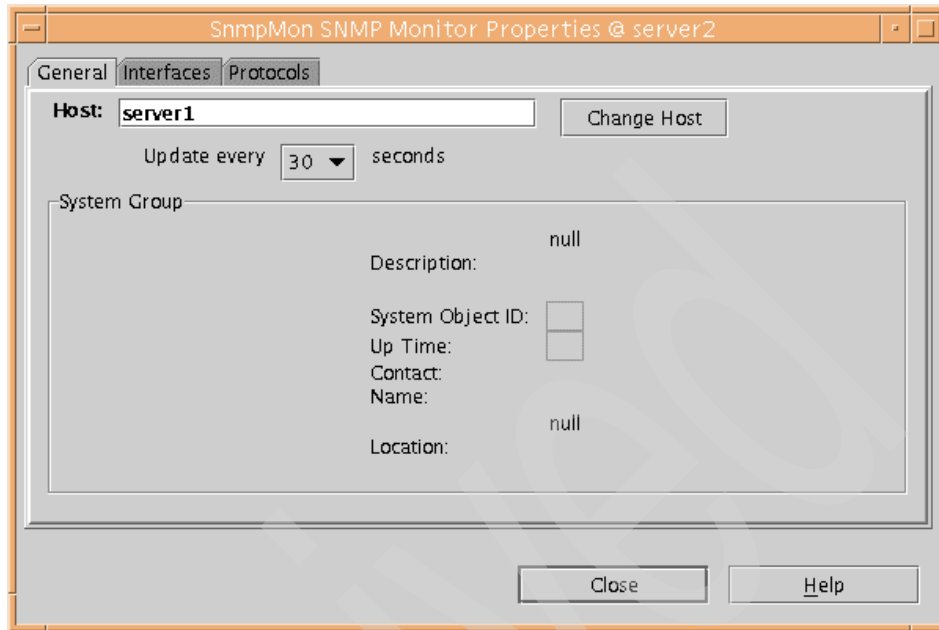


Figure 6-21 SNMP monitor configuration through Web-based System Manager

Enterprise management framework integration

In AIX 5L, there is a new way to launch the Web-based System Manager: It can be context launchable from the tool palette and tool menu from Tivoli NetView NT and AIX.

In environments that already have the Tivoli NetView server running, AIX 5L servers can be easily integrated and remotely managed through any Tivoli Netview servers launching the Web-based System Manager.

6.2.3 Web-based System Manager PC Client (5.1.0)

Web-based System Manager PC Client provides an installable application for the Windows PC Client. The Web-based System Manager console is provided for clients on Windows NT, Windows 2000, and Windows Me.

The Web-based System Manager console running on a PC will provide remote system administration support for AIX 32-bit and 64-bit systems.

Configuring the managed machine

In order to support the Web-based System Manager PC Client, the server must have the following software installed:

- ▶ IHS 1.3.12
- ▶ Java 1.3
- ▶ Web-based System Manager 5.1
- ▶ bos.net.*

The applet mode is configured using the IBM HTTP Server (IHS), using the **configassist** command (/usr/bin/configassist). This script will create all necessary links in the /usr/HTTPServer/htdocs. Running this script will prompt you with the configuration assistant task, as shown in Figure 6-22.

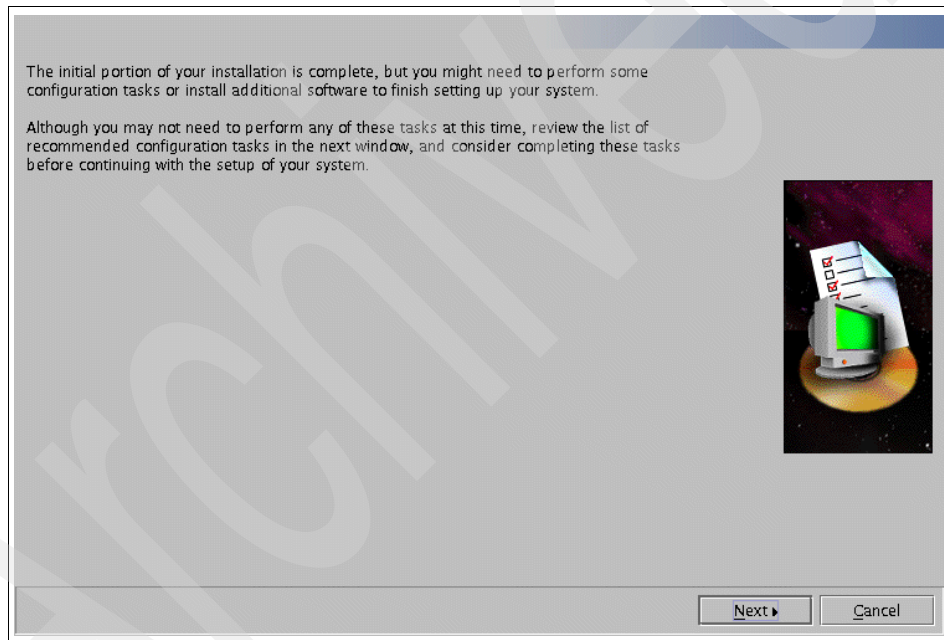


Figure 6-22 Configassist: Configuration task manager

Choose the option **Configure a Web server** to run Web-based System Manager in a browser, as shown in Figure 6-23 on page 348.

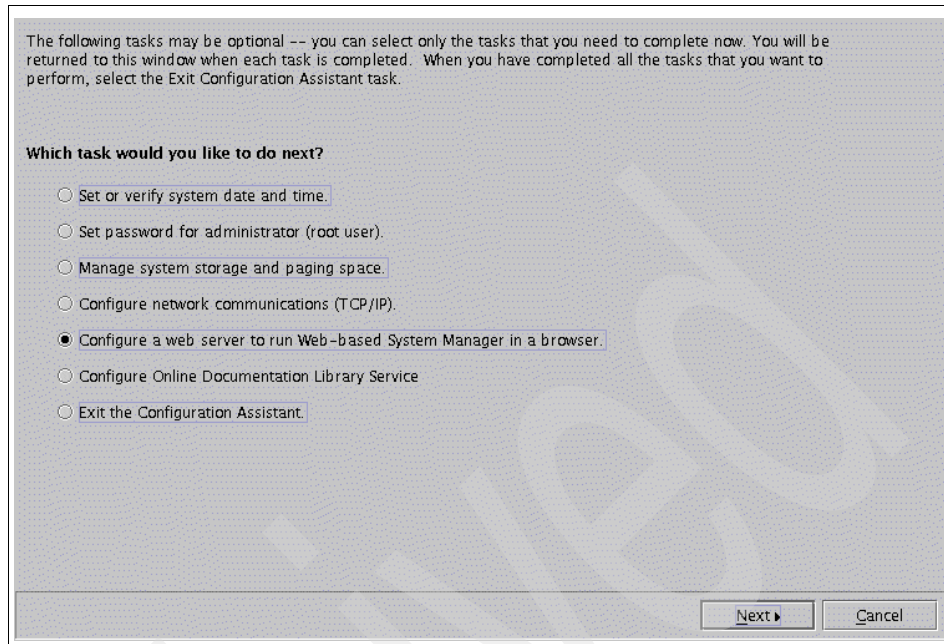


Figure 6-23 Web server to run Web-based System Manager in a browser

You will have the option of which Web browser you want to use, as shown in Figure 6-24 on page 349.

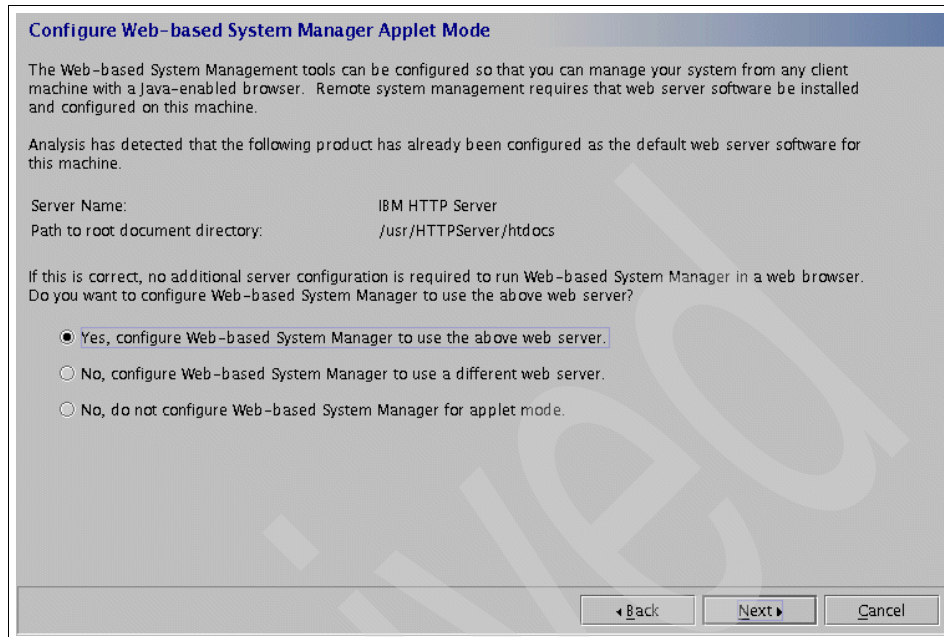


Figure 6-24 Configure Web-based System Manager Applet mode

You can exit the configuration assistant by selecting **Exit the Configuration Assistant**.

Set the default browser depending on what browser you are using on your PC. The default browser can be set through **SMIT -> System Environments, Internet and Documentation Services -> Change/Show Default Browser**.

Configuring Web-based System Manager PC Client

In order to configure the Web-based System Manager PC Client, you need around 35 MB or free disk space on your PC. Start your browser and go to `http://configured_mm/pc_client/setup.htm`, with `configured_mm` being your AIX server name. The InstallShield Multi-Platform will lead you through the setup of your Web-based System Manager PC Client, as shown in Figure 6-25 on page 350 and Figure 6-26 on page 350.

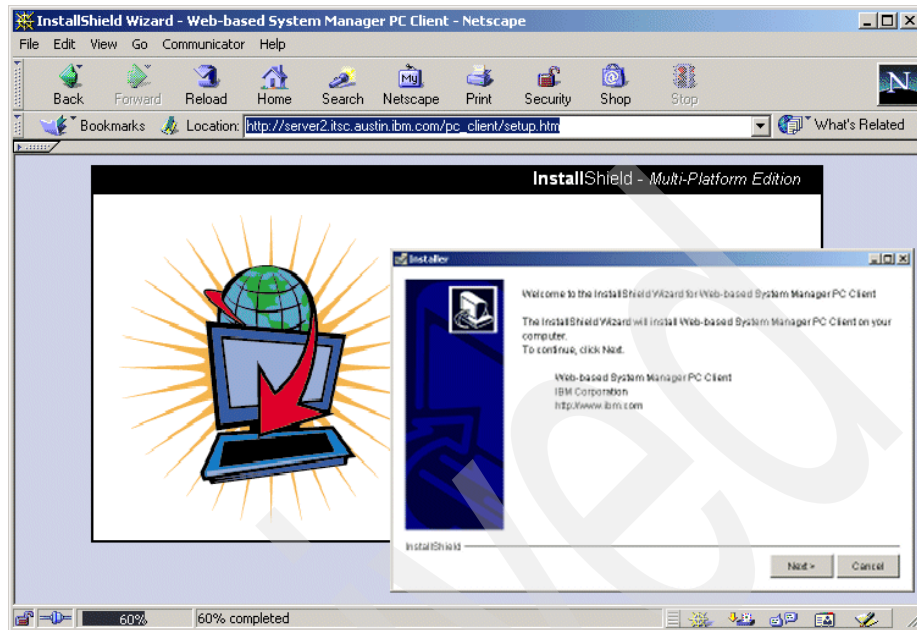


Figure 6-25 InstallShield Multi-Platform for PC Client

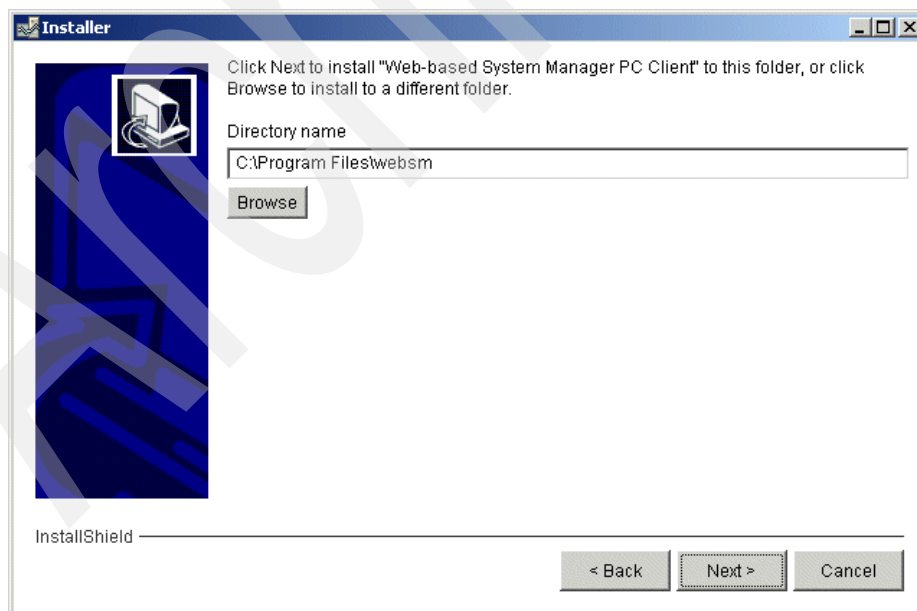


Figure 6-26 Installation of Web-based System manager PC Client

When the installation is finished, you can launch the Web-based system Manager PC Client through **Start -> Programs -> Web-based System Manager PC Client**. You will receive a login screen, as shown in Figure 6-27.

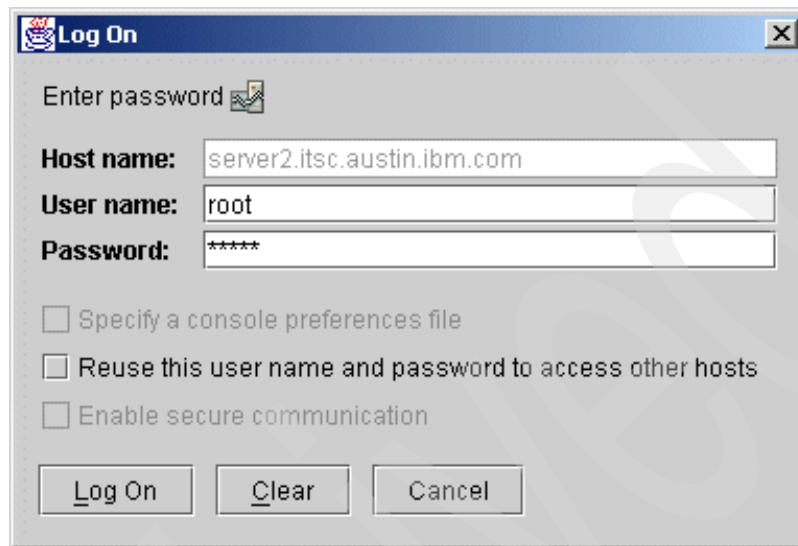


Figure 6-27 Log On screen for Web-based System Manager PC Client

Once you are logged in, Web-based System Manager will run and you are able to manage your AIX operating system from your PC, as shown in Figure 6-28 on page 352.

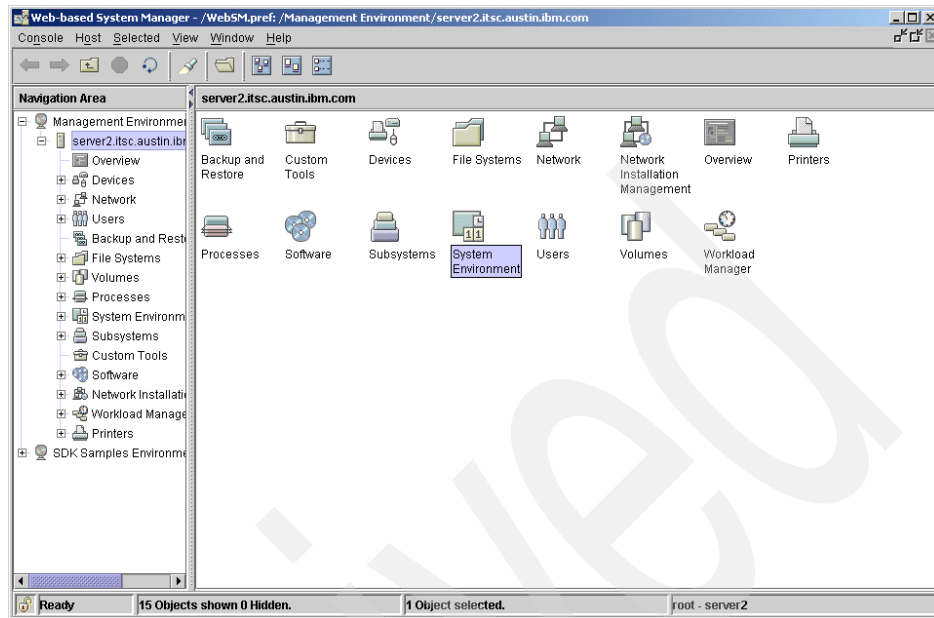


Figure 6-28 Web-based System Manager PC Client

6.2.4 Web-based System Manager Client for Linux (5.2.0)

Support has been added to the Web-based System Manager Client for the Linux platform. Since the Web-based System Manager is a platform-independent Java application, the Linux client is identical to the Web-based System Manager Client for Windows. It is supported on Red Hat 7.2 or Red Hat 7.3 Linux. It allows you to remotely manage AIX and HMC systems.

In the following sections, step-by-step instructions are given that enable you to quickly get the Web-based System Manager Client running on Linux.

To install the Web-based System Manager Client for Linux over the network, an AIX 5L Version 5.2 system needs to be configured with a Web server. After the Web server is properly set up, the installation will be started from a Web browser on the Linux system. This is done by the following steps:

1. Run the `ls1pp -L sysmgt.websm.webaccess` command to verify that `sysmgt.websm.webaccess` is installed. If it is installed the fileset will be listed.
2. Install IBM HTTP-Server (IHS) from the Expansion Pack CD with the `installp -acY -d /dev/cd0 http_server.base` command.

3. Run the `/usr/bin/configassist` command and select the task Configure a Web server to run Web-based System Manager in a browser, click **Next**, accept the default values on the next dialog, and click **Next** again.
4. On the Red Hat Linux system launch a Web browser and connect to the previously configured Web server by specifying the fully qualified domain name in the following URL:
`http://server2/remote_client.html`
5. On the Web page click the Linux link and save the `wsmlinuxclient.exe` file in a directory of your choice, for example, `/root`.
6. On the Linux command line run the following commands:

```
cd /root
chmod +x wsmlinuxclient.exe
/root/wsmlinuxclient.exe
```
7. Start Web-based System Manager with the `wsm` command.

6.2.5 Accessibility for Web-based System Manager

Because the Web-based System Manager in AIX 5L is using Java 2 Standard Edition 1.3, or more specifically the Java Foundation Classes, which are a default part of this version, you can now operate most of the panels, menus, screen controls, and dialogs without using a mouse or other pointing device.

Limited mobility users will welcome this function as well as any experienced administrator.

Two accessibility features are provided by default: Mnemonics and accelerators. Mnemonics allow you to execute a certain action on a visible dialog without pressing the space bar or Enter key by simultaneously holding down the Alt key and the underlined letter designated in the label belonging to the desired action. Accelerators, on the other hand, are always available, even if the dialog or menu panel with the accompanying action is not visible. These accelerators or shortcuts are usually a combination of the Ctrl, Alt, or Shift key, or a combination of these with a regular letter key or special keys (such as Tab or function keys).

A Keys Help provides a complete list of navigation and windowing keys, and the mnemonics and accelerators for menus are shown in the user interface.

Figure 6-29 on page 354 shows an example for the mnemonic key. In this example, pressing Alt+R selects the entry Remotely with **rlogin** and **telnet** commands in the Enable login group, regardless of where the cursor is currently located. The Ctrl+Q key shortcut exits the Web-based System Manager, independent of which dialog is currently active.

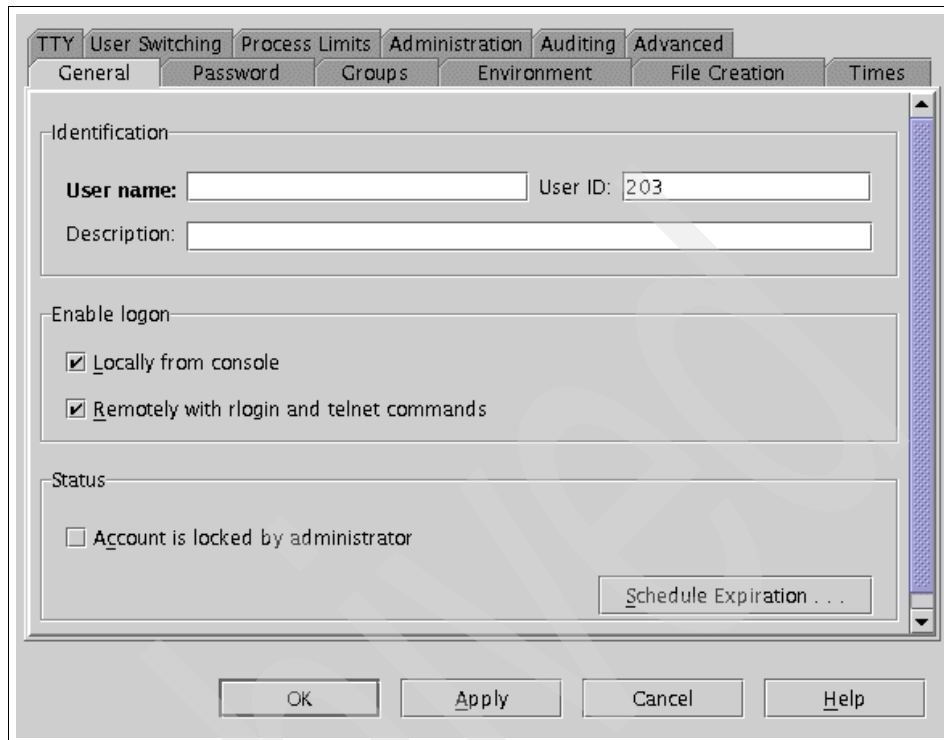


Figure 6-29 Accessibility example

6.3 Documentation search-engine enhancement

The Documentation Library Service in AIX 5L uses a new search engine. The Text Search Engine (TSE) is replacing the NetQuestion Version 1.2.3 (IMNSearch) that was presented in AIX Version 4.3.3.

Some of the enhancements of the Text Search Engine over NetQuestion include:

- ▶ Use of a single search engine for both single byte or double byte character sets, instead of one engine for each type of character.
- ▶ The Text Search Engine does not need a writeable index file, so you can have the Documentation CD-ROM mounted and do all the searches through the mounted CD-ROM without file write permission problems.
- ▶ The new Text Search Engine supports Russian Language through the ISO-8859-5 Russian codeset.
- ▶ The Text Search Engine is installed by default with the AIX base installation unless Minimal Install is used.

The Text Search Engine provides binary compatibility, and can read all NetQuestion search indexes. From a migration path point of view, AIX Version 4.3 machines will be able to upgrade to this new version without problems. However, rebuilding old user-created documents using the new engine will significantly improve search performance.

6.4 Information Center (5.2.0)

The IBM @server pSeries Information Center is a Web site that serves as a focal point for all information pertaining to pSeries and AIX (Figure 6-30 on page 356). It provides a link to the entire pSeries library. In addition, it provides access to the AIX Versions 4.3, 5.1, and 5.2 documentation. A message database is available to search error numbers, identifiers, and LEDs. FAQs, How-To's, a troubleshooting guide, and many more features are provided.

To access the Information Center you have three options:

- ▶ Open the URL:
http://publib16.boulder.ibm.com/pseries/en_US/infocenter/base
- ▶ Run the **infocenter** command from the command line. This command starts the default browser with the URL previously mentioned.
- ▶ Start the Information Center with the Information Center icon located on the Help panel of the CDE desktop.



Figure 6-30 Information Center

6.4.1 AIX online message database

For system administrators, application developers, or service personnel of all skill levels and experience who are troubleshooting error messages, a new message database is implemented on an IBM Web site. This database can be accessed using a browser and contains the seven-digit error messages for AIX 5L Version 5.2 and also includes other types of error messages such as LEDs, error identifiers, trace hood IDs, and more. The message database will be updated on a regular basis and we encourage customers to provide feedback on current message information and ask for additional information or provide tips on messages that they have received and worked through.

Figure 6-31 shows the main panel of the AIX message database. It is part of the new online Information Center.

Figure 6-31 View of search interface of the AIX message database

6.5 Software license agreement enhancements (5.1.0)

AIX 5L Version 5.1 has been enhanced to handle electronic software license agreements. There are new features to administer license agreements and associated documents. Information about all available license agreements on the system is kept in the `/usr/lib/objrepos/lag` agreement database file. The agreement database only includes license agreement information and no information about usage licenses such as administered by LUM. The agreement text itself is stored in the `/usr/swlag/locale` directory. The license agreement database is designed so that license information from non-IBM installation programs can be integrated.

The content of a license agreement file might appear similar to the following:

```
# more /usr/swlag/en_US/BOS.li
International Program License Agreement
```

Part 1 - General Terms

PLEASE READ THIS AGREEMENT CAREFULLY BEFORE USING THE PROGRAM. IBM WILL LICENSE THE PROGRAM TO YOU ONLY IF YOU FIRST ACCEPT THE TERMS OF THIS AGREEMENT. BY USING THE PROGRAM YOU AGREE TO THESE TERMS. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, PROMPTLY RETURN THE UNUSED PROGRAM TO THE PARTY (EITHER IBM OR ITS RESELLER) FROM WHOM YOU ACQUIRED IT TO RECEIVE A REFUND OF THE AMOUNT YOU PAID.

The Program is owned by International Business Machines Corporation or one of its subsidiaries (IBM) or an IBM supplier, and is copyrighted and licensed, not sold.

The term "Program" means the original program and all whole or partial copies of it. A Program consists of machine- readable instructions, its components, data, audio-visual content (such as images, text, recordings, or pictures), and related licensed materials.

This Agreement includes Part 1 - General Terms, Part 2 - Country-unique Terms, and "License Information" and is the complete agreement regarding the use of this Program, and replaces any prior oral or written communications between you and IBM. The terms of Part 2 and License Information may replace or modify those of Part 1.

1. License

Use of the Program

IBM grants you a nonexclusive license to use the Program You may 1) use the Program to the extent of authorizations you have acquired and 2) make and install copies to support the level of use authorized, providing you reproduce the copyright notice and any other legends of ownership on each copy, or partial copy, of the Program.

If you acquire this Program as a program upgrade, your authorization to use the Program from which you upgraded is terminated.

You will ensure that anyone who uses the Program does so only in compliance with the terms of this Agreement.

You may not 1) use, copy, modify, or distribute the Program except as provided in this Agreement; 2) reverse assemble, reverse compile, or otherwise translate the Program except as specifically permitted by law without the possibility of contractual waiver; or 3) sublicense, rent, or lease the Program.

6.5.1 The **inulag** command

The **inulag** command is a frontend to the subroutines to manage license agreements. Options other than listing the contents of the database can only be done by root, since the agreement database is writable only by root. The **inulag** command has several flags; for detailed information, see the man pages or the online documentation.

The -l flag, for example, lists all available software license agreements:

```
# inulag -l
=====
                        Installed License Agreements
=====

The installed software listed below contains license agreements
which have been accepted.

-----
-----
Fileset:  bos.rte
Product ID:
Description:
Agreement File:  /usr/swlag/en_US/BOS.li
Date:  Tue Feb 27 10:25:43 CST 2001
Machine ID:  000BC6FD4C00
```

6.5.2 The installp command enhancements

The **installp** command has been modified to recognize, display, require, and log software license agreements. The -E flag has been added to display software license agreements. The -Y flag is used to agree to the required software license agreements for software to be installed. For further or more detailed information, refer to the man pages or online documentation.

Using SMIT

The SMIT install panels have been enhanced with two new fields to handle the software license agreements, as shown in Figure 6-32 on page 360.

Install Software			
Type or select values in entry fields. Press Enter AFTER making all desired changes.			
* INPUT device / directory for software	[Entry Fields]		
* SOFTWARE to install	/dev/cd0		
PREVIEW only? (install operation will NOT occur)	[_all_latest]	+	
COMMIT software updates?	no	+	
SAVE replaced files?	yes	+	
AUTOMATICALLY install requisite software?	no	+	
EXTEND file systems if space needed?	yes	+	
OVERWRITE same or newer versions?	no	+	
VERIFY install and check file sizes?	yes	+	
Include corresponding LANGUAGE filesets?	no	+	
DETAILED output?	yes	+	
Process multiple volumes?	no	+	
ACCEPT new license agreements?	yes	+	
Preview new LICENSE agreements?	no	+	
F1=Help	F2=Refresh	F3=Cancel	F4=List
F5=Reset	F6=Command	F7=Edit	F8=Image
F9=Shell	F10=Exit	Enter=Do	

Figure 6-32 SMIT panel for accepting new software agreements using installp

Two of the fields shown are as follows:

► ACCEPT new license agreements?

If this field set to yes, the -Y flag is added to the **installp** command. If the value is no, the installation will fail.

► Preview new LICENSE agreements?

If yes, the -p and -E flags are added to the **installp** command. This results in an installation preview only.

6.5.3 The ls1pp command enhancements

The **ls1pp** command has also been enhanced to display the license agreement information of the installed filesets.

If the -E option is specified with the `ls1pp` command, then the arguments will simply be passed through to `inulag -l` with an -n fileset argument for each fileset argument passed in, as shown in the following example:

```
# ls1pp -E bos.rte
=====
                        Installed License Agreements
=====

The installed software listed below contains license agreements
which have been accepted.

-----
-----
Fileset: bos.rte
Product ID:
Description:
Agreement File: /usr/swlag/en_US/BOS.li
Date: Tue Feb 27 10:25:43 CST 2001
Machine ID: 000BC6FD4C00
```

6.5.4 Additional information in the bosinst.data file

The bosinst.data file contains a new field named ACCEP_LICENSES. If the field is set to no, you have to accept all licenses after the first reboot. If ACCEP_LICENSES is set to yes, you will not be prompted after a new installation.

The following is an extract from the bosinst.data file:

```
# FORCECOPY = no, yes
# ALWAYS_ALLOW = no, yes
control_flow:
  CONSOLE = /dev/tty0
  INSTALL_METHOD = migrate
  PROMPT = no
  EXISTING_SYSTEM_OVERWRITE = yes
  INSTALL_X_IF_ADAPTER = yes
  RUN_STARTUP = yes
  RM_INST_ROOTS = no
  ERROR_EXIT =
  CUSTOMIZATION_FILE =
  TCB = no
  INSTALL_TYPE =
  BUNDLES =
  SWITCH_TO_PRODUCT_TAPE =
  RECOVER_DEVICES = yes
  BOSINST_DEBUG = no
  ACCEPT_LICENSES = no
```

```
INSTALL_64BIT_KERNEL = no
INSTALL_CONFIGURATION = Default

target_disk_data:
  PVID = 000bc6fdbff92812
  CONNECTION = scsi0//8,0
  LOCATION = 10-60-00-8,0
  SIZE_MB = 8678
  HDISKNAME = hdisk0
```

6.5.5 System installation (BOS install)

Installing or migrating to AIX 5L Version 5.1 when not using a **mksysb** backup or SPOT copy will cause you to always accept the software license agreements. See 6.5.6, “Accepting licenses after reboot” on page 362, for more information.

In the case of an installation from a **mksysb** backup or a SPOT copy, then the ACCEPT_LICENSES value will dictate whether you have to accept the license agreements manually. If ACCEPT_LICENSES=yes, then **inu1ag -A** will be invoked to accept the license agreements automatically. If ACCEPT_LICENSES=no, then **inu1ag -D** will be invoked to revalidate all license agreements. In that case you have to accept all agreements by the next system reboot. If ACCEPT_LICENSES was not set or set to some other value, then no **inu1ag** operation will take place.

6.5.6 Accepting licenses after reboot

After a migration to AIX 5L Version 5.1 or a new install, you have to accept all software license agreements, as shown in Figure 6-33 on page 363. If not, you probably used a **mksysb** or NIM Install, while the ACCEP_LICENSES stanza in the bosinst.data file was set to yes.

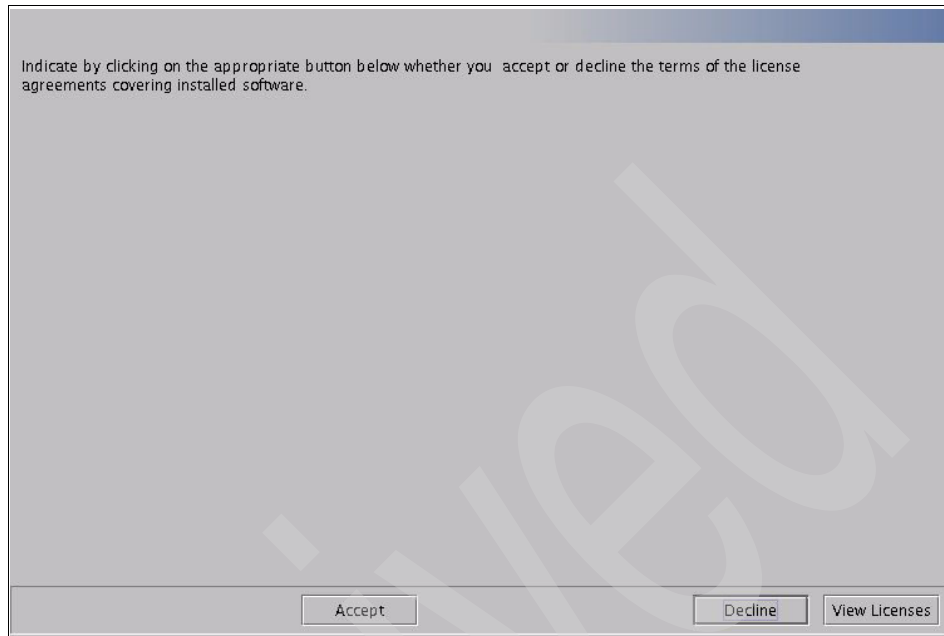


Figure 6-33 Configuration assistant, software license after reboot

Click the **View License** button to show all outstanding licenses or just click the **Accept** button to accept all licenses at once.

6.5.7 SMIT function enhanced

SMIT screens have been added to display the content of the license agreement database, as shown in Figure 6-34 on page 364.

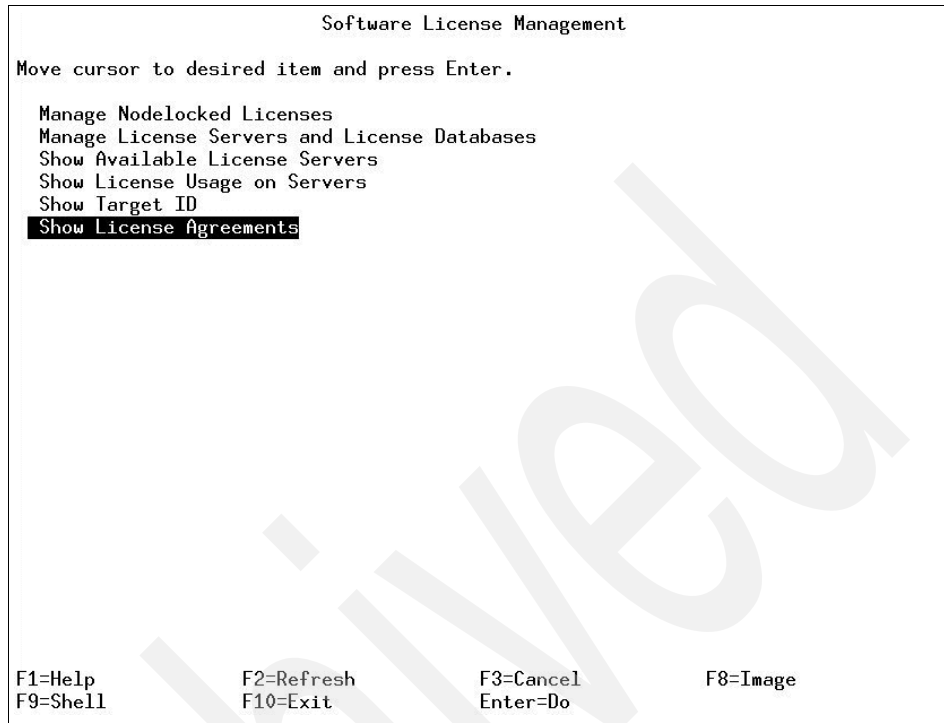


Figure 6-34 SMIT panel for license management

6.5.8 lslicense and chlicense enhancement (5.2.0)

The **lslicense** command has a new **-A** flag that shows how many available fixed licences you have currently on the system.

With the **chlicense** it is now possible to change the fixed license number without rebooting using the **-I** flag.

In the following example the fixed licenses are updated from four to 100 without reboot.

```
# lslicense -A
Maximum number of fixed licenses is 4.
Floating licensing is disabled.
Number of available fixed licenses is 3.
# chlicense -I -u100
# lslicense -A
Maximum number of fixed licenses is 100.
Floating licensing is disabled.
Number of available fixed licenses is 99
```

An enhancement to Web-based System Manager has also been made to change the system default, as shown in Figure 6-35.

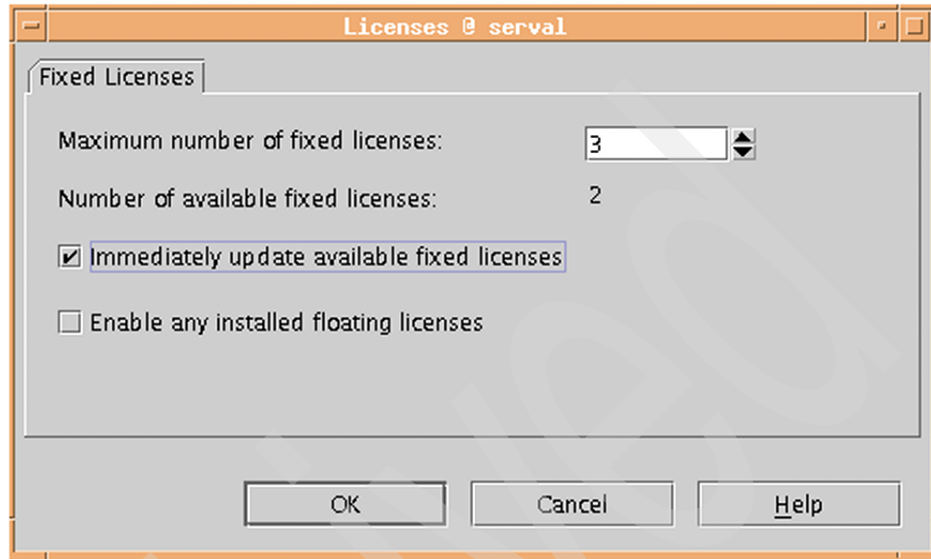


Figure 6-35 Licenses Web-based System Manager dialog

6.6 The bffcreate and lppmgr enhancement (5.2.0)

Before AIX 5L Version 5.2, when a system administrator had to download selective fixes and make an image to disk, the file names of those fixes are the name of the PTF. The **bffcreate** command has two new flags, namely the **-c** and the **-s** flags, that can be used to rename the PTF image file to the corresponding fileset names. A SMIT panel has also been enhanced to handle this new functionality. The following example shows how to rename the files using the SMIT panel, providing the following assumptions.

The following is an image file of four PTF files:

```
root@server1:/stuff/fix # ls -l
total 11488
-rw-r--r--  1 root    sys      729088 Sep 13 12:27 U476304
-rw-r--r--  1 root    sys      569344 Sep 13 12:27 U476306
-rw-r--r--  1 root    sys      4581376 Sep 13 12:28 U476314
root@server1:/stuff/fix #
```

To rename those files open the SMIT dialog (shown in Figure 6-36 on page 366) with the **SMIT maintain_software** command.

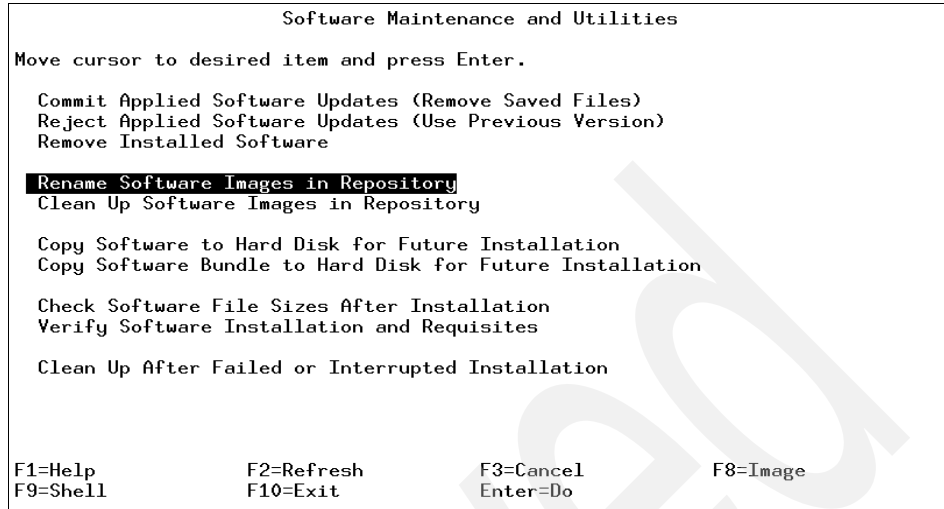


Figure 6-36 SMIT software maintenance and utilities panel

The rEname Software Images in Repository panel is shown in Figure 6-37.

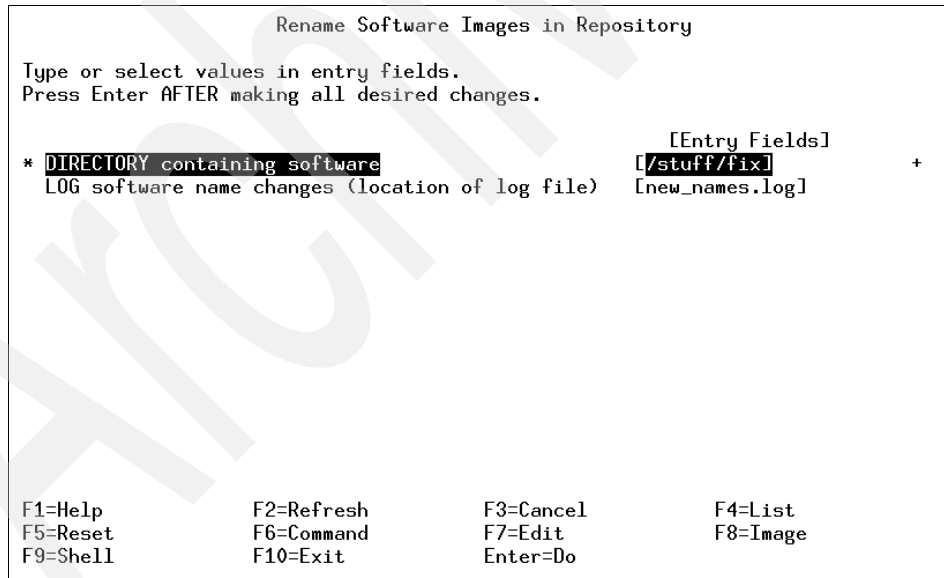


Figure 6-37 Rename software image repository

The files are renamed with the corresponding fileset name and level, which are more useful for the system administrator. The following lists the renamed files:

```
root@server1:/stuff/fix # ls -l
```

```
total 11520
-rw-r--r-- 1 root sys 9555 Sep 13 13:39 .toc
-rw-r--r-- 1 root sys 4581376 Sep 13 12:28 bos.mp.5.1.0.10.U
-rw-r--r-- 1 root sys 569344 Sep 13 12:27
bos.perf.tools.5.1.0.10.U
-rw-r--r-- 1 root sys 95 Sep 13 13:39 new_names.log
-rw-r--r-- 1 root sys 729088 Sep 13 12:27
perfagent.tools.5.1.0.10.U
root@server1:/stuff/fix #
```

In the previous example, a new file named `new_names.log` is created and contains the equivalence between the old file names and the new file names.

The **gencopy** and **bffcreate** commands now accept fileset names as well as package names for base images copy, providing the ability for the administrator to specify base image fileset names to copy to **bffcreate** or **gencopy**.

Another enhancement that helps the system administrator manage the maintenance of the system is the `/usr/lib/inst1/lppmgr` command. This command allows the system administrator to clean up software images in a directory that contains software for future installations by reducing the amount of space required to store them. The functions are the following:

- ▶ Remove duplicate updates.
- ▶ Remove duplicate base levels.
- ▶ Eliminate updates that are the same level as bases of the same fileset. These updates can create conflicts that lead to installation failure.
- ▶ Remove message and locale filesets other than the language you specify.
- ▶ Remove superseded filesets.
- ▶ Remove non-system images from a NIM lpp_source resource.

The syntax of the **lppmgr** command is as follows:

```
lppmgr -d DirectoryOrDevice [ -r | -m MoveDirectory ] { [ -x ] [ -X ] [ -l ]
[ -u ] [ -b ] [ -k LANG ] } [ -p ] [ -t ] [ -s ] [ -V ] [ -D ]
```

The most common flags are shown in Table 6-4.

Table 6-4 Most common flags of the lppmgr command

Flag	Description
-X	Remove system image for NIM lpp_source.
-u	Remove duplicate updates or updates which are the same level as bases of the same fileset.

Flag	Description
-b	Remove duplicate level.
-k	Remove extra languages and locale files.
-x	Remove superseded files.

The SMIT panels are also enhanced to support this enhancement. Figure 6-38 shows how to remove all of the languages except the en_US language, to get a significant gain of place on disk.

```

Clean Up Software Images in Repository

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                [Entry Fields]
* DIRECTORY containing software    [/usr/sys/inst.images]
PREVIEW only? (remove operation will NOT occur)  no      +
REMOVE DUPLICATE software          yes      +
REMOVE SUPERSEDED updates          yes      +
REMOVE LANGUAGE software           yes      +
PRESERVE language                  [en_US]
SAVE removed files                  no      +
    DIRECTORY for storing saved files  []
EXTEND file systems if space needed? yes      +

F1=Help      F2=Refresh      F3=Cancel      F4=List
F5=Reset     F6=Command     F7=Edit       F8=Image
F9=Shell     F10=Exit       Enter=Do

```

Figure 6-38 SMIT Clean Up Software Images in Repository panel

6.7 Comparison reports for LPPs (5.2.0)

Comparison reports are an easy way for you to manage the level of your systems regarding fixes and maintenance levels. It is possible to compare levels of different systems against a base system or a set of fixes. The comparison reports feature provides functionality, both through the **compare_report** command line and the SMIT menus, which allows you to compare the filesets installed on a system with the contents of an image repository or a service report that may be downloaded from the IBM support Web site.

The **compare_report** command generates comparison reports that will compare:

- ▶ Filesets installed on a system
- ▶ Filesets contained in a repository
- ▶ Filesets available from the IBM support Web site, both latest fix and maintenance levels

The different combinations that the **compare_report** command can handle are the following:

- ▶ The filesets installed on a system compared to filesets contained in a repository. Four lists can be generated:
 - A list of filesets on the system that are downlevel
 - Filesets in the image repository that are not installed on the system
 - A list of filesets on the system that are uplevel
 - Filesets installed on the system that are not in the image repository
- ▶ To compare the filesets installed on a system to the filesets available from the IBM support Web site. Three lists can be generated:
 - A list of filesets on the system that are downlevel from the latest levels available from the IBM support Web site.
 - A list of filesets on the system that are uplevel from the maintenance level available from the IBM support Web site.
 - A list of filesets on the system that are downlevel from the maintenance level available from the IBM support Web site.
- ▶ To compare the filesets contained in a repository to the filesets available from the IBM support Web site. One list can be generated: A list of filesets in the local image repository that are downlevel from the latest levels available from the IBM support Web site (filesets available from the IBM support Web site that are not in the image repository will be included).
- ▶ To compare the list of installed software (base system) to the list of installed software (other system). Four lists can be generated:
 - A list of base system-installed software that is at a lower level
 - Filesets not installed on the base system, but installed on the other system
 - A list of base system-installed software that is at a higher level
 - Filesets installed on the base system that are not installed on the other system

The following example shows a comparison between two systems. The two lists of the LPP have been produced by the **ls1pp -Lc** command and put into two files: The `complist.org` file (the base level) and the `complist` file (other system). As

follows, the comparison shows the base-installed LPPs that are at a lower level than the other system:

```
root@server1:/tmp # compare_report -b complist.org -o complist -l
#(baselower.rpt)
#Base System Installed Software that is at a lower level
#Fileset_Name:Base_Level:Other_Level
bos.docsearch.rte:5.2.0.0:5.3.0.0

root@server1:/tmp #
```

A SMIT panel has also been updated. To compare the filesets installed on your system to a fixed directory, run the **SMIT compare_report** command as shown in Figure 6-39

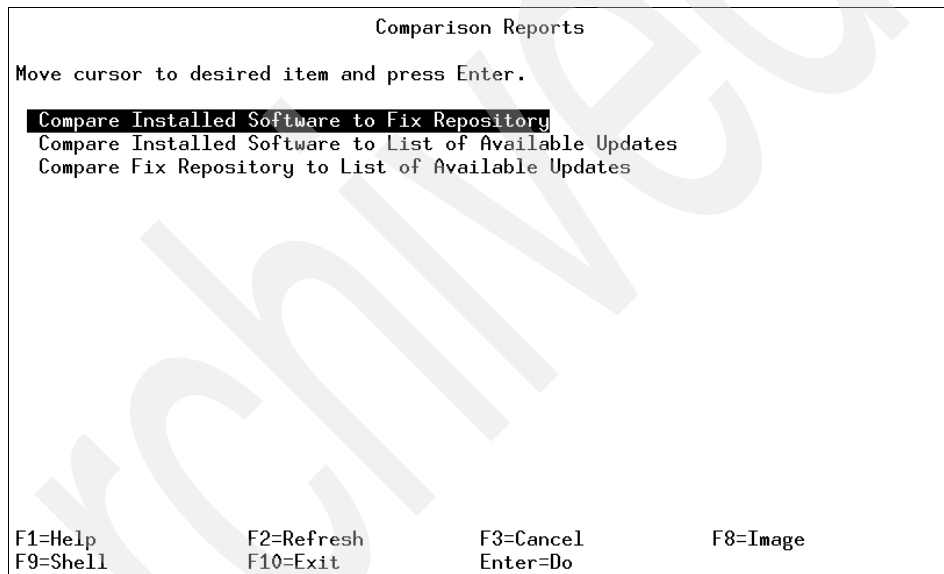


Figure 6-39 SMIT Comparison Reports panel

Then the Compare Installed Software to Fix Directory panel is selected, as shown in Figure 6-40 on page 371.


```

Compare Installed Software to Fix Repository

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]
* FIX REPOSITORY location [Entry Fields] [/stuff/0234A_520]

Select which reports to run.

Installed Software that is at a LOWER level (lowerlevel.rpt) yes +
Installed Software that is at a HIGHER level (higherlevel.rpt) yes +
Updates for filesets that are NOT INSTALLED (notininstalled.rpt) yes +
Installed Software with NO UPDATES found (no_update_found.rpt) yes +

[MORE...2]

F1=Help      F2=Refresh  F3=Cancel   F4=List
F5=Reset     F6=Command  F7=Edit    F8=Image
F9=Shell     F10=Exit   Enter=Do

```

Figure 6-40 SMIT Compare Installed Software to Fix Repository panel

Figure 6-41 shows the SMIT compare report result.

```

COMMAND STATUS

Command: OK          stdout: yes          stderr: no
TOP]
/usr/sbin/compare_report -s -i /stuff/0234A_520 -l -h -n -m -t /tmp

#(lowerlevel.rpt)
#Installed Software that is at a LOWER level
#
# Fileset Name                      Installed Level  Fix Level
#-----
#  xlsmp.rte                        1.3.4.0         1.3.6
#(notininstalled.rpt)
#Updates for filesets that are NOT INSTALLED.4.0
#
#(notinsta3led.rpt)
#Updates for filesets that are NOT INSTALLED
#
[MORE...3133]

F1=Help      F2=Refresh  F3=Cancel   F6=Command
F8=Image     F9=Shell    F10=Exit    /=Find
n=Find Next

```

Figure 6-41 SMIT Compare Installed Software to Fix Repository panel results

6.8 mksysb on CD or DVD (5.1.0)

CD (CD-R, CD-RW) and DVD (DVD-R, DVD-RAM) are devices supported as mksysb media on AIX 5L Version 5.1. As described in the following section, there are three types of CDs (the use of the term CD in this chapter will also imply DVD) that can be created:

- ▶ Personal system backup
- ▶ Generic backup
- ▶ Non-bootable volume group backup

6.8.1 Personal system backup

A personal mksysb CD will only boot and install the system where it was created. This type of mksysb backup is the same as the mksysb backup on a tape media.

6.8.2 Generic backup

A generic backup has the following platform-related condition.

Power-based system

This type of backup CD is used to boot and install any platform (rspc, rs6k, or chrp). It contains all three boot images and the device and kernel filesets to enable cloning. The bos.mp fileset will be automatically installed because the MP kernel is required to support booting both UP and MP systems. The MP kernel will not be made the running kernel if the system is a UP system. All device filesets will also be automatically installed for creation of CD file systems that support booting and installation on any system.

6.8.3 Non-bootable volume group backup

This type of backup CD is non-bootable and contains only a volume group image. If the image in the CD is a rootvg image, the CD can be used to install AIX after booting from a product CD-ROM. This CD can also be used as a source media for the `alt_disk_install` command. The CD-R and DVDs can be used as a backup media for the non-rootvg volume group and the volume group can be restored using the `restvg` command.

6.8.4 Tested software and hardware

Because IBM does not sell or support the AIX software to create CDs, they must be obtained from independent hardware and software vendors. Table 6-5 on page 373 lists the tested software and hardware, and their combinations,

required for this feature. There are many CD-R (CD recordable), CD-RW (CD ReWritable), DVD-R (DVD Recordable) and DVD-RAM (DVD Random access) drives available. IBM tested the listed drives in Table 6-5.

Table 6-5 Required hardware and software for backup CDs

Software	Hardware
GNU & Free Software Foundation, Inc. cdrecord Version 1.8a5 mkisofs Version 1.5	Yamaha CRW4416S - CD-RW Yamaha CRW8424S - CD-RW Ricoh MP6201SE 6XR-2X - CD-R Panasonic CW-7502-B - CD-R
Jodian Systems and Software, Inc. CDWrite Version 1.3 mkcdimg Version 2.0	Yamaha CRW4416S - CD-RW Ricoh MP6201SE 6XR-2X - CD-R Panasonic CW-7502-B - CD-R
Youngminds, Inc. MakeDisc Version 1.3-Beta2	Young Minds CD Studio - CD-R
Youngminds, Inc.	Young Minds Turbo Studio - DVD-R
GNU software	Matsushita LF-D291 - DVD-RAM IBM DVD-RAM

The listed software is used in conjunction with the **mkcd** command to make backups on CD-Rs and DVDs.

For information on how to obtain the software, see the readme file maintained in /usr/lpp/bos.sysmgmt/mkcd/README.oem_cdwriters or, as HTML, in the </usr/lpp/bos.sysmgmt/mkcd/README.html> file.

Note: Only the CHRP platform supports booting from DVD. However, a DVD media backup may be created or read on any platform (RSPC, RS6K, or CHRP) using a DVD device. Also, you may boot from other devices (CD, tape, or network) on any platform and then install from the DVD provided. The boot media's boot image contains support for DVD devices.

6.9 The **mkcd** command enhancement (5.2.0)

mksysb or **savevg** images are written to CD-Rs and DVDs using the **mkcd** command. The **mkcd** command has been extended to support two different formats, the ISO9660 format and the Universal disk Format (UDF) format. The **mkcd** command requires code supplied by third-party vendors so that it can create the RockRidge file system and write the backup image to CD media. This code must be linked to **/usr/sbin/mkrr_fs** (for creating the Rock Ridge format image) and **/usr/sbin/burn_cd** (for writing to the CD-R or DVD-RAM device). For

example, if you are using Youngminds software, you will need to create the following links:

```
ln -s /usr/samples/oem_cdwriters/mkrr_fs_youngminds /usr/sbin/mkrr_fs
ln -s /usr/samples/oem_cdwriters/burn_cd_youngminds /usr/sbin/burn_cd
```

6.9.1 ISO9660 format

The process for creating a mksysb CD (ISO9660) using the **mkcd** command is:

1. If file systems or directories are not specified, they will be created by **mkcd** and removed at the end of the command (unless the **-R** or **-S** flags are used). The **mkcd** command will create the following file systems:

- /mkcd/mksysb_image

Contains a mksysb image. Enough space must be free to hold the mksysb.

- /mkcd/cd_fs

Contains CD file system structures. At least 645 MB of free space is required (up to 4.7 GB for DVD).

- /mkcd/cd_image

Contains the final CD image before writing to CD-R. At least 645 MB of free space is required (up to 4.7 GB for DVD).

The /mkcd/cd_fs and /mkcd/cd_image may be required to have 4.7 GB of free space each, depending how big the mksysb is.

Note: The /mkcd/cd_images (with an “s”) may need to be even larger than 4.7 GB or 645 MB if the **-R** or **-S** flags were specified (if it is multi-volume) because there must be sufficient space to hold each volume.

User-provided file systems or directories can be NFS mounted.

The file systems provided by the user will be checked for adequate space and an error will be given if there is not enough space. Write access will also be checked.

2. If a mksysb image is not provided, **mkcd** calls **mksysb** and stores the image in the directory specified with the **-M** flag or in /mkcd/mksysb_image.
3. The **mkcd** command creates the directory structure and copies files based on the **cdfs.required.list** and the **cdfs.optional.list** files.
4. The mksysb image is copied to the file system. It determines the current size of the CD file system at this point, so it knows how much space is available for the mksysb. If the mksysb image is larger than the remaining space, multiple

CDs are required. It uses **dd** to copy the specified number of bytes of the image to the CD file system. It then updates the volume ID in a file.

5. The **mkcd** command then calls the **mkrr_fs** command to create a RockRidge file system and places the image in the specified directory.
6. The **mkcd** command then calls the **burn_cd** command to create the CD.
7. If multiple CDs are required, the user is instructed to remove the CD and put the next one in and the process continues until the entire mksysb image is put on the CDs. Only the first CD supports system boot.

The **mkcd** command now supports the Universal disk Format (UDF) for the DVD-RAM device. The advantage of the UDF is a significant gain of disk space.

6.9.2 UDF format

The following recreates the previous example of creating a mksysb image, but using the UDF format.

The following command is used:

```
mkcd -U -d /dev/cd0 -V rootvg
```

The **mkcd** will create the **/mkcd/mksysb_image** file system to store the mksysb files. Then the files are copied directly to the UDF file system without creating the CD structures and the CD image. The space needed is only the size of the mksysb files.

After the copy, you can also modify files such as **bosinst.data**, **image.data**, or **vgname.data** directly on the media and thus there is no need to use a diskette when you restore (or if the system does not have a diskette drive).

6.9.3 Additional flags for the mkcd command

The following is a list of additional flags for the **mkcd** command.

```
mkcd -r directory | -d cd_device | -S [ -m mksysb_image | -M mksysb_target  
| -s savevg_image | -v savevg_volume_group ] [ -C cd_fs_dir ]  
[ -I cd_image_dir ] [-V cdfs_volume_group ] [ -B ] [ -p pkg_source_dir ] [ -R |  
-S ] [ -i image.data ] [ -u bosinst.data ] [ -e ] [ -P ] [ -l package_list ]  
[ -L ] [ -b bundle_file ] [ -z custom_file ] [ -D ] [ -U ] [ -Y ]
```

Table 6-6 on page 376 provides a description of the flags.

Table 6-6 Additional flags of the `mkcd` command

Flag	Description
-L	Creates large DVD-sized images in ISO9660 format. The <code>mkcd</code> command expects the media to be 4.7 GB. Smaller media can be used, but if the backup exceeds the size of the media, the backup will be bad because <code>mkcd</code> will try to write 4 GB of data to the media even if it is 2.6 GB in size.
-r <i>dir</i>	Creates a CD file system image. If the -S or -R flags are not used, then the image will be burned to CD and removed. This flag is also a fast way to create a CD file system image based on a directory structure that already exists. It does not require extra space to create the CD file system, only the CD file system image. It is an easy way to back up data to CD.
-U	Create DVD-RAM in UDF format instead of ISO9660.

The following is an example of the -r flag:

```

/# mkcd -r /home -d /dev/rmt0 -L
/# mount -o ro /dev/cd0 /mnt
/# cd /mnt
/mnt# find . -print
./guest
./guest/perfagent.tools
./guest/bos.perf
./guest/xmwl1m.010216
./guest/xmwl1m.010315
./guest/.toc
./guest/nohup.out
./guest/xmwl1m.010316
./guest/short.rec
./antony
./antony/testfile
/mnt#

```

Additional information can be found in the `/usr/lpp/bos.sysmgmt/mkcd.README.txt` file.

6.10 Enhanced restore command (5.2.0)

Version 5.2 has enhanced the `restore` command to enable file attributes to be restored without the actual file contents. File attributes, otherwise known as metadata, refer to permissions, ownerships, timestamps, and ACLs.

6.10.1 Overview

The **restore** command reads files created in the **backup** format created either in file name or file system format. Files must be restored in the same manner as they were backed up. The **restore** command determines the backup format from the archive volume header and uses either **restbyname** or the **restbyinode**, respectively.

The **restore** command with the **-Pstring** option will allow you to extract the file attributes without actually restoring data. If the file whose attributes are to be restored does not exist in the target path, then the restore action skips the file with a warning message file does not exist and continues.

The new **-P** option allows the **restore** command to extract the following attributes on the file from the backup media:

- ▶ Permissions
- ▶ Ownership
- ▶ Timestamps
- ▶ ACLs

The **restore** command is the frontend command that calls **restbyname** or **restbyinode** for byname or byinode backups. The enhancement introduces the **-P** flag. The syntax of the command is shown as follows.

- ▶ To restore file attributes archived by file name:

```
restore -P string [B d qv] [ b Number] [ s SeekNumber] [-f Device]
[ File ... ]
```

- ▶ To restore file attributes archived by file system:

```
restore -P string [ hqv] [ b Number] [ s SeekNumber] [-f Device]
[ File ... ]
```

The flags that are applicable for this command are shown in Table 6-7.

Table 6-7 Most common flags for restore with **-P** option

Flag	Description
-P string	Restores only the file attributes. This option does not restore the file contents. If the file in the archive does not exist in the target path, then the file is not created and attributes are not extracted. Restores file attributes selectively depending upon the flags specified in the string parameter.
The valid set of sub-options supported with the -P options are the following:	
-A	Restore all attributes.

Flag	Description
-a	Restore only the permissions of the file.
-o	Restore only the ownership of the file.
-t	Restore only the timestamp of the file.
-c	Restore only the ACL attributes of the file.

Examples of this command are as follows:

- ▶ Restore only the permissions of the files on the archive:
`restore -Pa -vf backup.bak`
- ▶ Restore only the ACL attributes of the files on the archive:
`restore -Pc -vf backup.bak`
- ▶ To view the table of contents along with file permissions:
`restore -Ta -vf backup.bak`

Other than the -P option, the -a option is also introduced to the **restore** command. The new -a option along with the -T flag will allow the **restore** command to display permissions for the table of contents on the archive.

The syntax of the **restore** command for -T option is shown as follows.

- ▶ To list files archived by file name:
`restore -T [a q v] [-b Number] [-f Device] [-s SeekBackup]`
- ▶ To list files archived by file system:
`restore -t | -T [Bah q v y] [-b Number] [-f Device] [-s SeekBackup]
[File ...]`

6.11 Paging space enhancements

AIX 5L provides two enhancements for managing paging space. A new command, **swapoff**, allows you to deactivate a paging space. The -d flag, for the **chps** command, provides the ability to decrease the size of a paging space. For both commands, a system reboot is no longer required.

6.11.1 Deactivating a paging space

To deactivate a paging space with the **swapoff** command, you can either use:

```
# swapoff device name { device name ... }
```


Or a system management tool, such as SMIT (fast path swapoff), as shown in Figure 6-42.

Deactivate a Paging Space

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

PAGING SPACE name [Entry Fields]
paging00 +

F1=Help F2=Refresh F3=Cancel F4=List
F5=Reset F6=Command F7=Edit F8=Image
F9=Shell F10=Exit Enter=Do

Figure 6-42 *SMIT Deactivate a Paging Space panel*

This command may fail due to:

- ▶ Paging space size constraints
- ▶ I/O errors

Because it is necessary to move all pages (in use on the paging space) to be deactivated to other paging spaces, there must be enough space available in the other active paging spaces. Basically, this command pages in all active pages (after marking the paging space to be deactivated as unavailable) and allows the AIX VMM to page these pages out again to the other available paging spaces. In the case of I/O errors, you should check the error log, deactivate the paging space you are working on for the next system reboot with the **chps** command, and reboot the system. Do not try to reactivate paging spaces with I/O errors before you have checked the corresponding disk with the appropriate diagnostic tools. The **lspcs** command will display, in this case, the string I/O error in the column with the heading Active.

Using Web-based System Manager, a paging space can be deactivated by selecting that paging space from either the Paging Space, Logical Volume or Volume Groups plug-in and selecting **Stop...⁽²⁾** from the Selected pull-down or pop-up menu (Figure 6-43 on page 380).

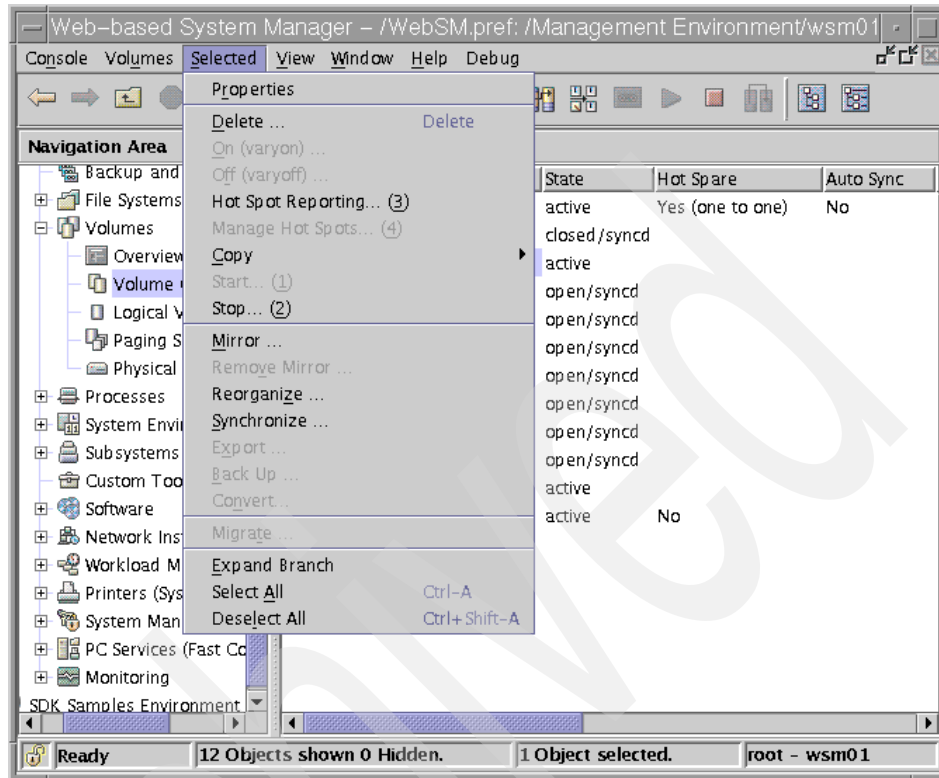


Figure 6-43 Selected pull-down for volume management

6.11.2 Decreasing the size of a paging space

By using the new -d flag, you can decrease the size of an existing paging space using the **chps** command as follows:

```
# chps -dLogicalPartitions PagingSpace
```

Or specify it on the SMIT panel (fast path chps), as shown in Figure 6-44 on page 381.

Change / Show Characteristics of a Paging Space

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

Paging space name	[Entry Fields]	
Volume group name	paging00	
Physical volume name	rootvg	
NUMBER of additional logical partitions	hdisk0	
Or NUMBER of logical partitions to remove	[]	#
Use this paging space each time the system is RESTARTED?	[]	#
	yes	+

F1=Help

F2=Refresh

F3=Cancel

F4=List

F5=Reset

F6=Command

F7=Edit

F8=Image

F9=Shell

F10=Exit

Enter=Do

Figure 6-44 SMIT panel for decreasing the size of a paging space

Using Web-based System Manager, a paging space can be dynamically decreased in size by selecting that paging space, bringing up the Properties dialog for that paging space, and inputting the size to deallocate in either Megabytes or physical partitions (Figure 6-45 on page 382). Web-based System Manager then issues the appropriate commands to perform the action and automatically notifies you of success or any error condition it encounters.

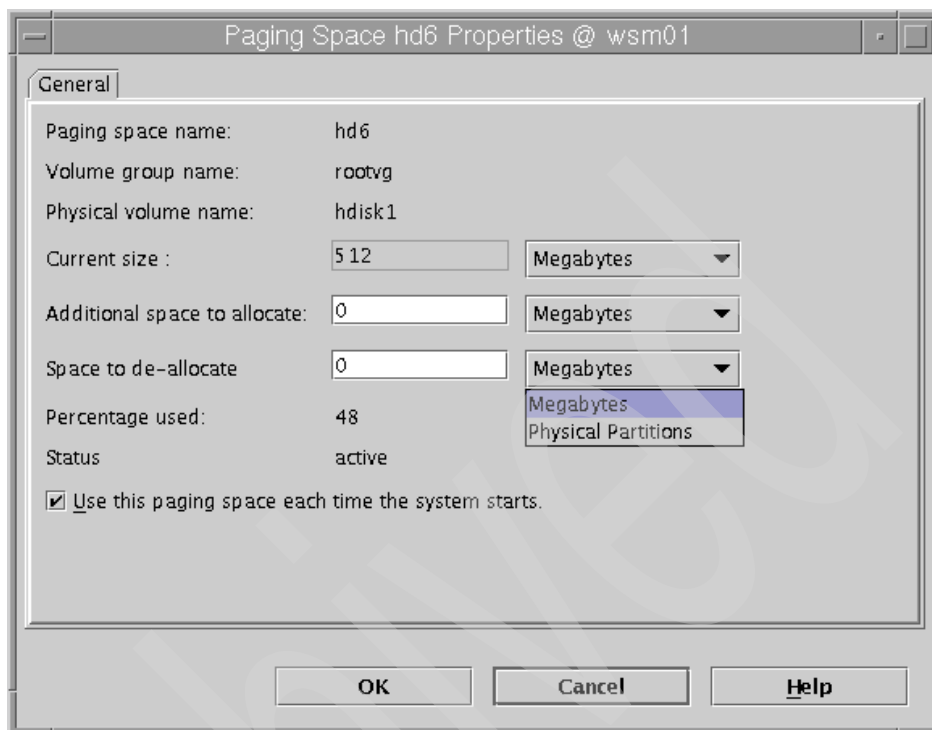


Figure 6-45 Properties dialog to increase page space

The actual processing is done by the shell script **shrinkps**. In the case of decreasing the size of an active paging space, **shrinkps** will create a temporary paging space, move all pages from the paging space to be decreased to this temporary one, delete the old paging space, recreate it with the new size, move all the pages back, and finally delete the temporary paging space. This temporary paging space is always created in the same volume group as the one you try to decrease. It is therefore necessary to have enough space available in the volume group for this temporary paging space. If you decrease the size of a deactivated paging space, the creation of a temporary paging space is not necessary and therefore omitted.

The following example shows the commands needed to remove one logical partition from paging01:

```
# lsps -a
Page Space  Physical Volume  Volume Group  Size  %Used  Active  Auto  Type
paging01    hdisk0              rootvg        48MB   1      yes    yes   lv
hd6         hdisk0              rootvg        32MB   11     yes    yes   lv
# chps -d 1 paging01
shrinkps: Temporary paging space paging00 created.
```

```
shrinkps: Paging space paging01 removed.  
shrinkps: Paging space paging01 recreated with new size.
```

```
# lsps -a
```

Page Space	Physical Volume	Volume Group	Size	%Used	Active	Auto	Type
paging01	hdisk0	rootvg	32MB	1	yes	yes	lv
hd6	hdisk0	rootvg	32MB	12	yes	yes	lv

As you can imagine from the above description, the deactivation or decrease in size of an active paging space can result in a noticeable performance degradation, depending on the size and usage of the paging space and the current system workload. But the main advantage is that there is no system reboot necessary to rearrange the paging space.

If you are working with the primary paging space (usually hd6), this command will prevent you from decreasing the size below 32 MB or actually deleting it. If you decrease the primary paging space, a temporary boot image and a temporary /sbin/rc.boot pointing to this temporary primary paging space will be created to make sure the system is always in a state where it can be safely rebooted.

Note: These command enhancements are not available through the Web-based System Manager. The Web-based System Manager allows you, by default, to specify the increase in size for a paging space in the Megabytes field.

6.12 The dd command enhancement (5.1.0)

The **dd** command now supports multiple volume spanning by using the **span=yes** option. In the case where **span=no**, **dd** does not span multiple volumes and functions as though the **span** option is omitted altogether. The following commands show an example of copying a source file onto multiple volumes using a 1.44 MB diskette drive:

```
# uuencode testfile testfile >testfile.uu  
# ls -l testfile.uu  
-rw-r--r--1rootssystem1839769Mar 1908:59testfile.uu  
# dd if=testfile.uu of=/dev/fd0 bs=720b conv=sync span=yes  
Insert next media on /dev/fd0 ,and press enter
```

```
Proceeding to next media for write  
8+0 records in.  
8+0 records out.
```

To restore from a multiple volume **dd** image, insert the first volume and perform the following procedure. Ensure that the diskettes are inserted in the correct consecutive order.

```
# dd if=/dev/fd0 of=restorefile bs=720b span=yes conv=sync
Insert next media on /dev/fd0, and press 'y' to continue or 'n' to quit
y
Proceeding to next media for read
Insert next media on /dev/fd0, and press 'y' to continue or 'n' to quit
n
8+0 records in.
8+0 records out.
```

Take note that the file size of `restorefile` is different from that of `testfile.uu`. The reason for this is that the `dd` command will dump the entire content of the diskette, including blank spaces, into the file. The file `restorefile` will have the size of two 1.44 MB diskettes. Using the `uudecode` command, the file is restored to its original size:

```
# uudecode restorefile
```

Note: Exercise care when selecting the block size since an incorrect value can result in data inconsistency or overlap. The correct block size should be a multiple of the physical volume size. Also, each volume should be externally labelled so that the volumes can be restored in the correct order.

6.13 shutdown enhancements

AIX 5L enhances the `shutdown` command with an `-l` flag to log the output (from select actions during the shut down) to the file `/etc/shutdown.log`. The contents of this file appear similar to the following:

```
# cat /etc/shutdown.log
```

```
Fri Aug 25 13:21:30 CDT 2000
shutdown: THE SYSTEM IS BEING SHUT DOWN NOW
```

```
User(s) currently logged in:
root
```

```
Stopping some active subsystems...
```

```
0513-044 The dpid2 Subsystem was requested to stop.
0513-044 The hostmibd Subsystem was requested to stop.
0513-044 The qdaemon Subsystem was requested to stop.
0513-044 The writesrv Subsystem was requested to stop.
0513-044 The wsmrefserver Subsystem was requested to stop.
```

```
Unmounting the file systems...
```

```
/usr/local unmounted successfully.  
/proc unmounted successfully.  
/home unmounted successfully.  
/tmp unmounted successfully.
```

Bringing down network interfaces:

```
detached en0 from the network interface list  
detached en1 from the network interface list  
detached et0 from the network interface list  
detached lo0 from the network interface list  
detached tr0 from the network interface list
```

The output of consecutive shutdowns (if the `-l` flag is used) is appended to the `/etc/shutdown.log` file. Therefore, this information is available even if there are problems with booting the system and the machine had to be shut down several times. The log file continues to grow until the system administrator intervenes.

6.14 Crontab enhancements (5.1.0)

AIX 5L Version 5.1 provides an enhancement in cron logging. The log file is mainly used for accounting and now has more detailed information, which is added by the new cron daemon. The `/var/adm/cron/log` now includes the following:

- ▶ The starting time of the daemon and the PID of the cron process
- ▶ The owner of the job run by the cron daemon
- ▶ The time of execution of the job
- ▶ The PID of the job
- ▶ The actual command line that is run to accomplish the job
- ▶ Whether the job has run successfully

The following display format is used:

```
User : CMD (actual command that is executed) : time when the job is executed :  
Cron Job with pid : Successful  
User : CMD (actual command that is executed) : time when the job is executed :  
Cron Job with pid : Failed
```

For example:

```
root      : CMD ( /usr/lib/ras/dumpcheck >/dev/null 2>&1 ) : Tue Feb. 20  
15:00:00 2001  
Cron Job with pid: 20664 Successful
```

Every time cron runs a job (either from the crontab file, for the system-related jobs, or from the /var/spool/cron/crontab/userfile, for user-related processes), all its activity will be logged into the /var/adm/cron/log file in the mentioned format.

6.15 Sendmail upgrade enhancements (5.1.0)

AIX 5L Version 5.1 uses Sendmail Version 8.11.0. This version has several enhancements and changes:

- ▶ The sendmail files sendmail.cf and aliases have been moved to the /etc/mail directory. Links exist on the POWER platforms that are required for the migration to AIX 5L Version 5.1 from earlier releases of AIX.

```
# ls -l /etc/sendmail.cf /etc/aliases
lrwxrwxrwx 1 root system 21 Mar 07 10:28 /etc/sendmail.cf
-> /etc/mail/sendmail.cf
lrwxrwxrwx 1 root system 17 Mar 07 10:28 /etc/aliases ->
/etc/mail/aliases
```

- ▶ Sendmail supports the Berkeley DB 3.1.14 format to more efficiently store the aliases.db database file. Other databases used can store their data in the Berkeley database formats.
- ▶ Support for message submission agents.
- ▶ Multiple queues, memory-buffered pseudo files, and more control over resolver time-outs improve performance.
- ▶ The ability to connect to servers running on named sockets.
- ▶ Better LDAP integration and support for LDAP-based routing.
- ▶ Improved support for virtual hosting.
- ▶ Even better anti-spam control features.
- ▶ Several new map classes, which include arith and macro.

More information on Sendmail Version 8.11.0 is available from the following Web site.

<http://www.sendmail.org>

6.15.1 Sendmail 8.11.0 supports the Berkeley DB

The Berkeley DB is an embedded database system that supports keyed access to data. The library includes support for the following access methods:

- ▶ Btrees
- ▶ Hashing

► Fixed and variable-length records

It also provides core database services, such as page cache management, transactions, locking, and logging. An API is provided that allows developers to easily embed database-style function and support into other objects or interfaces.

The Berkeley DB support is now available on AIX 5L Version 5.1 for Sendmail 8.11.0. As long as the aliases database is not rebuilt, `sendmail` will continue to read it in its old DBM format. This consists of two files: `/etc/mail/aliases.dir` and `/etc/mail/aliases.pag`. However, when the aliases database is rebuilt, `sendmail` will change this format to Berkeley DB. This file will be stored in `/etc/mail/aliases.db`.

In the `/etc/mail/alias` file, uppercase characters on the left-hand side of the alias are converted to lowercase before being stored in the aliases database. In the following example, mail sent to the `testalias` user alias fails, since `TEST` is converted to `test` when the second line is stored.

```
TEST: user@machine
testalias: TEST
```

To preserve uppercase in user names and alias names, add the `u` flag to the local mailer description in the `/etc/mail/sendmail.cf` file. Thus, in the previous example, mail to the `testalias` user alias would succeed. The `/etc/mail/sendmail.cf` for the local mailer would appear similar to the following:

```
Mlocal, P=/usr/bin/bellmail, F=lsDFMnu, S=10, R=20, A=mail $u
```

6.16 NCARGS value configuration (5.1.0)

In AIX 5L Version 5.1, the option has been added to allow the super user or any user belonging to the system group to dynamically change the value of the NCARGS parameters. In previous releases of AIX, these values were permanently defined as 24576, which resulted in a problem similar to that shown below when a large number of arguments are parsed to a command:

```
# rm FILE*
ksh: /usr/bin/rm: 0403-027 The parameter list is too long.
```

The value of NCARGS can be increased to overcome this problem. The value can be tuned anywhere within the range of 24576 to 524288 in 4 KB page size increments. To display the value, use the following command:

```
# lsattr -El sys0 |grep arg
ncargs12ARG/ENVlist size in 4K byte blocksTrue
```

Alternately, the SMIT system fast path can be used, as shown in Figure 6-46.

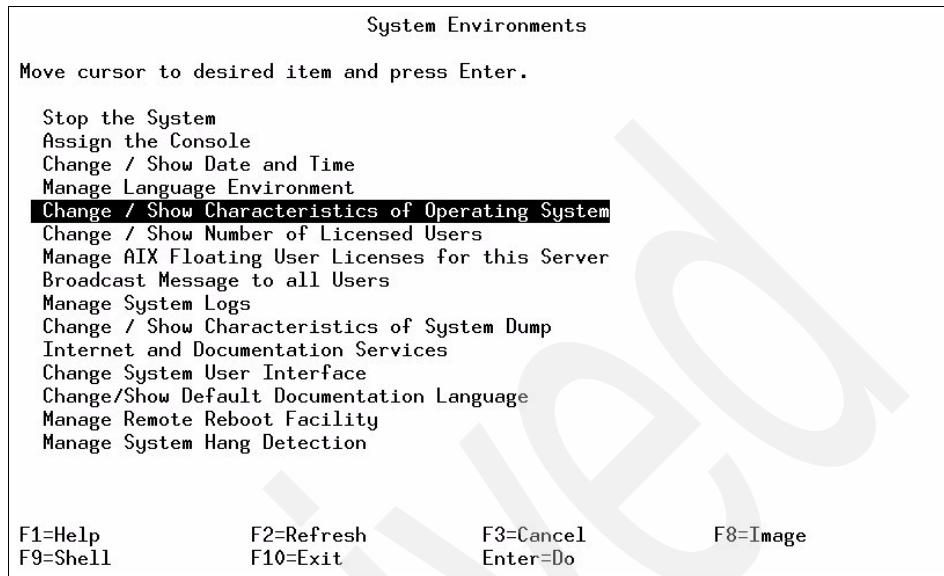


Figure 6-46 SMIT System Environment panel

Use the arrow keys on the keyboard to move to the Change/Show Characteristics of Operating System option and press Enter. The screen shown in Figure 6-47 on page 389 will be displayed. In this SMIT panel, the value can be changed.

Change / Show Characteristics of Operating System

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

	[Entry Fields]	
Maximum number of PROCESSES allowed per user	[128]	++
Maximum number of pages in block I/O BUFFER CACHE	[20]	++
Maximum Kbytes of real memory allowed for Mbufs	[0]	++
Automatically REBOOT system after a crash	false	+
Continuously maintain DISK I/O history	false	+
HIGH water mark for pending write I/Os per file	[0]	++
LOW water mark for pending write I/Os per file	[0]	++
Amount of usable physical memory in Kbytes	524288	
State of system keylock at boot time	normal	
Enable full CORE dump	false	+
Use pre-430 style CORE dump	false	+
CPU Guard	disable	+
ARG/ENV list size in 4K byte blocks	[6]	++

F1=Help

F2=Refresh

F3=Cancel

F4=List

F5=Reset

F6=Command

F7=Edit

F8=Image

F9=Shell

F10=Exit

Enter=Do

Figure 6-47 SMIT Change/Show Characteristics of Operating System panel

To change the value of NCARGS, the following command can be used:

```
# chdev -l sys0 -a ncargs='64'
```

Note: Increasing the values of NCARGS uses additional kernel memory and this may result in a performance issue on systems that have small memory sizes.

6.17 Extended host name support (5.1.0)

In AIX 5L Version 5.1, the maximum storage size has been increased for display of a remote host name. In the new version utmp.h and rhost.h, the ut_host string has been modified to display up to 256 characters, depending on commands that use ut_host.

The modified structure is as follows for utmp.h and rhost.h:

```
char ut_host[256];          /* host name */
```

For example, using the **who** command, AIX 5L Version 5.1 displays the following:

```
# who
root          pts/0          Feb 22 10:40          (ausres41.itso.austin.ibm.com)
```

Previous versions of AIX would appear as follows:

```
# who  
antonym      pts/0      Feb 23 03:43    (ausres41.itsc.au)
```

Other commands that use the ut_host string are **halt**, **reboot**, **acct**, **tsm**, and **uucp**.

6.18 OpenType font support (5.1.0)

In AIX 5L Version 5.1, the TrueType font rasterizer, available in AIX 5.0 and earlier, has been replaced by a version from the AGFA Corporation. Using a different TrueType rasterizer provides a better font quality.

6.18.1 TrueType rasterizer

A TrueType rasterizer generates character bitmaps for screens and printers. In order to do this, the following steps are required:

1. Decode the glyph from its compressed representation in the TrueType file and read the outline description of the character.
2. Scale the glyph according to the desired point size and output device.
3. Execute the glyph's hinting program, with the effect of distorting the glyph's control points.
4. Fill the hinted outline with pixels and make a bitmap image of the glyph.
5. Pass the bitmap to the system.

6.18.2 AGFA rasterizer enhancement (5.2.0)

In AIX 5L Version 5.2 the UFST code is updated to the latest official version from AGFA. This provides AIX with the latest quality and functional improvements from AGFA including support for embedded bitmap TrueType fonts.

Font rasterizers are known to have trouble creating nice looking characters when only a small number of pixels are available, such as when you are trying to create a font at a small point size. To get around this problem the font vendor can create the bitmaps for the font at a particular size, go back by hand and touch up the characters that need it, and then embed this back into the TrueType font. When the rasterizer is asked to produce characters at the designated size, rather than creating the characters on the fly, it will use the bitmaps instead.

6.19 Terminal support enhancements (5.1.0)

The terminal emulation in AIX 5L Version 5.1 has been enhanced to support the ANSI terminal type.

6.19.1 ANSI terminal support

The default emulation in Microsoft Windows telnet is VT-100/ANSI. There is no documented way to override the default emulation with command line options. One can, however, change the emulation after the session opens. When connecting to earlier AIX releases, the **telnet** command negotiates a terminal type of VT-100.

In AIX 5L Version 5.1, the telnet session negotiates a terminal type of ANSI, so the TERM environment variable gets set TERM=ansi. This helps reduce problems when opening a SMIT screen. Figure 6-48 shows a SMIT screen from a telnet session correctly displayed as a result of the TERM=ansi setting.

```
System Management
Move cursor to desired item and press Enter.
Software Installation and Maintenance
Software License Management
Devices
System Storage Management <Physical & Logical Storage>
Security & Users
Communications Applications and Services
Print Spooling
Problem Determination
Performance & Resource Scheduling
System Environments
Processes & Subsystems
Applications
Using SMIT <information only>

Esc+1=Help      Esc+2=Refresh   Esc+3=Cancel    Esc+8=Image
Esc+9=Shell     Esc+0=Exit     Enter=Do
```

Figure 6-48 Telnet session from Microsoft Windows 2000

After you have successfully logged in, the terminal environment variable has been set to TERM=ansi:

```
# echo $TERM
ansi
```

Note: You actually can manually set TERM to another value like vt100 or vt220. But be aware that your SMIT screen may be garbled when you are connecting from a Microsoft Windows system. Setting TERM to ANSI is not the same as setting to ansi (lower case).

6.20 New utmpd daemon (5.2.0)

Version 5.2 introduces a new daemon called utmpd, to manage the entries in the /etc/utmp file.

A number of commands read and write to the /etc/utmp file. The commands include, but are not limited to, the following: **date**, **lgout**, **users**, **uucp**, **who**, **w**, **init**, **penable**, **wall**, **login**, **rcvtty**, **dtlogin**, **xterm**, **aixterm**, **finger**, **rlogind**, **rexecd**, and **telnetd**.

When a user logs in to the system, an entry is made in /etc/utmp, and when the users logs out, the entry is removed. The daemon, utmpd, is dedicated to maintaining the consistency of this file by detecting that a process has terminated and ensuring that the corresponding entry in /etc/utmp is deleted. The utmpd daemon also processes the file to ensure that all entries are still valid.

The utmpd daemon can be started by init, and specified in the /etc/inittab file, although this entry is not provided by default. The default interval for the running of utmpd is 300 seconds, although this can be provided as a parameter. It is also possible to execute **utmpd** from the shell prompt. The syntax for the command is:

```
/usr/sbin/utmpd [Interval]
```

6.21 System information command (5.2.0)

The **getconf** command is enhanced with Version 5.2. The enhancement adds additional system configuration and path configuration parameters.

The command provides information about system configuration variables. The main information intended from the enhancement refer to: memory, disk size, last boot device, hardware check for 32-bit or 62-bit and the same for the kernel. The **getconf** command is enhanced to provide extra information that is currently available with the unsupported **bootinfo** command. The **getconf** command used the ODM library routines to extract information from the device configuration database. The **getconf** command issues a setuid root to access privileged configuration variables.

The syntax of the command is as follows:

```
getconf [ -v specification ] [ SystemwideConfiguration | PathConfiguration  
PathName ] [ DeviceVariable DeviceName]
```

Where the variable names are defined as provided in Table 6-8 on page 393.

Table 6-8 System-wide configuration names

Variable	Description
System-wide configuration names	
BOOT_DEVICE	Displays last boot device
MACHINE_ARCHITECTURE	Displays machine architecture type (chrp)
MODEL_CODE	Displays model code
KERNEL_BITMODE	Bit mode of the kernel, 32-bit or 64-bit
REAL_MEMORY	Real memory size in kilobytes
HARDWARE_BITMODE	Bit mode of the machine hardware, 32-bit or 64-bit
MP_CAPABLE	MP capability of the machine
Path configuration names	
DISK_PARTITION	Physical partition size of the disk
DISK_SIZE	Disk size in megabytes
Device variables names	
DISK_DEVNAME	Device name or location of the device

The values for the variables mentioned in Table 6-8 are also available from the `sysconf()`, `pathconf()`, or `confstr()` library calls.

An example of the `getconf` command is as follows:

```
# getconf KERNEL_BITMODE
64
# getconf HARDWARE_BITMODE
64
# getconf DISK_SIZE /dev/hdisk0
8678
```


Performance management

The topics within this chapter can be broken down into the AIX 5L enhancements in two areas:

- ▶ Performance tools
- ▶ AIX tuning framework

7.1 Performance tools

For AIX 5L the following tools and commands are available: **alstat**, **gennames**, **genkex**, **genkld**, **locktrace**, **truss**, **iostat**, **vmstat**, **sar**, **prof**, **tprof**, **gprof**, **emstat**, **filemon**, **fileplace**, **netpmon**, **pprof**, **rmss**, **svmon**, and **topas**.

The following tools have been withdrawn in AIX 5L: **bf** (bigfoot), **bfrpt**, **lockstat**, **stem**, and **syscalls**. Consult the man pages for **svmon**, **locktrace**, and **truss** to locate similar functions.

7.1.1 Performance tools repackaging (5.1.0)

In AIX 5L Version 5.1, the base performance tools are repackaged and moved from the `perfagent.tools` to the `bos.perf.tools` fileset.

To use the utilities in the `bos.perf.tools` fileset, you also have to install the following filesets:

- ▶ `bos.sysmgmt.trace`
- ▶ `bos.perf.perfstat`
- ▶ `perfagent.tools`

Tools that have been repackaged and are available in the `bos.perf.tools` fileset are provided in Table 7-1.

Table 7-1 Performance tools packaging versus platform

Performance utility	POWER-based
<code>/usr/bin/locktrace</code>	X
<code>/usr/bin/pprof</code>	X
<code>/usr/bin/rmss</code>	X
<code>/usr/bin/genkex</code>	X
<code>/usr/bin/gennames</code>	X
<code>/usr/bin/netpmon</code>	X
<code>/usr/bin/genkld</code>	X
<code>/usr/bin/fileplace</code>	X
<code>/usr/bin/ipfilter</code>	X
<code>/usr/bin/svmon</code>	X
<code>/usr/bin/tprof</code>	X

Performance utility	POWER-based
/usr/bin/emstat	X
/usr/bin/filemon	X
/usr/bin/topas	X
/usr/bin/stripnm	X
/usr/bin/genld	X
/usr/bin/alstat	X

The perfagent.tools fileset remains to support the PTX base dependencies. The perfagent.tools fileset has, as a prerequisite, bos.perf.tools and bos.perf.perfstat, so the basic performance tools will be automatically picked up and installed on the system.

7.1.2 Emulation and alignment detection

A new tool was added in the perfagent.tools fileset; in addition to the existing **emstat** command, **alstat** will count alignment interrupts while **emstat** will display emulation statistics.

Both commands can use the -v flag, which will display the statistics per CPU in SMP systems.

7.1.3 Performance monitor API

A new set of APIs is available to provide access to performance monitor data on selected processor types, namely 604, 604e, POWER3, POWER3-II, RS64-II, RS64-III, RS64-IV, POWER4, and POWER4+. Other processors of the POWER platform not listed are not supported by this API.

For AIX 5L Version 5.1, refer to “Performance Monitor API Programming Concepts” section in Chapter 10 “Programming on Multiprocessor Systems” of the Programming Guides publication in the Online Documentation Library (see this redbook Bibliography) for a complete list of API calls, as well as several sample programs.

For AIX 5L Version 5.2, refer to “Performance Monitor API Programming” in the *Performance Tools Guide and Reference* publication in the Online Documentation Library for a complete list of API calls, as well as several sample programs.

7.1.4 The locktrace command (5.1.0)

Starting with AIX 5L Version 5.1, the **lockstat** command is no longer supported. Tracing locks, including at class level, can now be done with the **locktrace** command, which is part of the bos.perf.tools and is shipped with the base AIX CD-ROMs for AIX POWER.

The **locktrace** command controls which kernel locks are being traced by the **trace** subsystem. The default is to trace none even if the machine has been rebooted after running the **bosboot -L** command. If **bosboot -L** was run, kernel lock tracing can be turned on or off for one or more (up to 32) individual lock classes, or for all lock classes. If **bosboot -L** was not run, lock tracing can only be turned on for all locks or none.

- ▶ On the regular kernel, **locktrace -S** allows the tracing of all locks regardless of their class membership, but will not set the classid.instance data word normally present in tracehook 112 (lock taken or unused) and 113 (lock released). The addresses of the locks and the addresses of the lock function caller will still be reported, allowing lock identification in many cases.
- ▶ On the **bosboot -L** kernel, **locktrace -S** also allows all locks regardless of their class membership, but will make the classid.instance data available in tracehooks 112 and 113.

Table 7-2 lists the flags that can be used with the **locktrace** command.

Table 7-2 The locktrace command flags

Flag	Description
-r <i>classname</i>	Turns off lock tracing for all the kernel locks belonging to the specified class. This option always fails if bosboot -L was not run.
-s <i>classname</i>	Turns on lock tracing for all the kernel locks belonging to the specified class. This option always fails if bosboot -L has not been executed.
-R	Turns off all lock tracing.
-S	Turns on lock tracing for all locks regardless of their class membership.
-l	Lists the kernel lock tracing current status.

Example of the locktrace command

This example describes a trace on a regular kernel. Start with enabling the lock tracing with the following command:

```
# locktrace -S
```

lock tracing enabled for all classes

Once the lock tracing is enabled, start the **trace** command:

```
#trace -a -T 768000 -L 10000000 -o /tmp/trace.out
```

Run a few commands, for example:

```
#crfs -v jfs -g datavg -a size='43' -m /test
#fsck /dev/ftptestlv
```

Stop the tracing and convert the output file:

```
# trcstop
# trcrpt /tmp/trace.out > /tmp/trace.rpt
```

The trace.rpt will have the locks listed and appears similar to the following:

```
Thu Mar 15 16:53:42 2001
System: AIX server1 Node: 5
Machine: 000BC6FD4C00
Internet Address: 0903F038 9.3.240.56
The system contains 4 cpus, of which 4 were traced.
Buffering: Kernel Heap
This is from a 32-bit kernel.
Tracing all hooks.
```

```
trace -a -T 768000 -L 10000000 -o trace.out
```

ID	ELAPSED_SEC	DELTA_MSEC	APPL	SYSCALL	KERNEL	INTERRUPT
112	0.000000000	0.000000			lock:	lock lock
addr=1F809BDC lock status=1B7D requested_mode=LOCK_SWRITE return addr=41CADC						
name=0000.0000						
113	0.000001132	0.001132			unlock:	lock addr=1F809BDC
lock status=0000 return addr=41CC0C name=0000.0000						

To start tracing the **SEM_LOCK_CLASS**, use the following command:

```
# locktrace -s SEM_LOCK_CLASS
```

7.1.5 Cmdstat tools enhancement (5.1.0)

The **cmdstat** commands are those software tools found in the bos.acct fileset that monitor system performance. The **cmdstat** commands include **vmstat**, **iostat**, and **sar**. The enhancements made have no impact on existing functions of the cmdstat tools. The enhancements are as follows.

- ▶ In previous releases of AIX, these commands made direct /dev/kmem reads. These reads from /dev/kmem have been replaced by calls to the perfstat

kernel extension. The APIs fetch the kernel statistics and populate the corresponding performance tool's data structure.

- The **cmdstat** commands in previous release (AIX 5L) required two different executables: One for 32-bit kernels and one for 64-bit kernels. AIX 5L Version 5.1 has performance tools' data structures used by the perfstat APIs that are not kernel bit sensitive.

For more information, see 7.1.19, "Perfstat API library (5.1.0 and 5.2.0)" on page 421.

7.1.6 The vmstat command enhancements

The **vmstat** command has two new flags in AIX 5L; these new flags add new controls and improve monitoring.

The **-l** flag outputs a report with the new columns **fi** and **fo**; these columns indicate the number of file pages in (**fi**) and out (**fo**). In this report, the **re** and **cy** columns are not displayed. A new **p** column displays the number of threads waiting for a physical I/O operation.

```
# vmstat -l 1 3
kthr      memory          page        faults        cpu
-----
r  b  p    avm    fre  fi  fo  pi  po  fr  sr    in    sy  cs  us  sy  id  wa
0  0  0 46391   228    0  0  0  0  0  0    2  108  156  20  1  0 99  0
0  1  0 46391   226    0  0  0  0  0  0    0  432  8080  53  1  1 98  0
0  1  0 46391   226    0  0  0  0  0  0    0  424   91  50  0  0 99  0
```

The **-t** flag shows a timestamp at the end of each line, as shown in the following:

```
# vmstat -t 1 3
kthr      memory          page        faults        cpu        time
-----
r  b  avm    fre  re  pi  po  fr  sr  cy  in    sy  cs  us  sy  id  wa  hr  mi  se
0  0 46905  5752    0  0  0  0  2  0 108  156  20  1  0 99  0 11:46:28
0  1 46905  5749    0  0  0  0  0  0 429 7264  72  1  1 98  0 11:46:29
0  1 46905  5749    0  0  0  0  0  0 434  165  60  0  0 99  0 11:46:30
```

7.1.7 The iostat command enhancements

The **iostat** command is enhanced with new parameters that provide a better presentation of the generated reports.

The **-s** flag adds a new line to the header of each statistic's data that reports the sum of all activity on the system.

```
# iostat -s 1 3
System: server1.itsc.austin.ibm.com
```

		Kbps	tps	Kb_read	Kb_wrtn
		9405.3	2351.3	28216	0

Disks:	% tm_act	Kbps	tps	Kb_read	Kb_wrtn
hdisk0	46.7	4693.3	1173.3	14080	0
hdisk1	24.0	2356.0	588.7	7068	0
hdisk2	0.0	0.0	0.0	0	0
hdisk3	24.3	2356.0	589.3	7068	0
hdisk4	0.0	0.0	0.0	0	0
cd0	0.0	0.0	0.0	0	0

The -a flag produces an output similar to the -s flag output, with the difference that it provides an adapter basis sum of activities. After displaying the adapter activity, it provides a per-disk basis set of statistics.

```
# iostat -a 1 3
```

tty:	tin	tout	avg-cpu:	% user	% sys	% idle	% iowait
	0.0	923.7		13.2	41.6	30.9	14.2

Adapter:		Kbps	tps	Kb_read	Kb_wrtn
scsi0		7030.4	1757.6	7048	0

Disks:	% tm_act	Kbps	tps	Kb_read	Kb_wrtn
hdisk0	43.9	4684.3	1171.1	4696	0
hdisk1	24.9	2346.1	586.5	2352	0
hdisk2	0.0	0.0	0.0	0	0
cd0	0.0	0.0	0.0	0	0

Adapter:		Kbps	tps	Kb_read	Kb_wrtn
scsi1		2346.1	585.5	2352	0

Disks:	% tm_act	Kbps	tps	Kb_read	Kb_wrtn
hdisk3	19.0	2346.1	585.5	2352	0
hdisk4	0.0	0.0	0.0	0	0

7.1.8 The netpmn and filemon command enhancements

New offline support allows you to generate **netpmn** reports with a normal trace report file and a **gennames** output for improved use and scalability on target systems.

To use the new function, you must generate a normal trace output (for example, through **smit trace** and then **start trace**), and then generate an unformatted trace file through the output trace file, as shown in the following example:

```
# trcrpt -r /var/adm/ras/trcfile > /tmp/newtrcfile
```

Immediately following the collection of the trace file, you should also run the **gennames** command and save its output:

```
# gennames > /tmp/gennames.out
```

When both files are correctly set, you can generate your offline report using the **-i** and **-n** flags, as shown in the following **netpmon** example:

```
# netpmon -i /tmp/newtrcfile -n /tmp/gennames.out
```

7.1.9 The **gennames** command (5.1.0)

The **gennames** command gathers all the information necessary to run the **tprof**, **filemon**, or **netpmon** commands in off-line mode.

The **gennames** command has been enhanced with a new **-f** flag. The **-f** flag is needed for processing offline **filemon** traces (to be added to the **gennames** output).

The following example shows how to run **filemon** in offline mode while using the **gennames** command:

```
# trace -a -T 768000 -L 10000000 -o trace.out -j  
000,000,001,002,003,005,006,139,102,10C,106,00A,107,  
101,104,10D,15B,12E,130,163,19C,154,3D3,1BA,1BE,1BC,10B,221,1C9,222,228,232,45B
```

Stop the trace after you have run the monitored application programs or system commands:

```
# trcstop
```

Create the **gennames** file:

```
# gennames -f > gennames.out
```

Format the trace file while using the **trcrpt** command:

```
# trcrpt -r trace.out > trace.rpt
```

Run **filemon** with both **-i** and **-n** flags:

```
filemon -i trace.rpt -n gennames.out -O all
```

7.1.10 The **svmon** command enhancements

The **svmon** command has been enhanced to display information about tiers, Superclasses, and Subclasses introduced with the Workload Manager in AIX 5L update.

Four new flags, discussed in the following sections, can be used in order to make use of this new function.

The -W flag

The -W flag is used to collect statistics for either an entire Superclass or only a specific Subclass. The following example is an output generated for a Superclass:

```
# svmon -W sv
```

Superclass	Inuse	Pin	Pgsp	Virtual
sv	2039	8	0	231

Vsid	Esid	Type	Description	Inuse	Pin	Pgsp	Virtual
5f4b	-	pers	/dev/hd2:43509	1082	0	-	-
48e8	-	pers	/dev/hd2:47134	182	0	-	-
e099	-	work		69	0	0	70
48ac	-	work		61	0	0	62

To display Subclass information, you must use `class.Subclass` for syntax:

```
# svmon -W sv.sv_sub
```

Class	Inuse	Pin	Pgsp	Virtual
sv.sv_sub	1929	6	0	124

Vsid	Esid	Type	Description	Inuse	Pin	Pgsp	Virtual
5f4b	-	pers	/dev/hd2:43509	1082	0	-	-
48e8	-	pers	/dev/hd2:47134	182	0	-	-
c8bc	-	work		74	2	0	73
2f45	-	pers	/dev/hd2:47128	54	0	-	-

The -e flag

The -e flag reports the statistics for the Subclasses of a Superclass. It only applies to Superclasses or tiers. The -e flag is only allowed with -T and -W. A sample output is shown in the following example:

Superclass	Inuse	Pin	Pgsp	Virtual
sv	1867	4	0	74

=====				
Class	Inuse	Pin	Pgsp	Virtual
sv.sv_sub	1769	0	0	0

Vsid	Esid	Type	Description	Inuse	Pin	Pgsp	Virtual
5f4b	-	pers	/dev/hd2:43509	1082	0	-	-
48e8	-	pers	/dev/hd2:47134	182	0	-	-
2f45	-	pers	/dev/hd2:47128	54	0	-	-

=====				
Class	Inuse	Pin	Pgsp	Virtual

sv.Default				98	4	0	74
Vsid	Esid	Type	Description	Inuse	Pin	Pgsp	Virtual
28c0	-	work		23	0	0	15
710b	-	work		21	0	0	13
e0f9	-	work		21	0	0	15
3043	-	work		14	2	0	14
3103	-	work		12	2	0	12
6109	-	work		7	0	0	5

Class	Inuse	Pin	Pgsp	Virtual
sv.Shared	0	0	0	0

The -T flag

The -T flag reports the statistics of all the classes in a tier. If a parameter is passed with the -T flag, then only the classes belonging to the tier will be analyzed. A list of tiers can be provided. When no parameter is specified, all the defined tiers of the class will be analyzed. Examples of flag interaction and command response follows.

The -T flag with no parameter provides the following results.

```
# svmon -T
```

Tier	Inuse	Pin	Pgsp	Virtual
0	87112	6650	11462	29167
Superclass	Inuse	Pin	Pgsp	Virtual
System	72109	6616	9197	25124
Shared	6535	0	878	2530
Unclassified	5950	10	5	20
Default	2518	24	1382	1493
Unmanaged	0	0	0	0
random	0	0	0	0
sequential	0	0	0	0
Tier	Inuse	Pin	Pgsp	Virtual
1	1853	2	0	74
Superclass	Inuse	Pin	Pgsp	Virtual
sv	1853	2	0	74

The -T flag with a specific tier value provides the following results:

```
# svmon -T 1
```

=====				
Tier	Inuse	Pin	Pgsp	Virtual
1	1902	4	0	130
=====				
Superclass	Inuse	Pin	Pgsp	Virtual
sv	1902	4	0	130

The -T flag with the -a flag indicating a specific Superclass provides the following results. All the Subclasses of the indicated Superclass in the tier *tiernumber* will be reported.

```
# svmon -a sv -T 1
```

=====					
Tier	Superclass	Inuse	Pin	Pgsp	Virtual
1	sv	2037	10	0	245
=====					
Class	Inuse	Pin	Pgsp	Virtual	
sv.sv_sub	1769	0	0	0	
sv.Default	268	10	0	245	

The -T flag with the -x flag will report all the Superclasses segment statistics of the specific tier and provides the following results.

```
# svmon -T 0 -x
```

Tier	Inuse	Pin	Pgsp	Virtual			
0	88106	6659	11462	30028			
=====							
Superclass	Inuse	Pin	Pgsp	Virtual			
System	73095	6625	9197	25982			
=====							
Vsid	Esid	Type	Description	Inuse	Pin	Pgsp	Virtual
db99	-	pers	large file /dev/lv04:23	27702	0	-	-
8010	-	work	misc kernel tables	3287	0	1210	3289
0	-	work	kernel seg	3134	1635	1919	3379
8811	-	work	kernel pinned heap	3087	1222	1226	3187
8af0	-	pers	/dev/hd2:112665	2316	0	-	-

As shown in the preceding examples, you can mix different flags to obtain different outputs. Refer to the **svmon** command man pages to check for other combinations.

7.1.11 The **svmon** command enhancements (5.2.0)

Reporting on large page memory support has been integrated into the **svmon** utility. The following section outlines the enhancements.

The **svmon** utility has previously reported the number of in-use, pinned, and virtual mapped physical memory pages, and assumed a 4-KB page size. Large page architecture allows the mixing of large and small 4-KB pages in an application address space. The **svmon** utility in Version 5.2 is now able to report large page information.

Large page processes and large page memory segments are supported by a statically defined pool of pinned physical memory. This pool is defined both by the allocation size used for large pages and by the number of large pages of the specified allocation size to be contained in the pool. Both can be specified by the **vmtune**, or on AIX 5L Version 5.2, the **vmo** command.

7.1.12 The **topas** command enhancements

The **topas** command is a performance monitoring tool that was introduced in AIX Version 4.3.3. In AIX 5L, it has several new enhancements, including Workload Manager support, an improved set of CPU usage panels, several new column sort options, NFS statistics, and per disk or adapter breakdown of network and disk usage.

Figure 7-1 on page 407 provides a sample **topas** main screen. This section is too brief to demonstrate all the features. It is recommended that the **topas** tool is given a complete exploration through hands-on use.

Topas Monitor for host: server2						EVENTS/QUEUES		FILE/TTY	
Tue Sep 19 16:29:45 2000						Interval: 1			
Kernel	0.0					Cswitch	28	Readch	149
User	1.0					Syscall	59	Writech	1605
Wait	0.0					Reads	2	Rawin	0
Idle	99.0	#####				Writes	2	Ttyout	0
						Forks	0	Igets	0
						Execs	0	Namei	1
						Runqueue	1.3	Dirblk	0
						Waitqueue	0.0		
Network	KBPS	I-Pack	O-Pack	KB-In	KB-Out				
trl	1.7	5.0	2.0	0.2	1.5				
lo0	0.0	0.0	0.0	0.0	0.0				
Disk	Busy%	KBPS	TPS	KB-Read	KB-Writ				
hdisk0	1.0	4.0	1.0	0.0	4.0				
hdisk1	0.0	0.0	0.0	0.0	0.0				
WLM-Class (Active)									
redbook		CPU%	Mem%	Disk-I/O%					
System		66	0	0					
		1	8	0					
Name	PID	CPU%	PgSp	Class					
aixterm	18326	1.0	0.5	System					
topas	18620	1.0	0.7	System					
expr	19180	0.0	0.0	redbook					
ksh	13928	0.0	0.2	redbook					
						PAGING		MEMORY	
						Faults	0	Real,MB	511
						Steals	0	% Comp	100.0
						PgspIn	0	% Noncomp	0.0
						PgspOut	0	% Client	0.0
						PageIn	0		
						PageOut	0	PAGING SPACE	
						Sios	0	Size,MB	0
						NFS (calls/sec)		% Used	1.0
						ServerV2	0	% Free	98.9
						ClientV2	0	Press:	
						ServerV3	0	"h" for help	
						ClientV3	0	"q" to quit	

Topas Monitor for host: server2				Interval: 1	Tue Sep 19 16:17:37 2000		
WLM-Class (Active)				CPU%	Mem%	Disk-I/O%	
redbook				2	0	0	
System				2	8	33	
Shared				0	4	0	
Default				0	0	0	
Unmanaged				0	5	0	
Unclassified				0	19	0	

USER	PID	PPID	PRI	NI	DATA RES	TEXT RES	PAGE SPACE	TIME	CPU%	PGFAULTS I/O	OTH	COMMAND
root	18620	17370	109	20	217	12	179	0:00	1.0	0	0	topas
rb	18906	13928	108	20	11	6	17	0:00	1.0	0	0	dd
rb	19674	18906	108	20	9	6	15	0:01	1.0	0	0	dd
root	1290	0	16	41	4	4134	4	0:00	0.0	0	0	wlmsched
root	1918	1	108	20	76	37	107	0:00	0.0	0	0	dtlogin
root	2080	0	108	20	4	4134	4	0:00	0.0	0	0	lvmbb
root	2672	1918	108	20	117	37	137	0:00	0.0	0	0	dtlogin
root	2908	1918	108	20	608	351	593	0:03	0.0	0	0	X
root	3190	1	108	20	101	19	93	0:00	0.0	0	0	AIXPowerMgt
root	3448	1	108	20	268	76	225	0:00	0.0	0	0	ttsession
root	3706	1	108	20	4	4134	4	0:00	0.0	0	0	HSCa

Figure 7-2 Workload Manager screen using the W subcommand

CPU display

By default, **topas** will display cumulative CPU usage as in previous releases. However, the **c** (lower case) command can toggle to a per-CPU usage view on SMP systems. The **c** command also toggles CPU monitoring off (see Figure 7-3 on page 409).

Topas Monitor for host: server1					EVENTS/QUEUES		FILE/TTY		
Tue Sep 19 16:38:20 2000					Interval: 1				
CPU	User%	Kern%	Wait%	Idle%	Cswitch	79	Readch	0	
cpu3	100.0	0.0	0.0	0.0	Syscall	1368	Writech	78	
cpu1	100.0	0.0	0.0	0.0	Reads	0	Rawin	0	
cpu2	100.0	0.0	0.0	0.0	Writes	0	Ttyout	0	
cpu0	1.0	1.0	0.0	98.0	Forks	0	Igets	0	
					Execs	0	Namei	0	
					Runqueue	3.0	Dirblk	0	
					Waitqueue	1.0			
Network	KBPS	I-Pack	O-Pack	KB-In	KB-Out	PAGING	MEMORY		
tr0	0.1	2.9	0.9	0.0	0.1	Faults	0	Real, MB	511
lo0	0.0	0.0	0.0	0.0	0.0	Steals	0	% Comp	100.0
Disk	Busy%	KBPS	TPS	KB-Read	KB-Writ	PgspIn	0	% Noncomp	0.0
hdisk2	0.0	0.0	0.0	0.0	0.0	PgspOut	0	% Client	0.0
hdisk0	0.0	0.0	0.0	0.0	0.0	PageIn	0		
hdisk1	0.0	0.0	0.0	0.0	0.0	PageOut	0	PAGING SPACE	
hdisk4	0.0	0.0	0.0	0.0	0.0	Sios	0	Size, MB	0
hdisk3	0.0	0.0	0.0	0.0	0.0	NFS (calls/sec)		% Used	2.8
WLM-Class (off)		CPU%	Mem%	Disk-I/O%		ServerV2	0	% Free	97.1
						ClientV2	0	Press:	
						ServerV3	0	"h" for help	
						ClientV3	0	"q" to quit	
Name	PID	CPU%	PgSp	Class					

Figure 7-3 topas with per-CPU usage enabled

7.1.13 FDPR binary optimizer

FDPR is a tool, first introduced in AIX 3.2, that optimizes binaries generated from xl compilers. It contains two major components: **aopt**, which is used for instrumenting and reordering AIX XCOFF executables; and **fdpr**, which is a more user-friendly interface to the **aopt** command.

This tool is continuously enhanced for each distribution of AIX.

7.1.14 The tprof command

The following section discusses the introduction and enhancements made to the **tprof** command in AIX 5L.

Introduction

The **tprof** command is a program counter sampling based profiler that reports CPU usage for individual programs and the system as a whole. It uses AIX trace, and includes an offline mode as a trace file post-processor. This command is a useful tool for anyone with a Java, C, C++, or FORTRAN program that might be CPU-bound and who wants to know which sections of the program are most heavily using the CPU, including object files, processes, threads, user mode subroutines, kernel mode subroutines, shared library subroutines, and program

source lines. The **tprof** command also reports the fraction of time the CPU is idle. These reports can be useful in determining CPU usage in a global sense.

Profiling concerns how much CPU time is used by subroutines. Micro-profiling concerns CPU time used by specific program source lines. To enable the former, no executable programs need to be modified, but for the latter, it is necessary to recompile in Version 5.1. In Version 5.2, recompilation is not necessary if a list file is available. Although best results for micro-profiling are achieved with both the list file and the source code available.

tprof support for Java profiling

In AIX 5L Version 5.1, the **tprof** command has been enhanced to do subroutine or method-level profiling for Java applications. The Java Virtual Machine Profiling Interface (JVMPPI), a new feature supported by Java 1.2 or later, has been enhanced to do class and method-level profiling for Java applications.

The **-j** flag was added to **tprof** to enable profiling for Java applications. The profiling report generated by **tprof** for Java applications is similar to that of a standard **tprof** profiling report.

The following example shows the profiling of a Java application named hello:

```
# tprof -j hello -x /usr/java130/bin/java -Xrunjpa hello
Starting Trace now
Starting java -Xrunjpa hello
Mon Mar 12 14:41:19 2001
System: AIX server1 Node: 5 Machine: 000BC6FD4C00
```

```
Big brother is watching you
Trace is done now
* Samples from __trc_rpt2
* Reached second section of __trc_rpt2
```

The profiling report adds a new column named JAVA. This column exists only if the **-j** option is set.

```
# more __hello.all
```

Process	PID	TID	Total	Kernel	User	Shared	Other	JAVA
=====	===	===	=====	=====	=====	=====	=====	=====
java	27726	60755	158	30	0	122	4	2
java	27726	60755	2	2	0	0	0	0
=====	===	===	=====	=====	=====	=====	=====	=====
Total			160	32	0	122	4	2

Segment :: 3 4

Process	FREQ	Total	Kernel	User	Shared	Other	Java
---------	------	-------	--------	------	--------	-------	------

=====	===	=====	=====	=====	=====	=====	=====
java	2	160	32	0	122	4	2
=====	===	=====	=====	=====	=====	=====	=====
Total	2	160	32	0	122	4	2

Total System Ticks: 1469 (used to calculate function level CPU)

Total JAVA ticks: 2 (ticks accumulated in Java Segment)

Total ticks for hello (JAVA) = 2

Class Name	Ticks	%	Source	Class ID
=====	=====	=====	=====	=====
java/io/OutputStreamWriter	1	0.1	OutputStreamWriter.java	3008f568
java/io/BufferedWriter	1	0.1	BufferedWriter.java	3008f178

Profile: java/io/OutputStreamWriter (OutputStreamWriter.java)

Method Name	Ticks	%	Method ID	Load Addr	Size
=====	=====	=====	=====	=====	=====
write[[CII)V	1	0.1	3454b8d8	346b0fec	7ac

Profile: java/io/BufferedWriter (BufferedWriter.java)

Method Name	Ticks	%	Method ID	Load Addr	Size
=====	=====	=====	=====	=====	=====
ensureOpen[()]V	1	0.1	34554ed8	346af3bc	314

New tprof implementation (5.2.0)

Version 5.2 introduces a completely new implementation of **tprof** using the trace and SymLib APIs. The trace APIs provide the interface to decode trace files. The SymLib APIs are the interface to the SymLib library, which contains the routines to capture and retrieve the symbol information for the kernel, kernel extension, shared libraries, and user applications.

This replaces the use of **genkld** and **genkex** for obtaining the loading addresses of shared libraries and kernel extension, and the use of **stripnm** to obtain symbol information for object files (executeables, libraries, kernel extensions, and kernel).

tprof enhancement (5.2.0)

The **tprof** command has been enhanced to include the following new features:

- ▶ Rename all input and output files using Rootstring.* names.
- ▶ Supports multiple program profiling and micro-profiling in a single pass.

- ▶ Full thread support, including thread breakdown profiles within one or more processes.
- ▶ Optional instruction-level annotation of listing file.
- ▶ Detailed address-level report.
- ▶ Improved front-end options to collect trace and name mapping information.
- ▶ A re-postprocessing mode that supports online and offline data collection.
 - Optional cooking produces processed trace and symbol name files.
- ▶ Enhanced symbol mapping replaces gennames format.
 - Uses new **gensyms** command offline
 - Online mode generates same format when cooking is selected
- ▶ Performance and other improvements.

Assembly level program profiling is now available. This is referred to as nano profiling. If a .list file is provided, **tprof** will profile down to assembly lines when micro-profiling is turned on. Therefore, if the source and a list file is available, the microfile reports will contain hot lines broken down by source line and each source line broken down to assembly.

The Java Profiler Agent has been redesigned to use the APIs provided by SymLib to dump the Java classes and method information in an ASCII file using the gensyms format. The **tprof** command uses the SymLib APIs to read the java symbol file and for java symbol lookups.

Profiling options

The **tprof** design is multi-threaded and has three phases, which are:

- | | |
|-------------------|--|
| Collection | Starts trace utility and collects trace events. |
| Processing | Processes the events and finds the type of trace hooks, using the appropriate callbacks to process them. |
| Reporting | Generates user-friendly reports. |

The **tprof** command now has the following modes of operation:

- ▶ Real time

The collection and processing phases work in parallel. Once they are complete, the reporting takes over and symbolic information is collected by **tprof** and a report (named Rootstring.prof) is generated.
- ▶ Automated offline

The **tprof** command starts tracing and logs the trace and gensyms output to files. The files are then processed in the same way as in real time mode. File

names created in the current working directory are `RootString.syms` and `RootString.trc[-cpuid]` unless the `-c` flag is specified, which creates cooked files. In this case file names are `RootString.csyms` and `RootString.ctrac[-cpuid]`.

► Manual offline

This mode can post-process regular trace and symbol files previously captured. These files can be produced by either the automated offline mode with no cooking or from the manual running of the **trace** and **gensyms** commands.

► Post-processing

For this mode to run, a previous call to `tprof` must have created cooked (pre-processed) files, which **tprof** can process much faster. File names created in this case are `RootString.csyms` and `RootString.ctrac[-cpuid]`. These files are created by any of the three previous modes when specifying the `-c` flag.

The syntax of the **tprof** command has changed considerably with Version 5.2 and is as follows:

```
tprof [ -c ] [ -C { all | CPUList } ] [ -d ] -D [ -e ] [ -F ] [ -j ] [ -k ]
[ -l ] [ -m ObjectsList ] [ -M SourcePathList ] [ -p ProcessList ]
[ -P { all | PIDsList } ] [ -s ] [ -S SearchPathList ] [ -t ] [ -T BufferSize ]
[ -u ] [ -v ] [ -V VerboseFileName ] [ -z ] { { -r RootString } |
{ [ -A { all | CPUList } ] [ -r RootString ] -x Program } }
```

Examples of **tprof** in action are shown below, with various options:

► Real-time mode trace (`-x` but no `-A`) output will be in `find.prof` in the current working directory:

```
#tprof -skeuj -x find /usr -name file
Tue Sep 3 15:39:45 2002
System: AIX 5.2 Node: server1 Machine: 0001810F4C00
Starting Command find /usr -name file
/usr/bin/file
stopping trace collection.
Generating find.prof
```

► Automated offline mode trace (`-x` and `-A` flag specified) with cooking; files mentioned in the command output are in the current working directory.

```
#tprof -c -A all -x find /usr -name file
Starting Command find /usr -name file
/usr/bin/file
stopping trace collection.
Tue Sep 3 15:41:19 2002
System: AIX 5.2 Node: server1 Machine: 0001810F4C00
Generating find.ctrac
```

```
Generating find.prof
Generating find.csyms
```

- ▶ Automated offline mode trace, with per-CPU profiling and overwrite of existing generated cooked files (using the -F flag). If this was specified without the -x flag, it would force the manual offline mode (which would process cooked files RootString.ctrc and RootString.csyms).

```
#tprof -c -A all -C all -F -x find /usr -name file
Starting Command find /usr -name file
/usr/bin/file
stopping trace collection.
Tue Sep 3 15:47:12 2002
System: AIX 5.2 Node: server1 Machine: 0001810F4C00
Generating find.ctrc-0
Generating find.ctrc-1
Generating find.ctrc-2
Generating find.ctrc-3
Generating find.ctrc
Generating find.prof-3
Generating find.prof-1
Generating find.prof-0
Generating find.prof-2
```

- ▶ Depending on the type of trace files available, cooked or non-cooked, the following command would run a manual offline report or a post-processing report (neither -A or -x are specified):

```
tprof -r find
Tue Sep 3 16:31:52 2002
System: AIX 5.2 Node: server1 Machine: 0001810F4C00
Generating find.prof
```

This command will run in post-processing mode if cooked files are found and it will run in manual offline mode if they are in non-cooked format. If the command were run with the -F flag, and both non-cooked and cooked files exist, the report would be generated using the non-cooked trace files.

The following section details the finer points of post-processing and manual offline mode.

Clarifying manual offline mode and post-processing mode

There are two ways to create reports from already existent trace files: Manual offline mode or post-processing mode. The **tprof** command will look for and process cooked files (RootString.csyms and RootString.ctrc) over non-cooked trace files (RootString.syms and RootString.trc).

If both file types exist, the cooked files will be processed so the report will be run in post-processing mode (as this uses cooked files). If both file types exist the but

manual offline mode is required (reports generated from non-cooked trace files), the user must specify the **-F** flag, as this forces **tprof** to use the manual offline mode and hence the non-cooked trace file format. Figure 7-4 illustrates the logic behind this.

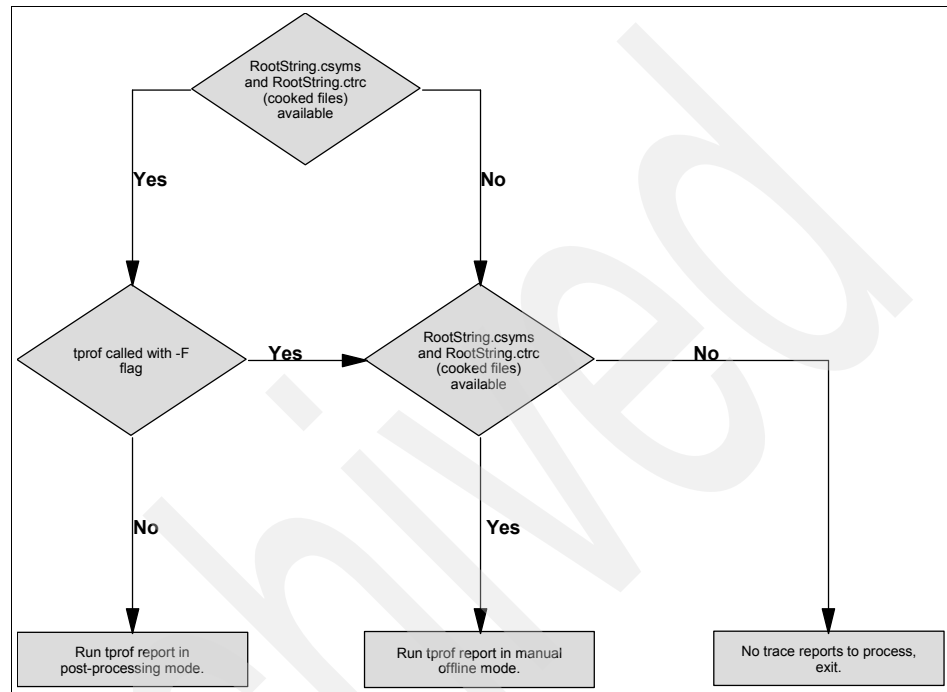


Figure 7-4 Logic flow for post-process mode and manual offline mode

Below is the **ls** command output of the directory containing the cooked files and the directory containing the non-cooked files for the trace of the **find** command used in this example. This shows the size requirements of each. It is worth noting the size of **find.csyms** and **find.syms**. The time difference for the creation of the report files between cooked and non-cooked trace files is significant, in that the cooked files were much quicker in their processing:

► Cooked files:

```

-rw-r--r-- 1 root system 2164 Sep 03 17:18 find.prof
-rw-r--r-- 1 root system 530893 Sep 03 17:18 find.csyms
-rw-r----- 1 root system 16384 Sep 03 17:18 find.ctrc
  
```

► Non-cooked files:

```

-rw-rw-rw- 1 root system 3568 Sep 03 17:19 find.trc
-rw-rw-rw- 1 root system 8340 Sep 03 17:19 find.trc-3
-rw-rw-rw- 1 root system 8488 Sep 03 17:19 find.trc-2
  
```

-rw-rw-rw-	1	root	system	8276	Sep 03 17:19	find.trc-1
-rw-rw-rw-	1	root	system	8368	Sep 03 17:19	find.trc-0
-rw-r--r--	1	root	system	21699553	Sep 03 17:19	find.syms
-rw-r--r--	1	root	system	1868	Sep 03 17:20	find.prof

Additional tprof features

The **tprof** command now allows multiple process profiling with the use of the **-p** flag. When using a flag, either one process or a process list can be specified.

The **-T** flag can be used to specify the trace buffer size (in realtime or automated modes).

If multiple reports are required, it is best to specify the **-c** flag to enable the output files to be cooked, as **tprof** is able to process these file faster than standard files.

Enhanced **tprof** is able to generate old style reports for backward compatibility, with the use of the **-z** flag, which in addition to default reports CPU usage in ticks and also adds the address and bytes column in subroutine reports.

7.1.15 The gensyms command (5.2.0)

The **gensyms** command is similar to the **gennames** command. It provides a mapping between memory addresses and names. This information is needed for the **tprof** command running in offline mode. In this case the **tprof** command needs a file like *filename.syms*. This file can be generated as in the following example:

```
gensyms >/tmp/filename.syms
```

7.1.16 The pstat command (5.2.0)

The **pstat** command, which displays many system tables such as a process table, inode table, or processor status, has been ported to AIX 5L Version 5.2 from AIX Version 4.3.3 with the same functionality. This command was missing in previous versions of AIX 5L.

7.1.17 CPU Utilization Reporting Tool (curt) (5.2.0)

The **curt** tool takes an AIX trace file and an optional address mapping file as input and produces a number of statistics related to processor (CPU) utilization and process/thread activity. It works with both uniprocessor and multiprocessor AIX traces if the processor clocks are properly synchronized.

The **curt** tool is contained in the bos.perf.tools fileset.

The syntax of the **curt** tool is as follows:

```
curt -i inputfile [-o outputfile] [-n gennamesfile] [-m trcnmfile]
      [-a pidnamefile] [-f|-l timestamp] [-ehpst]
```

The most important flags for the **curt** command are described in Table 7-3.

Table 7-3 The **curt** command flags

Flag	Description
-i <i>inputfile</i>	Specifies the input AIX trace file to be analyzed
-o <i>outputfile</i>	Specifies the output file (default is stdout)
-n <i>gennamesfile</i>	Specifies a names file produced by gennames
-m <i>trcnmfile</i>	Specifies a names file produced by trcnm
-a <i>pidnamefile</i>	Specifies a PID to process name mapping file
-f <i>timestamp</i>	Starts processing trace at <i>timestamp</i> seconds
-l <i>timestamp</i>	Stops processing trace at <i>timestamp</i> seconds
-e	Outputs elapsed time information for system calls
-p	Outputs detailed process information
-s	Outputs information about errors returned by system calls
-t	Outputs detailed thread-by-thread information
-h	Displays usage text (this information)

The AIX trace file, which is gathered using the **trace** command, should contain at least the trace events (trace hooks) listed below. These are the events **curt** looks at to calculate its statistics:

```
HKWD_KERN_SVC, HKWD_KERN_SYSCRET, HKWD_KERN_FLIH, HKWD_KERN_SLIH,
HKWD_KERN_SLIHRET, HKWD_KERN_DISPATCH, HKWD_KERN_RESUME, HKWD_KERN_IDLE,
HKWD_SYSC_FORK, HKWD_SYSC_EXECVE, HKWD_KERN_PIDSIG, HKWD_SYSC_EXIT,
HKWD_SYSC_CRTHREAD
```

This means that, if you specify the -j flag on your **trace** command, you must include these numbers for **curt**:

```
-j 100,101,102,103,104,106,10C,119,134,135,139,200,465
```

Or you can use **-J curt** instead.

The report **curt** creates has the following content:

- **curt** and AIX trace information

- ▶ System summary
- ▶ Per-processor summary
- ▶ Application and kernel summary
- ▶ kproc summary
- ▶ System calls summary
- ▶ First level interrupt handler (FLIH) summary
- ▶ Second level interrupt handler (SLIH) summary
- ▶ Detailed process information, if -p is specified
- ▶ Detailed thread information, if -t is specified

For example, to take a five-second trace and create a report with the **curt** command, run the following command sequence:

```
trace -aJ curt -o /mypath/trcfile; sleep 5; trcstop
curt -i /mypath/trcfile
```

The output produced by the **curt** command is similar to the following:

```
Run on Mon Sep 16 10:58:22 2002
Command line was:
curt -i /var/adm/ras/trcfile
----
AIX trace file name = /var/adm/ras/trcfile
AIX trace file size = 556024
AIX trace file created = Mon Sep 16 10:57:07 2002

Command used to gather AIX trace was:
  trace -aJ curt
```

System Summary			
processing total time (msec)	percent total time (incl. idle)	percent busy time (excl. idle)	processing category
906.91	1.05	1.35	APPLICATION
57.37	0.07	0.09	SYSCALL
12.77	0.01	0.02	KPROC
66062.02	76.63	98.55	FLIH
0.00	0.00	0.00	SLIH
5.49	0.01	0.01	DISPATCH (all procs. incl. IDLE)
2.65	0.00	0.00	IDLE DISPATCH (only IDLE proc.)
67031.79	77.75	100.00	CPU(s) busy time


```

19182.81      22.25      IDLE
-----
86214.60      TOTAL
Avg. Thread Affinity =      0.95

...

Application Summary (by Tid)
-----
-- processing total (msec) --      -- percent of total processing time
--
combined application      syscall      combined application      syscall
name (Pid Tid)
=====
=====
839.9648      838.3623      1.6025      0.9743      0.9724      0.0019
ndpd-host(7234 12137)
21.3272      10.1350      11.1922      0.0247      0.0118      0.0130
dtterm(8654 20413)
17.2555      6.3741      10.8814      0.0200      0.0074      0.0126
trcstop(13982 41211)

...

```

7.1.18 Simple Performance Lock Analysis Tool (splat) (5.2.0)

The Simple Performance Lock Analysis Tool (**splat**) is a software tool that post-processes AIX trace and gennames output files to produce reports on all possible types of locking contention (kernel simple locks, kernel complex lock, mutex, condition variables, rwlocks).

The **splat** tool is contained in the bos.perf.tools fileset.

The syntax of the **splat** tool is as follows:

```

splat -i file [-n file] [-o file] [-k kexList] [-d[bfta]] [-l address]
        [-c class] [-s[acelmsS]] [-C#] [-S#] [-t start] [-T stop]
splat -h [topic]
splat -j

```

The description of the most important flags is provided in Table 7-4.

Table 7-4 The *splat* command flags

Flag	Description
-i <i>inputfile</i>	Specifies the input AIX trace file to be analyzed.
-o <i>outputfile</i>	Specifies the output file (default is stdout).

Flag	Description
-n <i>namefile</i>	Specifies a names file produced by gennames
-d <i>detail</i>	<i>detail</i> can be one of: [b]asic: summary and lock detail (default) [f]unction: basic + function detail [t]hread: basic + thread detail [a]ll: basic + function + thread detail
-t <i>starttime</i>	Time offset in seconds from the beginning of the trace to the start of analyzing trace data
-T <i>stoptime</i>	Time offset in seconds from the beginning of the trace to the stop of analyzing trace data
-h [<i>topic</i>]	Helps on usage or a specific topic
-j	Prints a list of trace hooks used by splat

The **splat** command takes as primary input an AIX trace file that has been collected with the AIX **trace** command. Before analyzing a trace with **splat**, you need to make sure that the trace is collected with an adequate set of hooks, including the ones given when running the **splat -j** command. To collect the trace with the adequate set of hooks one may also specify the -J splat flag to the **trace** command. These hooks include several lock and unlock trace events.

Capturing these lock and unlock trace events can cause serious performance degradation due to the frequency that locks are used in a multiprocessor environment. Therefore, lock trace event reporting is normally disabled. In order to enable lock trace event reporting, the following steps must be taken before a trace can be collected, which will include lock trace events that **splat** requires:

1. **bosboot -ad /dev/hdisk0 -L**
2. **shutdown -Fr**
3. **locktrace -S**
4. **mkdir temp.lib; cd temp.lib**
5. **ln -s /usr/ccs/lib/perf/libpthreads.a**
6. **export LIBPATH=\$PWD:\$LIBPATH**

Steps 1 and 2 enable the kernel-lock class information in the trace hooks and are optional (see the **locktrace** command for details). Step 3 enables kernel-lock tracing, whereas steps 4–6 enable the user-lock tracing.

The report **splat** creates has the following content:

- Report summary

- ▶ Lock summary
- ▶ Lock detail
- ▶ Function detail
- ▶ Thread detail

For example, to take a five-second trace and create a report with the **splat** command run, the following command sequence:

```
trace -aJ splat -o /mypath/trcfile; sleep 5; trcstop
splat -i /mypath/trcfile
```

The following shows an excerpt from the output produced by the **splat** command:

```
splat Cmd:      splat -i /var/adm/ras/trcfile

Trace Cmd:  trace -aJ splat
Trace Host: server2 (000BC6FD4C00) AIX 5.2
Trace Date: Mon Sep 16 11:41:27 2002
```

```
Elapsed Real Time:      2.215229
Number of CPUs Traced:  4          (Indicated):1
Cumulative CPU Time:    8.860915
```

...

Lock Activity w/Interrupts Enabled (mSecs)

SIMPLE	Count	Minimum	Maximum	Average	Total
++++++	++++++	+++++	+++++	+++++	+++++
LOCK	140	0.000675	0.765470	0.059083	8.271590
SPIN	0	0.000000	0.000000	0.000000	0.000000
UNDISP	0	0.000000	0.000000	0.000000	0.000000
WAIT	0	0.000000	0.000000	0.000000	0.000000
PREEMPT	0	0.000000	0.000000	0.000000	0.000000

...

7.1.19 Perfstat API library (5.1.0 and 5.2.0)

A set of new APIs is available for easy access to kernel performance metrics. The APIs are in the `bos.perf.libperfstat` fileset. The goal of these APIs is to eliminate the need for an ISV to use `/dev/kmem` and avoid dependencies on kernel data structures, which can change from release to release. The APIs will, most likely, be enhanced in future releases, but binary compatibility will be preserved, therefore virtually eliminating the need for ISVs to port their system

monitoring tools to each new AIX release. The performance APIs are provided in Table 7-5.

Table 7-5 New performance APIs

API	Purpose
perfstat_cpu	Retrieves individual CPU usage statistics (5.1.0)
perfstat_cpu_total	Retrieves global CPU usage statistics (5.1.0)
perfstat_disk	Retrieves individual disk usage statistics (5.1.0)
perfstat_disk_total	Retrieves global disk usage statistics (5.1.0)
perfstat_diskadapter	Retrieves individual disk adapter usage statistics (5.2.0)
perfstat_memory_total	Retrieves global memory usage statistics (5.1.0)
perfstat_netinterface	Retrieves individual network interface usage statistics (5.1.0)
perfstat_netinterface_total	Retrieves global network interface usage statistics (5.1.0)
perfstat_protocol	Retrieves different protocol types' statistics, such as ICMP, ICMPv6, IP, IPv6, TCP, UDP, RPC, NFS, NFSv2, and NFSv3 (5.2.0)
perfstat_pagingspace	Retrieves individual paging space usage statistics (5.2.0)
perfstat_alloc	Retrieves different allocation counts depending on their size (5.2.0)

At the time of writing, the perfstat_diskadapter API does not support MPIO devices.

7.1.20 Xprofiler analysis tool (5.2.0)

The X-Windows-based profiler (Xprofiler) is now included with the AIX 5L Version 5.2 operating system. Xprofiler is a tool that allows you to analyze your parallel and serial applications. It uses procedure profiling information to construct a graphical display of the functions in your application. The graphical user interface (GUI) gives you a general overview of your application and allows you to focus on CPU-intensive sections of your application.

In order to enable profiling, you must compile and link your application with the -pg compiler flags. When your application executes, the CPU usage data is written to one or more files. Serial applications generate only one output file

named gmon.out, while parallel applications generate multiple output files with the name gmon.out.XX, where XX is the task ID assigned by the parallel operating environment (POE). An overview of preparing your application for profiling can be found in the following example:

```
$ cc -pg -c func1.c
$ cc -pg -c func2.c
$ cc -pg func1.o func2.o -o mytest
$ mytest
program output removed
...
$ ls gmon.out*
gmon.out
xprofiler mytest gmon.out
```

To install Xprofiler, you must install the ppe.xprofiler filesset from the AIX installation media. The Xprofiler command is located at /usr/bin/xprofiler. See Figure 7-5 on page 424 for an example of the Xprofiler application displaying the execution statistics of an application called mytest.

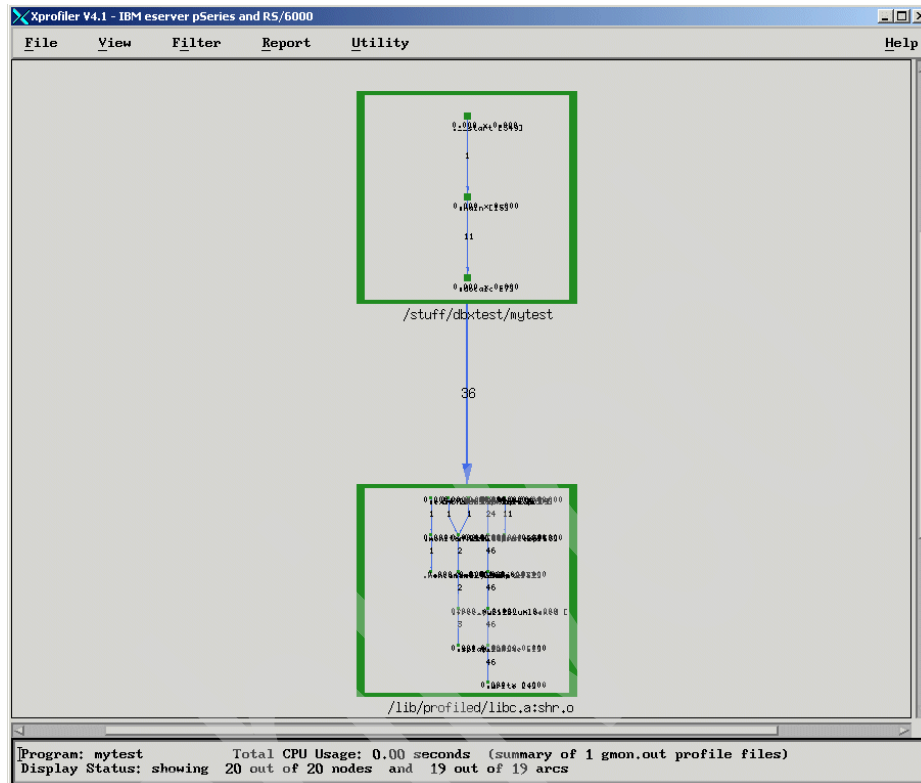


Figure 7-5 Xprofiler applications

7.2 AIX tuning framework (5.2.0)

Prior to AIX 5L Version 5.2, all the performance parameters that can be set by the **vm tune**, **sched tune**, **no**, or **nfso** command were lost at the next system reboot. The syntax and the output of those commands were also completely different. In AIX 5L Version 5.2, a complete review of the performance management has been made and the following enhancements provided:

- ▶ Support of permanent and reboot values for tuning parameters in a new `/etc/tunables` directory. This directory consists of the following files:
 - `/etc/tunables/nextboot` ASCII file using a stanza format with one stanza per command and one line per parameter to be changed from its default value. An additional information stanza provides general information about the file.

- /etc/tunables/lastboot contains values for each parameter set during the last reboot. The default values are marked.
- /etc/tunables/lastboot.log logs all changes made or impossible to make. The lastboot file contains a checksum for the lastboot.log to detect file corruption.
- Other files can be stored in this directory; however, only the nextboot file will be applied at boot time.
- Files can be copied from one machine to another, applied, edited, or created using SMIT, Web-based System Manager, or an editor such as vi.
- ▶ New commands have been created to manage these files, as discussed in the following section.
- ▶ All the tuning commands have been enhanced to have a consistent syntax and interface. They all interact with the /etc/tunables/nextboot file. These enhancements are part of the bos.perf.tune fileset.

7.2.1 The /etc/tunables commands

To manage its files in the /etc/tunables directory, new commands have been added to AIX. They are as follows:

- ▶ The **tuncheck** command
This command validates a file either to be applied immediately or at reboot time (-r flag). It checks the ranges, dependencies, and prompts to run bosboot if required. Run this command if you copy a file to a new system, or edit it with an editor such as vi.
- ▶ The **tunsave** command
This command saves all current values to a file, including optionally the nextboot file.
- ▶ The **tunrestore** command
This command applies values from a file, either immediately, or at the next reboot (-r flag). With the -r flag, it validates and copies the file over the current nextboot file.
- ▶ The **tundefault** command
This command resets all parameters to their default value. It can be applied at the next reboot with the -r flag.

7.2.2 Tuning commands enhancement

All the tuning commands (**vmo**, **ioo**, **schedo**, **nfso**, and **no**) now have common flags, described in Figure 7-6 on page 426.

Table 7-6 Common flags of the tuning commands

Flag	Description
-a	Displays values for all tunable parameters, one per line value.
-h	Displays command help or displays help about tunables.
-d	Resets tunables to default value.
-D	Resets all tunables to their default value.
-o	Tunable=value, sets tunable to specified value.
-p	Makes changes apply to both current and reboot values; modify the /etc/tunables/nextboot file in addition to updating the current value.
-r	Makes changes apply to reboot values only. Only modify the /etc/tunables/nextboot file.
-L	Prints header and characteristics of one or all tunables, one tunable per line.

The **vmtune** and the **schedtune** command, which use a syntax very incompatible with the syntax shown in the previous table, are being phased out. The **vmtune** command is replaced by the two new **vmo** and **ioo** commands. The **schedtune** command is replaced by the new **schedo** command. For compatibility reasons, the **vmtune** command and the **schedtune** command have been replaced by a shell script that calls the new commands.

The following example lists the **vmo** command values for the system including the current, default, and next reboot values; the minimum and the maximum value that a parameter can take; the unit of the value; the value type; and the dependencies.

```
#vmo -L
Name          Current  Default  Reboot   Minimum  Maximum  Unit      Type
Dependencies
value         value    value    value    value    value
memory_frames 262144   4000     4992     4992     8        204800    4KB pages S
minfree
maxfree

memory_frames 5000     128      128      16       204800    4KB pages D
maxfree
minfree

memory_frames
```


minperm%	20	20	20	1	100	% memory	D
maxperm%							
minperm	48630		48630				S
maxperm%	80	80	80	1	100	% memory	D
minperm%							
maxclient%							
maxperm	194520		194520				S
strict_maxperm	0	0	0	0	1	boolean	D
maxpin%	80	80	80	1	99	% memory	D
maxpin	209716		209716				S
maxclient%	80	80	80	1	100	% memory	D
maxperm%							
lrubucket	131072	131072	131072	65536		4KB pages	D
defps	1	1	1	0	1	boolean	D
nokilluid	0	0	0	0	4294967295	uid	D
numpsblks	131072		131072			4KB pages	S
npskill	1024	1024	1024	1	131071	4KB pages	D
npswarn	4096	4096	4096	0	131071	4KB pages	D
v_pinshm	0	0	0	0	1	boolean	D
pagecoloring	0	0	0	0	1	boolean	B
framesets	0	2	2	1	10		B
mempools	0	1	1	1	4		B
lgpg_size	0	0	0	0	268435456	bytes	B
lgpg_regions							
lgpg_regions	0	0	0	0			B
lgpg_size							
num_spec_dataseg	n/a	0	0	0			B
spec_dataseg_int	n/a	512	512	0			B
memory_affinity	n/a	0	0	0	1	boolean	B

In the previous example, note that the Type field is shown with different values. The S means that this parameter is static and cannot be changed, the D means that the parameter can be change dynamically, the R means a reboot is necessary to apply the new value to the system, the B means bosboot must be called and the machine rebooted to apply the new value to the system, and the M means that the file systems need to be unmounted and mounted. The current and reboot values of the above example have been changed with the following command:

```
vmo -p -o minfree=4000 -o maxfree=5000.
```

To display some of the fields of the **vmtune -a** command such as fsbufwaitcnt, use the **vmstat -v** command.

7.2.3 Web-based System Manager access

The Web-based System Manager has been enhanced to support the new performance tuning commands.

The Figure 7-6 is the main panel for system tuning.

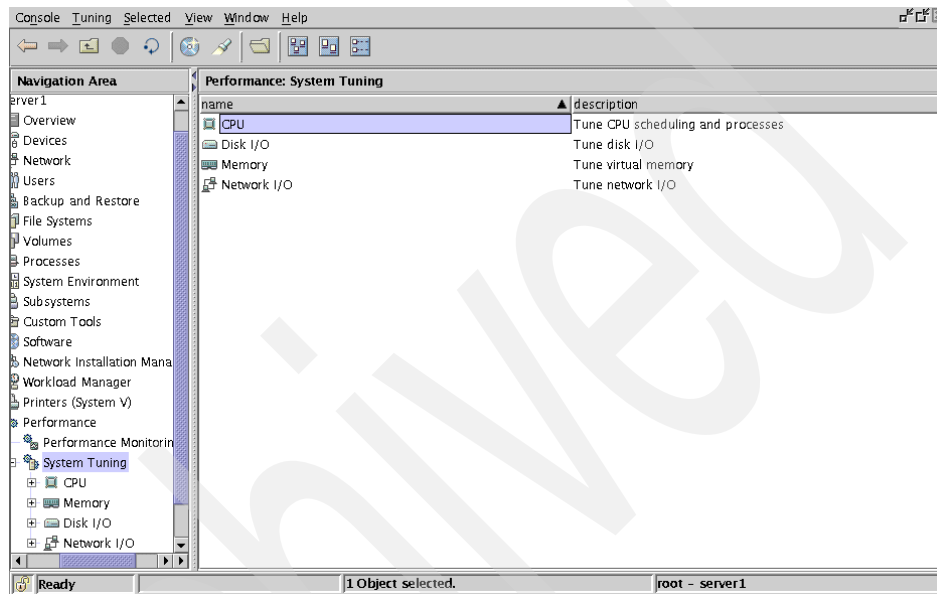


Figure 7-6 System performance main panel

Figure 7-7 on page 429 shows the I/O parameter tuning table.

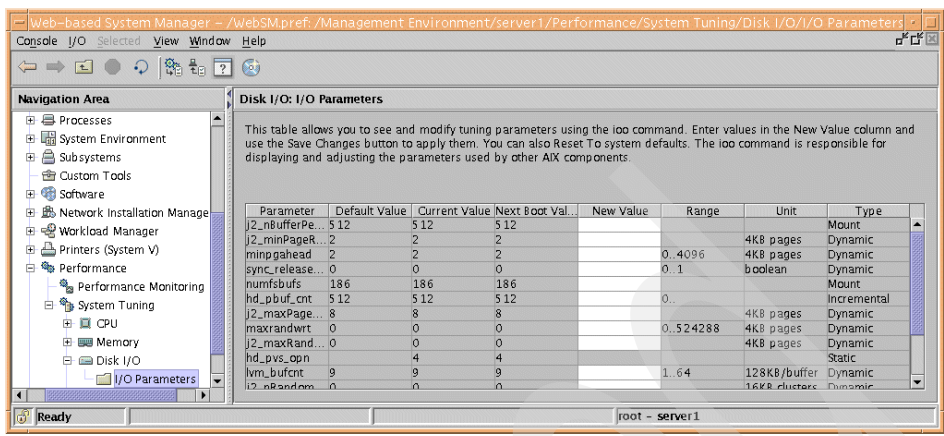


Figure 7-7 I/O parameters

7.2.4 SMIT access

A new SMIT panel handles the new AIX performance management commands. It can be accessed with the smitty tuning fast path, as shown in Figure 7-8.

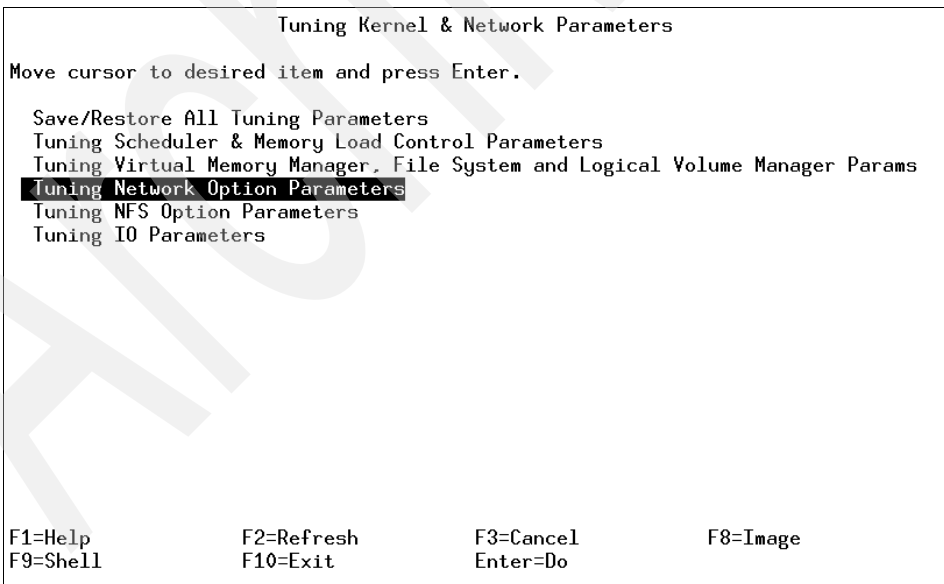


Figure 7-8 The smitty tuning fast path

Figure 7-9 shows how to reset to next boot the default network values using SMIT.

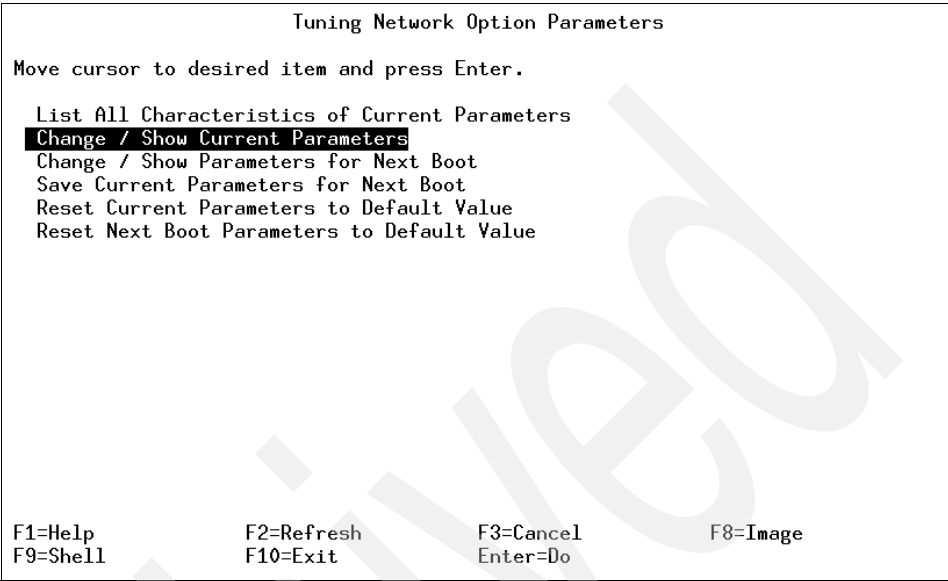


Figure 7-9 Tuning Network Option Parameters dialog

Figure 7-10 shows how to display the network parameters.

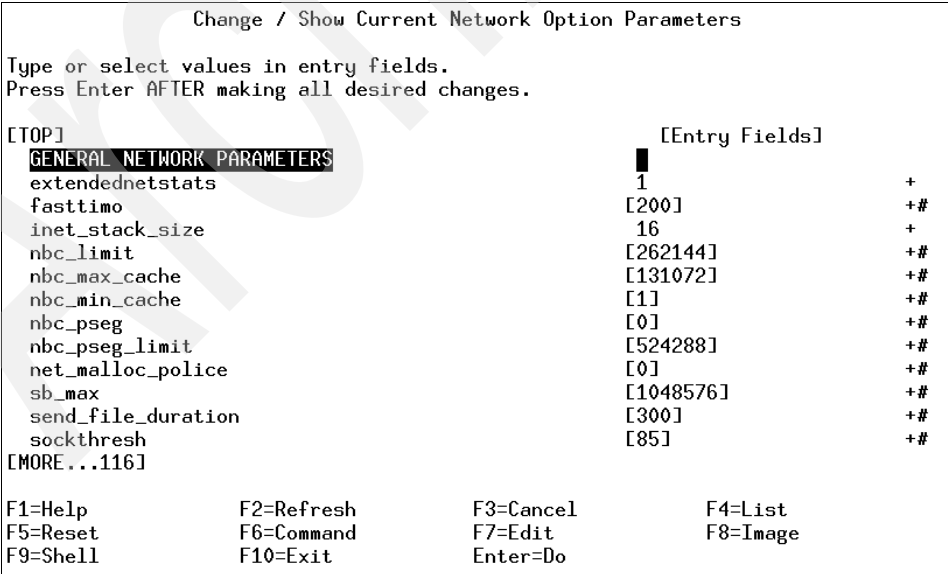


Figure 7-10 Change/Show Network Current Option Parameters dialog

Networking

AIX 5L provides many enhancements in the networking area. They are described in this chapter. Topics include:

- ▶ Quality of service
- ▶ BIND Version 9
- ▶ TCP/IP enhancements
- ▶ Virtual IP support
- ▶ Network buffer cache
- ▶ Mobile IPv6
- ▶ SMB enhancements
- ▶ IKE enhancements
- ▶ ATM enhancements
- ▶ EtherChannel support
- ▶ IPv6

8.1 Quality of Service support

A new method for regulating network traffic flows named Quality of Service (QoS) was introduced in AIX Version 4.3.3. The demand for QoS arises from applications such as digital media or real-time applications and the need to manage bandwidth resources for arbitrary administratively defined traffic classes.

AIX 5L further enhances the QoS implementation to support overlapping policies in the QoS manager. Directly related to this feature is the new and additional capability to specify a priority for a given policy. To improve the manageability of a QoS configuration, AIX 5L also offers four new commands to add, delete, modify, and list QoS policies.

8.1.1 QoS manager overlapping policies

The QoS implementation in AIX 5L offers, among other features, a policy-based network traffic categorization and conditioning for the Differentiated Services (DS) and Integrated Services (IS) QoS model. In order for network equipment to provide QoS features from various vendors that interoperate correctly, it is necessary to standardize the underlying policy scheme for QoS. The AIX policy schema is based on the Internet Draft, draft-rajana-policy-qosschema-01.txt, of the Internet Engineering Task Force (IETF).

A policy condition is a characteristic of an IP packet, and a policy action is an action the packet receives when it meets a policy condition. A policy condition is defined by five characteristics of a packet. They are source IP address, source port number, destination IP address, destination port, and protocol type (TCP or UDP). A policy action includes token bucket parameters and a TOS byte value defining in-profile traffic.

From an administrator's point of view, a policy is essentially a collection of configuration parameters to regulate certain types of traffic flow.

There are two core components of the QoS subsystem that are relevant to the policy-based networking function:

- ▶ QoS kernel extension (/usr/lib/drivers/qos)

The QoS kernel extension resides in /usr/lib/drivers/qos and is loaded and unloaded using the cfgqos and ucfgqos configuration methods. This kernel extension enables QoS support and provides the QoS manager functionality.

- ▶ Policy agent (/usr/sbin/policyd)

The policy agent is a user-level daemon (/usr/sbin/policyd). It provides support for policy management and interfaces with the QoS kernel extension

(QoS manager) to install, modify, and delete policy rules. Policy rules may be defined in the local configuration file (/etc/policyd.conf), retrieved from a central network policy server using LDAP, or both. AIX 5L also offers a command line interface to manage and administer policy rules.

Each policy definition requires a ServicePolicyRules and a ServiceCategories object within the /etc/policyd.conf file. The ServicePolicyRules object establishes the policy condition and the ServiceCategories object determines the policy action. The structure for the ServicePolicyRules object is shown in the following example:

Used conventions:

i : integer value
s : a character string
a : IP address format B.B.B.B
(R) : Required parameter
(O) : Optional parameter

```
ServicePolicyRules s
{
    SelectorTag          s      # Required tag for LDAP Search
    ProtocolNumber       i      # Transport protocol id for the policy rule
    SourceAddressRange   a1-a2
    DestinationAddressRange a1-a2
    SourcePortRange      i1-i2
    DestinationPortRange i1-i2
    PolicyRulePriority    i      # Highest value is enforced first
    ServiceReference      s      # Service category name for this policy
rule
}
```

where

s	(R): is the name of this policy rule
SelectorTag	(R): required only for LDAP to Search object class
ProtocolNumber	(R): default is 0 which causes no match, must explicitly specify
SourceAddressRange	(O): from a1 to a2 where a2 >= a1, default is 0, any source address
SourcePortRange	(O): from i1 to i2 where i2 >= i1, default is 0, any source port
DestinationAddressRange	(O): same as SourceAddressRange
DestinationPortRange	(O): same as SourcePortRange
PolicyRulePriority	(O): Important to specify when overlapping policies exist
ServiceReference	(R): service category this rule uses

Note that the newly introduced attribute `PolicyRulesPriority` and each `ServicePolicyRules` object is associated with a unique instance of the `ServiceCategory` referred to by the `ServiceReference` attribute.

During the start of the QoS subsystem, the policy agent installs the defined policies to be used by the QoS manager. Previous AIX releases took a conservative approach toward overlapping policies by completely disallowing them. This had implications for deployment and actual usage, where the system administrator may want to specify or assume a given ordering between the potentially overlapping policies. In AIX releases prior to AIX 5L, the QoS manager effectively searched for a matching policy in a way that did not allow a priority among the policies.

One example to illustrate the issues related to overlapping policies is as follows.

A customer desires to configure simultaneous policies for application audio (AppA) and application video (AppV). The first application (AppA) may select a valid port number for the source port and a wild card for the destination, while the second application (AppV) selects a wild card for the source port and a valid port number for the destination. The five attributes of the related `ServicePolicyRules` objects (source IP address, source port number, destination IP address, destination port, and either TCP or UDP) that are used by the QoS Manager to identify specific policy rules, may all have fields identical, with the exception of source and destination port for the two applications. When installing the policy definitions for both applications under AIX Version 4.3.3, the second policy in the installation sequence was found to be overlapping, an error was flagged, and the policy was not installed. While the policies were overlapping, if the system allowed the installation of both policies, the two applications would not have assigned conflicting ports. The policies would not have overlapped, because the application (AppA) that uses the source port would not have assigned a destination port overlapping with the second application (AppV) and vice versa.

This may happen with different applications in other scenarios. Even though the policies are allowed to install in practice, they may overlap, so order of policy installation becomes important.

In order to allow the installation of overlapping policies, the order in which the policies are input to the QoS Manager needs to be preserved. The highest priority policy in the overlapping case will be input to the QoS Manager from the policy agent last, and that order is maintained for proper policy enforcement. The last policy installed from the policy agent that matches will be enforced over previously installed policies in the overlapping case.

The policy agent's capability was extended to allow system administrators to set priorities for policies, so that they get installed in a desired order onto the QoS kernel extension. In order to do this, an attribute called `PolicyRulePriority` was

added to the ServicePolicyRules structure. The ServicePolicyRules objects are defined in the /etc/policyd.conf configuration file. The PolicyRulePriority attribute can be set to any positive integer. If no value is specified, the default is set to 0. The absolute value of this attribute has no meaning and only the relative values are important. The policies are installed onto the AIX 5L kernel in the order of the highest priority first. Every time a new policy is added to the policy agent, it is inserted into the policies list based on its priority, and finally the whole list is installed onto the QoS manager stack.

The priority for any specific policy can be specified by manually editing the ServicePolicyRules stanzas in the /etc/policyd.conf policy agent configuration file. Alternatively, you can use the new command line interface as described in 8.1.2, “QoS manager command line support” on page 435.

QoS is an optionally installable feature and packaged with the bos.net.tcp.server fileset.

8.1.2 QoS manager command line support

Beginning with AIX 5L, four new commands are available to add, modify, delete, or list Quality of Service policies. These AIX commands operate on the /etc/policyd.conf policy agent configuration file, so the use of a text editor is not required to manage policies. Once an **add**, **modify**, or **remove** command is executed, the change takes effect immediately and the local configuration file of the policy agent is updated to permanently keep the change. The **list** command will prompt the policy agent to query its internal indexed list to provide the information about ServiceCategories and ServicePolicyRules, which define the active policies. Also, a flag will be available for the command line programs to allow prioritization of policies, so the correct order of enforcement can be determined in the event of a policy overlap. The policy agent must input the policies to the QoS Manager in the order of lowest priority first.

The QoS command line interface consists of the commands provided in the following sections, with their given syntax and usage.

The qosadd command

The **qosadd** command adds the specified service category or policy rule entry in the policyd.conf file and installs the changes in the QoS Manager.

To add a service category or a policy rule:

```
#qosadd
usage: qosadd  -s ServiceCategory  [-t OutgoingTOS] [-b MaxTokenBucket]
               [-f Flow ServiceType] [-m MaxRate] service
usage: qosadd  -s ServiceCategory  -r ServicePolicyRules
               [-l PolicyRulePriority] [-n ProtocolNumber] [-A SrcAddrRange]
```

```
[-a DestAddrRange] [-P SrcPortRange] [-p DestPortRange] policy
```

The qosmod command

The **qosmod** command modifies the specified service category or policy rule entry in the policyd.conf file and installs the changes in the QoS Manager.

To modify an existing service category or policy rule:

```
# qosmod
usage: qosmod -s ServiceCategory [-t OutgoingTOS] [-b MaxTokenBucket]
          [-f Flow ServiceType] [-m MaxRate] service
usage: qosmod -s ServiceCategory -r ServicePolicyRules
          [-l PolicyRulePriority] [-n ProtocolNumber] [-A SrcAddrRange]
          [-a DestAddrRange] [-P SrcPortRange] [-p DestPortRange] policy
```

The qoslist command

The **qoslist** command lists the specified service category or policy rule. The **qoslist** command lists all service categories or policy rules if no specific name is given. The syntax is:

```
#qoslist
usage: qoslist [ServiceCategory][Policy Rule] <policy or service>
```

The qosremove command

The **qosremove** command removes the specified service category or policy rule entry in the policyd.conf file and the associated policy or service in the QoS Manager. The syntax is:

```
#qosremove
usage: qosremove <ServicePolicyRule or ServiceCategory> <policy or service>
```

8.1.3 Quality of Service enhancements (5.2.0)

The Quality of Service component of the AIX network stack has been enhanced to remove its dependency on the policy agent daemon (policyd), dynamic modifications to policy information of connections in flight, and new parameters for the **qosremove** command.

Prior to Version 5.2, the policy agent managed all the policy management information and used a socket to communicate with the kernel. If the policy agent was stopped or ended abnormally, QoS would stop functioning. In Version 5.2, the policy management information is still managed in the policy agent, but the policy agent publishes the policy management information into the QoS manager in the kernel. Because the QoS manager has a copy of the policy management information in pinned memory, QoS will still function if the policy agent is not running.

The **qosadd** command notifies the policy agent about a new service category or policy information. Then the policy agent publishes the new information into the QoS manger and then modifies the `/etc/policyd.conf` file, if that was successful.

The following example shows how to use the **qosadd** command to define a service category named `serviceCategory1` and a QoS policy named `interactive`. The following interactive policy marks all packets for any **telnet** session to the 192.168.1.6 machine with the service category named `serviceCategory1`.

```
# qosadd -s serviceCategory1 -t 10000001 -b 81 -f ControlledLoad -m 41 service
# qosadd -s serviceCategory1 -r interactive -l 2 -n 6 -a 192.168.1.6 -p 23
policy
# qoslist service
ServiceCategories serviceCategory1:
    OutgoingTOS (binary) 10000001
    MaxRate (Kbps) 41
    MaxTokenBucket (Kb) 81
    FlowServiceType 5
# qoslist policy
ServicePolicyRule interactive
    PolicyRulePriority 2
    ProtocolNumber 6
    SourceAddressRange 0.0.0.0
    SourcePortRange 0
    DestinationAddressRange 192.168.1.6
    DestinationPortRange 23
    ServiceReference serviceCategory1
```

The following example shows how to use the **qosadd** command to define a service category named `serviceCategory2` and a QoS policy named `shaper`. The following shaper policy marks all packets for any **ftp** (data) session to the 192.168.1.6 machine with the service category named `serviceCategory2`. Note that the flow service type is 2 (guaranteed), indicating that rate shaping is turned on.

```
# qosadd -s serviceCategory2 -t 10000001 -b 1000 -f Guaranteed -m 1100 service
# qosadd -s serviceCategory2 -r shaper -l 1 -n 6 -a 192.168.1.6 -p 21 policy
# qoslist service
ServiceCategories serviceCategory2:
    OutgoingTOS (binary) 10000001
    MaxRate (Kbps) 1100
    MaxTokenBucket (Kb) 1000
    FlowServiceType 2
# qoslist policy
ServicePolicyRule shaper
    PolicyRulePriority 1
    ProtocolNumber 6
    SourceAddressRange 0.0.0.0
    SourcePortRange 0
```

```
DestinationAddressRange 192.168.1.6
DestinationPortRange 21
ServiceReference serviceCategory2
```

The **qosremove** command now supports the all parameters. This will cause the policy agent to delete all policy and service category entries the QoS manager in the `/etc/policyd.conf`. The following example shows using the **qosremove** command with the all parameters.

```
# qosstat
Policy Rule Handle 1:
Filter specification for rule index 1:
  PolicyRulePriority:                2
  protocol:                        TCP
  source IP addr:                   INADDR_ANY
  destination IP addr:              192.168.1.6
  source port:                      ANY_PORT
  destination port:                 23
Flow Class for rule index 1:
  service class:                    Diff-Serv
  peak rate:                        100000000 bytes/sec
  average rate:                     5248 bytes/sec
  bucket depth:                    10368 bytes
  TOS (in profile):                 129
  TOS (out profile):                0
Statistics for rule index 1:
  total number of connections:      1
  total bytes transmitted:          30
  total packets transmitted:        26
  total in-profile bytes transmitted: 30
  total in-profile packets transmitted: 26

# qosremove all
# qosstat
No rules installed
```

The **qosmod** command and the policy agent have been enhanced to allow modifying any of the QoS fields in the service categories or policy rules. Prior to Version 5.2, the **qosmod** command would only allow you to change the type of service (TOS) field. When a policy is modified with the **qosmod** command, the policy agent will notify the kernel about the new policy. The kernel will have to reclassify all connections using the modified policy. Instead of reclassifying all the connections immediately, the kernel will only reclassify a connection when data is sent or received, to prevent degrading system performance. Connections with frequent traffic will be reclassified quickly while idle connections could take some time. After the policy is successfully modified in the QoS manager, the policy agent will update the `/etc/policyd.conf` file. The following example shows how to modify the destination port for the interactive policy from telnet to ssh, port 22.

```

# qosmod -s serviceCategory1 -r interactive -p 22 policy
# qosstat
Policy Rule Handle 1:
Filter specification for rule index 1:
    PolicyRulePriority:                2
    protocol:                        TCP
    source IP addr:                  INADDR_ANY
    destination IP addr:             INADDR_ANY
    source port:                     ANY_PORT
    destination port:                22
Flow Class for rule index 1:
    service class:                    Diff-Serv
    peak rate:                        100000000 bytes/sec
    average rate:                     5248 bytes/sec
    bucket depth:                     10368 bytes
    TOS (in profile):                 129
    TOS (out profile):                 0
Statistics for rule index 1:
    total number of connections:      0
    total bytes transmitted:           224
    total packets transmitted:         182
    total in-profile bytes transmitted: 224
    total in-profile packets transmitted: 182

```

8.2 BIND 9 enhancements (5.2.0)

AIX 5L Version 5.2 now includes Version 9.02.0 of the Berkeley Internet Name Domain (BIND). The BIND daemon implements the domain name service (DNS) protocols, which maps IP addresses to host names and the reverse. Version 5.2 supports BIND Versions 4, 8, and 9. BIND version 9 includes improvements in DNS security, IPv6 support, DNS protocol enhancements, and support for views.

The DNS security enhancements include DNS security (DNSSEC) and transaction signature (TSIG) support. These extensions provide data integrity and authentication through the use of digital signatures. DNSSEC allows a security-aware client to verify that the data received from a name server is valid and authentic. TSIG uses symmetric keys for server-to-server and administrator-to-server operations such as zone transfers, dynamic updates, and remote administration of the name server daemon. Prior to TSIG, you were only able to restrict these operations by IP address, which has been shown to be insecure.

The IPv6 enhancements include support for two new resource records, A6 and DNAME. Bitstring labels and BIND can answer DNS queries on IPv6 sockets.

The existing DNS protocols such as incremental zone transfer (IXFR), dynamic DNS (DDNS), and Notify have been enhanced. IXFR allows the name server to transfer only the changes in a zone file, not the entire file. DDNS was updated to support BIND 9 and TSIG. Notify was enhanced to allow the master servers to notify the slave servers of zone file updates, reducing the time the master and slave zone files are out of sync.

BIND 9 now supports the concept of views, which allows you to easily set up split DNS servers. Views allow a DNS server to respond differently depending on the address of the client. This is useful when you have a split DNS set up with public and private zone files. With split DNS you would normally have two BIND instances running and administer them separately. With views, servers can serve the private zones to a specific address range and the public zones to another address range.

The following sections show how the average company might install and configure BIND 9 taking advantage of the new features. The company's top level domain name is mycompany.example. One department has sufficient need for its own DNS zone and was assigned mydept.mycompany.example. There are two DNS servers named ns1.mycompany.example and ns2.mycompany.example. Their IP addresses are 192.168.1.5 and 192.168.1.6, respectively. You should configure the master and slave DNS server using the following section. After both BIND servers are running, the master and slave server will be configured independently.

Common example server configuration for BIND 9

By default, AIX 5L Version 5.2 uses BIND Version 4. To change to BIND 9 you need to change the symlinks for /usr/sbin/named and /usr/sbin/nsupdate to point to /usr/sbin/named9 and /usr/sbin/nsupdate9, respectively. Use the following commands to change the symlinks.

```
# ln -sf /usr/sbin/named9 /usr/sbin/named
# ln -sf /usr/sbin/nsupdate9 /usr/sbin/nsupdate
# ls -l /usr/sbin/nsupdate /usr/sbin/named
lrwxrwxrwx  1 root    system      16 Sep 10 00:20 /usr/sbin/named ->
/usr/sbin/named9
lrwxrwxrwx  1 root    system      19 Sep 10 00:20 /usr/sbin/nsupdate ->
/usr/sbin/nsupdate9
```

BIND 9 is now set up as the default DNS server when you start the named subsystem. Before we can start BIND 9, you need to set up the base environment and create the minimal named.conf file. The base directory for DNS in this example is /etc/dns. Use the following commands to set up the BIND environment.

```
# mkdir /etc/dns
# mkdir /etc/dns/master /etc/dns/slave /etc/dns/logs
```

```
# ln -sf /etc/dns/named.conf /etc/named.conf
```

Copy the following section into the file /etc/dns/named.conf:

```
//
// named logging option
//
logging {
    channel security {
        file "logs/security.log";
        print-category yes;
        print-severity yes;
        print-time yes;
    };

    channel messages {
        file "logs/messages.log";
        print-category yes;
        print-severity yes;
        print-time yes;
    };

    // All unspecified categories are sent to channel messages
    category default { messages; default_syslog; default_debug; };

    // Send all messages related to security to security channel
    category security { security; default_syslog; default_debug; };
};

//
// named server options
//
options {
    directory      "/etc/dns";
    dump-file      "logs/named_dump.db";
    pid-file       "named.pid";
    statistics-file "logs/named.stats";
};

// *****
// Zone list (master)
//
zone "." {
    type hint;
    file "master/db.root";
};

zone "0.0.127.in-addr.arpa" {
    type master;
```

```

        file "master/db.127.0.0";
    };

```

Now that the `named.conf` file is set up, you need to get the appropriate root name server list for your environment. Generally, if you are on an intranet or behind firewalls you will need to create your own root zone file. For information on how to do this, refer to the AIX 5L publications. If you are connected to the Internet, download the root server list from the Internic at the following URL and store it in the `/etc/dns/master/db.root` file.

<ftp://ftp.rs.internic.net/domain/named.root>

Copy the following information into the file `/etc/dns/master/db.127.0.0`.

```

$TTL 3600
@ in SOA ns1.mycompany.example. hostmaster.mycompany.example. (
                                1997112100      ; Serial number
                                10800            ; Refresh
                                3600             ; Retry
                                604800          ; Expire
                                3600 )          ; Minimum TTL

                                IN NS   ns1.mycompany.example.
1                                IN PTR  localhost.

```

BIND 9 requires all the master and slave DNS servers to have their time synchronized for the enhanced security features to work. The maximum allowed time skew is five minutes, before DNSSEC and TSIG break. Synchronize your clocks using your preferred method, for example `xnptd`. If you do not synchronize your clock normally, you can perform a quick one-time synchronization of your clock using the `ntpdate` or `setclock` commands. See the following example on how to use these commands to synchronize with host *TIMESERVER*.

```

# ntpdate TIMESERVER
10 Sep 16:43:59 ntpdate[32390]: adjust time server 9.45.125.42 offset -0.076524
sec

```

```

# setclock TIMESERVER
Tue Sep 10 16:44:40 2002

```

Start the DNS server using the `startsrc` command and then use the `lssrc` command to see if the server started properly. If the server did not start, check the log files in `/etc/dns/logs` for information on what did not work. The following example shows how to start the BIND server with the `startsrc` command and how to display the status of the named subsystem with the `lssrc` command.

```

# startsrc -s named
0513-059 The named Subsystem has been started. Subsystem PID is 30604.
# lssrc -ls named
Subsystem      Group          PID           Status

```


named	tcpip	30604	active
Debug	Inactive		
Type	Zone		Source File or Host
master	0.0.127.in-addr.arpa		master/db.127.0.0

Now that the DNS server is started, you need to set up support for the **rndc** (remote name daemon control) command. The **rndc** command allows you to administer the name server remotely. The **rndc** command uses symmetric keys instead of IP addresses to authenticate the administrator. This is done by running the **rndc-confgen** command to generate the configuration stanzas to copy into the `/etc/rndc.conf` and `/etc/named.conf`. The output of the **rndc-confgen** command looks like the following.

```
# /usr/sbin/rndc-confgen -r /dev/random

# Start of rndc.conf
key "rndc-key" {
    algorithm hmac-md5;
    secret "yBt9AGOUdmU/AM7Gbhy2iQ==";
};

options {
    default-key "rndc-key";
    default-server 127.0.0.1;
    default-port 953;
};
# End of rndc.conf

# Use with the following in named.conf, adjusting the allow list as needed:
# key "rndc-key" {
#     algorithm hmac-md5;
#     secret "yBt9AGOUdmU/AM7Gbhy2iQ==";
# };
#
# controls {
#     inet 127.0.0.1 port 953
#         allow { 127.0.0.1; } keys { "rndc-key"; };
# };
# End of named.conf
```

Copy the appropriate section of the **rndc-confgen** command output into the appropriate file. These symmetric keys are sensitive data and the file permissions should only allow root to read `/etc/rndc.conf` and `/etc/dns/named.conf`.

The following stanza configures the **rndc** client with the address, port, and secret key to administer the BIND server. Copy the following into `/etc/rndc.conf`:

```
# Start of rndc.conf
key "rndc-key" {
    algorithm hmac-md5;
    secret "yBt9AG0UDMU/AM7Gbhy2iQ==";
};

options {
    default-key "rndc-key";
    default-server 127.0.0.1;
    default-port 953;
};
# End of rndc.conf
```

The following stanzas configure the BIND server to allow **rndc** access only from localhost and using the correct key. Copy the following into `/etc/dns/named.conf`:

```
key "rndc-key" {
    algorithm hmac-md5;
    secret "yBt9AG0UDMU/AM7Gbhy2iQ==";
};

// Allow RNDc access from localhost with the rndc-key
controls {
    inet 127.0.0.1 port 953
    allow { 127.0.0.1; } keys { "rndc-key"; };
};
```

Protect both of these configurations files with the following command:

```
# chmod 600 /etc/rndc.conf /etc/dns/named.conf
```

Restart the BIND server using the **startsrc** and **stopsrc** commands to have the modifications take effect.

```
# stopsrc -s named
0513-044 The named Subsystem was requested to stop.
# startsrc -s named
0513-059 The named Subsystem has been started. Subsystem PID is 50000.
```

Run the following **rndc** commands to check if the configuration was successful.

```
# rndc status
number of zones: 2
debug level: 0
xfers running: 0
xfers deferred: 0
soa queries in progress: 0
query logging is OFF
server is up and running

# rndc reload
```

To enable the DNS security extensions, you must install the OpenSSL library and then symlink the secure DNS libraries. The OpenSSL RPM can be downloaded from the AIX Toolbox for Linux Applications home page located at the following URL:

<http://www.ibm.com/servers/aix/products/aixos/linux/>

Install the OpenSSL RPM packages by running the following **rpm** commands:

```
# rpm -i openssl-0.9.6e-2.aix4.3.ppc.rpm
# rpm -q openssl
openssl-0.9.6e-2
```

In order for BIND 9 to have access to the correct security libraries, you must now symlink the libcrypto.a and libdns_secure.a libraries using the following commands:

```
# ln -fs /usr/lib/libdns_secure.a /usr/lib/libdns.a
# ln -s /usr/linux/lib/libcrypto.a /usr/lib
```

Restart the BIND server to have changes take effect.

After the DNS master server is set up, complete the same tasks to complete the common configuration of the slave server.

Configuring additional zone files

On the master DNS server (ns1.mycompany.example), you need to add the additional zones required by mycompany. The two forward zones are mycompany.example and mydept.mycompany.example and the reverse zone is 1.168.192.in-addr.arpa. Copy the following three zone files into the specified file located in the /etc/dns/master directory.

```
# cat /etc/dns/master/db.mycompany.example
$TTL 180
@ in SOA ns1.mycompany.example. hostmaster.mycompany.example. (
    2002071802      ; Serial
    10800           ; Refresh after 3 hours
    3600            ; Retry after 1 hour
    604800          ; Expire after 1 week
    180 )           ; TTL in seconds

IN NS ns.mycompany.example.

ns1      IN A 192.168.1.5
ns2      IN A 192.168.1.6

localhost IN A 127.0.0.1

ca       IN A 192.168.1.5
```

```

ldap          IN A      192.168.1.5

# cat /etc/dns/master/db.mydept.mycompany.example
$TTL 180
@ IN SOA ns1.mycompany.example. hostmaster.mycompany.example. (
                                2002071804      ; Serial
                                10800             ; Refresh after 3 hours
                                3600              ; Retry after 1 hour
                                604800           ; Expire after 1 week
                                180 )            ; TTL in seconds

                                IN NS      ns1.mycompany.example.
                                IN NS      ns2.mycompany.example.

ns1            IN A      192.168.1.5
ns2            IN A      192.168.1.6

localhost     IN A      127.0.0.1

www            IN A      192.168.1.5
server1       IN A      192.168.1.5
server2       IN A      192.168.1.6

# cat /etc/dns/master/db.192.168.1
$TTL 3600
@ SOA ns1.mycompany.example. hostmaster.mycompany.example. (
                                1997112100      ; Serial number
                                10800             ; Refresh
                                3600              ; Retry
                                604800           ; Expire
                                3600 )            ; Minimum TTL

                                IN NS      ns1.mycompany.example.
                                IN NS      ns2.mycompany.example.

5              IN PTR      server1.mydept.mycompany.example.
6              IN PTR      server2.mydept.mycompany.example.

```

Now that the zone files are created, you need to add the zone file definitions to the named.conf file on the master server. Copy the following three stanzas into the named.conf file on the master.

```

zone "mycompany.example" {
    type master;
    file "master/db.mycompany.example";
};

zone "mydept.mycompany.example" {
    type master;

```

```

        file "master/db.mydept.mycompany.example";
    };

    zone "1.168.192.in-addr.arpa" {
        type master;
        file "master/db.192.168.1";
    };

```

Refresh the BIND server and then use the **lsrrc** command to see the list of zones the server is configured for. The output should look similar to the following.

```

# refresh -s named
0513-095 The request for subsystem refresh was completed successfully.
# lsrrc -ls named

```

Subsystem	Group	PID	Status
named	tcpip	30604	active


```

Debug      Inactive

Type      Zone                      Source File or Host
master    0.0.127.in-addr.arpa        master/db.127.0.0
master    mydept.mycompany.example    master/db.mydept.mycompany.example
master    1.168.192.in-addr.arpa      master/db.192.168.1
master    mycompany.example          master/db.mycompany.example

```

Configuring zone transfer with TSIG security

Now that all the zones are configured on the master server, you must configure the slave server to zone transfer these zones. Prior to BIND 9, zone transfers could be restricted by IP address. Now the preferred method is to restrict zone transfers using TSIG. You must first create a DNSSEC host key using the **dnssec-keygen** command. The following command creates a 128-bit HMAC-MD5 key with a name of ns1-ns2. (a period at the end is intentional). The key name ns1-ns2. was chosen to indicate this key is for server communication between ns1 and ns2. Run the following **dnssec-keygen** command to generate the DNSSEC key.

```

# dnssec-keygen -a hmac-md5 -b 128 -n HOST -r /dev/urandom ns1-ns2.
Kns1-ns2.+157+57454
# cat Kns1-ns2.+157+57454.private
Private-key-format: v1.2
Algorithm: 157 (HMAC_MD5)
Key: Scb/CEcH4+/zJaEe/qXUIA==

```

Now that the ns1-ns2. key was generated, you need to add the following key stanza into the named.conf file on the master and slave. The algorithm and secret attributes in the key stanza are created from the ns1-ns2. private key file. Add the allow-transfer attribute to the existing options stanza and new slave server stanza to the /etc/named.conf on the master. The allow-transfer attribute

specifies what keys are allowed to zone transfer with this server. The server stanza specifies the key to use when contacting server 192.168.1.6, the slave server.

```
options {
...
    allow-transfer { key ns1-ns2.; };
...
};

// Server stanzas
server 192.168.1.6 {
    keys { ns1-ns2.; };
};

// Authentication keys
key ns1-ns2. {
    algorithm hmac-md5;
    secret "Scb/CEcH4+/zJaEe/qXUIA==";
};
```

Now configure the slave server by adding the new server stanza for the master server, the ns1-ns2. key and the following zone stanzas to the /etc/named.conf. The server stanza is now specifying an address of ns1, 192.168.1.5.

```
// Server stanzas
server 192.168.1.5 {
    keys { ns1-ns2.; };
};

// Authentication keys
key ns1-ns2. {
    algorithm hmac-md5;
    secret "Scb/CEcH4+/zJaEe/qXUIA==";
};

zone "mycompany.example" {
    type slave;
    file "slave/db.mycompany.example";
    masters { 192.168.1.5; };
};

zone "mydept.mycompany.example" {
    type slave;
    file "slave/db.mydept.mycompany.example";
    masters { 192.168.1.5; };
};

zone "1.168.192.in-addr.arpa" {
    type slave;
```

```

        file "slave/db.192.168.1";
        masters { 192.168.1.5; };
};

```

Restart both the master and slave BIND servers using the **refresh** command. The slave server will now transfer the zones from the master server. Look in the `/etc/dns/slave` directory for the transferred zone files and look in the message.log for any errors.

You can test that the secure zone transfers are set up correctly by using the **dig** command to transfer a zone from the master server. The first **dig** command below attempts to zone transfer `mycompany.example` without the `ns1-ns2.` key, which will fail. The second **dig** command specifies the `ns1-ns2.` key using the `-y` flag and will complete successfully.

```

# dig @ns1.mycompany.example AXFR mycompany.example
; <<>> DiG 9.2.0 <<>> @ns1.mycompany.example AXFR mycompany.example
;; global options: printcmd
; Transfer failed.

# dig @ns1.mycompany.example -y ns1-ns2.:Scb/CEcH4+/zJaEe/qXUIA== AXFR
mycompany.example
; <<>> DiG 9.2.0 <<>> @ns1.mycompany.example -y ns1-ns2. AXFR mycompany.example
;; global options: printcmd
mycompany.example.      180      IN        SOA       ns1.mycompany.example.
hostmaster.mycompany.example. 2002071802 10800 3600 604800 180
mycompany.example.      180      IN        NS        ns.mycompany.example.
mycompany.example.      180      IN        KEY       256 3 3
BLBVQ589+LR69sbiaWopX5DQsWc7917QF2ynmFJX2NmHT8EsV21EiIHu
cdYIkBY+BYtcn4CrXhENTVVtFqHX9np71Yj/bMSJFeLh7zvMKOC55e35
Qd8mYPSS/pA8/X58p+iQ5DpSGWHwBEQufbkyPsx/9b6BbTQ7FNbyD4G1
UfzwprovpeZE4GjVY51GSoIN11A3n5ro1Ar850nSxbDUnRVvf9gsBXAZ
iMSLWIueZjB1q4fryv0jKp/HBzu8oc5o/97gWP1HadTknzJHno9TJha
FG3QM32apKW5Qb73nEtP/LLOGopeuRu0dd3jduHKUsq9fmaqMXewfeq5
5VF57+kfTrZiYrA1vtOgQwL4MF6Hh1U05/8ayswUSbC4SaEqVMPIJ8TW +jMYeZd11Zi1At5VTI14
ca.mycompany.example.  180      IN        A         192.168.1.5
ldap.mycompany.example. 180      IN        A         192.168.1.5
localhost.mycompany.example. 180 IN      A         127.0.0.1
ns1.mycompany.example. 180      IN        A         192.168.1.5
ns2.mycompany.example. 180      IN        A         192.168.1.6
mycompany.example.      180      IN        SOA       ns1.mycompany.example.
hostmaster.mycompany.example. 2002071802 10800 3600 604800 180
ns1-ns2.                0        ANY      TSIG      hmac-md5.sig-alg.reg.int.
1031633977 300 16 Hu4Gy0avkRNoq1m55bzzRg== 2035 NOERROR 0
;; Query time: 5 msec
;; SERVER: 192.168.1.5#53(ns1)
;; WHEN: Mon Sep  9 23:59:37 2002
;; XFR size: 10 records

```

Signing the trusted root zone file

Now that zone transfers are working correctly between the master and slave server you should set up DNSSEC to sign the zones. When a zone is signed you allow security-enabled DNS clients to validate that the data was not tampered with. Normally you would sign the root zone in your organization and then sign all the children zones with the key of the child's parent zone. This builds a chain of trust, allowing clients that have obtained a public key higher in your DNS hierarchy to follow the chain of trust to validate your child zones. In this example, the zone `mycompany.example` is the trusted root zone and the zone `mydept.mycompany.example` is the child zone.

You must first generate a DNSSEC zone key for the `mycompany.example` zone using the `dnssec-keygen` command. Run the following commands on the DNS master server in the `/etc/dns/master` directory:

```
# dnssec-keygen -a DSA -b 1024 -n ZONE -r /dev/urandom mycompany.example
Kmycompany.example.+003+09992
```

```
# cat Kmycompany.example.+003+09992.key
mycompany.example. IN KEY 256 3 3
CMg2e8gHPHPYIXdQNeIEn6sY7IoNqxqWSYW1eJwyV+Sb/Y53q/aQHBPW
ngvSQiywJ+gBUrsoOp+JbyY/VjweoTR6l62V3AoPHgEekpp9/o7w/Yp1
RU6/IqqGi fSCcaxX3AT1FYv9bbYCN7UmxYbNf/Ze5suCN3D1WQuwMJ1r
9B6Fr0gbnoNfjfgPnDBNyFfwn8V4w6ODyr+CdvGB15n4E0ikSseidPHZ
V5Zs/C/fyP/khxBbc5F0Ujo2LqUnpg/9Sq/IrYhDeHsfPIPX5JcR91b/
mrPGTQ0wkjx1Ka1U/kNHpdT1oG1vquR50WmL880qnbQuM8h/1+9Rjka
i/XQqQ+X6+K60415mg481bp+0ApdxjxKVmRGc8A4ym+u0UJCgrBZ3j1s
y6A6/7obmcy0G17sGU1U1xDHr09IaLNwqA3WS/ROex3pEZcZyDs/N5ik
d5o36vthfwAgubDiE67BFga/mUu/Ub3gyoZr7IYKjc1kC8o6I6sGNGSN
5fYTKuwu1AyWSWSZRgVHdsXxfgPEadYvqXWD
```

Publish the public key for the trusted root zone by adding following include line in the `db.mycompany.example` zone file on the master server.

```
$INCLUDE /etc/dns/master/Kmycompany.example.+003+09992.key
```

Now add the same public key to the trusted-keys stanza in the `named.conf` files on both the master and slave DNS server. The format of the public key in the `Kmycompany.example.+003+09992.key` needs to be modified before inserting it into the `named.conf` file. The following example shows the expected format of the trusted-key stanza. This trusted-key stanza specifies the public key for the trusted security root zone. Restart the master and slave BIND servers to have this take affect.

```
// Public key of our trusted root zone
trusted-keys {
```

```
    mycompany.example. 256 3 3
```



```

"CMg2e8gHPHPYIXdQNeIEn6sY7IoNqxqWSYW1eJwyV+Sb/Y53q/aQHBPW
ngvSQiywJ+gBUrso0p+JbyY/VjweoTR6l62V3AoPHgEekpp9/o7w/Yp1
RU6/IqqGi fSCcaxX3AT1FYv9bbYCN7UmxYbNf/Ze5suCN3D1WQuwMJ1r
9B6Fr0gbonfjfgPnDBNyFfwn8V4w60Dyr+CdvGB15n4E0ikSseidPHZ
V5Zs/C/fyP/khxBbc5F0ujo2LqUnpg/9Sq/IrYhDeHsfPIPX5JcR91b/
mrXPGTQOwkjx1Ka1U/kNHpdT1oG1vquR50WmL880qnbQuM8h/1+9Rjka
i/XQqQ+X6+K60415mg481bp+0ApxdjxKVmRGc8A4ym+u0UJCgrBZ3j1s
y6A6/7obmcy0G17sGU1U1xDHr09IaLNwqA3WS/R0ex3pEZcZyDs/N5ik
d5o36vthfwAgubDiE67BFga/mUu/Ub3gyoZr7IYKjc1kC8o6I6sGNGSN
5fYTKuwu1AyWSWSZRgVHdsXxfgPEadYvqXWD";
};

```

The zone is now ready to be locally signed using the **dnssec-signzone** command. Increment the serial number for the mycompany.example zone file so the slaves will get the updated signed zone. The **dnssec-signed** command generates a new zone file named db.mycompany.example.signed, which is the signed version of the mycompany.example zone. The named.conf file on the master needs to be modified to serve the signed mycompany.example zone instead of the unsigned version. The following example shows the **dnssec-signzone** command to generate the signed zone:

```

# dnssec-signzone -r /dev/random -o mycompany.example db.mycompany.example\
Kmycompany.example.+003+09992
db.mycompany.example.signed

```

Replace the existing mycompany.example zone stanza in the named.conf file on the master server with the following stanza to enable the signed zone. Restart the server for the updates to take effect.

```

zone "mycompany.example" {
    type master;
    file "master/db.mycompany.example.signed";
};

```

Signing additional child zones

Now that the trusted root zone is set up, all the child zones need to be signed by the parent. You must first create a zone key for the child zone and then package it into a keyset file. The keyset file must then be sent to the administrator of the parent zone to be signed.

The following example generates a zone key, using the **dnssec-keygen** command, for the mydept.mycompany.example zone. The a keyset is generated using the **dnssec-makekeyset** command. The keyset file is then sent to the parent zone administrator to be signed.

```

# dnssec-keygen -a DSA -b 1024 -n ZONE -r /dev/random mydept.mycompany.example
Kmydept.mycompany.example.+003+24329

```

```
# dnssec-makekeyset -t 172800 -r /dev/random
Kmydept.mycompany.example.+003+24329.key
keyset-mydept.mycompany.example.
```

You need to publish the zone key in the child zone mydept.mycompany.example by adding the following line in the db.mydept.mycompany.example zone file on the master server. The zone's serial number must be incremented.

```
$INCLUDE /etc/dns/master/Kmydept.mycompany.example.+003+24329.key
```

The parent zone administrator, after receiving the unsigned keyset for the mydept zone, must not run the **dnssec-signkey** command to sign the keyset. This creates a chain of trust from the parent zone to the child. The signed keyset should then be returned to the mydept zone administrator.

```
# dnssec-signkey -r /dev/random keyset-mydept.mycompany.example.
Kmycompany.example.+003+09992.key
signedkey-mydept.mycompany.example.
```

The signed keyset is now used to sign the mydept zone using the **dnssec-signzone** command. The following example will sign the mydept.mycompany.example zone and create a new signed zone file.

```
# dnssec-signzone -r /dev/random -o mydept.mycompany.example
db.mydept.mycompany.example
db.mydept.mycompany.example.signed
```

Replace the existing mydept.mycompany.example zone stanza in the named.conf file on the master server with the following stanza to enable the signed zone. Restart the server for the updates to take effect.

```
zone "mydept.mycompany.example" {
    type master;
    file "master/db.mydept.mycompany.example.signed";
};
```

Dynamic DNS enhancements (DDNS)

BIND 9 has enhanced the support for dynamic DNS by its support for BIND 9 servers, allowing update policies and using TSIG instead of IP addresses to restrict updates. DDNS is a protocol that allows applications to update dynamic zones on a master server using a standard protocol. The most common application to use DDNS is the dynamic host configuration protocol (DHCP) server, where clients receive an assigned IP addresses from a pool and, using DDNS, the DHCP server updates the forward and reverse dynamic zones with the new address. The DNS server stores updates to dynamic zones in journal files and synchronizes the zone file periodically or when the server is stopped with **rndc**. The command line interface to DDNS is the **/usr/sbin/nsupdate**

command. The enhanced DDNS support has one modified option, `allow-update`, and a new option, `update_policy`.

The `allow-update` option specifies that the specific zone allows dynamic DNS updates. The `address_match_list` parameter can now be a TSIG key. Prior to BIND 9, `address_match_list` would only support IP addresses. If `allow-update` is set, authorized clients can add or modify any resource record in the dynamic zone. The following example shows the syntax and how to use the `allow-update` option.

```
// allow-update { address_match_list } ;
zone "dynamic.zone" {
...
    allow-update { key nsupdate.; };
...
};
```

The `update-policy` option was added to give fine-grained control that restricts dynamic updates. It allows the administrator to configure rulesets to restrict specific identities to only update certain resource records. The following is the syntax for the `update-policy` option.

```
// update-policy { update_policy_rule [...] } ;
// update_policy_rules = ( grant | deny ) identity nametype name [ types ]
```

To enable secure dynamic updates, you must create a TSIG key and add a zone and key stanza to the `named.conf` on the master server. The following example shows how to create a TSIG named `nsupdate`. (the period is intentional), using the `dnssec-keygen` command.

```
# dnssec-keygen -a hmac-md5 -b 128 -n HOST -r /dev/urandom nsupdate.
Knsupdate.+157+30189
# cat Knsupdate.+157+30189.private
Private-key-format: v1.2
Algorithm: 157 (HMAC_MD5)
Key: C8RGxOWJlVuKtTo3PFqhmw==
```

The following example shows the key stanza defining the `nsupdate`. key and the zone stanza to define the dynamic zone. The `allow-update` zone option specifies that only DDNS clients with the `nsupdate`. key can make changes to this zone.

```
key nsupdate. {
    algorithm hmac-md5;
    secret "C8RGxOWJlVuKtTo3PFqhmw==";
};
zone "dynamic.mycompany.example" {
    type master;
    allow-update { key nsupdate.; };
    file "master/db.dynamic.mycompany.example";
};
```

The **nsupdate** command accepts commands either from standard input or from a file. The following example shows the DDNS command file. The commands tell the **nsupdate** to contact the ns1.mycompany.example server, delete all RR for the mycomputer.dynamic.mycompany.example from the dynamic.mycompany.example zone, and then add a new A record for mycomputer.dynamic.mycompany.example with the IP address 192.168.1.100.

```
# cat update100
server ns1.mycompany.example
zone dynamic.mycompany.example
update delete mycomputer.dynamic.mycompany.example
update add mycomputer.dynamic.mycompany.example 86400 A 192.168.1.100
show
send
```

The following example shows how to run the **nsupdate** command to execute the previous DDNS command file. The preferred method to supply authentication method for the **nsupdate** command is to use the **-k** flag, which specifies the key file. You can also specify the key name and password on the command line, but that is not secure because command line parameters are normally visible using the **ps** command and stored in command shell histories. The following example shows how to call the **nsupdate** command using the key file for authentication.

```
# nsupdate -k Knsupdate.+157+30189 update100
Outgoing update query:
;; ->HEADER<<- opcode: UPDATE, status: NOERROR, id:      0
;; flags: ; ZONE: 0, PREREQ: 0, UPDATE: 0, ADDITIONAL: 0
;; UPDATE SECTION:
mycomputer.dynamic.mycompany.example. 0 ANY ANY
mycomputer.dynamic.mycompany.example. 86400 IN A 192.168.1.100
```

Incremental zone transfers (IXFR)

BIND 9 now completely supports incremental zone transfers (IXFR). IXFR allows the slave to receive individual updates to the zone instead of a complete zone transfer. The BIND server tracks all updates to all master zones enabled for IXFR in the directory specified by the ixfr-directory option. In very dynamic zone files, the BIND server will sometimes decide that it is more efficient to do a complete zone transfer instead of incremental zone transfers. If a slave server is more than one increment behind the master, then a complete zone transfer will occur. The IXFR protocol is described in more detail in RFC1995.

The provide-ixfr option configures the local server, acting as a master, to honor or deny a request for IXFR from a specific slave server. This option can be globally defined in the options or in a server stanza. If you specify provide-ixfr in the server and options stanza, the provide-xfer in the server stanza will be used. The default is yes. The following example shows the syntax and how to use the provide-ixfr option.

```
// provide-ixfr yes_or_no;
server 192.168.1.6 {
    provide-ixfr no;
};
```

The request-ixfr option configures the local server, acting as a slave, to ask for incremental zone transfers from a specific master server. This option can be specified in the server and options stanza. The default is yes. The following example shows the syntax and to use the request-ixfr option.

```
// request-ixfr yes_or_no;
options {
...
    request-ixfr    yes;
...
};
```

Enhanced notification support

BIND 9 now has enhanced notification support. The notification protocol allows the master to notify the slave servers of an updated zone file to minimize the time the master and slave servers are out of sync. When a zone file is updated on the master server, notifications are sent to all the slave servers in the zone file. After receiving the notification the slave server can either choose to ignore it or initiate a zone transfer. For more information about the protocol, refer to RFC1996. The enhancements have added three new configuration options to the named.conf file: Allow-notify, notify, and notify-source.

The allow-notify option specifies the list of additional servers, besides the master server, to allow receiving notifications from. This option can be specified in the options or zone stanzas. The default is to only accept notifications from the zone's master. The following is the syntax for the allow-notify option:

```
// allow-notify { address_match_list };
```

The notify option specifies the list of servers to notify when a zone changes. The notify option can be specified in the options and zone stanza. If the notify is set to yes, the default, then notifications are sent to all servers with NS records in the zone and any server specified by the also-notify option. If notify is set to explicit, then only the servers listed in also-notify will be notified. If notify is set to no, no notifications will be sent. The only reason to turn off notifications is when the notification crashes slave servers. The following is the syntax for the notify option.

```
// notify yes_or_no | explicit ;
```

The notify-source option allows you to change the source address and port for notifications. This is normally used when the DNS server is multihomed or to

make filter definitions easier if the notifications need to pass through a packet-filtering firewall. The following is the syntax for the notify-source option:

```
// notify-source (ip4_addr | *) [port ip_port] ;
```

IPv6 enhancements

BIND 9 now supports the new RFC2874 addressing scheme. This RFC introduces new resource records (RRs) A6 and DNAME, a new domain for reverse lookups, and a new IPv6 address notation called bitstring.

The A6 RR record was introduced to store IPv6 addresses not as a single RR but as a chain of RRs. A6 RRs were designed to simplify the process of renumbering sites. The A6 RR replaces the AAAA record for forward resolution of IPv6 addresses. AAAA RR is still supported but is deprecated. It is useful to have both AAAA and A6 records for backwards compatibility when you have clients that do not support the newer A6 records.

The DNAME record was introduced to allow easy management of the reverse tree. A new reverse domain was introduced to replace the ip6.int domain, which is now deprecated but still supported. The new reverse domain ip6.arpa uses the new bitstring labels, while the old domain uses the nibble labels. For more information of IPv6 addressing, refer to RFC2874. The following example shows using the AAAA and A6 RRs:

```
$ORIGIN mydept.mycompany.example.
server1-V6      IN AAAA fe80::204:acff:fe7c:c3d8
                 IN A6   0 fe80::204:acff:fe7c:c3d8
server2-V6      IN A6   0 fe80::206:29ff:fec5:1d87
```

For reverse lookups, BIND 9 now supports specifying the IPv6 address with both in nibble and bitstring labels. Nibble labels are deprecated but still supported. The following example shows a IPv6 reverse zone with nibble labels, its named.conf zone stanza, and an abbreviated **dig** example.

```
$ORIGIN 0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.int.
7.b.0.2.9.b.e.f.f.9.2.6.0.2.0      IN PTR
server1-V6.mydept.mycompany.example.
7.8.d.1.5.c.e.f.f.9.2.6.0.2.0      IN PTR
server2-V6.mydept.mycompany.example.

zone "0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.int" {
    type master;
    file "master/db.ipv6rev-nibble";
};

# dig -n
7.b.0.2.9.b.e.f.f.9.2.6.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.int PTR
```

```
; <<>> DiG 9.2.0 <<>> -n
7.b.0.2.9.b.e.f.f.f.9.2.6.0.2.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.int PTR
...
;; ANSWER SECTION:
7.b.0.2.9.b.e.f.f.f.9.2.6.0.2.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.int. 3600
IN PTR server1-V6.mydept.mycompany.example.
...
```

The preferred method of expressing IPv6 addresses is now using the bitstring labels. The bitstring labels use hexadecimal characters in natural order, making it much easier to read and much more compact. The following example shows the previous zone file using bitstring labels, its named.conf zone stanza, and an abbreviated **dig** example.

```
$ORIGIN \[xfe80000000000000/64].ip6.arpa
\[x020629fffeb920b7/64] IN PTR server1-V6.mydept.mycompany.example.
\[x020629fffec51d87/64] IN PTR server2-V6.mydept.mycompany.example.

zone "\[xfe80000000000000/64].ip6.arpa" {
    type master;
    file "master/db.ipv6rev-bitstring";
};

# dig \[xfe80000000000000020629fffeb920b7/128].ip6.arpa PTR
; <<>> DiG 9.2.0 <<>> \[xfe80000000000000020629fffeb920b7/128].ip6.arpa PTR
...
;; ANSWER SECTION:
\[xFE80000000000000020629FFFEb920B7/128].ip6.arpa. 3600 IN PTR
server1-V6.mydept.mycompany.example.
...
```

BIND 9 added two new configuration options specifically for IPv6 support: Allow-v6-synthesis and listen-on-v6.

The listen-on-v6 option is used to specify the port in which BIND will listen for queries sent using IPv6. Unlike IPv4, BIND 9 does not bind a separate socket for each IPv6 address. Instead, it always listens on the IPv6 wildcard address. The only valid values for address_match_list are { any; } or { none; }. You may specify multiple listen-on-v6 options to listen on more than one port. The default is BIND and does not listen on any IPv6 addresses. The following example shows the syntax and how to use the listen-on-v6 option.

```
// listen-on-v6[portip_port]{address_match_list};
listen-on-v6 { any; };
```

The allow-v6-synthesis option allows the BIND 9 server to support older stub resolvers that only support DNS lookups as defined in RFC1886, instead of the newer RFC2874. RFC1886 uses AAAA records for forward lookups and *nibble labels* in the ip6.int domain for reverse lookups, while RFC2874 uses A6 and

DNAME for forward lookups and bitstring notation in the ip6.arpa domain for reverse lookups. If this option is enabled, the server will automatically convert RFC1886 queries into RFC2874 queries and return the results in AAAA and ip6.int PTR records. This option is disabled by default and can be enabled per client address using the *address_match_list* parameter. The following example shows the syntax and how to use the allow-v6-synthesis option.

```
// allow-v6-synthesis{ address_match_list };
allow-v6-synthesis { any; };
```

If allow-v6-synthesis is disabled and the client requests a reverse address in the ip6.int domain, the server will respond with an NXDOMAIN error, which is a non-existent domain. The following is the output from **dig** for a reverse address on the ip6.int domain with allow-v6-synthesis disabled:

```
# dig -n
7.b.0.2.9.b.e.f.f.f.9.2.6.0.2.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.int PTR
; <<> DiG 9.2.0 <<> -n
7.b.0.2.9.b.e.f.f.f.9.2.6.0.2.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.int PTR
;; global options: printcmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NXDOMAIN, id: 10786
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;7.b.0.2.9.b.e.f.f.f.9.2.6.0.2.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.int. IN
PTR
...
```

If allow-v6-synthesis is enabled, the server would accept this request and then return a valid answer in the old RFC1886 style. The following example shows the results of the same **dig** command with allow-v6-synthesis enabled:

```
# dig -n
7.b.0.2.9.b.e.f.f.f.9.2.6.0.2.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.int PTR
; <<> DiG 9.2.0 <<> -n
7.b.0.2.9.b.e.f.f.f.9.2.6.0.2.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.int PTR
...
;; ANSWER SECTION:
7.b.0.2.9.b.e.f.f.f.9.2.6.0.2.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.int. 0 IN
PTR server1-V6.mydept.mycompany.example.
...
```

Enabling the allow-v6-synthesis option also allows queries for non-existing AAAA to be mapped to A6 queries and the results returned as an AAAA record. This feature allows you to remove old style IPv6 addresses from your zone files, while still supporting older stub resolvers. The following example shows a **dig** request for a non-existent AAAA record, but the server will respond with the A6 record synthesized as a AAAA record.


```
# dig server2-V6.mydept.mycompany.example AAAA
; <<> DiG 9.2.0 <<> server2-V6.mydept.mycompany.example AAAA
...
;; ANSWER SECTION:
server2-V6.mydept.mycompany.example. 0 IN AAAA fe80::206:29ff:fec5:1d87
...
```

For more information on IPv6 addressing, refer to the following RFCs:

- ▶ RFC2373 - IP Version 6 Addressing Architecture
- ▶ RFC2874 - DNS Extensions to support IPv6 Address Aggregation and Renumbering
- ▶ RFC2673 - Binary Labels in the Domain Name System

You can download RFCs from the IETF home page at the following URL:

<http://www.ietf.org>

Views support

BIND 9 now supports the concept of views. Views allow one single BIND server to answer requests differently depending on the requesting client. A view is just a collection of zones that is given a name, for example `private`. You can have a specific view only visible to clients with certain IP addresses and have all the other clients use another view. If you do not specify a view, all zone files are included in the default view. If you do specify, a view all zones must be included in a view. If several views share the same zones and db files, then it is easiest to put those common zones in a separate file. Then use the `include` command in each view to load the common zones from that file.

The following example shows how to create two views, one `private` and one `public`. The `private` zone contains resource records that should only be available to clients on the `192.168.1.0/24` network, but not exposed to the public clients. Notice the different zone files names with the same zone name.

```
view "private" {
    match-clients { 192.168.1.0/24; };
    recursion      yes;
    ...
};
zone "mycompany.example" {
    type master;
    file "master/db.mycompany.example.private";
};

view "public" {
    match-clients { any; };
};
```

```

        recursion      yes;
...
zone "mycompany.example" {
    type master;
    file "master/db.mycompany.example.public";
};
};

```

8.3 TCP/IP routing subsystem enhancements

AIX 5L offers multipath routing and dead gateway detection (DGD) as new features of the TCP/IP routing subsystem. They are intended to enable administrators to configure their systems for load balancing and failover.

Multipath routing provides the function necessary to configure a system with more than one route to the same destination. This is useful for load balancing by routing IP traffic over different network segments, or to specify backup routes to use with dead gateway detection. Section 8.3.1, “Multipath routing” on page 460 covers the details on this new routing feature.

Dead gateway detection enables a system to discover if one of its gateways is down and use an alternate gateway. DGD offers an active and a passive mode of operation to account for different kinds of customer requirements (in respect to performance and availability). Section 8.3.2, “Dead gateway detection” on page 466, provides more in-depth information about this enhancement to the TCP/IP routing subsystem.

Both new routing features are implemented for IP Version 4 (IPv4) and IP Version 6 (IPv6).

8.3.1 Multipath routing

Prior to AIX 5L, a new route could be added to the routing table only if it was different from the existing routes. The new route would have to be different by either destination, netmask, or group ID. The sample output of the **netstat** command, depicted in the following, shows two routing table entries that have the same netmask. However, the route for the token-ring interface differs from the route associated with the Ethernet interface by the destination:

```

# netstat -rn
Routing tables
Destination Gateway  Flags Refs  Use   If PMTU  Exp  Groups

Route tree for Protocol Family 2 (Internet):
9.3.21/24   9.3.21.22 U    106   17412 tr1-   .

```

```
9.3.22/24    9.3.22.37 U    0    266344 en0-    .
```

The following **netstat** command output was taken from a system where two routes for two different gateways are defined with the same destination but for different netmasks.

```
# netstat -rn
Routing tables
Destination Gateway Flags Refs Use If PMTU Exp Groups

Route tree for Protocol Family 2 (Internet):
10/24 9.3.21.22 UGc 0 0 tr1 - - =>
10/23 9.3.22.37 UGc 0 0 en0
```

In the case where the destination address is the same but the netmask is different, the most specific route that matches will be used. In the previous example, packets sent to 10.0.0.1–10.0.0.255 would use the 10/24 route, since it is more specific, while packets sent to 10.0.1.1–10.0.1.255 would use the 10/23 route, since they do not match the 10/24 route but do match the 10/23 route.

The third possible differentiator for a unique route definition is given by the group ID list. The groups associated with a route are listed in the column of the **netstat -r** output, which is labeled with the keyword *groups*. These groups are comprised of AIX group IDs, and they determine which users have permission to access the route. This feature is used by system administrators to enforce security policies or to provide different classes of service to different users.

With the new multipath routing feature in AIX 5L, routes no longer need to have a different destination, netmask, or group ID list. If there are several routes that equally qualify as a route to a destination, AIX will use a cyclic multiplexing mechanism (round-robin) to choose between them. The benefit of this feature is twofold:

- ▶ Enablement of load balancing between two or more gateways.
- ▶ Feasibility of load balancing between two or more interfaces on the same network can be realized. The administrator would simply add several routes to the local network, one through each interface.

In order to implement multipath routing, AIX 5L allows you to define a user-configurable cost attribute for each route and offers the option to associate a particular interface with a given route. These enhancements are configurable by the parameters **-hopcount** and **-if** of the **route** command. In the following, you find an excerpt of the manual page for the **route** command.

Note the new `-active_dgd` parameter that turns on the active DGD for a given route, which will be described later on in “Active dead gateway detection” on page 472:

```
route [ -n ] [ -q ] [ -v ] Command [ Family ] [ [ -net | -host ] Destination  
[-prefixlen n] [-netmask] [ Address ] ] Gateway ]  
[ Arguments ]
```

Flags

The following is a list of the common flags and their definitions.

-n	Displays host and network names numerically, rather than symbolically, when reporting results of a flush or of any action in verbose mode.
-q	Specifies quiet mode and suppresses all output.
-v	Specifies verbose mode and prints additional details.
-net	Indicates that the destination parameter should be interpreted as a network.
-netmask	Specifies the network mask to the destination address. Make sure this option follows the destination parameter.
-host	Indicates that the destination parameter should be interpreted as a host.
-prefixlen n	Specifies the length of a destination prefix (the number of bits in the netmask).

Parameters

The following is a list of the common parameters and their definitions.

Arguments	Specifies one or more of the following arguments. Where <code>n</code> is specified as a variable to an argument, the value of the <code>n</code> variable is a positive integer.
-active_dgd	Enables Active dead gateway detection on the route.
-hopcount n	Specifies maximum number of gateways in the route.
-if ifname	Specifies the interface (<code>en0</code> , <code>tr0</code> , ...) to associate with this route so that packets will be sent using this interface when this route is chosen.
Commands	Specifies one of six possibilities: Add, flush, delete, change, monitor, or get.
Family	Specifies the address family (<code>inet</code> , <code>inet6</code> , or <code>xns</code>).
Destination	Identifies the host or network to which you are directing the route.

Gateway

Identifies the gateway to which packets are addressed.

User-configurable cost attribute of routes

The user-configurable cost of a route is specified as a positive integer value for the variable associated with the `-hopcount` parameter. The integer can be any number between 0 and the maximum possible value of `MAX_RT_COST`, which is defined in the `/usr/include/net/route.h` header file to be `INT_MAX`. The value of `INT_MAX` is defined in `/usr/include/sys/limits.h` to be 2147483647. The header files will be on your system after you install the `bos.adt.include` fileset. The `-hopcount` parameter existed in the past, and the assigned integer value was supposed to reflect the number of gateways in the route. However, in previous AIX releases, the parameter value given during the configuration of the route had no effect on how the route was used.

Even so, the `-hopcount` parameter in AIX 5L refers historically to the number of gateways in the route; the number configurable by the system administrator can be totally unrelated to the actual presence or absence of any real gateways in the network environment. The user-configurable cost attribute's sole purpose is to establish a metric, which is used to create a priority hierarchy among the entries in the routing table.

If the routing table offers several alternative routes to the desired destination, the operating system will always choose the route with the lowest distance metric as indicated by the lowest value for the current cost. In the case where multiple matching routes have equal current cost, a lookup mechanism chooses the most specific route. When both criteria are equal for multiple routes, AIX 5L will round-robin select between them. Higher-cost routes ordinarily will never be used; they are only there as backups. If the lower-cost routes are deleted or their costs are raised, the backup routes will be used. This provides a mechanism for marking bad routes when a gateway failure is detected; indeed, the DGD feature, as described in 8.3.2, "Dead gateway detection" on page 466, exploits this particular feature.

The kernel resident routing table is initialized when interface addresses are set by making entries for all directly connected interfaces. The routing entry structure `rtenry` is defined in the `route.h` header file, which will be located in the `/usr/include/net/` directory after you optionally install the `bos.adt.include` fileset.

The behavior of the code to select routes has only changed when duplicate routes exist. For nodes with multiple routes, the duplicated route is followed until a route that matches is found. If there are other entries with the same cost and netmask, the route that was last used is skipped and the next one chosen.

The costs on all routes can be displayed using the new `-C` flag on the `netstat` command, as indicated by the following example.

With the -C flag set, the **netstat** command shows the routing tables, including the user-configured and current costs of each route. The user-configured cost is set using the -hopcount flag of the **route** command. The current cost may be different from the user-configured cost if, for example, the dead gateway detection has changed the cost of the route. For further details on DGD, refer to 8.3.2, “Dead gateway detection” on page 466.

```
# netstat -Cn
Routing tables
Destination      Gateway          Flags    Refs  Use  If    Cost  Config_Cost

Route tree for Protocol Family 2 (Internet):
9.3.149.96/28     9.3.149.100     U        5     23   en1    0      0
9.3.149.160/28    9.3.149.163     U        1      5   tr0    0      0
9.53.150/23       9.3.149.160     UGc      0      0   tr0    0      0 =>
9.53.150/23       9.3.149.97      UGc      0      0   en1    1      1
127/8             127.0.0.1       U        1    130425 lo0    0      0

Route tree for Protocol Family 24 (Internet v6):
::1              ::1             UH        0      0   lo0    0      0
```

Interface-specific routes

The implementation of TCP/IP routing in previous AIX releases did not provide any mechanism to associate a specific interface with a route. When there were multiple interfaces on the same network, the same outgoing interface for all destinations accessible through that network was always chosen. In order to configure a system for network traffic load balancing, it is desirable to have multiple routes so that the network subsystem routes network traffic to the same network segment by using different interfaces. AIX 5L introduces the new -if argument to the **route** command, which can be used to associate a particular interface with a specific route.

The -if parameter argument must not be mistaken for the -interface parameter argument of the **route** command. The -interface argument specifies that the route being added is an interface route, which means it is a direct route to the local network and does not go through a gateway.

The following example shows the usage of the **route** command to establish an interface-specific host route from a given computer on one network to its counterpart on a different network:

```
route add 192.100.201.7 192.100.13.7 -if tr0
```

The 192.100.201.7 address is that of the receiving computer (destination parameter) and the 192.100.13.7 address is that of the routing computer (gateway parameter). The -if argument assigns the static host route to the token ring interface tr0.

Deletion and modification of routes

The **route** command, used in conjunction with the **delete qualifier** command, examines the entries in the kernel route table and deletes only the specified route in the routing table if a unique route has been successfully identified. In previous AIX releases, this command could only fail if no route entry matched the specified command line parameters. Since AIX 5L offers the option to specify multiple routes to the same destination, but with different gateways or interfaces, the **route delete** command may fail, because more than one route matches the criteria for deletion. If the attempt to delete a route fails, an error message is returned (as always), but this message explicitly mentions that there are now two possible error conditions that have to be considered. The following example shows the error message returned by the **route delete** command on a system with more than one defined default route:

```
# route delete default
0821-279 writing to routing socket: The process does not exist.
default net default: route: not in table or multiple matches
```

In order to account for the possible existence of multiple routes to the same destination but with different gateways or interfaces in AIX 5L, similar modifications were implemented for the command to change a route. This means that the **route change** command will return an error message whenever no unique route could be identified, regardless of the absence of a given route or the existence of multiple routes to the same destination. Note that only the user-configurable cost, gateway, and interface of a route can be changed.

Limitations for multipath routing

You must completely understand the limitations when using Multipath routing in conjunction with the path maximum transfer unit (PMTU) discovery feature of AIX.

When the destination of a connection is on a remote network, the operating system's TCP, by default, advertises a maximum segment size (MSS) of 512 bytes. This conservative value is based on a requirement that all IP routers support an MTU of at least 576 bytes.

The optimal MSS for remote networks is based on the smallest MTU of the intervening networks in the route between source and destination. In general, this is a dynamic quantity and could only be ascertained by some form of path MTU discovery.

The AIX 5L operating system supports a path MTU discovery algorithm as described in RFC1191. Path MTU discovery can be enabled for TCP and UDP applications by modifying the `tcp_pmtu_discover` and `udp_pmtu_discover` options of the **no** command. When enabled for TCP, path MTU discovery will automatically force the size of all packets transmitted by TCP applications to not

exceed the discovered path MTU size. Since UDP applications themselves determine the size of their transmitted packets, UDP applications must be specifically written to utilize path MTU information by using the `IP_FINDPMTU` socket option, even if the `udp_pmtu_discover` network option is enabled. By default, the `tcp_pmtu_discover` and `udp_pmtu_discover` options are disabled on Version 4.2.1 through Version 4.3.1, and enabled on Version 4.3.2 and later.

When the path MTU has been discovered for a network route, a separate host route is cloned for the path. These cloned host routes, as well as the path MTU value for the route, can be displayed using the `netstat -r` command. Accumulation of cloned routes can be avoided by allowing unused routes to expire and be deleted. Route expiration is controlled by the `route_expire` option of the `no` command. Route expiration is disabled by default on Version 4.2.1 through Version 4.3.1, and set to one minute on Version 4.3.2 and later.

Beginning with AIX 5L, you may have several equal-cost routes to a given network, but with different associated gateways, on a system for which PMTU discovery is enabled. When traffic is sent to a host on that specific network, a host route will be cloned from whichever network route was chosen by the cyclic multiplexing code of the multipath routing algorithm. Because the cloned host route is always more specific than the original network route of which the clone was derived, all traffic to that host will use the same gateway as long as the cloned route exists and, consequently, no cyclic multiplexing among the different gateways associated with the equal-cost route to the specific network will take place.

Since PMTU discovery is enabled by default in AIX 5L, system administrators may consider disabling the network options `tcp_pmtu_discover` or `udp_pmtu_discover` to turn off route cloning (in order to take full advantage of the new multipath routing feature). This measure will prevent the creation of the cloned host routes and will instead allow cyclic multiplexing between equal-cost routes to the same network.

8.3.2 Dead gateway detection

The new dead gateway detection (DGD) feature in AIX 5L implements a mechanism for hosts to detect a dysfunctional gateway, adjust its routing table accordingly, and reroute network traffic to an alternate backup route if available. DGD is generally most useful for hosts that use static rather than dynamic routing.

Overview

AIX releases prior to AIX 5L did not permit you to configure multiple routes to the same destination. If a route's first hop gateway failed to provide the required routing function, AIX continued to try to use the broken route and address the

dysfunctional gateway. Even if there was another path to the destination that would have offered an alternative route, AIX did not have any means to identify and switch to the alternate route unless a change to the kernel routing table was explicitly initiated, either manually or by running a routing protocol program, such as **gated** or **routed**. Gateways on a network run routing protocols and communicate with one another. So if one gateway goes down, the other gateways will detect it and adjust their routing tables to use alternate routes. (Only the hosts continue to try to use the dead gateway.)

The new DGD feature in AIX 5L enables host systems to sense and isolate a dysfunctional gateway and adjust the routing table to make use of an alternate gateway without the aid of a running routing protocol program.

AIX 5L implements DGD based on the requirements given in RFC1122, sections 3.3.1.4 and 3.3.1.5, and RFC816. These RFCs contain a number of suggestions on mechanisms for doing DGD, but currently no completely satisfactory algorithm has been identified. In particular, the RFCs require that pinging to discover the state of a gateway be avoided or extremely limited, and they recommend that the IP layer receive *hints* that a gateway is up or down from transport and other layers that may have some knowledge of whether a data transmission succeeded. However, in one of the two possible modes (active mode) for the AIX 5L DGD feature, status information of a gateway is collected with the help of pinging, and hence the AIX 5L DGD implementation is not fully compliant with the RFCs mentioned above.

DGD utilizes the functions of AIX 5L multipath routing. The multipath routing feature allows for multiple routes to the same destination, which can be used for load balancing and failover. Refer to 8.3.1, “Multipath routing” on page 460, for further details.

The DGD implementation in AIX 5L offers the flexibility to address two distinct sets of customer requirements:

- ▶ Requirement for minimal impact on network and system environment in respect to compatibility and performance. The detection of a dysfunctional gateway and the switch from the associated route over to an alternate gateway route must be accomplished without any significant overhead.
- ▶ Requirement for maximum availability of network services and connections. If a gateway goes down, a host must always discover that fact within a few seconds and switch to a working gateway.

Since both sets of requirements cannot be satisfied by a single mechanism, AIX 5L DGD offers a passive and an active mode of operation.

The passive dead gateway detection addresses the need for minimal overhead, while the active dead gateway detection ensures maximum availability while

imposing some additional workload onto network segments and connected systems. Passive DGD is disabled system wide by default, but active DGD is defined as an attribute for a particular route, and therefore requires being enabled on a route-to-route basis.

Passive dead gateway detection

One of the two modes for dead gateway detection will work without actively pinging the gateways known to a given system; therefore, this mode is referred to as passive DGD.

Passive DGD will take action to use a backup route if a dysfunctional gateway has been detected. The backup route can have a higher current cost than the route associated with the dysfunctional gateway, which allows you to configure primary (lower cost) gateways and secondary (higher cost) backup gateways. As such, DGD expands the TCP inherent failover between alternate equal cost routes, as introduced by the new AIX 5L multipath routing feature.

The passive DGD mechanism depends on protocols that provide information about the state of the relevant gateways. If the protocols in use are unable to give feedback about the state of a gateway, a host will never know that a gateway is down and no action will be taken.

The Transmission Control Protocol (TCP), in conjunction with the Address Resolution Protocol (ARP), is able to give the necessary feedback about the state of a specific gateway. It is important to note that these two protocols give different types of feedback, and that you have to use both protocols to receive the full benefit of the passive DGD feature.

TCP identifies round-trip traffic that is not getting through. It will correctly detect that the gateway in question is down if it is indeed no longer forwarding traffic. However, it may incorrectly report that the gateway is down if there is a temporary routing problem elsewhere in the network that the first-hop gateway will soon detect and adjust to, or if the address it is sending to is unreachable or nonexistent.

On the other hand, ARP still perceives a gateway to be up even if it is no longer forwarding traffic. The only thing ARP can detect with certainty is whether the first-hop gateway can be reached, but it does not sense whether the network traffic is forwarded and reaches its final destination. So transitory problems elsewhere in the network cannot cause ARP to mistake a functional for a dysfunctional gateway.

Because TCP cannot detect if the destination for the network traffic is supposed to be reachable, the decisions about a gateway's state cannot be based only on

TCP. Instead, TCP is used to prompt dead gateway detection under certain conditions to determine the state of a gateway based on feedback from ARP.

Note: For IPv6, it is not necessary to use passive dead gateway detection. The Neighbor Discovery Protocol (NDP) contains all the functions that passive DGD adds for IPv4.

Multipath routing in AIX 5L allows you to specify a distance metric or cost associated with a route. Routes to the same destination with equal cost will be selected by a cyclic multiplexing algorithm. Routes with a higher cost will not be used unless there is a problem with the lower-cost routes. Passive DGD exploits the multipath routing feature to provide failover for dysfunctional gateways.

If feedback is received from ARP that a gateway might be down, the current costs of all routes using that gateway will be increased to the maximum value `MAX_RT_COST` (refer to “User-configurable cost attribute of routes” on page 463 for further details). The user-configurable cost will not be changed, but eventually will be used in the future to restore the current cost to the original value if the gateway comes up again. If alternative routes to the same destination with a cost equal to the original cost of the deprecated route are defined, the TCP/IP subsystem will use those exclusively, and the route whose current cost was increased is no longer addressed. If there were no other routes to the destination, the original route is still the lowest-cost route, and the system will continue to try to use it.

When the current cost of a route is increased, as described previously, a timer will be set for a configurable period of time. This will be specified by a new network option called `dgd_retry_time`. The default value for this network option is set to five minutes, since that is about the amount of time it will take a gateway that has crashed to reboot. Use the `no -o` command to display or change the `dgd_retry_timer` network option. The `no` command output in the following example shows the value for the `dgd_retry_timer` on a system where this specific network option is set to the default of 5:

```
# no -o dgd_retry_time
dgd_retry_time = 5
```

Note that the network options set by the `no` command are only in effect until the next reboot. If you would like to use the customized settings for the network options permanently, you will have to include the appropriate `no` commands in the network startup script `/etc/rc.net`. This script is executed during the boot process and will establish the network options with the customized values of your choice.

When the timer expires, the cost will be restored to its original user-configured value. If the gateway did not come up in the meantime, the next attempt to send traffic will raise the current cost for the routes in question again to the maximum value and the timer is reset for another five minute wait. If the gateway is back up, that route will continue to be used. The only exception to this is when active DGD is in use, as described in “Active dead gateway detection” on page 472. In this case, a flag on the route will indicate that active detection is in use, and passive detection should not restore the cost to its original value.

ARP requests are only sent out if the ARP cached entry has expired. By default, ARP entries expire after 20 minutes. So if a gateway goes down, it may take quite a long time (relative to transaction events that require responsive networks) before DGD senses any problem with a given gateway through ARP protocol. For this reason, the DGD mechanism monitors to see if TCP retransmits packets too many times, and in the case where it suspects that a gateway is down, it deletes the ARP entry for that gateway. The next time any traffic is sent along the given route, an ARP request is initiated, which provides the necessary information about the state of the gateway to DGD.

TCP is not supposed to initiate a change of the cost associated with a route, because it does not know whether the gateway is actually down or if the destination is just unreachable. For this reason, TCP indirectly initiates an ARP request by deleting the ARP cache entry for the gateway in question. On the other hand, TCP is aware of any particular failing connection. So, TCP explores (independently of the feedback of the initiated ARP requests) if there is any other route to its destination with a cost equal to the one it is currently using. If TCP identifies alternate routes, it tries to use them. This way the connection in question will still recover right away if the gateway really was down.

It is important to carefully choose the criteria for deciding that a gateway is down. A failover to a backup gateway just because a single packet was lost in the network must be avoided, but in the case of an actual gateway failure, network availability must be restored with as little delay as possible. The number of lost packets needed before a gateway will be suspected or considered as dysfunctional is user-configurable by the new network option named `dgd_packets_lost`. The network option `dgd_packets_lost` can be displayed and changed by the `no -o` command and is set to 3 by default. The `no` command output in the following example shows the value for the `dgd_packets_lost` on a system where this specific network option is set to the default of 3:

```
# no -o dgd_packets_lost
dgd_packets_lost = 3
```

The same restrictions that were mentioned before in respect to the `dgd_retry_timer` network option apply for the `dgd_packets_lost` network option.

If TCP retransmits the same packet as many number of times as defined by `dgd_packets_lost` and gets no response, it deletes the ARP entry for the gateway route it was using and tries to use an alternative route. When the next attempt is made to send a packet along the gateway route, no ARP cache entry is found, and ARP sends out a request to collect the missing information. The value for `dgd_packets_lost` also determines how often no response to an ARP request is tolerated before a gateway finally will be considered to be down and the costs of all routes using it will be increased to the maximum possible value.

The control flow for DGD as described implies that DGD will work even when non-TCP traffic occurs. Under this condition, DGD depends on the ARP protocol feedback only, and the related relatively long lifetime values for ARP cache entries will slow down the detection of dysfunctional gateways. However, DGD will still allow you to configure primary (lower cost) and secondary (higher cost) gateways, and it handles the failover from a dysfunctional primary gateway to the secondary backup gateway.

One important aspect in respect to passive DGD must be considered in security sensitive environments. There are many cases where TCP could mistake a functional gateway for being dysfunctional: The destination that TCP is trying to reach may be turned off, has crashed, be unreachable, or be non-existent. Also, packets may be filtered by a firewall or an other security mechanism on the way to the destination to name just one possibility. In these cases, the ARP entry for the gateway in use will be deleted in order to force dead gateway detection to be initiated and to find out if the gateway is actually down. This will cause extra overhead and traffic on the network for the ARP packets to be sent, and also for other connections to wait for an ARP response. In general, this extra overhead will be fairly minimal. It does not happen very often that a connection will be attempted to an unreachable address, and the overhead associated with an ARP is quite small. However, the possibility exists that malicious users could continually try to connect to addresses they knew to be unreachable to purposely degrade performance for other users on the system and generate extra traffic on the network.

To protect systems and users against these types of attacks, a new network option named `passive_dgd` was introduced with the implementation of DGD in AIX 5L. The `passive_dgd` default value is 0, indicating that passive DGD will be off by default. The network option `passive_dgd` can be displayed and changed by the `no -o` command. The `no` command output in the following example shows the value for the `passive_dgd` on a system where this specific network option is set to the default of 0:

```
# no -o passive_dgd
passive_dgd = 0
```

If you want to permanently enable passive DGD, you will have to include the following command line in the network startup script `/etc/rc.net`:

```
no -o passive_dgd=1
```

Active dead gateway detection

Passive dead gateway detection has low overhead and is recommended for use on any network that has redundant gateways. However, passive DGD is done on a best-effort basis only. Some protocols, such as UDP, do not provide any feedback to the host if a data transmission is failing, and in this case, no action can be taken by passive DGD. Passive DGD detects that a gateway is down only if it does not respond to ARP requests.

When no TCP traffic is being sent through a gateway, passive DGD will not sense a dysfunctional state of the particular gateway. The host has no mechanism to detect such a situation until TCP traffic is sent or the gateway's ARP entry times out, which may take up to 20 minutes. But this situation does not modify route costs. In other words, a gateway not forwarding packets is not considered dead.

This behavior is unacceptable in information technology environments with very strict availability requirements. AIX 5L offers a second DGD mechanism, specifically for these environments, named Active dead gateway detection. Active DGD will ping gateways periodically, and if a gateway is found to be down, the routing table is changed to use alternate routes to bypass the dysfunctional gateway.

A new network option called `dgd_ping_time` will allow the system administrator to configure the time interval between the periodic ICMP echo request/reply exchanges (ping) in units of seconds. The network option `dgd_ping_time` can be displayed and changed by the `no -o` command and is set to 5 seconds by default. The `no` command output in the following example shows the value for `dgd_ping_time` on a system where this specific network option is set to the default of 5:

```
# no -o dgd_ping_time
dgd_ping_time = 5
```

You should include an appropriate `no` command line in the `/etc/rc.net` file to ensure that a value for this network option, which deviates from the default, stays in effect across reboots of your system.

Active dead gateway detection will be off by default and we recommend that you use it only on machines that provide critical services and have high-availability requirements. Since active DGD imposes extra network traffic, network sizing and performance issues have to receive careful consideration. This applies

especially to environments with a large number of machines connected to a single network.

Active DGD operates on a per-route basis, and it is turned on by the new parameter argument `-active_dgd` of the **route** command. The following example shows how the **route** command is used to add a new default route through the 9.3.240.58 gateway with a user-configurable cost of 2, and which is under the surveillance of active DGD:

```
# route add default 9.3.240.58 -active_dgd -hopcount 2
```

The **netstat -C** command lists the routes defined to the system, including their current and user-configurable cost. The new flag A, as listed for the default route through the 9.3.240.58 gateway, indicates that the active DGD for this particular route is turned on.

```
# netstat -C
Routing tables
Destination      Gateway          Flags    Refs      Use    If    Cost Config_Cost

Route Tree for Protocol Family 2 (Internet):
default          9.3.240.59      UG        3    104671  tr1    2        2 =>
default          9.3.240.58      UGA       0         0  tr1    2        2
9.3.240/24       server2         U        32    67772  tr1    0        0
127/8            loopback        U         6     1562  lo0    0        0

Route Tree for Protocol Family 24 (Internet v6):
::1              ::1             UH         0         0  lo0    0        0
```

The kernel will keep a list of all the gateways that are subject to active DGD. Each time `dgd_ping_time` seconds pass, all the gateways on the list will be pinged. A pseudo-random number is used to slightly randomize the ping times. If several hosts on the same network use active DGD, the randomized ping times ensure that not all of the hosts ping at exactly the same time. If any gateways fail to respond, they will be pinged several times repeatedly with a 1 second pause between pings. The total number of times they are pinged will be determined by the `dgd_packets_lost` network option. This network option was already introduced in “Passive dead gateway detection” on page 468, but note that this option has a slightly different meaning for passive DGD compared to active DGD.

The network option `dgd_packets_lost` in passive DGD refers to the number of TCP packets lost (if any) in the course of data transmission, whereas for active DGD, the option is specifically related to the packets used in an ICMP echo request/reply exchange (ping) to sense the state of the gateways that are under the surveillance of active DGD.

If the gateway does not respond to any of these pings, it will be considered to be down, and the costs of all routes using that gateway will be increased to the

maximum value, which is defined to be MAX_RT_COST. MAX_RT_COST in turn is equal to INT_MAX=2147483647, the highest possible value for an integer. These definitions can be examined in the /usr/include/net/route.h and the /usr/include/sys/limits.h header files, which are optionally installed on your system as part of the bos.adt.include fileset.

The gateway will remain on the list of gateways to be pinged, and if it responds at any point in the future, the costs on all routes using that gateway will be restored to their user-configured values.

Passive DGD does not decrease the cost on any route for which active detection is being done, as active detection has its own mechanism for recovery when a gateway comes back up. However, passive DGD is allowed to increase the cost on a route for which active detection is in use, as it is quite likely that passive detection will discover the outage first when TCP traffic is being sent.

DGD network options and command changes

Four new network options are defined for dead gateway detection and all of them are runtime attributes that can be changed at any time. Table 8-1 provides details of the attributes of these options.

Table 8-1 Network options for dead gateway detection

Network option	Default	Description
dgd_packets_lost	3	Specifies how many consecutive packets must be lost before dead gateway detection decides that a gateway is down.
dgd_ping_time	5	Specifies how many seconds should pass between pings of a gateway by active dead gateway detection.
dgd_retry_time	5	Specifies how many minutes a route's cost should remain raised when it has been raised by passive dead gateway detection. After this number of minutes passes, the route's cost is restored to its user-configured value.
passive_dgd	0	Specifies whether passive dead gateway detection is enabled. A value of 0 turns it off, and a value of 1 enables it for all gateways in use.

If the customized DGD network attributes are intended to be permanent, the system administrator must include the appropriate **no** command in /etc/rc.net. Otherwise, the customized network options will be reset to their default during a system boot.

For example, if you want to turn on passive DGD permanently, you have to include the following line in `/etc/rc.net`:

```
# The following no command enables passive Dead Gateway Detection
# after each system boot
if [ -f /usr/sbin/no ] ; then
    /usr/sbin/no -o passive_dgd=1
fi
```

DGD sample configuration

Figure 8-1 on page 476 depicts the basic system environment that will be used throughout this section to give an example for active dead gateway detection. Server1 attached to the token-ring network 9.3.240.0 (netmask 255.255.255.0) has two default routes to the Client1 computer in the Ethernet segment 10.47.0.0 (netmask 255.255.0.0). One route goes through the Gateway1, which has a token-ring interface `tr0` with the IP address 9.3.240.58 and an Ethernet interface `en0` with the IP address 10.47.1.1. The second route uses Gateway2, which is configured to have a token-ring interface `tr0` with the IP address 9.3.240.59 and an Ethernet interface `en0` with the IP address 10.47.1.2. The **no -o ipforwarding=1** command was used on both gateway systems to enable the gateway function. The Ethernet interface of Client1 has the IP address of 10.47.1.3. Server1 and Client1 run AIX 5L, and on both systems, the **no -o tcp_pmtu_discover=0** and the **no -o udp_pmtu_discover=0** commands were used to disable dynamic PMTU discovery interference with multipath routing. Also on both computers, the `passive_dgd` network option was set to 1 by the **no -o passive_dgd=1** command to enable passive DGD. It is not required to have passive DGD enabled in order to use the active DGD function, but for TCP-based network traffic, passive DGD may initiate the failover to the backup gateway earlier than active DGD normally would. If the network traffic is not TCP-based, then the active ping of the gateways by active DGD will get the information about the state of the gateway faster than passive DGD potentially could get it through the expiration of the ARP cache entry.

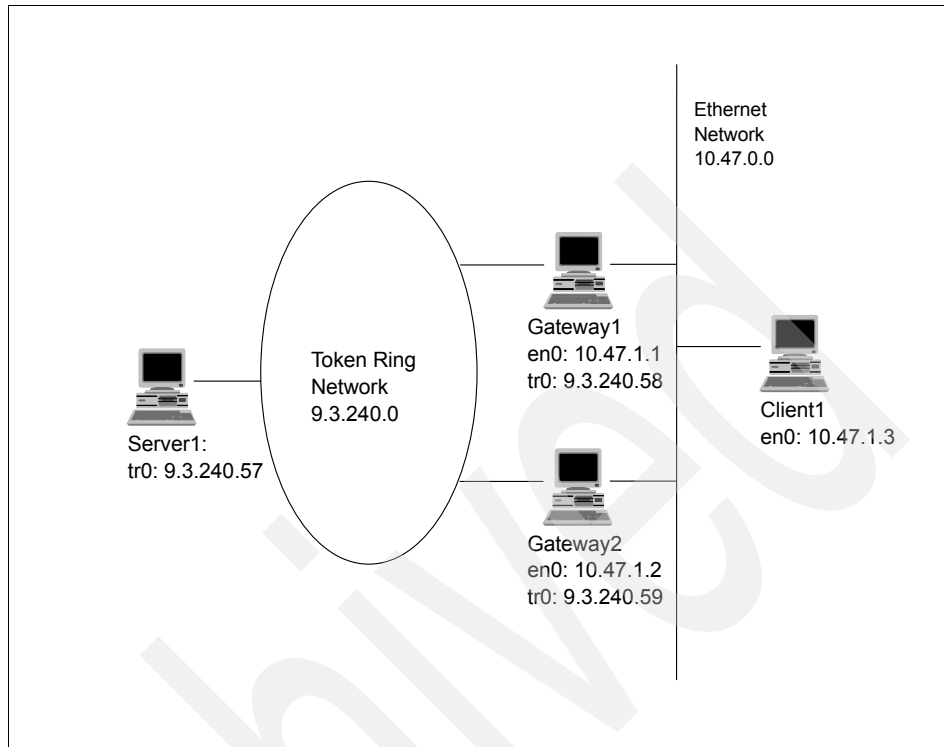


Figure 8-1 DGD sample configuration

For Server1 and Client1, the default routes were configured through the SMIT menu Add Static Route, which you can access directly with the **smit mkroute** command. The default routes were defined to have the same user-configurable cost, but to use different gateways. The underlying SMIT script, which is associated with the Add Static Route SMIT task, uses the **chdev** command for the **inet0** device to permanently define routes. The **route** command affects only the current kernel routing table, and all additions and changes applied to the routing table will be lost after a system boot.

The **netstat -Cn** command output, shown in the following lines, reflects the routing table entries that were made. The reference count for both gateway routes is 2, because after the set up of the routing environment, four telnet sessions to Client1 were initiated from Server1. Multipath routing ensured (through cyclic multiplexing) that the sessions are divided evenly among the two default routes. The flag A in the Flags column indicates that active DGD is set for both default routes:

```
# netstat -Cn
Routing tables
```

Destination	Gateway	Flags	Refs	Use	If	Cost	Config_Cost
Route Tree for Protocol Family 2 (Internet):							
default	9.3.240.58	UGA	2	154	tr1	2	2 =>
default	9.3.240.59	UGA	2	177	tr1	2	2
9.3.240/24	9.3.240.57	U	4	160	tr1	0	0
127/8	127.0.0.1	U	4	190	lo0	0	0
Route Tree for Protocol Family 24 (Internet v6):							
::1	::1	UH	0	0	lo0	0	0

To test the active DGD feature, the **ifconfig tr0 down** command was used to disable the gateway function of Gateway1. After the takeover has been completed, **netstat -Cn** returns the following output:

```
# netstat -Cn
Routing tables
```

Destination	Gateway	Flags	Refs	Use	If	Cost	Config_Cost
Route Tree for Protocol Family 2 (Internet):							
default	9.3.240.59	UGA	4	604	tr1	2	2 =>
default	9.3.240.58	UGA	0	245	tr1	MAX	2
9.3.240/24	9.3.240.57	U	5	479	tr1	0	0
127/8	127.0.0.1	U	0	190	lo0	0	0
Route Tree for Protocol Family 24 (Internet v6):							
::1	::1	UH	0	0	lo0	0	0

The reference count for the route through Gateway1 has dropped from 2 to 0 and both associated connections are now handled by the backup route through Gateway2. In order to mark the dysfunctional gateway as unusable, the current cost of that route was set to the maximum possible value, as indicated by the keyword MAX.

8.3.3 User interface for multipath routing and DGD

System management tasks that are related to the new multipath routing and DGD features are supported on the command line interface level by new parameters and flags to the **route** and **netstat** commands.

Two parameters were added to the **route** command in order to support the multipath routing feature. The -hopcount argument of the route parameters requires a positive integer as the variable value. The variable value refers to the user-configurable cost for a given route and supposedly relates to the maximum number of gateways in the route. However, the ultimate objective in introducing the user-configurable costs for a route is to implement a priority hierarchy among the defined routes. The new -if argument must be supplemented by a variable

that takes a defined network interface as the variable value. The **-if** argument specifies the interface to associate with a route so that packets will be sent using this interface when the given route is chosen.

In addition to the two new parameters that support multipath routing, one parameter was specifically added to the **route** command to implement active DGD. The name of this parameter is **active_dgd**, and whenever this parameter is given during the definition of a route, active DGD will be enabled for the particular route.

Note that the **route** command only changes the kernel routing table but does not permanently change the attributes of the **inet0** device.

To preserve route definitions across system boot processes, you have to change the attributes of the **inet0** device either by using the **chdev** command or with the aid of the Add Static Route SMIT menu.

Table 8-2 provides an overview of the new parameters added to the **route** command that support the new routing features in AIX 5L.

Table 8-2 The route command parameters for multipath routing and DGD

Parameter argument	Argument variable	Description
-active_dgd	NA	Enables active DGD on given route
-hopcount	n	Specifies relative cost of a given route if the n variable is a positive integer
-if	ifname	Specifies the interface ifname (en0, tr0, ...) to associate with this route so that packets will be sent using this interface when this route is chosen

The new **-C** flag (as shown in Table 8-3 on page 479) was added to the **netstat** command to provide additional routing table information. The **netstat -C** command displays the routing tables, including the user-configured and current costs of each route.

The current cost is either dynamically determined during the route definition process and reflects the number of gateways in the route or it is equal to the user-configured cost. The user-configurable costs can be set just for the routes in the current kernel routing table using the **route** command with the **-hopcount** parameter, or they are permanently defined by the appropriate **chdev** command as attributes of the **inet0** device. The current cost may be different than the user-configured cost if dead gateway detection has changed the cost of the route.

Table 8-3 New netstat command flag

Command	Description
netstat -C	Shows the routing tables, including the user-configured and current costs of each route. The user-configured cost is set using the -hopcount flag of the route command. The current cost may be different than the user-configured cost if dead gateway detection has changed the cost of the route.

More details about the command line interfaces for multipath routing and DGD are given in “Passive dead gateway detection” on page 468, “Active dead gateway detection” on page 472, and in the standard AIX documentation library.

In addition to the command line interface for configuration and administration of the multipath routing and DGD feature, AIX 5L provides graphical user interface support for the relevant systems management tasks through SMIT and the Web-based System Manager tool.

The menus of the System Management Interface Tool (SMIT), which assists the addition of a static route for IP Version 4 (IPv4) and for IP Version 6 (IPv6), were changed to accommodate the new user-configurable metric (cost) option, to account for the added flexibility needed to associate a particular interface with a specific route, and to support dead gateway detection.

In the SMIT menus, Add a Static Route and Add an IPv6 Static Route, three new fields were added to take input for the underlying SMIT script, which in turn uses the **chdev** command to set the route attribute for the inet0 Internet network extension. Refer to Table 8-4 for further details about the field definition.

Table 8-4 Static Route and Add an IPv6 Static Route SMIT menu new fields

Field	Description
Network Interface (interface to associate route with)	Specifies the interface (en0, tr0, ...) to associate with this route so that packets will be sent using this interface when this route is chosen
COST	User-configurable distance metric for route
Enable Active Gateway Detection	Enables active DGD on the route

In order to add an alternate default route to your system, you will have to use the keyword default as the destination address in the SMIT input panel.

The SMIT fast paths mkroute and mkroute6 bring you directly to the SMIT menus for IPv4 and IPv6 (that are related to the systems management task) to add a static route. Figure 8-2 on page 480 depicts the SMIT menu Add Static Route, which supports the IPv4 specific task.

Add Static Route

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

Destination TYPE	[Entry Fields]	
* DESTINATION Address (dotted decimal or symbolic name)	[]	+
* Default GATEWAY Address (dotted decimal or symbolic name)	[]	
COST	[0]	#
Network MASK (hexadecimal or dotted decimal)	[]	
Network Interface (interface to associate route with)	[]	+
Enable Active Dead Gateway Detection?	no	+

F1=Help
F5=Reset
F9=Shell

F2=Refresh
F6=Command
F10=Exit

F3=Cancel
F7=Edit
Enter=Do

F4=List
F8=Image

Figure 8-2 Add Static Route SMIT menu

The Web-based System Manager environment for multipath routing and DGD is accessible through the following sequence of menu selections on the Web-based System Manager console:

1. Select **Network -> TCPIP (IPv4 and IPv6) -> Protocol Configuration -> TCP/IP**.
2. Select **Configure TCP/IP -> Advanced Methods**. Click **Static Routes**.
3. Complete the following in the Add/Change a Static Route menu: Destination Type, Gateway address, Network interface name (drop-down menu), Subnet mask, Metric (Cost), and the Enable active dead gateway detection check box.
4. Click **Add/Change Route**.

Figure 8-3 on page 481 shows the Web-based System Manager menu for static route management related tasks.

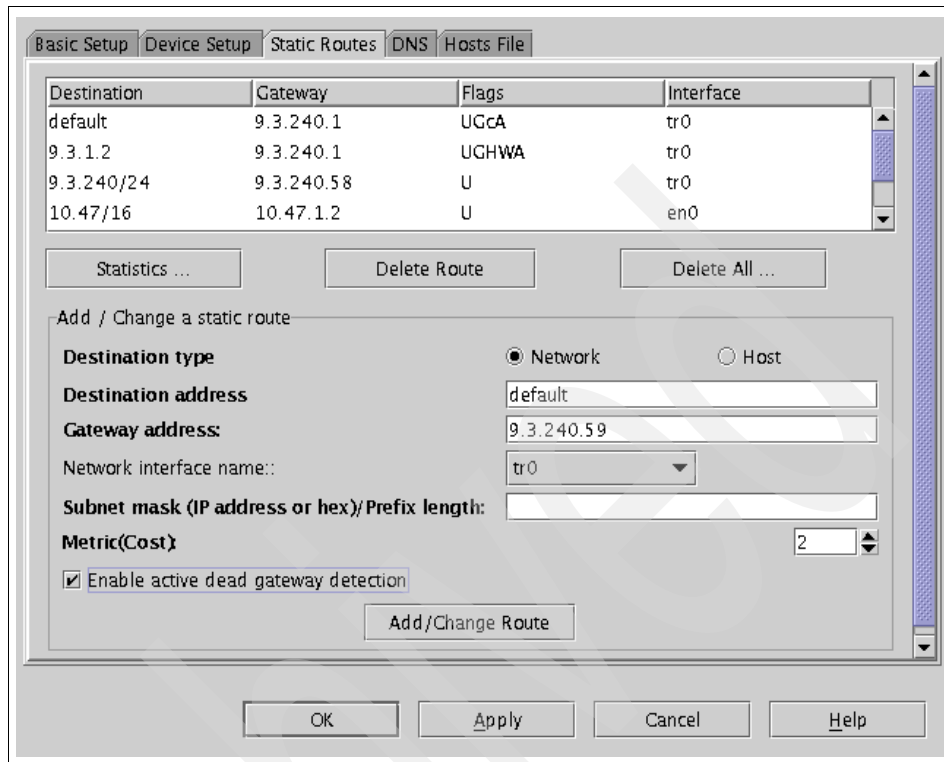


Figure 8-3 Web-based System Manager menu for static route management

8.4 TCP/IP general enhancements

The following are the enhancements for TCP/IP on AIX 5L.

8.4.1 Split-connection proxy systems (5.1.0)

Many designs for Internet services use split-connection proxies, in which a proxy machine is interposed between the server and the client machines in order to mediate the communication between them. Split-connection proxies have been used for everything from HTTP caches to security firewalls to encryption servers. Split-connection proxy designs are attractive because they are backwards compatible with existing servers, allow administration of the service at a single point (the proxy), and typically are easy to integrate with existing applications.

Current application layer proxies suffer major performance penalties, as they spend most of their time moving data back and forth between connections,

context switching, and crossing protection boundaries for each chunk of data they handle. For more information, please visit:

<http://www.cs.umd.edu/~pravin/publications/publist.htm>

8.4.2 TCP splicing (5.1.0)

TCP splicing is a feature that pushes the data-relaying function of a proxy application into the kernel. This improves the performance by avoiding the context switches and data copying between kernel space and user space. This feature benefits any split-connection proxy system. A logical diagram is shown in Figure 8-4.

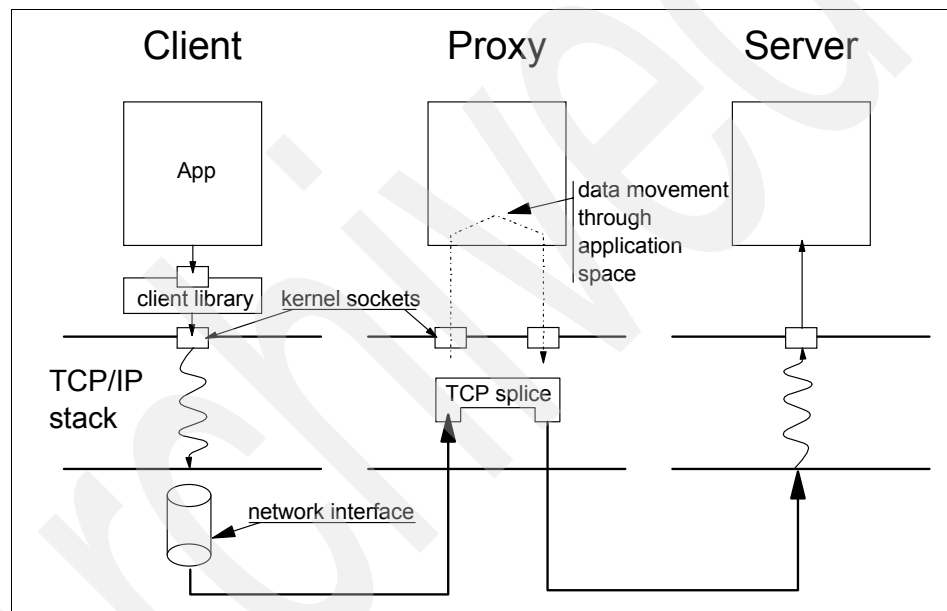


Figure 8-4 Basic architecture of split-connection application layer proxies

Splice subroutine

TCP splicing has been implemented by the splice() system call. The splice subroutine lets TCP manage two sockets that are in a connected state, thus relieving the caller from moving data from one socket to another. After the splice subroutine returns successfully, the caller needs to close the two sockets.

Syntax

The syntax of the splice() subroutine is:

```
#include <sys/types.h>
#include <sys/socket.h>
```



```
int splice(socket1, socket2, flags)
    int socket1, socket2;
    int flags;
```

Parameters

The following is a list of the parameters and their settings:

socket1, socket2	Specifies a socket that had gone through a successful connect() or accept(). The two sockets should be of type SOCK_STREAM and protocol IPPROTO_TCP. Specifying a protocol of zero also works.
flags	Set to zero. Currently ignored. In the future, different values could get supported.

Return values

Upon successful completion, splice() subroutine returns zero. On error, it returns -1. An errno will indicate the specific error.

Error Codes

The following are the available error codes and their definitions.

EBADF	socket1 or socket2 is not valid.
ENOTSOCK	socket1 or socket2 refers to a file, not a socket.
EOPNOTSUPP	socket1 or socket2 is not of type SOCK_STREAM.
EINVAL	The parameters are invalid.
EEXIST	socket1 or socket2 is already spliced.
ENOTCONN	socket1 or socket2 is not in connected state.
EAFNOSUPPORT	The sockets (socket1 or socket2) address family not supported for this subroutine.

Note: At the time of writing, no application is using the new socket system call splice(); therefore, basic performance numbers are not available. But it is expected that for proxy-type applications, the performance gain should be significant when a large amount of data is transferred. For short sessions, there may not be any gain.

8.4.3 UDP fragmentation (5.1.0)

With UDP data transfers, fragmentation occurs. The datagram in AIX 5L Version 5.1 is reassembled before the driver layer. Instead of individual packets being

sent to the driver, a chain of packets is sent, which overcomes multiple trips through the IP layer for each fragment, thus improving performance.

8.4.4 TCB headlock (5.1.0)

In previous versions of AIX, the global lock TCBHEAD_LOCK is part of a critical code path that impedes performance in loaded systems. The TCBHEAD_LOCK has been removed and replaced with an array of hash lists each with its own lock.

8.4.5 Explicit Congestion Notification (5.1.0)

The Explicit Congestion Notification (ECN) feature for TCP can be enabled by the new network option `tcp_ecn` with the **no** command.

Note: ECN capability is only available on the TCP layer.

Normally, TCP uses packet drops as an indication of congestion. With Explicit Congestion Notification, routers do not have to drop packets to notify congestion. An ECN-capable TCP receiver would notify the TCP sender of the congestion by setting a bit in the TCP header. On receipt of this notification from the TCP receiver, the TCP sender's congestion control response should be the same as its response to a dropped packet. Adding ECN capability to the TCP layer helps applications that are sensitive to delays or packet loss.

For TCP, ECN has three new functions:

- ▶ Negotiation between the end points during connection set up to determine if they are both ECN-capable
- ▶ An ECN-Echo (ECE) flag in the TCP header, so that the data receiver can inform the data sender when a Congestion Experienced (CE) packet has been received
- ▶ A Congestion Window Reduced (CWR) flag in the TCP header, so that the data sender can inform the data receiver that the congestion window has been reduced

This feature is created under the assumption that the source TCP uses the standard congestion control algorithms of slow-start, fast retransmit, and fast recovery (RFC2001).

Two new flags are created in the Reserved field of the TCP header. The TCP mechanism for negotiating ECN-capability uses the ECN-Echo (ECE) flag in the TCP header. Bit 9 in the Reserved field of the TCP header is designated as the

ECN-Echo flag. The location of the 6-bit Reserved field in the TCP header is shown in Figure 8-5.

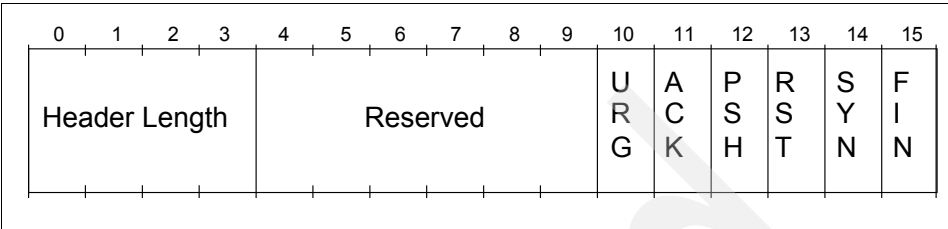


Figure 8-5 The previous definition of bytes 13 and 14 of the TCP header

To enable the TCP receiver to determine when to stop setting the ECN-Echo flag, a second new flag in the TCP header, the CWR flag, is introduced. The CWR flag is assigned to bit 8 in the Reserved field of the TCP header.

This specification of these fields leaves the Reserved field as a 4-bit field using bits 4–7, as shown in Figure 8-6.

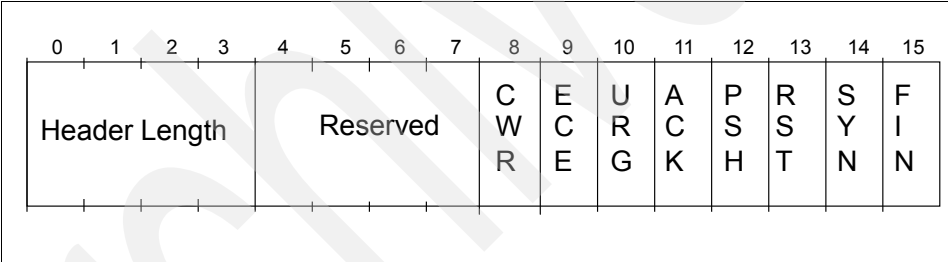


Figure 8-6 The new definition of bytes 13 and 14 of the TCP header

ECN uses the ECN Capable Transport (ECT) and CE flags in the IP header for signaling between routers and connection end points, and uses the ECN-Echo and CWR flags in the TCP header for TCP-endpoint to TCP-endpoint signaling.

For a TCP connection, a typical sequence of events in an ECN-based reaction to congestion is as follows:

1. The ECT bit is set in packets transmitted by the sender to indicate that ECN is supported by the transport entities for these packets.
2. An ECN-capable router detects impending congestion and detects that the ECT bit is set in the packet it is about to drop. Instead of dropping the packet, the router chooses to set the CE bit in the IP header and forwards the packet.
3. The receiver receives the packet with the CE bit set, and sets the ECN-Echo flag in its next TCP ACK sent to the sender.

4. The sender receives the TCP ACK with ECN-Echo set, and reacts to the congestion as if a packet had been dropped.
5. The sender sets the CWR flag in the TCP header of the next packet sent to the receiver to acknowledge its receipt of and reaction to the ECN-Echo flag.

For more detailed information about Explicit Congestion Notification, refer to

<http://www.aciri.org/floyd>

<http://www.ietf.org>

8.4.6 IPv6 API upgrade (5.1.0)

Starting with AIX 5L Version 5.1, the IPv6 protocol has been enhanced with three new library routines (getipnodebyname, getipnodebyaddr, and freehostent) as part of RFC2553. The fileset affected by these new routines is bos.rte.libc.

The getipnodebyname subroutine allows the caller more control over the types of addresses required and is thread safe and serves for node name-to-address translation. It also does not need a global option like RES_USE_INET6. The name argument can be either a node name or a numeric (either a dotted-decimal IPv4 or colon-separated IPv6) address.

The parameters of the getipnodebyname subroutine are listed in Table 8-5. In order to obtain a more detailed list of the flags used, refer to RFC2553.

Table 8-5 Parameters of getipnodebyname

Parameter	Description
name	Specifies either a node name or a numeric (either a dotted-decimal IPv4 or colon-separated IPv6) address
af	Specifies the address family, which is either AF_INET or AF_INET6
flags	Controls the types of addresses searched for and the types of addresses returned
error_num	Returns argument to the caller with the appropriate error code

The getipnodebyaddr subroutine serves for address-to-node name translation and is thread safe. The getipnodebyaddr subroutine is similar in its name query to the gethostbyaddr subroutine except in one case. If af equals AF_INET6 and the IPv6 address is an IPv4-mapped IPv6 address or an IPv4-compatible address, then the first 12 bytes are skipped over and the last 4 bytes are used as an IPv4 address with af equal to AF_INET to look up the name.

The parameters of the getipbnodeaddr subroutine are listed in Table 8-6 on page 487.

Table 8-6 Parameters of *getipnodebyaddr* subroutine

Parameter	Description
src	Specifies a node address. It is a pointer to either a 4-byte (IPv4) or 16-byte (IPv6) binary format address.
af	Specifies the address family, which is either AF_INET or AF_INET6.
len	Specifies the length of the node binary format address.
error_num	Returns argument to the caller with the appropriate error code.

The *freehostent* subroutine serves to free memory allocated by *getipnodebyname* and *getipnodebyaddr*. It frees any dynamic storage pointed to by elements of *ptr*. This includes the *hostent* structure and the data areas pointed to by the *h_name*, *h_addr_list*, and *h_aliases* members of the *hostent* structure.

8.4.7 Performance enhancements (5.2.0)

In AIX 5L Version 5.2, there have been several performance enhancements in the communications subsystem. With the introduction of machines with many processors, many interface cards, and a large number of hosts on a network, performance bottlenecks have been identified and removed in the following areas.

The address resolution protocol (ARP) table was enhanced by removing the single global lock protecting the table. With the large number of adapters and large number of hosts on the network, the ARP table was getting larger and the global lock was becoming a bottleneck. The single ARP table lock was removed and replaced with a lock for each ARP bucket. The lock granularity is reduced and so ARP bucket operations can now proceed in parallel.

Applications running on SMP machines that use the loopback interface (lo0) for socket communications hit another bottleneck. The loopback interface dequeues the data on an off-level interrupt generally only on one CPU. If the CPU handling the loopback interface is busy, data will be backed up waiting for the handler to run. This was fixed by performing loopback processing and interrupt handling for the loopback handler on a per-CPU basis.

Servers that create or service a significant number of UDP read/writes or extensive use of interface lookups, will experience a bottleneck on the *INIFADDR_LOCK*. This has been fixed by creating a hashed interface to the address entries and multiple locks for each bucket. The lock granularity is reduced and so *INIFADDR* bucket operations can now proceed in parallel.

8.4.8 TCP/UDP inpcb hash table tunable enhancements (5.2.0)

The communication subsystem in Version 5.2 has been enhanced to allow independent tuning of the TCP and UDP inpcb hash tables. AIX stores all connection-related information for sockets in the protocol control block (PCB) structures in the inpcb hash tables.

Prior to Version 5.2, the TCP and UDP inpcb hash tables were both fixed to be the same size. The fixed hash table size did not allow the administrator to tune the table size based on the number of connections the machines handled or for the popularity of the TCP protocol over the UDP protocol.

In Version 5.2, you can now independently tune the TCP and UDP hash table sizes to reflect the workload and network protocol usage on the machine. The network options for the TCP and UDP hash table size are `tcp_inpcb_hashtab_siz` and `udp_inpcb_hashtab_siz`. You change these network options with the `no` command. The machine must be rebooted to have the changes take effect.

The following example shows how to set the size of the TCP hash table to 31000 and the UDP hash table to 21000. You must use the `-r` flag with the `no` command so these changes will take effect on the next reboot.

```
# no -r -o tcp_inpcb_hashtab_siz=31000 -o udp_inpcb_hashtab_siz=21000
# no -L tcp_inpcb_hashtab_siz -L udp_inpcb_hashtab_siz
```

	NAME	VALUE	DEFAULT	BOOT	MIN	MAX	UNIT	TP
	tcp_inpcb_hashtab_siz	24499	24499	31000	1	999999	numeric	R
	udp_inpcb_hashtab_siz	24499	24499	21000	1	83000	numeric	R

8.4.9 TCP keep alive enhancements (5.2.0)

Version 5.2 added three new TCP socket options (TCP_KEEPIDLE, TCP_KEEPINTVL, and TCP_KEEPCNT) to the `getsockopt` and `setsockopt` subroutines. This enhancement allows application developers to specify TCP keepalive parameters for each socket. These options are only valid when the `SO_KEEPAIVE` option is set. The following new options have been added to the `netinet/tcp.h` header file.

- TCP_KEEPIDLE

Specifies the number of seconds of idle time on a connection after which TCP sends a keepalive packet. The socket option value is inherited from the parent socket from the `accept` system call. The default value is 7200 seconds.
- TCP_KEEPINTVL

Specifies the interval of time between keepalive packets, measured in seconds. This socket option is inherited from the parent socket from the `accept` system call. The default value is 75 seconds.

TCP_KEEPCNT Specifies the maximum numbers of keepalive packets to be sent to validate a connection. This socket option value is inherited from the parent socket. The default is 8.

A new network tunable option for TCP keepalive count was added. This option represents the number of keepalive probes that could be sent before terminating the connection. The default value of this option is 8 and the maximum value is 32. To modify this value use the **no** command.

```
# no -L tcp_keepcnt
```

	NAME	VALUE	DEFAULT	BOOT	MIN	MAX	UNIT	TP
	tcp_keepcnt	8	8	8	0	32MAX	numeric	D

8.4.10 Asynchronous accept() routine supported (5.2.0)

Version 5.2 now supports the `accept()` routine for the I/O completion port (IOCP) mechanism to implement asynchronous I/O.

Normally when a server is listening on a socket and it calls `accept()`, it will block, which is wasteful of computational resources. If the server calls an asynchronous `accept()`, the program can continue to process other tasks immediately. When the accept is completed, the application is notified about the completion of the accept, through threads performing `GetQueuedCompletion Status` on the IOCP. The application can then choose how to handle the event.

In order to use the IOCP mechanism, you must install the `bos.iocp.rte` filesset using `installp`, SMIT, or the Web-Based System Manager. You must then enable the IOCP interface either using the command line or the SMIT interface. The SMIT interface can be located using the following fast path `iocp`. The following example shows how to configure the `iocp0` device using the `mkdev` command.

```
# mkdev -l iocp0
# lsdev -C -l iocp0
iocp0 Available I/O Completion Ports
```

8.4.11 IPv6 functional update (5.2.0)

The following section discusses the enhancements to IPv6 made in AIX 5L Version 5.2.

New socket options

AIX 5L Version 5.2 introduces two new socket options, `IPV6_CHECKSUM` and `ICMP6_FILTER`, to be used with the `getsockopt` and `setsockopt` subroutines.

The IPV6_CHECKSUM socket option specifies that the kernel computes checksums over the IPv6 pseudo headers and the data for a raw socket. The kernel will compute checksums for outgoing packets and verify checksums on incoming packets on that socket. Incoming packets with incorrect checksums will be discarded. The user must specify an offset into user data where the checksum is to be stored. The following example shows how to use the **netstat** command to display the invalid checksum packet count.

```
# netstat -s -p  ipv6
ipv6:
    112 total packets received
...
    0 packets dropped due to the full socket receive buffer
    0 packets not delivered due to bad raw IPv6 checksum
    0 message responses generated
```

The ICMP6_FILTER socket option allows the user to filter incoming ICMPV6 messages by the ICMPV6 type field. The following section shows the macros defined in `netinet/icmp6.h` to assist developers with modifying the ICMP6_FILTER option.

```
ICMP6_FILTER_SETPASS(type, filterp)
ICMP6_FILTER_SETBLOCK(type, filterp)
ICMP6_FILTER_WILLPASS(type, filterp)
ICMP6_FILTER_WILLBLOCK(type, filterp)
ICMP6_FILTER_SETPASSALL(filterp)
ICMP6_FILTER_SETBLOCKALL(filterp)
```

getaddrinfo subroutine update

The following flags were added to the `getaddrinfo` subroutine.

AI_NUMERICSERV If this flag is specified, the supplied servname is a numeric port string. Otherwise, an EAI_NONAME error is returned. This flag prevents any type of name resolution from being invoked.

AI_V4MAPPED If this flag is specified along with an `ai_family` of `AF_INET6`, the `getaddrinfo` subroutine returns IPv4-mapped IPv6 addresses when no matching IPv6 addresses are found.

AI_ALL If this flag is used with the `AI_V4MAPPED` flag, the `getaddrinfo` subroutine returns all matching IPv6 and IPv4 addresses. IPv4 addresses, if any, will be returned in the IPv4-mapped IPv6 address format.

AI_ADDRCONFIG If this flag is specified, a query for AAAA or A6 records should occur only if the node has at least one IPv6 source address configured. A query for A records should occur

only if the node has at least one IPv4 source address configured.

The `getaddrinfo` and `getnameinfo` subroutines no longer return `EAI_NODATA`, they now return `EAI_NONAME`.

autoconf6 command update

The **autoconf6** command has been enhanced to allow IPv6 to be started without having IPv4 configured.

Prior to Version 5.2, the `-i iflist` flag would only configure the interfaces specified in `iflist` that already had IPv4 addresses. This behavior has been enhanced in Version 5.2, where the `-i iflist` flag now configures the specified interfaces with IPv6 addresses even if IPv4 is not configured on them.

The `-a` flag configures all interfaces that already have IPv4 addresses configured. A new flag, `-A`, configures all interfaces whether or not the interface has an IPv4 address configured. If the `-a`, `-i`, or `-A` flag is not specified, then IPv6 will be started only on the interfaces that have IPv4 addresses configured.

Prior to Version 5.2, the default behavior of the **autoconf6** command is to always load the `sit()` interface. In Version 5.2, running **autoconf6** with the `-A` or `-i` flags will only configure the `sit()` interface if an IPv4 address is configured on the system. If the `-A` or `-i` flags are not used, the `sit()` interface will be configured by default.

8.5 TCP/IP RAS enhancements (5.1.0)

The TCP/IP Reliability, Availability, and Serviceability (RAS) is extended with enhancements described in this section.

8.5.1 Snap enhancement

The **snap** command is modified to provide more configuration files when running the `-t` flag. For a detailed listing of the TCP/IP configuration files, see “The snap command enhancements” on page 277.

8.5.2 Network option enhancements

The **no** command, used to set network options, has been enhanced in AIX 5L Version 5.1.

Use of syslog to log messages

The **no** command logs a message to the syslog using the LOG_KERN facility when any networking kernel option is set. This message includes the option name, value, time, and UID value.

For example, the **no** option rfc2414 is set to 1 and then back to 0. Make sure the syslog daemon is running and the destination of the output of the syslog daemon is defined in the /etc/syslog.conf file. The output of the log file would appear similar to the following:

```
Mar 12 16:14:17 server3 syslogd: restart
Mar 12 16:14:21 server3 no[22084]: Network option rfc2414 was set to the value
1
Mar 12 16:14:26 server3 no[22086]: Network option rfc2414 was set to the value
0
```

The sodebug network option

A new network option named sodebug is added to the options of the **no** command. This option sets the SO_DEBUG flag on any socket that is created. The TCP protocol records outgoing and incoming packet events when the socket used has had the SO_DEBUG option turned on for the socket.

New Reno algorithm for Fast Recovery

In the typical implementation of the TCP Fast Recovery algorithm (first implemented in the 1990 BSD Reno release, and referred to as the Reno algorithm), the TCP data sender only retransmits a packet after a retransmit timeout has occurred, or after three duplicate acknowledgments have arrived triggering the Fast Retransmit algorithm. A single retransmit timeout might result in the retransmission of several data packets, but each invocation of the Reno Fast Retransmit algorithm leads to the retransmission of only a single data packet.

The network option tcp_newreno enables the modification the TCP's Fast Recovery algorithm, as described in RFC2582. This fixes the limitation of TCP's Fast Retransmit algorithm to quickly recover from dropped packets when multiple packets in a panel are dropped. In AIX 5L Version 5.1, the default of tcp_newreno is on (1).

RFC2414: Increasing TCP's initial window

The **no** option rfc2414 enables the increasing of TCP's initial window, as described in RFC2414. The default is off (0). Set this to 1 to turn it on. When it is on, the initial window will depend on the setting of the tunable option tcp_init_window.

Initial TCP window

The network option `tcp_init_window` is only used when `rfc2414` is turned on. If `rfc2414` is on and this value is zero, then the initial window computation is done according to RFC2414. If this value is not zero, the initial (congestion) window is initialized for a number of maximum sized segments equal to `tcp_init_window`.

Explicit Congestion Notification

The network option `tcp_ecn` enables TCP level support for Explicit Congestion Notification, as described in RFC2481. The default is off (0). Turning it on (1) will make all connections negotiate ECN capability with the peer. For this feature to work, you need support from the peer TCP and also IP-level ECN support from the routers in the path.

For more detailed information, see 8.4.5, “Explicit Congestion Notification (5.1.0)” on page 484.

Limited transmit for TCP loss recovery

Limited transmit is a new Transmission Control Protocol (TCP) mechanism that is used to more effectively recover lost segments when a connection's congestion window is small, or when a large number of segments is lost in a single transmission window. The Limited Transmit algorithm calls for sending a new data segment in response to each of the first two duplicate acknowledgments that arrive at the sender. Transmitting these segments increases the probability that TCP can recover from a single lost segment using the fast retransmit algorithm, rather than using a costly retransmission timeout. Limited transmit can be used both in conjunction with, and in the absence of, the TCP selective acknowledgment (SACK) mechanism.

The network option `limited_transmit` enables the enhanced TCP's loss recovery. The default is on (1).

8.5.3 The `iptrace` command enhancement

The `iptrace` command has been modified to keep track of the number of bytes of data written. If a log file limit is specified and the number of bytes written reaches this limit, the current log file will be renamed with the `.old` extension and data will be written to the new file without the extension. When `iptrace` is started with the log limit set, it will rename any existing log file to one with the `.old` extension. When the log limit option is not specified using the `-L` option, then `iptrace` behavior is the same as the past version.

Using `iptrace` with the `-P` flag, the command expects a comma-separated list of protocols.

Using the **iptrace** command with the **-p** flag, the command expects a comma-separated list of ports.

The syntax is as follows:

```
/usr/sbin/iptrace [ -a ] [ -e ] [ -PProtocol_list ] [ -iInterface ]
[ -pPort_list ] [ -sHost [ -b ] ] [ -dHost [ -b ] ] [ -L Log_size ] LogFile
```

Table 8-7 lists the flags of the **iptrace** command.

Table 8-7 The iptrace command flags

Flag	Description
-P Protocol_list	Records packets that use the protocol specified by the Protocol_list variable, which is a comma-separated list of protocols. The protocols can be a decimal number or a name from the /etc/protocols file.
-p Port_list	Records packets that use the port number specified by the Port_list variable, which is a comma-separated list of ports. The port variable can be a decimal number or a name from the /etc/services file.
-L Log_size	This option causes iptrace to log data so that the LogFile is copied to LogFile.old at the start and also every time it becomes approximately Log_size bytes long.

8.5.4 Trace enhancement

The following enhancements may help network problem determination. For more information on **trace**, see 5.2.1, “The trace command enhancements” on page 267.

The -C flag enhancement

Running the **trace** command with the **-C** flag traces one set of buffers per CPU in the CPUList. The CPUs can be separated by commas, or enclosed in double quotation marks and separated by commas or blanks. To trace all CPUs, specify all.

Since this flag uses one set of buffers per CPU, and produces one file per CPU, it can consume large amounts of memory and file space, and should be used with care. The files produced are named trcfile, trcfile-0, trcfile-1, and so on, where 0, 1, and so on are the CPU numbers. If **-T** or **-L** are specified, the sizes apply to each set of buffers and each file. On a uniprocessor system, you may specify **-C all**, but **-C** with a list of CPU numbers is ignored. If **-C** is used to specify more than one CPU, such as **-Call** or **-C "0 1"**, the associated buffers are not put into the system dump.

Additional trace hooks

A trace hook identifier is a three-digit hexadecimal number that identifies an event being traced. You specify the trace hook identifier in the first twelve bits of the hook word.

Trace hook identifiers are defined in the `/usr/include/sys/trchkid.h` file. The values `0x010` through `0x0FF` are available for use by user applications. All other values are reserved for system use. The currently defined trace hook identifiers can be listed using the `trcrpt -j` command.

The hook type identifies the composition of the event data and is user-specified.

Beginning with AIX 5L Version 5.1, the trace hooks `HKWD_TCPIP` and `HKWD_SOCKET` are replaced by the following hooks:

HKWD_SOCKET(252)	Only socket calls
HKWD_TCP (25B)	Only TCP function trace
HKWD_UDP (25C)	Only UDP function trace
HKWD_IP (25D)	Only IP function trace
HKWD_IP6 (25E)	Only IP6 function trace
HKWD_PCB (25F)	Traces all PCB related functions
HKWD_SLOCKS (253)	Traces all locks in socket and TCP/IP functions

8.6 Virtual IP address support

In previous AIX releases, an application had to bind to a real network interface in order to get access to a network or network services. If the network became inaccessible or the network interface failed, the application's TCP/IP session was lost, and the application was no longer available.

To overcome application availability problems as described, AIX 5L offers support for virtual IP addresses (VIPA) for IPv4 and IPv6. The VIPA-related code is part of the `bos.net.tcp.client` fileset, which belongs to the `BOS.autoi` and `MIN_BOS.autoi` system bundles, and therefore will always be installed on your AIX system.

With VIPA, the application is bound to a virtual IP address, not a real network interface that can fail. When a network or network interface failure is detected (using routing protocols or other schemes), a different network interface can be used by modifying the routing table. If the rerouting occurs fast enough, then TCP/IP sessions will not be lost.

A traditional IP address is associated with a specific network adapter. Virtual IP addresses are supported by a network interface that is not associated with any particular network adapter. The VIPA system management tasks are supported by the appropriate changes and additions to the interface-related high-level operating system commands **mkdev**, **chdev**, **rmdev**, **lsdev**, **lsattr**, **ifconfig**, and **netstat**. Also, all VIPA management tasks are covered by SMIT and the Web-based System Manager tool.

The following example shows how to configure a virtual interface (vi0) for the Internet address 9.3.160.120 with the netmask of 255.255.255.0, using the **mkdev** command.

The virtual interface belongs to the device class **if**, the Subclass **VI**, and the device type **vi**.

```
# mkdev -c if -s VI -t vi -a netaddr='9.3.160.120' -a netmask='255.255.255.0'
-w 'vi0' -a state='up'
```

You can also use the SMIT fast path **mkinetvi** (**smit mkinetvi** command) to get access to the relevant SMIT menu, as shown in Figure 8-7.

Add a Virtual IP Address Interface

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

<p>* INTERNET ADDRESS (dotted decimal)</p> <p>* Network MASK (hexadecimal or dotted decimal)</p> <p>* Network Interface</p> <p>* ACTIVATE the Interface after Creating it?</p>	<p>[Entry Fields]</p> <p>[9.3.160.120]</p> <p>[255.255.255.0]</p> <p>[vi0]</p> <p>yes</p>	<p>+</p>
--	---	----------

F1=Help	F2=Refresh	F3=Cancel	F4=List
F5=Reset	F6=Command	F7=Edit	F8=Image
F9=Shell	F10=Exit	Enter=Do	

Figure 8-7 Add a Virtual IP Address Interface SMIT menu

The **lsdev** command will list the virtual network interface and the traditional network interfaces as members of the interface class **if**:

```
# lsdev -HCc if -F 'name class subclass type status description'
name class subclass type status description
```

en0	if	EN	en	Available	Standard Ethernet Network Interface
en1	if	EN	en	Defined	Standard Ethernet Network Interface
et0	if	EN	ie3	Defined	IEEE 802.3 Ethernet Network Interface
et1	if	EN	ie3	Defined	IEEE 802.3 Ethernet Network Interface
lo0	if	LO	lo	Available	Loopback Network Interface
tr0	if	TR	tr	Available	Token Ring Network Interface
vi0	if	VI	vi	Available	Virtual IP Address Network Interface

Also, the **netstat** command reports the existence of the newly defined interface:

```
# netstat -in
```

Name	Mtu	Network	Address	Ipkts	Ierrs	Opkts	Oerrs	Coll
lo0	16896	link#1		191957	0	191961	0	0
lo0	16896	127	127.0.0.1	191957	0	191961	0	0
lo0	16896	::1		191957	0	191961	0	0
en0	1500	link#2	0.6.29.c5.1d.68	28048	0	2580	0	0
en0	1500	10.47	10.47.1.2	28048	0	2580	0	0
tr0	1492	link#3	0.6.29.be.d2.a2	155075	0	42520	0	0
tr0	1492	9.3.240	9.3.240.58	155075	0	42520	0	0
vi0	0	link#4		0	0	0	0	0
vi0	0	9.3.160	9.3.160.120	0	0	0	0	0

System administrators can use the **lsattr** command to examine the device attributes for virtual network interfaces, and the **ifconfig** command is enabled to handle the new network interface type:

```
# lsattr -El vi0
```

netaddr	9.3.160.120	N/A	True
state	up	Standard Ethernet Network Interface	True
netmask	255.255.255.0	Maximum IP Packet Size for This Device	True
netaddr6		Maximum IP Packet Size for REMOTE Networks	True
alias6		Internet Address	True
prefixlen		Current Interface Status	True
alias4		TRAILER Link-Level Encapsulation	True

```
# ifconfig vi0
vi0: flags=84000041<UP,RUNNING,64BIT>
    inet 9.3.160.120 netmask 0xffffffff00
```

As indicated by the example, virtual network interfaces are similar to traditional network interfaces in most ways. A virtual interface is apparently configured and customized using the same system management commands as for real network interfaces. A system administrator has the option to define multiple virtual interfaces and can choose to associate aliases with them.

One of the main advantages of choosing a virtual device, as opposed to defining aliases to real network interfaces, is that a virtual device can be brought up or down separately without having any effect on the real interfaces of a system.

Furthermore, it is not possible to change the address of an alias (aliases can only be added and deleted), but the address of a virtual interface can be changed.

For applications and processes, the difference between a real and a virtual IP address is completely transparent, and therefore they can bind to a virtual interface just like to any other network interface.

However, a virtual address takes precedence over other interface addresses in a source address selection if an application locally binds to a wildcard address. (Telnet would be an example for an application having this binding characteristic.) This enables applications to make use of VIPA without any changes. In situations where there are multiple virtual addresses, the address of the first virtual interface on the list of interfaces will be chosen.

Since a virtual interface does not have a device associated with it, no route pointing to this interface will be added at configuration time. It is not possible to add routes on your local system that point to a virtual interface.

The gated process, which provides the gateway routing function in AIX, does not add a route for any virtual interface; also, gated will not send advertisements over the virtual interface, like it does for the other interfaces. However, gated does include the virtual interface in its advertisement to its neighboring routers, which enable these routers to add a host route for the virtual address.

Because the virtual interface does not relate to any real network interface, packets will never go in or out of the interface, and, consequently, the packet count for the virtual interface will always be zero. For the same reason, the virtual network interface will not respond to ARP requests.

Considering all the information given in the paragraphs above, you can complete the description of the data and control flow for network traffic through a virtual interface.

When an application locally bound to a wildcard address connects to a remote host, a VIPA is selected as its source address. The interface the outgoing packet actually uses is determined by the route table based solely on the destination address. The remote host receives the packet and then tries to send a response to the host using the virtual address. The remote host and all routers along the way must have a route that will send the packet with the virtual address to one of the network interfaces of the host with the virtual address.

Either gated running on the host with VIPA will send information, which enables the adjacent routers and the remote host to add a host route for the virtual address, or the intermediate routes have to be configured manually along the route.

8.6.1 Virtual IP address enhancement (5.2.0)

The virtual IP address (VIPA) feature in Version 5.2 has been enhanced to give the administrator greater control to select the source address for outgoing packets that have the source address unset.

The behavior of the source address selection rules depends on whether the outgoing packets have the source address set. The source address could be unset if a server process binds to the ANY IP address, also called the wildcard address. Outgoing packets from telnet or FTP clients, for example, will not specify a source address. If the outgoing packets' source address is unset, the network stack will use these rules to assign one.

The source address selection rules for AIX without VIPAs configured are as follows. If the source address of the outgoing packet is unset, the source address is set to the IP address of the interface the packet is being sent on. If the source address is set, then the address is left as is.

In Version 5.1, the source address selection rules with VIPAs configured are as follows. If the source address of the outgoing packet is unset, the source address is set to the IP address of the first virtual IP address configured. If the source address is set, then the address is left as is.

In Version 5.2, you are now able to assign physical network adapters to a specific VIPA. Each physical network adapter can only be assigned to one VIPA.

The source address selection rules with VIPAs configured and the source address of the outgoing packet is unset is as follows:

- ▶ If the physical interface the packet is being sent on is assigned to a VIPA, the source address will be set to that VIPA.
- ▶ If the physical interface the packet is being sent on is not assigned to a VIPA, the source address will be set to the IP address of the interface the packet is being sent on.

To emulate Version 5.1's source address selection rules on Version 5.2, you just need to add a VIPA and assign all the physical interfaces to that VIPA.

If all the physical interfaces are assigned to specific VIPAs, you can still create more VIPAs but you can't assign them to any physical interfaces. Your application server must bind specific to the new VIPA, otherwise the source address will be different than the VIPA.

The following example creates a VIPA named vi0 with an IP address of 192.168.3.100, netmask of 255.255.255.0, and assigned physical interfaces en0 and en2.

To add this VIPA using the **mkdev** command, you must run the following.

```
# mkdev -c if -s VI -t vi -a netaddr='192.168.3.100' -a netmask='255.255.255.0'
\      -w 'vi0' -a state='up' -a interface_names='en0,en2'
```

To use the SMIT interface, use the SMIT fast path **mkinetvi**. See Figure 8-8 for this same example using SMIT.

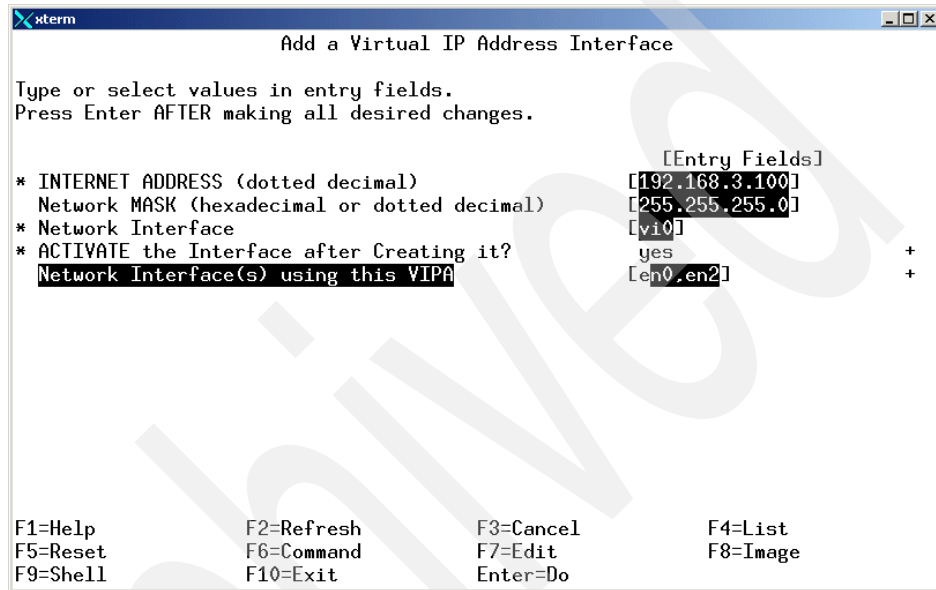


Figure 8-8 SMIT Add a Virtual IP Address Interface panel

After the VIPA is created there are several ways to visualize its configuration. The following examples show the output of the **netstat**, **ifconfig**, and **lsattr** commands with VIPA.

```
# netstat -in -I vi0
Name Mtu Network Address Ipkts Ierrs Opkts Oerrs Coll
vi0 0 link#5
vi0 0 192.168.3 192.168.3.100 0 0 0 0 0
# ifconfig vi0
vi0: flags=84000041<UP,RUNNING,64BIT>
inet 192.168.3.100 netmask 0xffffffff
iflist : en0 en2
# lsattr -E -l vi0
netaddr 192.168.3.100 N/A True
state up Standard Ethernet Network Interface True
netmask 255.255.255.0 Maximum IP Packet Size for This Device True
netaddr6 Maximum IP Packet Size for REMOTE Networks True
alias6 Internet Address True
```

prefixlen	Current Interface Status	True
alias4	TRAILER Link-Level Encapsulation	True
interface_names en0,en2	N/A	True

You can use the `vipa_iflist` and `-vipa_iflist` flags on the **ifconfig** command to temporarily add and remove interfaces assigned to the VIPA. The changes made with the **ifconfig** command will not be saved when the machine is rebooted. The following examples show how to use the **ifconfig** command to unassign an interface and then reassign the `en0` interface.

```
# ifconfig vi0 -vipa_iflist en0
# ifconfig vi0
vi0: flags=84000041<UP,RUNNING,64BIT>
    inet 192.168.3.100 netmask 0xffffffff00
    iflist : en2
# ifconfig vi0 vipa_iflist en0
# ifconfig vi0
vi0: flags=84000041<UP,RUNNING,64BIT>
    inet 192.168.3.100 netmask 0xffffffff00
    iflist : en0 en2
```

To make persistent changes to the VIPA interface you can use either the **chdev** command or go through SMIT `chinet`. The following example shows how to remove the `en0` interface from the `vi0` VIPA. To make this change with the **chdev** command run the following command.

```
# chdev -l vi0 -a interface_names='-,en0'
```

See Figure 8-9 on page 502 for this same example using the SMIT interface.

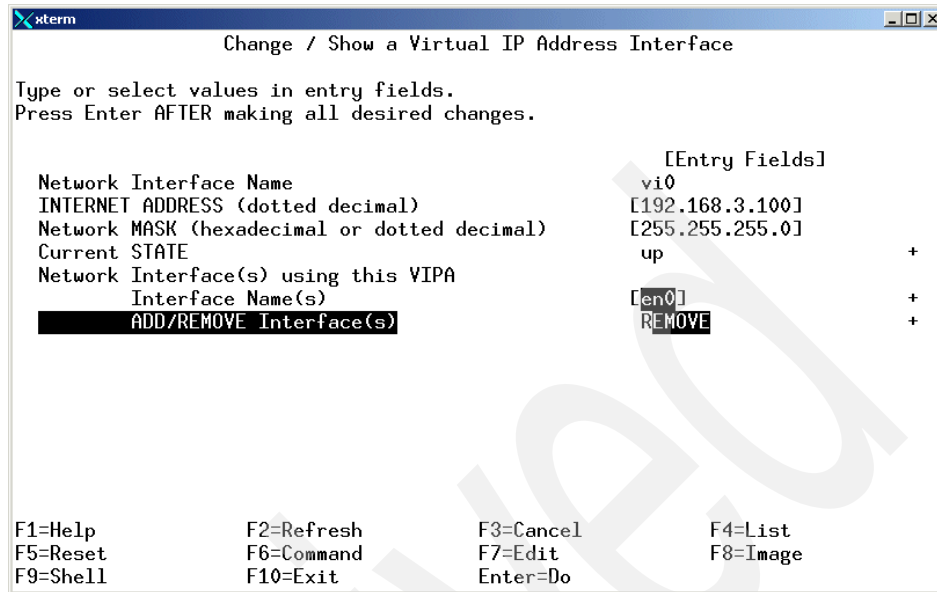


Figure 8-9 SMIT Change/Show a Virtual IP address Interface panel

8.7 Mobile IPv6 (5.2.0)

Mobile IPv6 allows systems to keep the same Internet address all over the world, and allows applications using that address to maintain transport and upper-layer connections when you change locations. It allows mobility across homogenous and heterogeneous networks.

To understand mobile IPv6, the understanding of the following concepts is required.

- Mobile node** A node that can change its point of attachment from one link to another, and still be reachable using its home address.
- Correspondent node** A peer node with which a mobile node is communicating.
- Home agent node** A router on a mobile node's home link with which the mobile node has registered its current care-of address. While the mobile node is away from home, the home agent intercepts packets on the home link destined to the mobile node's home.

Each mobile node has a home address and a care-of address. The care-of address, which is an IPv6 address, can be assigned by any method including

autoconfiguration, manual configuration, or DHCPv6. The home address is a permanent IP address that identifies the mobile node regardless of its location. When a mobile node arrives to a visited network, it must acquire a care-of address, which will be used during the time that the mobile node is under this location in the visited network. The care-of address changes at each new point of attachment and provides information about the mobile node's current situation. There must be at least one home agent configured on the home network, and the mobile node must be configured to know the IP address of its home agent. The mobile node sends a packet containing a binding update destination option to the home agent. The home agent gets the packet and makes an association between the home address to the mobile node and the care-of address it received.

Figure 8-10 shows the different interactions that take place in mobile IPv6.

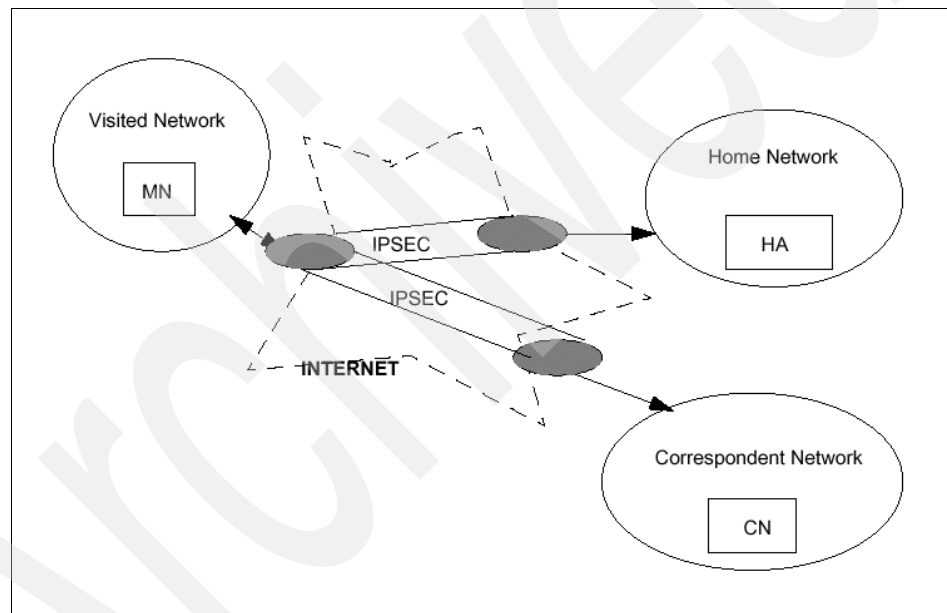


Figure 8-10 The different mobile IPv6 nodes

The mobile node (MN) in Figure 8-10 is in a visited local area network. The home agent (HA) which is in the home from where LAN handles the location information of the MN while it is away from home and redirects packets to the mobile node. The correspondent node (CN) is a node the MN communicates with.

In AIX, the nodes can be configured as home agent or correspondent node. To perform this configuration, a new SMIT panel has been added (Figure 8-11).

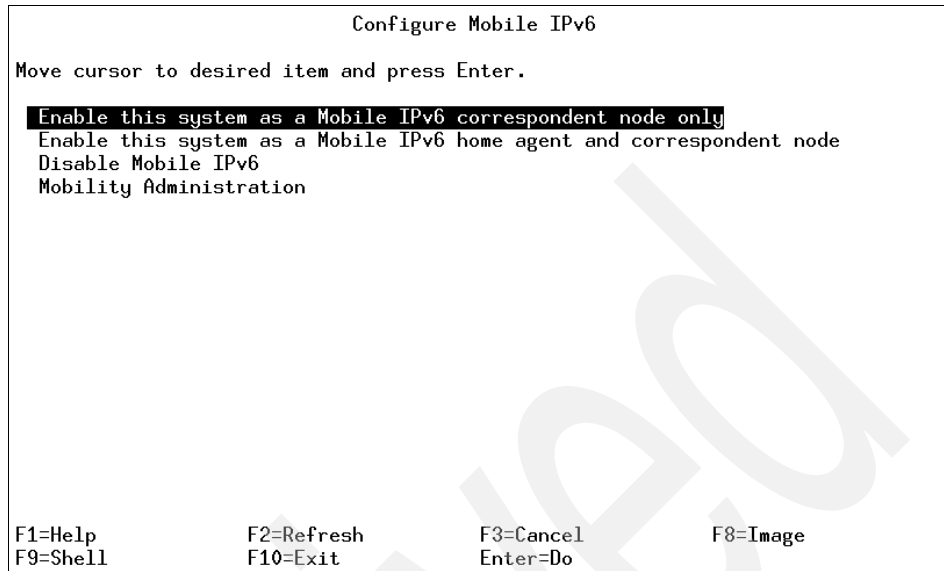


Figure 8-11 SMIT Configure Mobile IPv6 panel

The options in Figure 8-11 allow you to enable the system as correspondent node only or home agent and correspondent node and add a new line in the `/etc/inittab` file. For a home agent and correspondent node the following line is added:

```
rcmobip6:23456789:wait:/etc/rc.mobip6 start -H > /dev/console 2>&1 # Mobile IPv6
```

The `mobip6ctr1` command can also be used to configure and manage the mobile IPv6 home agent and correspondent node. It is possible, for example, to add or delete *home address* or *care-of address* in a home agent node.

8.8 DHCP enhancements (5.2.0)

In AIX 5L Version 5.2, the dynamic host configuration protocol (DHCP) server was enhanced to support the following RFCs:

- ▶ RFC2241 - DHCP Options for Novell Directory Services
- ▶ RFC2610 - DHCP Options for Service Location Protocol
- ▶ RFC2937 - The Name Service Search Option for DHCP
- ▶ RFC3011 - The IPv4 Subnet Selection Option for DHCP

For more information, these Requests for Comments (RFC) can be found on the Internet Engineering Task Force (IETF) Web site at the following URL:

<http://www.ietf.org/rfc.html>

Prior to AIX 5L Version 5.2, the DHCP server could be made to support RFC2241, RFC2610, and RFC2937 options, but it is difficult to set up and administer. The option data has to be entered as hexadecimal numbers and it needs to be prefixed with 0x and the data's length. Examples of the old style configuration are included alongside the new configuration stanzas in the following examples. For backwards compatibility, the DHCP server still supports the old style configuration.

RFC2241 introduces three new DHCP options to configure clients to use Novell Directory Services (NDS). Option number 85 specifies one or more IP addresses for the location of the NDS servers. Option number 86 specifies the name of the NDS tree the client should contact. Option 87 specifies the initial NDS context the clients should use. The following example configures a client to contact the NDS server at the address 192.168.1.5, and if that fails, it will try to connect to 192.168.2.5. After connecting to the NDS server, the client will use the NDS tree mycompany_inc using the initial context mydept.mycompany.

```
# RFC2241 - DHCP Options for Novell Directory Service
# option 85      IPaddress1 IPaddress2 IPaddress3 IPaddress4
# option 86      NDS tree name
# option 87      Initial NDS Context
option 85        192.168.1.5 192.168.2.5
option 86        mycompany_inc
option 87        mydept.mycompany

# Old hexadecimal style
# option 85      0x08C0A80105C0A80205
# option 86      0x0D6D79636F6D70616E795F696E63
# option 87      0x106D79646570742E6D79636F6D70616E79
```

RFC2610 introduces two new DHCP options to configure clients to use Service Location Protocol (SLP). Option 78 specifies a mandatory byte and one or more IP addresses for the location of the SLP servers. Option 79 specifies a mandatory byte and the default scope. The following example configures a client to contact the SLP server at the address 192.168.1.10, and if that fails, it will try to connect to 192.168.2.10. After connecting to the SLP server, the client will use a scope of mycompany_scope.

```
# RFC2610 - DHCP Options for Service Location Protocol
# option 78      Mandatory Byte IPaddress1 IPaddress2 IPaddress3 IPaddress4
# option 79      Mandatory Byte Default Scope
option 78        0 192.168.1.10 192.168.2.10
option 79        0 mycompany_scope
```

```
# Old hexadecimal style
# option 78      0x090008C0010A08C0020A
# option 79      0x0F6D79636F6D70616E795F73636F7065
```

RFC2937 introduces a new DHCP option to configure that the order name services are consulted when the client attempts to resolve an address or host name. The parameters for option 117 are a list of name services in order. RFC2937 specifies the following possible name services. DHCP clients might not support all of these options. The AIX DHCP client does not support option 117.

Name Service	Value
Local Name Resolution	0
Domain Name Server Option	6
Network Information Servers Option	41
NetBIOS over TCP/IP Name Server Option	44
Network Information Service+ Servers Option	65

The following example shows how to configure a client to use DNS for name resolution first. If DNS is not available, network information services (NIS) will be consulted.

```
# RFC2937 - The Name Service Search Option for DHCP
# option 117   Name Service1 Name Service2 ... NameService N
option 117     6 41
```

RFC3011 introduced a new DHCP option numbered 118, which allows a DHCP client to request an address from a specific subnet. This option would override the DHCP server's default method for selecting the subnet to allocate an address on. Normally the DHCP server will determine the subnet of the original DHCP request and allocate an address on that same subnet. In some applications, such as remote access servers (RAS), the clients would not have direct access to the DHCP server. The RAS device would then make DHCP requests on behalf of its clients using the client subnet specified in option 118. The DHCP server would allocate an address on the client subnet and reply to the RAS device with the client's address. Without option 118, the DHCP server would allocate an address on the same subnet as the RAS device.

This option is enabled in the DHCP server configuration file using the `supportoption118` option in the global container. The `supportoption118` option accepts one parameter to determine the scope of option 118 support. If `supportoption118` is set to `global`, then all subnet containers will support option 118. If `supportoption118` is set to `subnetlevel`, then you must specifically enable option 118 in each subnet container.

The following example specifies that the option `supportsubnetselection` in the global container is set to `subnetlevel`. The subnet container `192.168.1.0` does not support option 118, as the `supportoption118` is set to `no`. The subnet container `192.168.2.0` does support option 118, as `supportoption 118` is set to `yes`.


```
# supportsubnetselection {global | subnetlevel | no }
supportsubnetselection subnetlevel

subnet 192.168.1.0 255.255.255.0 192.168.1.50-192.168.1.254 {
    supportoption118 no
    ...
}
subnet 192.168.2.0 255.255.255.0 192.168.2.50-192.168.2.254 {
    supportoption118 yes
    ...
}
```

If you need more information about the NDS and SLP options, refer to your Novell documentation.

The following file is a sample DHCP server configuration file for use with the samples within this publication.

```
numLogFiles          4
logFileSize          100
logFileName           /usr/tmp/dhcpd.log

logItem              SYSERR
logItem              OBJERR
logItem              PROTERR
logItem              WARNING
logItem              EVENT
logItem              ACTION
logItem              INFO
logItem              ACNTING
logItem              TRACE

leaseTimeDefault      30 minutes
leaseExpireInterval   3 minutes
supportBOOTP          yes
supportUnlistedClients yes

ignoreInterface       9.3.4.97

# supportsubnetselection {global | subnetlevel | no }
supportsubnetselection subnetlevel

option 6              192.168.1.20 192.168.2.20      # DNS name servers
option 15             mydept.mycompany.example      # DNS domain name

# RFC2937 - The Name Service Search Option for DHCP
# option 117          Name Service1 Name Service2 ... NameService N
option 117            6 41
```

```

# RFC2241 - DHCP Options for Novell Directory Service
# option 85      IPaddress1 IPaddress2 IPaddress3 IPaddress4
# option 86      NDS tree name
# option 87      Initial NDS Context
option 85        192.168.1.5 192.168.2.5
option 86        mycompany_inc
option 87        mydept.mycompany

# Old hexadecimal style
# option 85      0x08C0A80105C0A80205
# option 86      0x0D6D79636F6D70616E795F696E63
# option 87      0x106D79646570742E6D79636F6D70616E79

# RFC2610 - DHCP Options for Service Location Protocol
# option 78      Mandatory Byte IPaddress1 IPaddress2 IPaddress3 IPaddress4
# option 79      Mandatory Byte Default Scope
option 78        0 192.168.1.10 192.168.2.10
option 79        0 mycompany_scope

# Old hexadecimal style
# option 78      0x090008C0010A08C0020A
# option 79      0x0F6D79636F6D70616E795F73636F7065

subnet 192.168.1.0 255.255.255.0 192.168.1.50-192.168.1.254 {
    supportoption118 no

    option 1      255.255.255.0          # subnet mask
    option 3      192.168.1.1           # default gateway
    option 28     192.168.1.255         # broadcast address

    option 79     0 mydept_scope

    # Old hexadecimal style
    # option 79    0x0C6D79646570745F73636F7065
}

subnet 192.168.2.0 255.255.255.0 192.168.2.50-192.168.2.254 {
    supportoption118 yes

    option 1      255.255.255.0          # subnet mask
    option 3      192.168.2.1           # default gateway
    option 28     192.168.2.255         # broadcast address
}

```

8.9 FTP server enhancements (5.2.0)

The `ftpd` server has been enhanced to allow the administrator to display messages before and after **ftp** login, restrict **ftp** login to specific hosts, restrict what directories users can read or write into, and support login of anonymous restricted users. To use the new `ftpd` enhancements, you must create the `ftpd` configuration file `/etc/ftpaccess.ctl`. If this file does not exist, the `ftpd` server will behave as normal.

The configuration keywords can be broken up into four different groups: Notification, host restriction, directory restriction, and restricted users. The complete list of the supported keywords and their expected parameters are listed in the following. Lines starting with unsupported keywords are silently ignored.

```
# Notification
herald: filename
motd: on|off

# Host restrictions
allow: hostname, hostname, ...
deny: hostname, hostname, ...

# Directory restrictions
readonly:dirname, dirname, ... | ALL | NONE
writeonly: dirname, dirname, ... | ALL | NONE
readwrite: dirname, dirname, ... | ALL | NONE

# Restricted users
useronly: username, username, ...
grouponly: groupname, groupname, ...
```

The notification group *keywords* allows the administrator to configure the `ftpd` server to send messages to the FTP client before and after the user logs in. The `herald` keyword configures `ftpd` to send a message to the client before logging in. The `herald` keyword requires one parameter, the name of the file containing the message to send. The `motd` keyword configures the `ftpd` to send the contents of the message of the day file (`motd`) after the FTP user logs in. The `motd` keyword requires one parameter either `on` or `off`. The `motd` file must be located in the home directory of the user. The following example shows how to use the `herald` and `motd` keywords and the contents of the `ftpaccess.ctl`, `ftpheard.txt`, and `root's` `motd` file.

```
# cat /etc/ftpaccess.ctl
herald: /etc/ftpheard.txt
motd: on

# cat /etc/ftpheard.txt
```

Welcome to our FTP Server

```
#cat /motd
```

This is roots's MOTD

```
# ftp ftp.mycompany.example
Connected to ftp.mycompany.example.
220-
220-Welcome to our FTP Server
220-
220 ftp.mycompany.example FTP server (Version 4.1 Mon Aug 19 21:52:59 CDT 2002)
ready.
Name (ftp.mycompany.example:root):
331 Password required for root.
Password:
230-Last unsuccessful login: Sun Sep  8 01:32:26 CDT 2002 on /dev/pts/4 from
sig-9-65-80-212.mts.ibm.com
230-Last login: Sun Sep  8 23:35:25 CDT 2002 on /dev/pts/4 from
here.mycompany.example
230-
230- This is roots's MOTD
230-
230 User root logged in.
ftp> quit
```

The host restriction keywords, `allow` and `deny`, allow the administrator to restrict the hosts that are allowed to connect to the FTP server. Use the `allow` keyword to deny all hosts from connecting except the ones specifically allowed. Use the `deny` keyword to allow all hosts to connect except the ones specifically denied. The `allow` and `deny` keywords are mutually exclusive and should not be used at the same time. The following examples show how the `allow` and `deny` keywords are used.

```
# Comments - Allow these specific hostname and addresses
# All other hosts are denied.
allow: myhost1.mycompany.example, myhost3.mycompany.example
allow: myhost4.mycompany.example, myhost9.mycompany.example
allow: 192.168.1.50

# Comments - Allow all hosts to connect except for the bad ones
deny: badhost1.othercompany.example, badhost2.othercompany.example
```

The following example shows an FTP client being denied by the FTP server. Hosts can be denied explicitly with the `deny` keyword or implicitly with the `allow` keyword.

```
# ftp ftp.mycompany.example
Connected to ftp.mycompany.example.
```

```
521 Connection refused by server
ftp>
```

The directory restriction keywords, `readonly`, `writable`, and `readwrite`, allow the administrator to restrict what directories FTP users are allowed to read and write in. The read and write operations map to FTP's `get/mget` and `put/mput` commands. These keywords are enforced for regular FTP users. If an anonymous user has a `ftpaccess.ctl` file in the `/etc` directory (accessed with `chroot`), then the `motd`, `readonly`, `writable`, and `readwrite` keywords are enforced.

The `readonly` keyword prevents FTP users from writing (**put**) into the specified directories. The `writable` keyword prevents FTP users from reading (**get**) files from the specified directories. The `readwrite` keyword allows FTP users to *only* read and write to the specified directories. The `writable` keyword does not prevent the directory from being displayed with `ls`. To restrict the directory listing remove the read permission attribute from the directory.

The behavior of the `readonly` and `writable` keywords depends upon whether the `readwrite` keyword is used. If the `readwrite` keyword is not specified, FTP users will have unrestricted access to all directories not specifically mentioned by the `readonly` and `writable` keywords. If the `readwrite` keyword is used, FTP users will only have read access to the directories specified by the `readwrite` and `readonly` keywords and write access to the directories specified by the `readwrite` and `writable` keywords. Additionally, all other directories will not allow read or write.

The following examples show how to configure the `ftpaccess.ctl` file.

```
# setup a dropoff directory where users can write (put) files
# but are unable to read (get) them.
writable: /home/dist/incoming
```

```
# setup a software distribution directory where users can
# only read(get) from /home/dist/pub only. No other access is
# permitted
readonly: /home/dist/pub
readwrite: NONE
```

```
# setup a software distribution directory where users can
# only read(get) from /home/dist/pub only. No other access is
# permitted
readonly: /home/dist/pub
readwrite: NONE
```

The restricted user keywords, `useronly` and `grouponly`, allow the administrator to configure anonymous restricted users that are restricted to their home directories. The `useronly` keyword specifies the list of users to be restricted to

their home directory. The `grouponly` keyword specifies the list of groups of users that should be restricted to their home directory.

When a restricted user logs in, the FTP server uses **chroot** to restrict the user to his home directory. The restricted user directories must be set up similar to the traditional anonymous FTP user. For more information on the required permissions and directory structure refer to the AIX 5L Version 5.2 ftpd documentation. The sample script `/usr/samples/tcpip/anon.users.ftp` makes the account and directory creation process easier. See the following example on how to create the restricted user `ftp3`.

```
# /usr/samples/tcpip/anon.users.ftp ftp3
Creating ftp3 as an anonymous ftp user.
Added user ftp3.
Are you sure you want to modify /home/ftp3?
y
Made /home/ftp3/bin directory.
Made /home/ftp3/etc directory.
Made /home/ftp3/pub directory.
Made /home/ftp3/lib directory.
Made /home/ftp3/dev/null entry.
Made /home/ftp3/usr/lpp/msg/en_US directory.
```

Note: When enhanced ftpd functions are enabled, the server checks the existence of the reverse IP address of the FTP client. If the IP address does not exist the client will receive a 521-connection refused by the server message.

8.10 Network buffer cache dynamic data support

The network buffer cache (NBC) was introduced in AIX Version 4.3.2. to improve the performance of network file servers, such as the Web server, FTP server, and SMB server. In AIX Version 4.3.3, the NBC design was improved to allow the use of 256-MB private memory segments for caching additional data. This design was chosen to eliminate the need to use pinned kernel heap and the network memory pools that had size restrictions. The use of private segments allows a system limit, set by the **no** option `nbc_pseg`, of 2^{20} segments. A setting should not exceed 2^{19} , because file systems, processes, and other applications also require segments. Therefore, the total amount of data can be 256×2^{19} or the limit set by the `nbc_pseg_limit` option. Only as much physical memory is consumed as data exists in a segment.

With the same AIX release, a second key for the cache access mechanism was introduced to support the HTTP GET kernel extension in conjunction with the Fast Response Cache Architecture (FRCA).

AIX 5L further enhances the network buffer cache kernel extension to facilitate a dynamic data buffer cache and to support an expiration time per cache object. Also, internal memory usage code optimizations were applied to expand the caching capacity of NBC.

Within the scope of the kernel address space, NBC uses network memory for caching data, which is accessed frequently through networks. For example, by enabling and using the NBC, the IBM HTTP Server can cache frequently referenced Web pages to eliminate the repetitive costs of moving data among the file buffers, user buffers, and networking buffers. NBC, as a kernel component, provides kernel services for its users to take advantage of the network buffer cache. In the NBC context, the term *users* refers to other kernel components or kernel extensions. Application-level users have to go through APIs provided by those kernel components or kernel extensions to interact with the NBC.

There are two ways for an application to exploit the NBC feature:

- ▶ Using the `send_file()` system call
- ▶ Using the Fast Response Cache Architecture (FRCA) API

The new AIX 5L NBC enhancements are only accessible for applications through the FRCA API.

8.10.1 Dynamic data buffer cache

In previous AIX releases, there was only one type of cache object that is cached in the NBC. Each cache object held copies of original data already existing in the file subsystem and, therefore, the related cache object type was named `NBC_NAMED_FILE`. Since the NBC was designed to improve the performance of typical network file servers, this single cache type was sufficient to improve the performance of Web servers in static Web page access scenarios. However, more and more Web pages consist of dynamically generated data and contents. These Web pages are not necessarily saved in files, and they are much more volatile than static file pages. For these reasons, NBC's capability was expanded to accommodate dynamically generated data (for example, dynamic pages or page fragments) generated by user-level applications.

Beginning with AIX 5L, NBC offers support for caching data buffers created and given by kernel users. The most prominent kernel user that depends on NBC is the FRCA kernel extension. FRCA utilizes the NBC and provides a platform-independent API for Web servers to add and delete dynamic data buffer caches on AIX systems. FRCA also accesses the NBC cache whenever an HTTP GET request can be satisfied by the cache in the system interrupt context.

The new NBC features provide adequate kernel services for FRCA to improve the overall IBM HTTP Web Server performance.

To the NBC, the dynamic data buffer cache is a group of buffers that were allocated and given by other kernel extensions or kernel components. These buffers are in the mbuf chain format for keeping and accessing from the NBC. The buffers are pinned in memory, and the cache object creators have the responsibility of keeping this memory pinned for the lifetime of the cache. These buffers can be allocated from regular mbuf pool (`m_get()`, `net_malloc()`, etc); from kernel heap (`xmalloc()`); or from private segments. When the buffers are given to the NBC for caching, it is the responsibility of the kernel extension or kernel component using NBC to build up an mbuf chain and set up the mbuf headers correctly for the corresponding buffers. The private segments do not have to be mapped by users at the time of adding, but they have to be pinned all the time.

The buffer cache is subject to the previously existing NBC flushing control. All caches are on the least recently used (LRU) list in the NBC. When the total cache size reaches the NBC system limits (multiple configured network options), any buffer cache may get removed from the NBC just like other caches.

A new cache type, `NBC_FRCA_BUF`, will be the cache type for the dynamic buffer cache associated with the FRCA. A primary key for type `NBC_FRCA_BUF` is generated and controlled by FRCA to uniquely identify each piece of cache within the `NBC_FRCA_BUF` type in the NBC.

Three new statistics were added for keeping track of the cache objects of the new cache type in the NBC:

- ▶ **Current total `NBC_FRCA_BUF` entries:** Number of cache entries with `NBC_FRCA_BUF` type that currently exist in the cache
- ▶ **Maximum total `NBC_FRCA_BUF` entries:** Highest number of cache entries with `NBC_FRCA_BUF` type that have ever been created in cache
- ▶ **Current total user buffer size:** Byte count of the total buffer size currently in the NBC that is not accounted in either the mbuf pool memory or the private segments

Use the `netstat -c` command to display the NBC statistics that are related to the new cache type, as in the following example:

```
# netstat -c
```


Network Buffer Cache Statistics:

```
-----  
Current total cache buffer size: 256  
Maximum total cache buffer size: 256  
Current total cache data size: 0  
Maximum total cache data size: 0  
Current number of cache: 1  
Maximum number of cache: 1  
Number of cache with data: 1  
Number of searches in cache: 1  
Number of cache hit: 0  
Number of cache miss: 1  
Number of cache newly added: 1  
Number of cache updated: 0  
Number of cache removed: 0  
Number of successful cache accesses: 0  
Number of unsuccessful cache accesses: 0  
Number of cache validation: 0  
Current total cache data size in private segments: 0  
Maximum total cache data size in private segments: 0  
Current total number of private segments: 0  
Maximum total number of private segments: 0  
Current number of free private segments: 0  
Current total NBC_NAMED_FILE entries: 0  
Maximum total NBC_NAMED_FILE entries: 0  
Current total NBC_FRCA_BUF entries: 1  
Maximum total NBC_FRCA_BUF entries: 1  
Current total user buffer size: 131072
```

8.10.2 Cache object-specific expiration time

In previous AIX releases, the NBC provides cache invalidation based on a time limit specified by the cache access client, not the creator. In other words, once the cache is loaded, it is assumed to be good; the frequency of invalidation checking or updating is up to the client's tolerance. This is acceptable with a cache object that is expected to be reasonably static. For dynamic data, however, it is necessary to support an expiration time per cache object.

In AIX 5L, the NBC will invalidate the buffer cache according to a time-to-live value specified by the creator. Each buffer cache object has a live-time limit specified when it is first added to the NBC. When the cache is accessed, and if the age of the cache object exceeds the live-time limit, the NBC will remove this particular piece of cache and return NULL to the client. The client can also specify a time to make sure that the cache object is not older than expected. If the cache is older than the client's time limit, the NBC will return a NULL; the cache object, however, is still considered valid. The resolution for both time limit values is in units of seconds.

8.11 Direct I/O and callbacks for NBC (5.2.0)

The network buffer cache is used to cache files in the kernel space to avoid costly user-to-kernel space copying. The network buffer cache can be used by applications such as ftp/ftpd and FRCA (an in-kernel Web serving technology). Until this enhancement, the code was not aware until after a specific duration of time whether the files were changed, removed, or the file system was unmounted. When caching files that change rapidly this design was not practical. Therefore, new kernel services are provided where the application can register the files of any file system it caches and request notification on changes to the files. A kernel service is also provided to request notification if a JFS file system gets unmounted.

Furthermore, a kernel service to provide direct I/O to NBC for JFS file systems is provided so that NBC can read files directly from disk without going through the file system layers.

8.11.1 Callback for NBC

The notification is done by the callback routine `nbc_locate()`. The new parameter type `NBC_DELE_CACHE` is used in the case of removing, renaming, copying, or editing a file (for simplicity referred to as *file change* in the following). The new parameter type `NBC_UMOUNT_FS` is used in the case of a JFS file system being unmounted; the JFS device is passed as `parm1`.

The following pseudo code shows how the callback function could be extended to be made aware of these new parameters:

```
callback_function(...)
{
    ...
    switch()
    {
        ...
        case NBC_DELE_CACHE:
        {
            int oval;
            vnode_t *vp = key;
            hp = &ofile_hash_table[NBC_OFILE_HASH(vp)];
            hpri = disable_lock(PL_IMP, &HASH_LOCK);
            /* lookup for the file in cache */
            NBC_LOOKUP_OFILE(vp, hp, fp);
            if (fp) { /* found */
                /* mark it OF_FLUSHING */
                oval = fp->state;
                compare_and_swap(&fp->state, &oval, OF_FLUSHING);
            }
        }
    }
}
```

```

        /* If not found, we do nothing, but it shouldn't happen */
        unlock_enable(hpri, &HASH_LOCK);
        vp->v_flag &= ~V_NBC;
    }

    case NBC_UMOUNT_FS:
    {
        dev_t dev = *parml;
        loop through every entry of the NBC cache {
            if( nbc_vnode_in_dev(vp, dev) ) {
                /* this vnode is in the device, flush the entry */
                flush this NBC cache entry
            }
        }
        ...
    }
    ...
}

```

The new `nbc_vnode_in_dev(vnode_t *vp, dev_t dev)` function in the pseudo code above is used to check whether the file pointed to with the vnode pointer `vp`, is in the JFS file system given by `dev`.

Before the NBC code can be notified by the kernel, the application has to register the file that it caches to the kernel. To register, two functions are provided and a description of the parameters is provided:

- ▶ `nbc_vno_flag(vnode_t *vp, int cmd)`
 - `vp`: The vnode pointer for this file we are trying to send;
 - `cmd`:
 - `CLR_NBC_FLAG`: 0 - Unset the `V_NBC` flag.
 - `SET_NBC_FLAG`: 1 - Set the `V_NBC` flag.
 - `CHK_NBC_FLAG`: 2 - Check the `V_NBC` flag; if it is set, return 1 (true); otherwise, return 0 (false).
- ▶ `nbc_vfs_flag(vnode_t *vp, int cmd)`
 - `vp`: The vnode pointer for this file we are trying to send.
 - `cmd`:
 - `CLR_NBC_FLAG`: 0 - unset the `CHK_NBC_FLAG` flag for the vfs pointed by this `vp`.
 - `SET_NBC_FLAG`: 1 - Set the `CHK_NBC_FLAG` flag for the vfs pointed by this `vp`.
 - `CHK_NBC_FLAG`: 2 - Check the `CHK_NBC_FLAG` flag for the vfs pointed by this `vp`; if it is set, return 1 (true), otherwise, 0.

The first function is to request notification in the case of a file change event and the second is to request notification in the case of a JFS file system being unmounted. The functions should get called right after the NBC cache entry was created.

8.11.2 Direct I/O for NBC

To use memory mapped (direct) I/O, a new kernel service is provided to map a JFS file pointer to a new memory segment:

```
nbc_vptosid(vnode_t *vp, vmid_t *vmid)
```

The key parameters are defined as follows:

vp The vnode pointer for this file we are trying to send

vmid The vmid constructed from srval for this virtual address

To use the above kernel service in NBC code, the code to read the file directly from memory could look like the following:

```
...
/* Call nbc_vptosid to map the file into a new memory segment */
nbc_vptosid((vnode_t *) vp, &vmid);
/* Be sure to attach the segment before we start reading
 * the file.
 */
vaddr = vm_att(SRVAL(vmid, 0, journ), 0)
read the while file from vaddr;
/* Be sure to detach the segment after we finish reading the file
 * or we might have a segments overflow problem later
 */
vm_det(vaddr);
...
```

8.12 HTTP GET kernel extension enhancements

Starting with AIX Version 4.3.2, the Fast Response Cache Architecture (FRCA) with the HTTP GET kernel extension was introduced to AIX.

AIX 5L improves the FRCA HTTP GET kernel extension to support HTTP 1.1 persistent connections. Other enhancements to the HTTP GET kernel extension include an external 64-bit ready API (to give every user space program access to the existing function of the HTTP GET kernel extension) and additional support for a new cache type based on memory buffers.

The FRCA utilizes the AIX network buffer cache (NBC) to greatly improve the Web server response time for HTTP GET requests. Figure 8-12 illustrates the FRCA data flow for an incoming request, which refers to a Web page located on a given Web server. The HTTP GET requests are intercepted and the response is sent directly from the AIX NBC on the input interrupt. No data is copied between kernel and user space, and no user context switch is necessary. If the HTTP GET request can be serviced by the engine, the user space Web server is not contacted and never sees the request. GET requests that cannot be serviced by the kernel engine are passed to the user space Web server.

The logic of FRCA is shown in Figure 8-12.

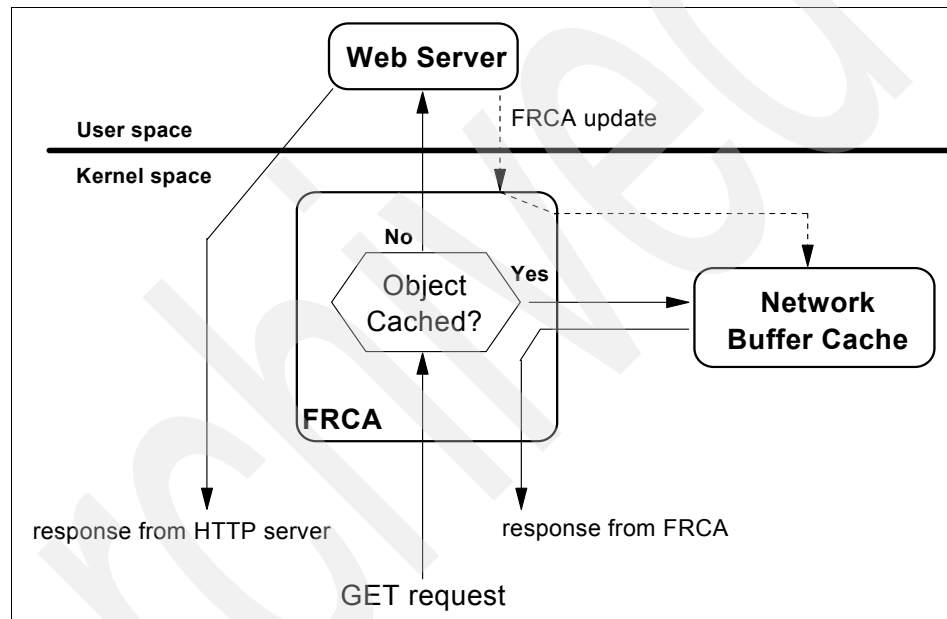


Figure 8-12 FRCA GET data flow

8.12.1 HTTP 1.1 persistent connections support

When AIX Version 4.3.2 was released, the predominant protocol in use was HTTP Version 1.0, with a major part of all requests referring to static content. Since then, a shift toward HTTP Version 1.1 has taken place. One of the major differences between the two versions of HTTP is the newer version's well-defined ability to handle multiple requests per connection while the previous version almost always closes a connection, after a single request. Keeping a connection established for several requests allows the underlying transport layer protocol (TCP) to make better use of the available bandwidth by adapting to it over time.

The implementation of the HTTP GET kernel extension prior to AIX 5L either transparently redirected the pending request to a user space Web server, or it closed the connection after serving a single request.

With HTTP 1.1, a well-defined way of imposing entity boundaries on the exchanged HTTP data has been introduced, which will rapidly result in widespread use of persistent connections. For that reason, AIX 5L adds support for HTTP 1.1 persistent connections to the FRCA feature.

The support for persistent connections was such that the HTTP GET kernel extension parses an incoming packet like before, but with only a little addition to the previously used code path. As the packet may contain multiple requests, it loops over the data and marks down the number of bytes from the input buffer that belong to the current request, the request's protocol version, and the absence of a connection header that includes the connection-token *close*.

On a per request basis, the kernel extension then acts according to the following rules:

- ▶ If the protocol version of the current request is not HTTP 1.1, then in case of a cache hit, it adds the response to the response buffer, sends the buffer, and closes the connection; in case of a cache miss, it sends the buffer and reconnects the connection to the user space Web server.
- ▶ If the protocol version of the current request is HTTP 1.1 and the close token has been detected, then in case of a cache hit, it adds the response to the response buffer, sends the buffer, and closes the connection; in case of a cache miss, it sends the buffer and reconnects the connection to the user space Web server.
- ▶ If the protocol version of the current request is HTTP 1.1 and the close token has not been detected, then in case of a cache hit, it adds the response to the response buffer, sends the buffer, and keeps the connection in kernel space; in case of a cache miss, it sends the buffer and reconnects the connection to the user space Web server.

8.12.2 External 64-bit FRCA API

Beginning with AIX 5L, an external 64-bit FRCA API is offered to allow more user space applications to exploit the existing function of the HTTP GET kernel extension.

The external API largely follows the structure of the internal API, which consists of a set of functions to create and control an FRCA instance and another set of functions to create and fill a cache for a given FRCA instance. It is implemented as a layer on top of the internal API, which results in no changes to the previously existing HTTP GET kernel extension itself. The API will cover only the major part

of the existing function of the HTTP GET kernel extension, but not all of it. Functions specific to the AIX platform, such as control over the amount of time that the HTTP GET kernel extension may spend on interrupt, will not be covered by the external API, and are left to the existing `frcactrl` program. The `frcactrl` command controls and configures the FRCA kernel extension.

As the internal API continues to exist unchanged, all currently existing code developed against the internal API continues to work without a single change required.

AIX 5L provides a 64-bit version of the external API library to accommodate 64-bit applications. The following services that compose the external API are defined in `/usr/include/net/frca.h`. They are made available to user space applications through the `libfrca.a` library:

FrcaCtrlCreate	Creates a FRCA control instance
FrcaCtrlDelete	Deletes a FRCA control instance
FrcaCtrlStart	Starts the interception of TCP data connections for a previously configured FRCA instance
FrcaCtrlStop	Stops the interception of TCP data connections for a FRCA instance
FrcaCtrlLog	Modifies the behavior of the logging subsystem
FrcaCacheCreate	Creates a cache instance within the scope of a FRCA instance
FrcaCacheDelete	Deletes a cache instance within the scope of a FRCA instance
FrcaCacheLoadFile	Loads a file into a cache associated with a FRCA instance
FrcaCacheUnloadFile	Removes a cache entry from a cache that is associated with a FRCA instance

8.12.3 Memory-based HTTP entities caching

AIX 5L adds new services to the internal FRCA API to support caching of HTTP entities that are based on memory buffers and have no association with a file. The underlying NBC data cache provides the related NBC cache object type `NBC_FRCA_BUF`. The `NBC_FRCA_BUF` type in NBC refers the new dynamic data buffer cache, which is introduced with AIX 5L in order to expand the NBC caching capabilities to allow for Web pages with dynamically generated data and contents. For further details about the new NBC cache object type, refer to 8.10, “Network buffer cache dynamic data support” on page 512.

The previous implementation of the HTTP GET kernel extension only handled cache objects with content data that is tightly coupled to files in the local file system. This works fine in the case of static HTML pages that are stored in the local file system, but it does not handle semi-dynamic content very well. The term *semi-dynamic* refers to content that is static to a certain degree (for example, a dynamically rendered HTML page that changes only once a minute, but has a reasonably higher access rate, such as once a second).

Although the semi-dynamic content could be written to a file, which in turn could be loaded into the HTTP kernel extension using the existing API, this involves some overhead, especially when the code that renders the content is executed on a different machine.

AIX 5L introduces a new service to the internal API to support caching of memory-based HTTP cache objects, which allows FRCA to handle caching of HTTP data that is not represented in the file system. One of the main purposes of the service is to accommodate application-level cache managers residing on remote systems.

8.13 Packet capture library

Previous AIX operating system releases and AIX 5L offer the Berkeley Packet Filter (BPF) as a packet capture system. AIX 5L introduces, in addition to that, a Packet Capture Library (libpcap.a), which provides a high-level user interface to the BPF packet capture facility. The AIX 5L Packet Capture Library is implemented as part of the libpcap library, Version 0.4 from Lawrence Berkeley National Laboratory (LBNL).

The Packet Capture Library user-level subroutines interface with the existing BPF kernel extensions to allow users access for reading unprocessed network traffic. By using the new 24 subroutines of this library, users can write their own network-monitoring tools.

To accomplish packet capture, follow this procedure:

1. Decide which network device will be the packet capture device. Use the `pcap_lookupdev` subroutine to do this.
2. Obtain a packet capture descriptor by using the `pcap_open_live` subroutine.
3. Choose a packet filter. The filter expression identifies which packets you are interested in capturing.
4. Compile the packet filter into a filter program using the `pcap_compile` subroutine. The packet filter expression is specified in an ASCII string.

5. After a BPF filter program is compiled, notify the packet capture device of the filter using the `pcap_setfilter` subroutine. If the packet capture data is to be saved to a file for processing later, open the previously saved packet capture data file, known as the savefile, using the `pcap_dump_open` subroutine.
6. Use the `pcap_dispatch` or `pcap_loop` subroutine to read in the captured packets and call the subroutine to process them. This processing subroutine can be the `pcap_dump` subroutine, if the packets are to be written to a savefile, or some other subroutine you provide.
7. Call the `pcap_close` subroutine to clean up the open files and deallocate the resources used by the packet capture descriptor.

The current implementation of the libpcap library applies to IP Version 4 and only the reading of packets is supported. Applications using the Packet Capture Library subroutines must be run as root user. The files generated by libpcap applications can be read by `tcpdump` and vice-versa. However, the `tcpdump` command in AIX 5L does not use the libpcap library.

The Packet Capture Library `libpcap.a` is located in the `/usr/lib` directory after you have optionally installed the `bos.net.tcp.server` fileset. The `bos.net.tcp.server` fileset also provides the BPF kernel extension (`/usr/lib/drivers/bpf`), which is used by the libpcap subroutines. The library-related header file `pcap.h` can be examined in the `/usr/include/` directory, if you choose to install the `bos.net.tcp.adt` fileset. The libpcap sample code, which is also part of the `bos.net.tcp.adt` fileset, can be found in `/usr/samples/tcpip/libpcap`.

Further information about BPF can be found in *UNIX Network Programming, Volume 1: Networking APIs: Sockets and XTI*, Second Edition, by W. Richard Stevens.

8.14 Firewall hooks enhancements

The AIX TCP/IP stack provides a way for other kernel extensions to insert themselves into the stack at specific points using hooks.

AIX 5L introduces two new firewall hooks that expand the functional spectrum of the already existing hooks for IP filtering and offers additional potential to improve the performance of firewalls. The new hooks will be part of the existing `netinet` kernel extension, which is packaged in `bos.net.tcp.client`.

The firewall hook routines provide kernel-level hooks for IP packet filtering, enabling IP packets to be selectively accepted, rejected, or modified during reception, transmission, and decapsulation. These hooks are initially `NULL`, but

are exported by the netinet kernel extension and will be invoked if assigned non-NULL values.

The following routines are included in AIX 5L as hooks for IP packet filtering:

- ▶ `ip_fltr_in_hook`
- ▶ `ip_fltr_out_hook`
- ▶ `ipsec_decap_hook`
- ▶ `inbound_fw` (new in AIX 5L)
- ▶ `outbound_fw` (new in AIX 5L)

The `ip_fltr_in_hook` routine is used to filter incoming IP packets, the `ip_fltr_out_hook` routine filters outgoing IP packets, and the `ipsec_decap_hook` routine filters incoming encapsulated IP packets.

The new AIX 5L `inbound_fw` and `outbound_fw` firewall hooks allow kernel extensions to get control of packets at the place where IP receives them. The `outbound_fw` hook was added exactly at the point where IP is entered when transmitting packets and the `inbound_fw` hook at the point where IP is called to process receive packets. The two new firewall hooks in AIX 5L are supplemented by additional methods to call the main IP code and to save firewall hook arguments in order to inject the filtered packets into the network at a later time. Also, some changes to existing routines were made alongside with the implementation of the new firewall hooks.

The code of following existing functions has been changed:

`ipintr_noqueue2` The `ipintr_noqueue2` hook itself and all references to `ipintr_noqueue2` are removed. The function of `ipintr_noqueue2` is provided by passing a null NDD parameter to `ipintr_noqueue`.

`ipintr_noqueue` Most of `ipintr_noqueue`'s code was moved to `ipintr_noqueue_post_fw`.

`ip_output` Most of `ip_output`'s code was moved to `ip_output_post_fw`.

The following new functions were added in AIX 5L to support the new firewall hooks:

`ipintr_noqueue_post_fw` The `ipintr_noqueue_post_fw` hook contains the code that used to be in `ipintr_noqueue` and may be called from either `ipintr_noqueue` or from the firewall hook routine pointed at by `inbound_fw`.

inbound_fw_save_args	The inbound_fw_save_args hook gives a firewall hook routine, called through the inbound_fw variable, the ability to save a copy of the inbound_fw_args_t *args. This copy can be used to call ipintr_noqueue_post_fw at a later time.
inbound_fw_free_args	The inbound_fw_free_args hook frees a inbound_fw_args_t created by inbound_fw_save_args.
ip_output_post_fw	The ip_output_post_fw hook largely contains the code that used to be in ip_output.
outbound_fw_save_args	The outbound_fw_save_args hook creates a copy of outbound_fw_args_t *args. In doing so, it also makes sure all the things pointed at by *args remain valid indefinitely, either by copying or making references.
outbound_fw_free_args	The outbound_fw_free_args hook frees a outbound_fw_args_t created by outbound_fw_save_args. It also frees and removes references from anything pointed at by outbound_fw_args_t *args.

If inbound_fw is set, ipintr_noqueue, the IP input routine, calls inbound_fw and then exits. If not, ipintr_noqueue calls ipintr_noqueue_post_fw and then exits. If the inbound_fw hook routine wishes to pass the packet into IP, it can call ipintr_noqueue_post_fw. The inbound_fw hook may copy its args parameter by calling inbound_fw_save_args, and may free its copy of its args parameter by calling inbound_fw_free_args.

Similarly, ip_output calls outbound_fw if it is set, and calls ip_output_post_fw if not. The outbound_fw hook can call ip_output_post_fw if it wants to send a packet. The outbound_fw hook may copy its args parameter by calling outbound_fw_save_args, and later free its copy of its args parameter by calling outbound_fw_free_args.

8.15 Fast Connect enhancements

IBM AIX Fast Connect provides support for the Server Message Block (SMB) protocol to deliver file and print serving to PC clients. In AIX 5L, there are several improvements that will be discussed in this section.

8.15.1 Locking enhancements

Some applications require shared files between AIX server-based applications and PC client applications. The file server requires lock mechanisms to protect these files against multiple modifications at the same time. Because of this, Fast Connect implements UNIX locking in addition to internal locking, to allow exclusions based on file locks taken by PC clients. AIX 5L implements the following lock enhancements:

- ▶ Opportunistic locks put an exclusive lock on the file when the exclusive opportunistic lock is granted and the file will be unlocked when the opportunistic lock is broken.
- ▶ SMB share modes are implemented with a UNIX lock consistent with the granted open mode and share mode.

8.15.2 Per-share options

Several advanced features of AIX Fast Connect are available as per-share options. These options are encoded as bit fields within the `sh_options` parameter of each share definition. These options must be defined when the share is created with the **net share /add** command, or set through system management tools.

Per-share options currently allowed by **net share /add** are shown in Table 8-8.

Table 8-8 Per-share value options

Parameter	Values	Default	Description
sh_oplockfiles	(0,1)	1	oplocks=1 enables opportunistic lock on this share.
sh_searchcache	(0,1)	0	searchcache=1 enables search caching on this share.
sh_sendfile	(0,1)	0	sendfile=1 enables sendfile API on this share.
mode	(0,1)	1	Mode=1 enables read/write access. mode=0 enables read only access.

8.15.3 PC user name to AIX user name mapping

When a client tries to access resources on the server, it needs to establish an SMB/CIFS session. The SMB/CIFS session setup can use either user-level security or share-level security.

In case of user-level security, clients must present their user names. In previous Fast Connect releases, it was required that the user name match the one on AIX exactly. In many situations, this one-to-one mapping of user names is not possible.

AIX Fast Connect on AIX 5L allows the server administrators to configure the mapping of PC user names to AIX user names. When enabled, AIX Fast Connect tries to map every incoming client user name to a server user name, and then uses that server user name for further user authentication and AIX credentials.

Figure 8-13 shows the SMIT panel with the user name mapping option highlighted.

Attributes		
Type or select values in entry fields. Press Enter AFTER making all desired changes.		
[MORE...10]	[Entry Fields]	
Passthrough Authentication Server	[]	
Backup Passthrough Authentication Server	[]	
Allow DCE/DFS access	[no]	+
Enable network logon server for client PCs	[enabled]	+
Client startup script file name	[startup.bat]	
Guest logon support	[enabled]	+
Guest logon ID	[smb]	+
Enable client user name mapping	[yes]	+
Enable share level security	[no]	+
Share level security user login	[nobody]	+
Enable opportunistic locking	[yes]	+
Enable search caching	[no]	+
Enable send file API support	[no]	+
[BOTTOM]		
F1=Help	F2=Refresh	F3=Cancel
F5=Reset	F6=Command	F7=Edit
F9=Shell	F10=Exit	Enter=Do
		F4=List
		F8=Image

Figure 8-13 SMIT panel with user name mapping option highlighted

If the user name mapping function is enabled, then you can define mapping between client user name (Windows) and server user name (AIX) using the following SMIT dialog: **SMIT -> Communications Applications and Services -> AIX Fast Connect -> Configuration -> Fast Connect Users -> Map a User**. The mapping information is stored in `/etc/cifs/cifsPasswd`. Figure 8-14 on page 528 shows the smit panel for this function.

Map a Client User Name to a Server User Name

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

* Client user name * Server user name Description Active	[Entry Fields] [] [] [] [yes]	+ +
--	---	--------------------

F1=Help
 F5=Reset
 F9=Shell

F2=Refresh
 F6=Command
 F10=Exit

F3=Cancel
 F7=Edit
 Enter=Do

F4=List
 F8=Image

Figure 8-14 Map a Client User Name to a Server User Name panel

8.15.4 Windows Terminal Server support

Windows Terminal Server from Microsoft and other similar products allow support of multiple users on one Windows NT machine. When a multiuser NT machine connects to a Fast Connect server for file and print services, it can use multiple SMB sessions over one transport session. In AIX 5L, Fast Connect allows multiple SMB sessions over one transport session. In previous releases, Fast Connect was limited to one SMB session per transport connection.

8.15.5 Search caching

Generally, file search operation requests from a PC client take large amounts of resources, and performance issues may arise if a large number of clients does file search operations at the same time.

In AIX 5L, Fast Connect allows you to enable search caching. If enabled, all the cached structures will compare their time stamps to the original files to check for modifications periodically. This feature improves file searching significantly.

Figure 8-15 on page 529 shows the SMIT panel with the Enable search caching option highlighted. Search caching must be enabled for the share by enabling the per-share option in addition to the global parameter shown.

Attributes

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[MORE...10]	[Entry Fields]	
Passthrough Authentication Server	[]	
Backup Passthrough Authentication Server	[]	
Allow DCE/DFS access	[no]	+
Enable network logon server for client PCs	[enabled]	+
Client startup script file name	[startup.bat]	
Guest logon support	[enabled]	+
Guest logon ID	[smb]	+
Enable client user name mapping	[yes]	+
Enable share level security	[no]	+
Share level security user login	[nobody]	+
Enable opportunistic locking	[yes]	+
Enable search caching	[yes]	+
Enable send file API support	[no]	+
[BOTTOM]		

F1=Help	F2=Refresh	F3=Cancel	F4=List
F5=Reset	F6=Command	F7=Edit	F8=Image
F9=Shell	F10=Exit	Enter=Do	

Figure 8-15 SMIT panel with Enable search caching option highlighted

8.15.6 Memory-mapped I/O (5.1.0)

AIX 5L Version 5.1 allows files to be mapped to memory. A region of memory is reserved for these files. This region allows access to mapped files, which is much faster and CPU efficient. The `shmat()` system call is used to maximize performance.

Mapping can be used to reduce the overhead involved in writing and reading the contents of files. Once the contents of a file are mapped to an area of user memory, the file may be manipulated as if it were data in memory, using pointers to that data instead of input/output calls. The copy of the file on disk also serves as the paging area for that file, saving paging space. Because mapped files can be accessed more quickly than regular files, the system can load a program more quickly if its executable object file is mapped to a file.

By default, the memory-mapped I/O function is not exploited. To enable this function, insert the following entry in `/etc/cifs/cifsConfig`. Currently, there is no system management tool to do this for you.

```
mmapfiles = 1
```

8.15.7 send_file API

AIX Fast Connect provides the functionality to exploit the send_file routine since AIX Version 4.3.3 and AIX Fast Connect 2.1. The send_file is an API to reduce system overhead, sending cached files directly from being cached in NBC to the connection socket. By default, this functionality is disabled, so to enable this function, you have to select yes in the Enable send file API support field in the following SMIT panel. It is also possible to turn on this function per-share; please refer to 8.15.2, “Per-share options” on page 526.

Figure 8-16 shows the smit panel to set these attributes.

Attributes	
Type or select values in entry fields. Press Enter AFTER making all desired changes.	
	[Entry Fields]
* Server Name	[kepukepu]
* Start Server	[Now] +
* Domain Name	[WORKGROUP]
Description	[Fast Connect Server]
Server alias(es)	
WINS Address	[]
Backup WINS address	[]
Proxy WINS Server	[off] +
NetBIOS Name Server (NBNS)	[on] +
Use Encrypted Passwords	[Negotiate Encryption] +
Passthrough Authentication Server	[]
Backup Passthrough Authentication Server	[]
Allow DCE/DFS access	[no] +
Enable network logon server for client PCs	[disabled] +
Client startup script file name	[startup.bat]
Guest logon support	[disabled] +
Guest logon ID	[nobody] +
Enable client user name mapping	[no] +
Enable share level security	[no] +
Share level security user login	[nobody] +
Enable opportunistic locking	[yes] +
Enable search caching	[yes] +
Enable send file API support	[yes] +

F1=Help	F2=Refresh	F3=Cancel	F4=List
F5=Reset	F6=Command	F7=Edit	F8=Image
F9=Shell	F10=Exit	Enter=Do	

Figure 8-16 Send_file attributes

8.16 SMB file system support (5.2.0)

Server Message Block File System (SMBFS) allows access to shares on SMB servers as a local file system on AIX. Furthermore, you can create, delete, read, write, and modify the access times of files and directories. The owner or access mode of files and directories cannot be changed.

SMBFS can be used to access files on an SMB server. The SMB server is a server running Samba; an AIX server running AIX Fast Connect; or a Windows XP, Windows NT, or Windows 2000 server or workstation. Each of these server types allows a directory to be exported as a share. This share can then be mounted on an AIX system using SMBFS.

To use SMBFS to access a share on an SMB server, the SMBFS needs to be installed and the remote file system mounted.

8.16.1 Installing SMBFS

The SMBFS can be installed from the base operating system CD by using the following command. The `bos.cifs_fs` is on the second install CD.

```
installp -ac -d /dev/cd0 bos.cifs_fs
```

When installing the `bos.cifs_fs` fileset, the following components are installed:

- ▶ SMIT panels
- ▶ The `/usr/lib/drivers/nsmbdd` device driver
- ▶ The `/usr/lib/methods/cfgnsmb` configuration method
- ▶ The `/sbin/helpers/mount_cifs` mount helper
- ▶ The `/etc/mkcifs_fs` boot time script

Furthermore, the device `/dev/nsmb0` is created and always available. At boot time this device is made available by the `/etc/mkcifs_fs` script.

Note: SMBFS is only supported on a 32-bit kernel, and therefore the installation on a 64-bit kernel will fail.

8.16.2 Mounting a file system

To mount an SMBFS file system, as with any other file system, the `mount` command should be used. For the mount of an SMBFS file system the following syntax is applicable:

```
mount [ -r ] -v cifs -n Node [ -o Options ] Share Directory
```

Table 8-9 on page 532 describes the flags applicable when mounting a SMBFS with the `mount` command.

Table 8-9 The mount command flags for SMBFS

Flag	Description
-r	Mounts the file system as a read-only file system.
-v <i>cifs</i>	Specifies the file system as defined by the VfsName parameter in the /etc/vfs file.
-n <i>Node</i>	Specifies the remote node that holds the share, the user name, and the password provided as a string: <i>hostname/username/password</i>
-o <i>Options</i>	Specifies <i>options</i> . Options you enter on the command line should be separated only by a comma, not a comma and a space. The options for the SMBFS file system are: fmode=octal mode for file and directory; default is 755. uid=uid that will be assigned as uid to all files in the mount point on the client, default is root. gid=gid that will be assigned as gid to all files in the mount point on the client; default is system.
Share	Specifies the share name on the node.
Directory	Specifies the mount point on the client.

For example, to mount the share *export* on the node *server* and connect to the share with the user name *dave* and the password *xyz123* under the mount point */mnt* the following command should be used:

```
mount -v cifs -n server/dave/abc123 /export /mnt
```

For the SMBFS, mount and unmount SMIT panels are provided that can be accessed through the following fast path:

```
smit cifs_fs
```

Note: The following notes will assist you in your installation.

- ▶ The SMBFS cannot be automatically mounted with the /etc/filesystem stanza. This limitation occurs due to the need for passwords.
- ▶ The host name specified must be a host name, not an IP address.
- ▶ The host name has to match the network ID or the netbios name of the server.

8.17 SNMPv3 (5.2.0)

AIX 5L Version 5.2 now supports Simple Network Management Protocol (SNMP) Version 3. Prior to Version 5.2, AIX only supported SNMPv1. From Version 5.2

on, the new SNMP agent is a SNMPv1/v2c/v3 compatible agent. SNMP provides a powerful and flexible framework for message level security and access control. This new framework introduced the user-based security model (USM) for message security and the view-based access control model (VACM) for access control. SNMPv3 now supports dynamic reconfiguration of the SNMP agent.

The user-based security model specified in RFC2574, User-based Security Model (USM) for version 3 of the Simple Network Management Protocol, defines the elements of procedure to providing SNMP message level security. USM uses a basic concept of a user, on whose behalf SNMP messages are generated. For USM to work, the user must be defined to both the manager and the agent. For an authenticated request on behalf of the user, both manager and agent must know a set of one or more *secrets* or keys to be used in processing the message. The authentication protocols that the SNMPv3 uses to generate the keys are HMAC-MD5 and HMAC-SHA. For message encryption, it supports CBC 56-bit DES, but it uses whichever protocol is selected for authentication for also processing the privacy keys. The message level security provides the following services:

- ▶ Data integrity
Ensures the data has not been altered in transit.
- ▶ Data origin authentication
Ensures that the message was in fact originated on behalf of the user from which it claims to have been originated.
- ▶ Message timeless and replay detection
Ensures that the message has not been replayed or retransmitted beyond what is normal in a connection-less transport protocol.
- ▶ Data confidentiality
Messages are encrypted to prevent the disclosure of the data in transit.

The view-based access control model, specified in RFC2575, View-based Access Control Model (VACM) for the Simple Network Management Protocol, involves defining collections of data called views, groups of users, and access statements that define which views a group can read, write, or receive traps.

SNMPv3 now supports the ability to dynamically configure the SNMP agent using SNMP SET commands against the MIB objects representing the agent's configuration. This dynamic configuration supports modification of configuration entries either locally or remotely. Because of the dynamic configuration functionality, if you want to manually edit the agent configuration file, it is recommended that you stop the SNMPv3 agent before making any modification to the agent configuration file. After you finish editing the agent configuration file, you must restart the SNMPv3 agent so that the new configuration will take effect.

In Version 5.2, there are three supported versions of SNMP. The three included versions of SNMP are as follows.

- ▶ **SNMPv1 agent**
This is a SNMPv1 only agent.
- ▶ **SNMPv3 agent without data privacy encryption**
This is a SNMPv1/v2c/v3 compatible agent. This is the default version of SNMP starting with AIX 5L Version 5.2 at the system boot time.
- ▶ **SNMPv3 agent with 56-bit DES for data privacy**
This is a SNMPv2/v2c/v3 compatible agent. This version is not installed by default.

You must use the **snmpv3_ssw** command to switch from one version to another. The **snmpv3_ssw** command supports three parameters **-e**, **-n**, and **-1**, which will enable SNMPv3 agent with encryption, SNMPv3 agent without encryption, and SNMPv1 agent, respectively. The following example shows how to enable SNMPv3 agent without encryption using the **snmpv3_ssw** command.

```
# snmpv3_ssw -n
In /etc/rc.tcpip file, comment out the line that contains: dpid2
In /etc/rc.tcpip file, remove the comment from the line that contains: snmpmibd
Stop daemon: snmpd
Make the symbolic link from /usr/sbin/snmpd
to /usr/sbin/snmpdv3ne
Make the symbolic link from /usr/sbin/clsnmp
to /usr/sbin/clsnmpne
Start daemon: snmpd
```

In order to use the SNMPv3 agent with encryption, you must install the **snmp.crypto** fileset from the AIX Expansion Pack. After the installation, the active running SNMP agent is SNMPv3 agent with encryption.

The SNMPv3 subsystem contains several components:

agent	The encrypted SNMPv3 agent is located at <code>/usr/sbin/snmpdv3e</code> , while the non-encrypted agent is located at <code>/usr/sbin/snmpdv3ne</code> . Both agents share the same configuration file <code>/etc/snmpdv3.conf</code> . The SNMPv1 only agent is located at <code>/usr/sbin/snmpdv1</code> . It uses configuration file <code>/etc/snmpd.conf</code> .
manager	The encrypted SNMP manager is located at <code>/usr/sbin/clsnmpe</code> , while the non-encrypted manager is located at <code>/usr/sbin/clsnmpne</code> . Both managers share the same configuration file <code>/etc/clsnmp.conf</code> .

subagents DPI2 subagent: There are three different subagents that use distributed protocol interface version 2 (DPI2) to communicate with the SNMPv3 agent. The hostmibd, snmpmibd, and aixmibd subagents handle requests for management data for specific MIBs. SMUX peer: Based on SNMP Multiplexing (SMUX) protocol. It is another easy way to extend SNMP without recompiling SNMP agent.

For more information about configuring SNMP on AIX, refer to the AIX 5L Version 5.2 system documentation. For more information about the SNMP protocols and standard MIBs, refer to the IETF home page at the following URL:

<http://www.ietf.org>

8.17.1 AIX SNMP subagent for enterprise MIB

Version 5.2 now supports an enterprise-specific MIB for instrumenting the AIX operating system for real-time monitoring, configuration, and events. The AIX enterprise MIB extension subagent is a daemon, aixmibd, that collects data from system for variables defined in the AIX enterprise-specific MIB. The subagent receives SNMP requests and sends data via the SNMP distributed protocol interface (DPI) API for communication with the main AIX snmpd daemon.

The AIX enterprise MIB's defined variables are classified into the following nine categories or groups. For detailed information on the AIX enterprise MIB, refer to the IBM-AIX-MIB definitions in the file /usr/samples/snmpd/aixmib.my.

System	Objects that describe the variables of the subagent, system environment, traps, and the generic trap
Physical and logical storage	Objects that model volume groups, physical volumes, logical volumes, and paging spaces
Printing Spooling	Objects that model printing queue and print job
Users and Groups	Objects that model users and groups
Services	Objects that model the sub-server and subsystem such as Telnet, FTP server with state, and log information
Files Systems	Variables that describe the state and usage of all file systems
Processes	Objects that model the processes in the system
Current login users	Objects that model the current login users

Devices

Objects that model printers/plotters, tapes, hard disks, memory, graphics adapters, SCSI adapters, and CDROM drives

The aixmibd subagent reads its configuration from the /etc/aixmibd.conf file. The preferred method for controlling the aixmibd subagent is with the **startsrc** and **stopsrc** commands.

8.18 Internet Key Exchange enhancements (5.1.0)

In AIX 5L Version 5.1, new features are added to Internet Security Association and Key Management Protocol (ISAKMP), also known as Internet Key Exchange or IKE.

The following topics are discussed in the subsequent sections:

- ▶ Security enhancements
- ▶ New serviceability features
- ▶ System management enhancements

8.18.1 Security enhancements

The Virtual Private Network (VPN) support has been enhanced with several new security features.

IKE group enhancement

VPN includes new functions, such as adding groups, default policies, and supporting wild cards. Support of wild cards, groups, and default policies simplifies the configurations for remote access and DHCP scenarios. You are able to specify one policy, then indicate a group of users or set of users whose remote IDs will use those policies. To manage the group, entries can be added to the group and key database without changing the security policy information.

A group must be defined before using that group name in a tunnel definition. Use the **ikedb** command to define groups. This command accepts XML text as input to create a group definition in the IKE databases. The group's size is limited to 1 KB. The part of the XML file used to create a group would appear similar to the following:

```
<!-- BEGIN IKEGroup P1_Group_1 -->
<IKEGroup
  IKE_GroupName="P1_Group_1">
  <IKEID
    Port="21"
    Protocol="6">
```

```

        <FQDN
          Value="test.austin.ibm.com">
        <IPV4_Address
          Value="9.3.97.191"/>
        </FQDN>
      </IKEID>
    <IKEID
      Port="21"
      Protocol="6">
    <IPV4_Address
      Value="9.3.97.76"/>
    </IKEID>
    <IKEID
      Port="21"
      Protocol="6">
    <User_FQDN
      Value="user@test.austin.ibm.com">
  <IPV6_Address
    Value="1:2:3:4:5:6:7:76"/>
    </User_FQDN>
  </IKEID>
  <IKEID
    Port="21"
    Protocol="6">
  <IPV6_Address
    Value="1:2:3:4:5:6:7:10"/>
  </IKEID>
</IKEGroup>
<!-- END IKEGroup P1_Group_1 -->

```

IKE command line interface

In AIX 5L Version 5.1, a new command line interface is available to retrieve, update, delete, import, and export information in the Internet Key Exchange (IKE) database. IKE tunnels have more complex policy parameters, and in most cases you must use the Web-based System Manager interface to configure IKE.

To perform a put, which writes to the database based on the given XML file, use the following command syntax:

```
# ikedb -p[F s] [ -e entity-file ] [ XML-file ]
```

To perform a get, which displays what is stored in the IKE database, use the following command syntax.

Output is sent to stdout and is in XML format, which is suitable for processing with **ikedb -p**.

```
# ikedb -g[r] [ -t type [ -n name | -i ID -y ID-type ] ]
```

To perform a delete on the specified item from the database, use the following command syntax. The flags are the same as for the `-g` flag, except that `-r` is not supported.

```
# ikedb -d -t type [ -n name | -i ID -y ID-type ]
```

The following is an example of **ikedb -g**:

```
# ikedb -g -t IKE Tunnel -n testtunnel | more
<?xml version="1.0"?>
<AIX_VPN>
<IKE Tunnel>
IKE_TunnelName="testtunnel"
IKE_ProtectionRef="testtunnel_TRANSFORM"
IKE_Flags_AutoStart="Yes"
IKE_Flags_MakeRuleWithOptionalIP="No">
<IKELocalIdentity>
<IPV4_Address Value="9.3.240.58"/>
</IKELocalIdentity>
<IKERemoteIdentity>
<IPV4_Address Value="9.3.240.57"/>
</IKERemoteIdentity>
</IKE Tunnel>
</AIX_VPN>
```

To perform a conversion from a Linux IPSec configuration file to an AIX IPSec configuration file in XML format, use the following command syntax. It requires one or two files from Linux as input, a configuration file, and, possibly, a secrets file with pre-shared keys.

```
# ikedb -c[F] [ -l linux-file ] [ -k secrets-file ] [ -f XML-file ]
```

To perform an expunge on the database, use the following command syntax. This empties out the database.

```
# ikedb -x
```

To perform an output of the DTD that specifies all elements and attributes for an XML file that is used by the **ikedb** command, use the following command syntax. The DTD is sent to stdout.

```
# ikedb -o
```

Import/export IPSEC configuration with Linux

FreeS/WAN, which is Open Source, is the most widely used VPN software for Linux. Although FreeS/WAN does not have the flexibility of AIX IPSec, it provides most of the commonly used functions.

FreeS/WAN 1.5 or higher is required to import the VPN definitions successfully in AIX.

The IPSEC configuration in Linux is defined in two different files (/etc/ipsec.conf and /etc/ipsec.secrets).

Since the IKE support on Linux is only a subset of what is supported on AIX, not all options are able to be imported from one platform to another.

Table 8-10 lists how the Linux VPN functions have been mapped to AIX.

Table 8-10 Linux versus AIX VPN function mapping

Linux keyword	AIX mapping	Default value
interfaces	None; not needed.	None
forwardcontrol	Not available, but can be simulated using the no command.	no
syslog	Not available, but can be simulated using the syslog.conf.	daemon.error
klipsdebug	Not available, but can be simulated using the trace.	None
plutodebug	Not available, but can be simulated using the logging feature of isakmpd and /etc/isakmpd.conf files.	None
dumpdir	No comparable function. Can be simulated by changing to that directory and starting from there.	None
dump	N/A.	None
pluto	No comparable function.	yes
plutoload	No comparable function. AIX loads all defined tunnels in db.	None
plutostart	Autostart	None
plutowait	No comparable function.	yes
plutobackgroundload	No comparable function.	no
prepluto	No comparable function.	None
postpluto	No comparable function.	None
type	tunnel/transport.	tunnel
auto	Autostart.	no
left	Local/Remote IP/ID.	None

Linux keyword	AIX mapping	Default value
leftid	Local/Remote ID.	The value of left
leftrsasigkey	No comparable function.	None
leftsubnet	Local/Remote subnet.	None
leftnexthop	Local/Remote subnet.	The value of right
leftupdown	No comparable function.	None
leftfirewall	No comparable function.	None
right	Local/Remote IP/ID.	None
rightid	Local/Remote ID.	The value of right
rightrsasigkey	No comparable function.	None
rightsubnet	Local/Remote subnet.	None
rightnexthop	Local/Remote subnet.	The value of left
rightupdown	No comparable function.	None
rightfirewall	No comparable function.	None
keyexchange	Redundant information.	ike
auth	AH/ESP in AIX.	ESP
authby	authentication	secret
pfs	pfs	yes
keylife	lifetime	8h
rekeyfuzz	No comparable function.	100%
keyingtries	No comparable function.	3
ikelifetime	lifetime	1h

To import a tunnel configuration from Linux to AIX, perform the following steps:

1. Copy the Linux configuration files (/etc/ipsec.conf and /etc/ipsec.secrets) to AIX.
2. Run the **ikedb** command with the -c option. This will convert the configuration and load it into the database.
3. Initiate the tunnel and verify the status.

In the following example, these steps were performed on a test system.

On the Linux machine

Perform the following steps on the Linux server.

1. Log in as root.
2. Enter `cd /etc`.
3. Open FTP transfer to the AIX system:
 - a. `ftp> cd /tmp`
 - b. `ftp> put ipsec.conf`
 - c. `ftp> put ipsec.secrets`
 - d. `ftp> quit`
4. Enter `# ipsec setup restart`.
5. Enter `# exit`.

On the AIX machine

Perform the following steps on the AIX server.

1. Log in as root.
2. Enter `# cd /tmp`.
3. Enter `#ikedb -c` or `ikedb -c -l ipsec.conf -k ipsec.secrets`.
4. Enter `#ike cmd=activate`.

With the **ikedb** command you can read or edit the IKE database. The input and output format is an Extensible Markup Language (XML) file.

For more details about the **ikedb** command, see “IKE command line interface” on page 537.

The **ikeconvert** utility reads the Linux configuration file and converts it into the XML format, which is suitable for loading in the AIX IKE database.

8.18.2 New serviceability features

To make system administration easier and to prevent file systems from filling up, the outputs have combined using **syslogd**. The **isakmpd** daemon reads the logging level from its own configuration file (`/etc/isakmp.conf`), but the log file name is taken from the **syslogd** configuration file (`/etc/syslog.conf`).

8.18.3 System management enhancements

New and enhanced Web-based System Manager dialogs provide a better way to configure and administer IKE, as shown in Figure 8-17 on page 542.

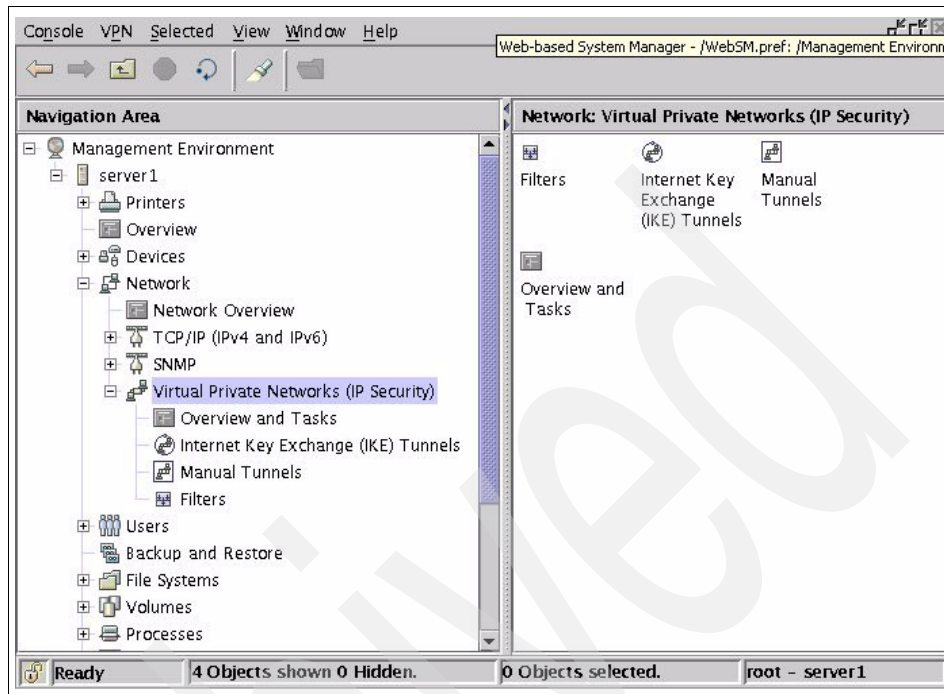


Figure 8-17 Web-based System Manager VPN screen

The Task and Overview panels allow you to perform several configuration tasks:

- ▶ Configure a basic tunnel connection.
- ▶ Manage certificates.
- ▶ Start IP security.
- ▶ Stop IP security.

You also get a quick status overview of the following services:

- ▶ IP security service
- ▶ Internet Key Exchange daemon
- ▶ Digital certificate support
- ▶ IP packet filtering

Selecting Overview and Tasks provides the menu shown in Figure 8-18 on page 543.

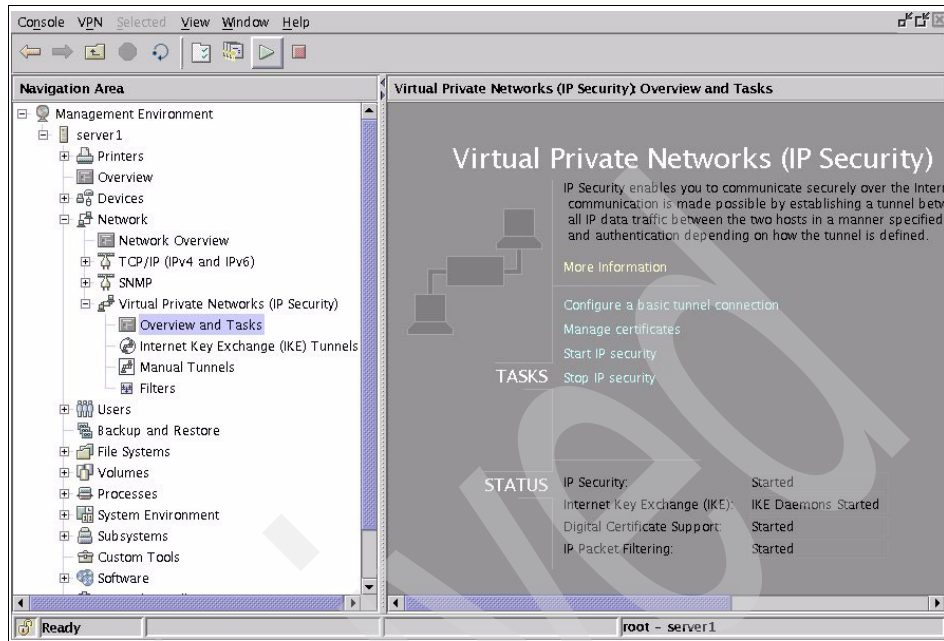


Figure 8-18 Web-based System Manager VPN Overview and Tasks panel

8.18.4 Notify messages

The notify messages enhancement provides additional error information when setting up Security Associations.

The Security Association Payload is used to negotiate security attributes and to indicate the Domain of Interpretation (DOI) and Situation under which the negotiation is taking place.

During Security Association (SA) negotiation, it is possible that errors may occur. The informational exchange with a Notify payload provides a controlled method of informing a peer entity that errors have occurred during protocol processing.

The Notification Payload can contain both ISAKMP and DOI-specific data, and is used to transmit informational data, such as error conditions, to an ISAKMP peer. It is possible to send multiple Notification Payloads in a single ISAKMP message. The Notification Payload contains notification data that specifies why an SA could not be established, such as NO-PROPOSAL-CHOSEN, INVALID-SIGNATURE, and AUTHENTICATION-FAILED.

When a Notify Payload is received, the receiving entity can take appropriate action according to its local policy. A user views any notification payload

information by turning the IKE logging level to EVENTS and viewing the payload information in the log. The NOTIFY information is useful in debugging when an IKE negotiation fails.

The following are the status-type notification messages:

- ▶ CONNECTED
- ▶ RESERVED (future use)
- ▶ DOI-specific codes
- ▶ Private Use

For more detailed information, refer to RFC2407, RFC2408, and RFC2409.

8.18.5 The syslog enhancements

The Internet Key Exchange (IKE) daemons are provided in Table 8-11.

Table 8-11 Web-based System Manager tunnel daemons

Daemon	Description
tmd	The tunnel manager daemon
isakmpd	The IKE daemon
cpsd	The certificate proxy daemon

The tmd and cpsd daemons log events to syslog, and starting with AIX 5L Version 5.1, the isakmpd daemon also logs events to syslog. The logging is enabled by configuring the syslog daemon and refreshing the daemons by issuing the command `ike cmd=log`. The `/etc/isakmpd.conf` configuration file can be set up to specify the logging level. The level can be configured as the following:

none	No logging (the default)
error	Only logging protocol and API errors
isakmp_events	Only logging IKE protocol events and errors
Information	Logging protocol and implementation information (recommended for debugging)

The setting of the log level can be done through the Web-based System Manager, IKE plug-in, as shown in Figure 8-19 on page 545.

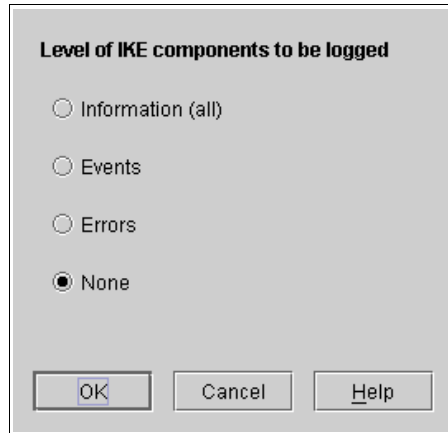


Figure 8-19 Level of IKE components to be logged

When the syslog daemon is running and debugging is turned on, isakmpd will send logging events to the output file of the syslog daemon. The log file is similar to the following example:

```
Mar 15 11:45:47 server3 isakmpd: error: logpipe failed to be ready for reading
Mar 15 11:48:18 server3 isakmpd:
entropy_src::entropy_src():stat(/usr/sbin/ikentropy):No such file or directory.
Mar 15 11:48:18 server3 isakmpd:
/usr/sbin/isakmpd:/usr/sbin/isakmpd:isakmpd:initcrypto dlopen of des failed
Mar 15 11:48:18 server3 isakmpd: isakmpdError number = 2
```

8.19 Dynamic Feedback Protocol (5.1.0)

In AIX 5L Version 5.1 the Dynamic Feedback Protocol (DFP) is now supported. The Dynamic Feedback Protocol provides a mechanism for reporting statistics to server load balancing (SLB) devices (for example, Cisco's Catalyst 4840G, Catalyst 6000, or LocalDirect), so that future connections can be handled by most available servers.

8.19.1 The dfpd agent

The DFP agent is available in the bos.net.tcp.server fileset. The agent is designed to be controlled using the system resource controller (SRC). To start the daemon, just use the normal SRC commands.

```
# startsrc -s dfpd
0513-059 The dfpd Subsystem has been started. Subsystem PID is 23218.
```

To start the DFP agent automatically, an entry in the `/etc/rc.tcpip` file is needed. The new entry is similar to the following:

```
# Start up the dfpd dynamic feedback protocol daemon
start /usr/sbin/dfpd "$src_running"
```

8.19.2 Configuration file

The configuration file of the Dynamic Feedback Protocol daemon (dfpd) is shown in the following:

```
# cat /etc/dfpd.conf
# @(#)20      1.1  src/tcpip/etc/dfpd.conf, dfp, tcpip510 10/3/00 15:56:33
# The md5key is the secret key (upto 64 bytes) that is the same as the one
# defined in the load manager configuration.
md5key 1234567890abcdefabcdef12345678901234567890abcdefabcdef1234567890

# This is the port that dfpd will listen on for load manager connections.
ldlistener 8002

# This is the time in seconds that between computations of cpu idle time.
pollidletime 30

# This is multiplication factor that is applied to the cpu idle time before
# sending it to the load manager. This is useful to rationalize the weights
# among machines of different capacities.
# The mfactor is a positive integer value.
mfactor 1
```

8.19.3 Reports

The DFP agent reports the statistics of the host it is running on. The agent collects the percent of time the CPU is idle. This CPU idle time gets multiplied with a factor (mfactor) specified in the configuration file to get the weight. This weight is being reported to the Load Manager. The multiplication factor is, by default, the number of CPUs if not specified in the configuration. It is possible to configure the interval between successive CPU idle time computations. The default value is 30 seconds. To smooth out the variations in CPU idle time, the average of the last two readings is used.

A DFP agent does not collect, maintain, or provide bind information to the Load Manager.

To ensure integrity of the data communication, the DFP Agent and the Load Manager share a secret key up to 64 bytes long.

The Load Manager sends a keepalive time when a connection is initiated. If the Load Manager does not provide a keepalive time, then a default of 60 seconds is assumed. The CPU idle time information will be sent to the Load Manager periodically with the period being the lower of the keepalive time and the time between CPU idle computations.

8.20 ATM LANE and MPOA enhancements

The ATM LAN Emulation device driver emulates the operation of Standard Ethernet, IEEE 802.3 Ethernet, and IEEE 802.5 token-ring LANs. It encapsulates each LAN packet and transfers its LAN data over an ATM network at up to OC12 speeds (622 megabits per second). This data can also be bridged transparently to a traditional LAN with ATM/LAN bridges, such as the IBM 2216. The logical presentation of an ATM system environment LAN Emulation is shown in Figure 8-20.

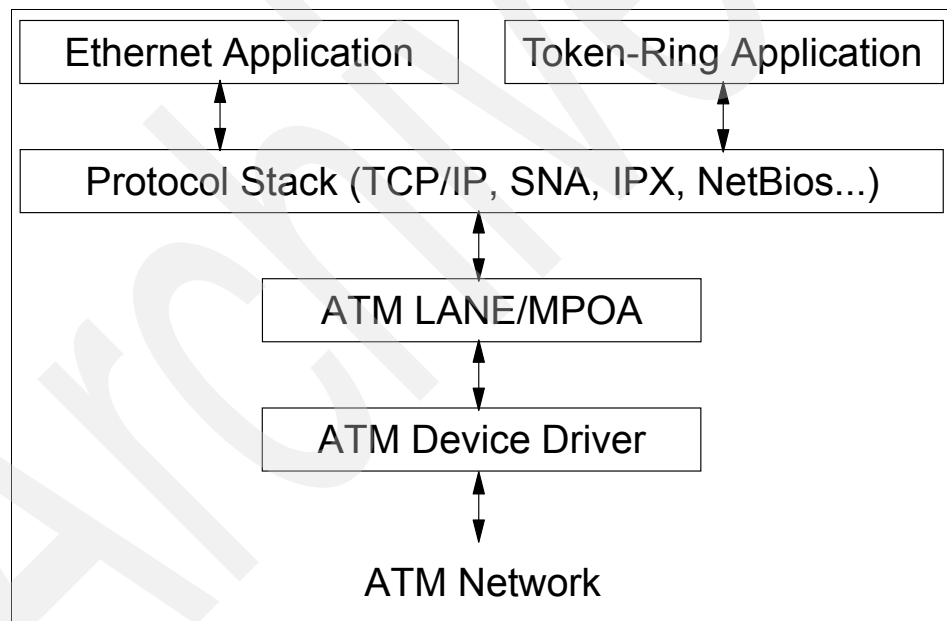


Figure 8-20 System environment ATM LAN Emulation

The ATM LANE device driver is a dynamically loadable device driver. Each LE client or multiprotocol over ATM (MPOA) client is configurable by the operator, and the LANE driver is loaded into the system as part of that configuration process. If an LE client or MPOA client has already been configured, the LANE driver is automatically reloaded at reboot time as part of the system configuration process.

8.20.1 Debug option (5.1.0)

In AIX 5L Version 5.1, the `debug_trace` option, when configuring the ATM LANE device driver, can be set to off.

The `debug_trace` option specifies whether the MPOA client should keep a real time debug log within the kernel and allow full system trace capability. Select **Yes** to enable full tracing capabilities for this client. Select **No** for optimal performance when minimal tracing is desired. The default is Yes (full tracing capability).

Toggling a LANE/MPOA trace off disables all normal flow trace points to both the system trace and the internal driver trace buffer. This will improve performance of the interface on large SMP systems. Error conditions will continue to trace to the system trace and the internal driver trace buffer.

There are different ways to toggle the debug option on and off. You can configure the LANE/MPOA client with SMIT and are able to select the full tracing, as shown in Figure 8-23 on page 552.

8.20.2 IP fragmentation (5.1.0)

The multiprotocol over ATM (MPOA) implementation supports IPv4 without options. In AIX 5L Version 5.1, MPOA has been enhanced to support IP fragmentation.

Having unlike protocols at each end of a shortcut (Figure 8-21 on page 549) poses a special problem, because they do not necessarily have the same maximum transmission unit (MTU) sizes defined at each end.

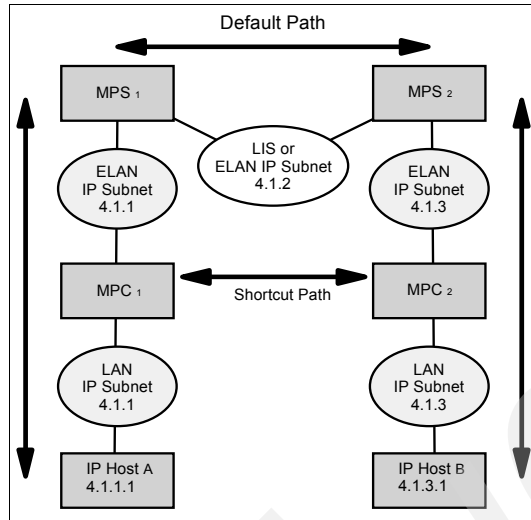


Figure 8-21 An example of an MPOA network

Ethernet has a LANE frame size of 1516 and MTU of 1500 bytes, while token ring can have LANE frame sizes of 4544 or 18190 bytes with subsequently larger MTUs. These are clearly incompatible and require the MPOA layer to do IP fragmentation.

Send IP packet to MPOA shortcut

A packet going out onto an MPOA shortcut will be fragmented if the following conditions are true:

1. The flags field in the IP header has the *Do not fragment* bit turned off.
2. The `ip_len` field in the IP header has a value larger than the MTU returned in the MPOA Resolution Reply.
3. MPOA IP fragmentation is enabled.
4. Mbufs can be obtained to create all the fragments.

If any of the above conditions are false, the packet will be sent down the LANE path. If fragmentation is performed, each fragment will have as large of an `ip_len` as possible that does not exceed the MTU returned in the MPOA Resolution Reply and does not violate the rules for IP fragmentation.

Receive IP packet from MPOA shortcut

A packet received on an MPOA shortcut that will be reassembled into an IEEE 802.3 frame format will be fragmented if the following conditions are true:

1. The flags field in the IP header has the *Do not fragment* bit turned off.
2. The ip_len field in the IP header has a value larger than the LE Client's NDD MTU, minus the size of the DLL header.

If a packet requiring fragmentation has the *Do not fragment* bit turned on in the flags field of the IP header, the MPOA client (MPC) will drop the packet and generate an ICMP message (ICMP Unreachable Error, Fragmentation Required). The ICMP message contains the largest IP MTU that the LE Client can handle.

Reassemble to IEEE 802.3 Ethernet format

The IEEE 802.3 frame format contains a length field that cannot have a value larger than 1500 bytes. For this reason, packets received on a shortcut to be reassembled into an IEEE 802.3 frame format must be fragmented to be received.

Reassemble to Standard Ethernet format

A packet received on an MPOA shortcut that will be reassembled into a Ethernet frame format will never be fragmented. The Ethernet frame format does not contain any length information, so there is no need to fragment these packets once they have been received. The only limitation is the packet cannot be larger than what IP can handle. Currently, IP can handle up to 64 KB. The current LANE maximum frame size is 18190 bytes, so this is not an issue.

Reassemble to token ring-format

A packet received on an MPOA shortcut that will be reassembled into a LANE token-ring frame format will never be fragmented. The token-ring frame format does not contain any length information, so there is no need to fragment these packets once they have been received. The only limitation is that the packet cannot be larger than what IP can handle. Currently, IP can handle up to 64 KB.

Configure IP fragmentation

To disable the IP fragmentation feature, you need a configured an available ATM LAN Emulation MPOA client adapter. Use the following command to check the available adapters:

```
# lsdev -Cc adapter
atm0  Available  10-68IBM PCI 155 Mbps ATM Adapter (14107c00)
atm1  Available  30-78IBM PCI 155 Mbps ATM Adapter (14107c00)
ent1  Available  ATM LAN Emulation Client (Ethernet)
mpc0  Available  ATM LAN Emulation MPOA Client
```

The IP fragmentation can be changed by using SMIT, as shown in Figure 8-22.

Change / Show an MPOA Client

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

	[Entry Fields]	
MPOA Client Device Name	mpc0	
Automatic Configuration via LECS	No	+
Shortcut Setup Frame Count	[10]	+#
Shortcut Setup Frame Time (seconds)	[1]	+#
Initial Request Retry Time (seconds)	[5]	+#
Maximum Request Retry Time (seconds)	[40]	+#
Failed request retry Hold Down Time (seconds)	[160]	+#
VCC Inactivity Timeout value (minutes)	[20]	+#
Debug Trace Enabled	Yes	+
Enable MPOA Fragmentation	Yes	+
Apply change to DATABASE only	no	+

F1=Help
F5=Reset
F9=Shell

F2=Refresh
F6=Command
F10=Exit

F3=Cancel
F7=Edit
Enter=Do

F4=List
F8=Image

Figure 8-22 SMIT panel for Change/Show an MPOA client

You can also verify the settings of the multi-protocol client (MPC) device by using the **lsattr** command:

```
# lsattr -El mpc0
auto_cfg      No  Auto Configuration with LEC/LECS
True
sc_setup_count 10  Shortcut Setup Frame Count
True
sc_setup_time  1   Shortcut Setup Frame Time in seconds
True
init_retry_time 5   Initial Request Retry Time in seconds
True
retry_time_max 40   Maximum Request Retry Time in seconds
True
hold_down_time 160  Failed Resolution request retry Hold Down Time in seconds
True
vcc_inact_time 20   VCC Inactivity Timeout value in minutes
True
debug_trace    Yes  Debug Trace Enabled
True
fragment       Yes  Enable MPOA Fragmentation
True
```

If MPOA fragmentation is enabled, outgoing packets will be fragmented if needed.

If MPOA fragmentation is disabled, the outgoing packages are never fragmented. If fragmentation is needed, the packets have to be sent down to the LANE.

Incoming packets will be fragmented when necessary, regardless of whether MPOA fragmentation is enabled.

Add an Ethernet ATM LE Client

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

	[Entry Fields]	
Local LE Client's LAN MAC Address (dotted hex)	<input type="text" value="[]"/>	
Automatic Configuration via LECS	No	+
If No, enter the LES ATM Address (dotted hex)	<input type="text" value="[]"/>	
If Yes, enter the LECS ATM Address (dotted hex)	<input type="text" value="[]"/>	
Local ATM Device Name	[atm0]	+
Emulated LAN Type	Ethernet/IEEE 802.3	+
Maximum Frame Size (bytes)	Unspecified	+
Emulated LAN Name	<input type="text" value="[]"/>	
Force Emulated LAN Name	No	+
Enable Forum MPOA and LANE-2 functions	No	+
MPOA Primary Auto Configurator	No	+
Debug Trace Enabled	Yes	+

F1=Help
F5=Reset
F9=Shell

F2=Refresh
F6=Command
F10=Exit

F3=Cancel
F7=Edit
Enter=Do

F4=List
F8=Image

Figure 8-23 SMIT panel for adding an ATM LE client

The same debug control is available with a token-ring ATM LE client or an MPOA client. You can select this through SMIT, as shown in Figure 8-23. Also, depending on the device driver type, one of the following commands can be used to toggle the debug tracing on and off dynamically while the client is operational:

# entstat -t	Toggles LANE Ethernet debug tracing on and off
# tokstat -t	Toggles LANE token-ring debug tracing on and off
# mpcstat -t	Toggles MPOA debug tracing on and off

8.20.3 Token-ring support for MPOA

AIX 5L Version 5.1 provides support for token ring for multiprotocol over ATM (MPOA). This also includes the capability to transfer shortcut data between unlike LAN IP protocol layers, such as token ring to Ethernet, or token ring to IEEE 802.3. The panel for adding this function is shown in Figure 8-24 on page 553.

Add a Token Ring ATM LE Client

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

	[Entry Fields]
Local LE Client's LAN MAC Address (dotted hex)	<input type="text"/>
Automatic Configuration via LECS	No +
If No, enter the LES ATM Address (dotted hex)	<input type="text"/>
If Yes, enter the LECS ATM Address (dotted hex)	<input type="text"/>
Local ATM Device Name	[atm0] +
Emulated LAN Type	Token Ring +
Maximum Frame Size (bytes)	Unspecified +
Emulated LAN Name	<input type="text"/>
Force Emulated LAN Name	No +
Enable Forum MPOA and LANE-2 functions	No +
MPOA Primary Auto Configurator	No +
Debug Trace Enabled	Yes +

F1=Help
F5=Reset
F9=Shell

F2=Refresh
F6=Command
F10=Exit

F3=Cancel
F7=Edit
Enter=Do

F4=List
F8=Image

Figure 8-24 SMIT panel for adding a token ring ATM LE client

8.20.4 ATM communications support for UNI and ILMI V4.0 (5.2.0)

The asynchronous transfer mode (ATM) communications subsystem has been enhanced to support the user-network interface (UNI) signaling specification Version 4.0 and integrated local management interface (ILMI) specification Version 4.0.

One of the features of the UNI Version 4.0 specification is that incoming add party requests are now supported. An Add party request allows multiple LAN Emulation Clients to operate on the same emulated LAN over a single port.

The ATM specifications can be found at the following URL:

<http://www.atmforum.com/standards/approved.html>

8.21 ATM network performance enhancements (5.2.0)

ATM for Version 5.2 has three main enhancements, which will be detailed in this section.

8.21.1 Changes to LANE2 timers design

Configuration parameters for the control and forward disconnect timer have been changed as follows.

Control timer

ATM Forum LANE Version 2 has changed client support for the control timeout and has added two new configuration parameters: Initial control timeout value and the control timeout multiplier.

- ▶ Control timeout value (C7): The default value has been changed from 120 seconds to 30 seconds. The configuration parameters are now:
 - Minimum: 1 second
 - Default: 30 seconds
 - Maximum: 300 seconds
- ▶ Initial control timeout value (C7i - new parameter) has the settings described below:
 - Minimum: 1 second
 - Default: 5 seconds
 - Maximum: 10 seconds
- ▶ Control timeout multiplier (C7x - new parameter) has the settings described below. This parameter is not user configurable and will always run with the default of 2. These parameters and how they interact are described below:
 - C7_wait: Timeout value that is sent to the response timer and is set to the Initial Control Timeout value.
 - C7_cumwait: Cumulative period derived from the backoff multiplier; will initially be set to C7_wait.
 - C7_retry: Number of retries that have already occurred; initially set to 0. If the retry timer expires without receiving a response, C7_wait is added to the C7_cumwait value, and C7_retry is incremented. When the value for control timeout is reached the control sequence has failed.

Forward disconnect timer

Forward disconnect timer is used to ensure that the BUS has a point-to-multipoint path back to the client at all times. This is initiated once the client starts the Multicast Send VCC. If a Multicast Forward VCC is not established on the BUS before the timer expires the Multicast Send is dropped, and a new Multicast Send is initiated. The new parameters are as follows:

Forward Disconnect Timer (C33)

- ▶ Minimum: 10 seconds
- ▶ Default: 60 seconds
- ▶ Maximum: 300 seconds

8.21.2 Changes to checksum offload design

Flags are used to identify, transmit, and receive packets that contain checksum information. They originate in the TCP layer for transmit, and in the ATM device driver on receive.

The call manager for the device driver is able to accept a protocol for each VC created (LANE Ethernet, LANE token ring, MPOA, or C/IP). The Call Manager and the ATM device driver are able to accept checksumming for both transmit and receive, or either transmit or receive, for a particular VC.

The adapter will only attempt to modify transmit packets that are set for checksum offloading and will only indicate that receive checksumming was completed on IP packets. LANE and MPOA are able to checksum on VCs for each of the LAN protocol types.

8.21.3 Changes to dynamic MTU design

This function allows dynamic maximum MTU support for devices that have MTU values that can be changed.

Typically when ATM LANE devices complete the JOIN process, the MTU size (ndd_mtu) has already been set to unspecified. Once joined, the network interface does not update this value even though the network value is then known.

Dynamic MTU allows this value to be revalidated against the ndd_mtu figure once the interface is up. This feature requires that the ndd_mtu value is set to its largest possible value when the device is first brought up for autosense devices. The ndd_mtu figure is then set to the network ndd_mtu value once it has joined the network.

This change to dynamic MTU affects ATM devices, token ring, and Ethernet. Token ring and Ethernet network interfaces will fail when a user MTU exceeds the ndd_mtu range, but also saves the ndd_mtu value, which is updated if a larger value is detected. If a user MTU is larger than the current figure, but is still within range, the operational MTU will be changed to fit within the current ndd_mtu.

8.22 EtherChannel enhancements (5.1.0)

EtherChannel is a network aggregation technology that allows you to produce a single large pipe by combining the bandwidth of multiple Ethernet adapters. In AIX 5L Version 5.1, the EtherChannel feature has been enhanced to support the detection of interface failures. This is called network interface backup.

EtherChannel is a trademark registered by Cisco Systems and is generally called multi-port trunking or link aggregation. If your Ethernet switch device has this function, you can exploit the support provided in AIX 5L Version 5.1. In this case, you must configure your Ethernet switch to create a channel by aggregating a series of Ethernet ports.

8.22.1 Network interface backup mode

In the network interface backup mode, the channel will only activate one adapter at a time. The intention is that the adapters are plugged into different Ethernet switches, each of which is capable of getting to any other machine on the subnet/network. When a problem is detected, either with the direct connection, or through inability to ping a machine, the channel will deactivate the current adapter and activate a backup adapter.

Note: The network interface backup feature is currently supported by 10/100 Ethernet FC 2968 and 4962 and gigabit Ethernet PCI card FC 2969 (devices.pci.23100020.rte, devices.pci.1410FF01.rte, and devices.pci.14100401.rte). If you are using other devices, you may receive unexpected results.

Configuring EtherChannel for network interface backup

Use SMIT either by choosing the SMIT fast path etherchannel or going through the menu (**Devices** -> **Communication** -> **EtherChannel**), as shown in Figure 8-25 on page 557. Note that these screens are specific to AIX 5L Version 5.1 and have received updates for AIX 5L Version 5.2.

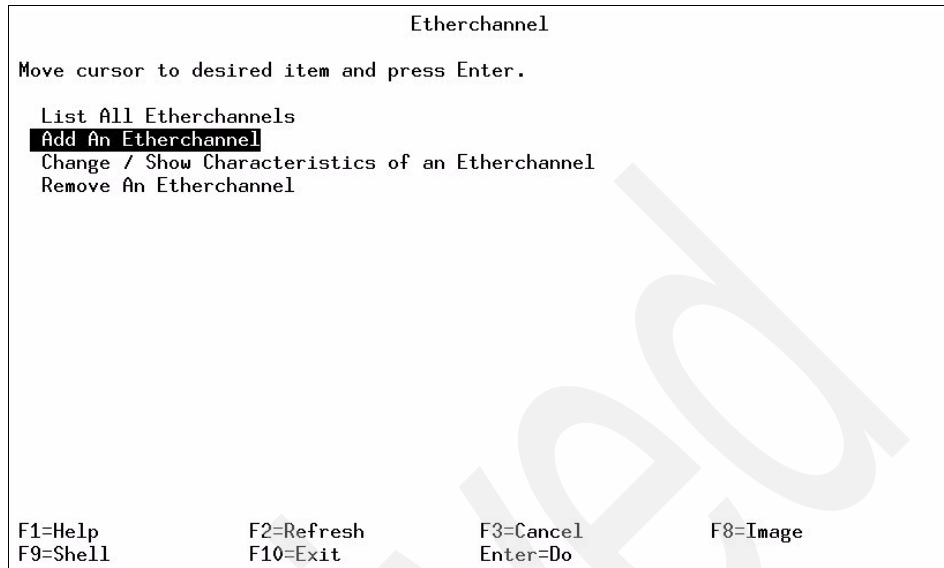


Figure 8-25 SMIT panel to add a new EtherChannel

Choose **Add An EtherChannel** to add a new EtherChannel definition to your system, as shown in Figure 8-26.

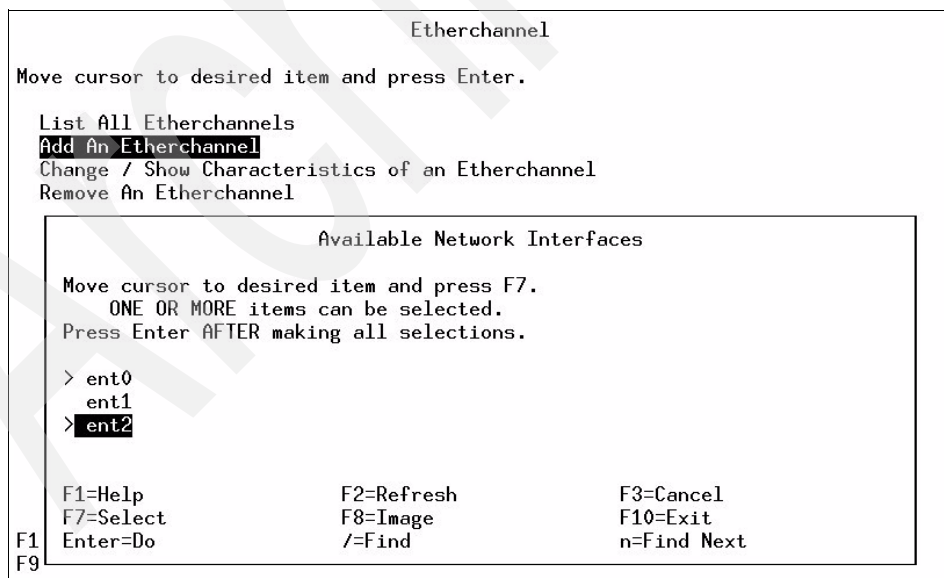


Figure 8-26 SMIT panel for choosing the adapters that belong to the channel

To create a new EtherChannel, you have to select the network interfaces that will be a part of the channel. If you select an interface that is in use or already part of another EtherChannel, you will receive an error similar to:

```
Method error (/usr/lib/methods/cfgech):
```

```
0514-001System error:
```

```
Method error (/usr/lib/methods/chgent):
```

```
0514-062cannot perform the requested function because the
specified device is busy.
```

Add An Etherchannel

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

Etherchannel Adapters	[Entry Fields]	
Enable ALTERNATE ETHERCHANNEL address	ent0 ent2	+
ALTERNATE ETHERCHANNEL address	no	+
Mode	[0x1234deadbeef]	+
Enable GIGABIT ETHERNET JUMBO frames	netif_backup	+
Internet Address to Ping	no	+
Number of Retries	[10.0.0.3]	#
Retry Timeout (sec)	[]	#

F1=Help
F5=Reset
F9=Shell

F2=Refresh
F6=Command
F10=Exit

F3=Cancel
F7=Edit
Enter=Do

F4=List
F8=Image

Figure 8-27 SMIT panel for configuring the EtherChannel

If you are using the non-gigabit adapters (FC 2968 or 4962 device 23100020 or 1410FF0), you should enable polling before adding these adapters to the EtherChannel so the adapters are able to detect changes to the link status and inform the EtherChannel. To do this, use the following command:

```
#chdev -l entx -a poll_link=yes
```

You should run this command for all FC 2968 or 2962 adapters in the EtherChannel. If you do not run this command, the EtherChannel will not work correctly.

Choose a valid alternate hardware address for the new EtherChannel, as shown in Figure 8-27. Change the EtherChannel mode to netif_backup to enable the network interface backup feature. In that mode, the channel will be informed of the adapter's link status. If the link status is not up (either due to a cable being unplugged, switch down, or device driver problem), the channel will switch to another adapter.

This mode is the only one that makes use of the Internet Address to Ping, Number of Retries, and Retry Timeout fields. The following list provides the meaning of the fields:

Internet Address to Ping The address will be pinged if the address field has a non-zero address and the mode is set to netif_backup. If the channel is unable to ping the address for the number of retries times in retry timeout intervals, the channel will switch adapters.

Number of Retries The number of retries is the number of ping response failures before the channel switches adapters. The default is three times.

Retry Timeout The retry timeout is the interval in seconds between the times when the channel will send out a ping packet. The default is one second intervals. The ping feature is design to detect failures on the entire network path to the host being pinged, not just failures between the adapter and switch. The address select for pinging must be an IP address that you always expect connectivity to.

Once the EtherChannel has been configured, the new adapter and interfaces are available, as shown in the following example:

```
server1:/home/root>lsdev -Cc adapter
tok0    Available 10-68 IBM PCI Tokenring Adapter (14103e00)
ent0     Available 10-78 IBM 10/100 Mbps Ethernet PCI Adapter (23100020)
ent1     Available 10-80 IBM PCI Ethernet Adapter (22100020)
ent2     Available 20-60 IBM 10/100 Mbps Ethernet PCI Adapter (23100020)
sioma0   Available 01-K1-01Mouse Adapter
ent4     Available Etherchannel
ent3     Available 10-70 3Com 3C905-TX-IBM Fast EtherLink XL NIC
```

```
server1:/home/root>lsdev -Cc if
en1 Defined 10-80 Standard Ethernet Network Interface
en2 Defined 20-60 Standard Ethernet Network Interface
et0 Defined 10-78 IEEE 802.3 Ethernet Network Interface
et1 Defined 10-80 IEEE 802.3 Ethernet Network Interface
et2 Defined 20-60 IEEE 802.3 Ethernet Network Interface
lo0 Available Loopback Network Interface
tr0 Available 10-68 Token Ring Network Interface
en3 Available 10-70 Standard Ethernet Network Interface
et3 Defined 10-70 IEEE 802.3 Ethernet Network Interface
en0 Defined 10-78 Standard Ethernet Network Interface
en4 Defined Standard Ethernet Network Interface
et4 Defined IEEE 802.3 Ethernet Network Interface
```

Configuring IP on the EtherChannel interface

The new interface can be configured like any other network interface. Use SMIT to define an IP address on the interface:

```
server1:/home/root>ifconfig en4
en4:
flags=e080863<UP,BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64BIT>
    inet 10.0.0.4 netmask 0xffffffff broadcast 10.0.0.255
```

Use the **ping** command to test the new IP connection:

```
server1:/home/root>ping 10.0.0.3
PING 10.0.0.3: (10.0.0.3): 56 data bytes
64 bytes from 10.0.0.3: icmp_seq=0 ttl=255 time=0 ms
64 bytes from 10.0.0.3: icmp_seq=1 ttl=255 time=0 ms
64 bytes from 10.0.0.3: icmp_seq=2 ttl=255 time=0 ms
```

8.23 EtherChannel backup (5.2.0)

Version 5.2 introduces support for the use of an EtherChannel backup adapter for EtherChannel installations.

8.23.1 EtherChannel overview

EtherChannel allows for multiple adapters to be aggregated into one virtual adapter, which the system treats as a normal Ethernet adapter. The IP layer sees the adapters as a single interface with a shared MAC and IP address. The aggregated adapters can be a combination of any supported Ethernet adapter, although they must be connected to a switch that supports EtherChannel. All connections must be full-duplex and there must be a point-to-point connection between the two EtherChannel-enabled endpoints.

EtherChannel provides increased bandwidth, scalability, and redundancy. The EtherChannel provides aggregated bandwidth with traffic being distributed over all adapters in the channel rather than just one. To increase bandwidth, the only requirement is to add more adapters to the EtherChannel, up to a maximum of eight physical devices. If an adapter in the EtherChannel goes down, then traffic is transparently rerouted. Incoming packets are accepted over any of the interfaces available. The switch can choose how to distribute its inbound packets over the EtherChannel according to its own implementation, which in some installations is user configurable. If all the adapters in the channel fail then the channel is unable to transmit or receive packets.

There are two policies for outbound traffic in Version 5.2: Standard and round robin. The standard policy is the default. This policy allocates the adapter to use

on the basis of the hash of the destination IP address. The round-robin policy allocates a packet to each adapter on a round-robin basis in a constant loop.

8.23.2 EtherChannel backup adapter

Version 5.2 introduces the concept of configuring a backup adapter to the EtherChannel. The backup adapter's purpose is to take over the IP and MAC address of the channel in the event of a complete channel failure, which is constituted by the failure of all adapters defined to the channel. It is only possible to have one backup adapter configured per EtherChannel.

All adapters that constitute the EtherChannel must be connected to the same switch. Version 5.2 can protect against a switch failure as it provides the capability for the backup to be connected to a different switch to the EtherChannel. Therefore, to guard against switch failure and introduce further resilience it is recommended that the backup adapter is connected by a separate Ethernet switch to the EtherChannel. Until takeover the backup adapter is idle.

The process is as follows:

- ▶ If all but one of the primary adapters fail, then no action is taken as the primary objective is to keep the EtherChannel open.
- ▶ If all primary adapters fail, the backup adapter is checked to see if it is functioning. If the backup adapter is down, the primary adapters stay as the active channel. This is because it is more likely that one of the EtherChannel adapters will come back up before the single backup adapter.
- ▶ If the backup adapter is up and all the primary adapters fail, then failover starts. All the adapters in the EtherChannel are disabled, and take on the MAC and IP address of the backup adapter. The backup adapter takes on the MAC and IP of the EtherChannel. All adapters are then re-enabled.
- ▶ Gratuitous ARPs are sent to ensure that the MAC associated with the EtherChannel port is now mapped to the backup adapter port.
- ▶ When at least one of the adapters in the EtherChannel becomes available, the MAC and IP are swapped back to the EtherChannel following the same process as before.

8.23.3 netif_backup mode

Prior to AIX 5L Version 5.2, there was another mode of operation called `netif_backup` (see 8.23.2, “EtherChannel backup adapter” on page 561). The functionality of the backup adapter is used to emulate what used to be `netif_backup` mode.

The netif_backup mode enabled the following features:

- ▶ Ability to connect every adapter to a different switch so that each can access all the machines in the same network.
- ▶ Failure could be detected by either noticing that the link status of an adapter is down or optionally pinging a remote machine.

In Version 5.2, the backup adapter function is used to emulate the netif_backup mode and retains the ping feature of the netif_backup mode.

8.23.4 Configuration

The EtherChannel has a new attribute for the backup adapter in the Object Data Manager (ODM), called backup_adapter. This is possible to see using the `lsattr` command on the EtherChannel.

There are also changes to the SMIT (fast path is etherchannel) screen for configuring EtherChannel. From there it is possible to select **Add An EtherChannel**. The results of this selection are shown in Figure 8-28.

Add An EtherChannel

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

	[Entry Fields]	
EtherChannel Adapters	ent0	+
Enable Alternate EtherChannel Address	no	+
Alternate EtherChannel Address	[]	+
Enable Gigabit Ethernet Jumbo Frames	no	+
Mode	standard	+
Backup Adapter		+
Internet Address to Ping	[ent2]	
Number of Retries	[10]	+#
Retry Timeout (sec)	[10]	+#

F1=Help
F5=Reset
F9=Shell

F2=Refresh
F6=Command
F10=Exit

F3=Cancel
F7=Edit
Enter=Do

F4=List
F8=Image

Figure 8-28 SMIT screen showing changes to allow EtherChannel backup

As shown in Figure 8-28, the Number of Retries and Retry Timeout fields have been modified. Since this example has only defined a single adapter acting as the main channel and the backup adapter, the EtherChannel will function as if it were in netif_backup mode prior to AIX 5L Version 5.2. These are only relevant

for the ping feature when emulating the `netif_backup` mode. It is only possible to do this with one adapter defined to the main channel and one adapter as a backup.

The `mkdev` command also allows the specification of the field `backup_adapter` when used with the `-a` flag. For the configuration shown in the figure, the command would be:

```
mkdev -c adapter -s pseudo -t ibm_ech -a "adapter_names=ent0  
backup_adapter=ent2 num_retries=10 retry_time=10"
```

8.24 Virtual Local Area Network (5.1.0)

Virtual Local Area Networks (VLANs) can be thought of as logical broadcast domains. A VLAN splits up groups of network users on a real physical network into segments of logical networks. This implementation supports the IEEE 802.1Q VLAN tagging standard, with the capability to support multiple VLAN IDs running on Ethernet adapters. Each VLAN ID is associated with a separate Ethernet interface to the upper layers (for example, IP) and creates unique logical Ethernet adapter instances per VLAN, for example, `ent1`, `ent2`, and so on.

The IEEE 802.1Q VLAN support can be configured over any supported Ethernet adapters. If connecting to a switch, the switch must support IEEE 802.1Q VLAN.

You can configure multiple VLAN logical devices on a single system. Each VLAN logical device constitutes an additional Ethernet adapter instance. These logical devices can be used to configure the same Ethernet IP interfaces used with physical Ethernet adapters. As such, the `no` option, `ifsize` (default 8), needs to be increased to include not only the Ethernet interfaces for each adapter, but also any VLAN logical devices that are configured.

When configuring a VLAN network, ensure that all virtual adapters within the virtual network have the same VLAN ID.

Each VLAN can have a different maximum transmission unit (MTU) value, even if sharing a single physical Ethernet adapter.

VLAN support is managed through SMIT. Type the `smit vlan` fast path from the command line and make your selection from the main VLAN menu. Online help is available.

After you have configured a VLAN, configure the IP interface (for example, `en1`) for standard Ethernet or `et1` for IEEE 802.3, using Web-based System Manager, SMIT, or the command line interface.

The following command shows the SMIT fast path for the Local Virtual Area Network configuration methods:

```
# smitty vlan
```

The Add a VLAN panel is shown in Figure 8-28 on page 562.

Add A VLAN

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

VLAN Base Adapter

* **VLAN Tag ID**

[Entry Fields]

ent0

[5]

F1=Help
F5=Reset
F9=Shell

F2=Refresh
F6=Command
F10=Exit

F3=Cancel
F7=Edit
Enter=Do

F4=List
F8=Image

Figure 8-29 SMIT panel for adding a VLAN

The **lsdev** command will list the virtual LAN adapters as a member of the adapter class, as provided in the following output:

```
# lsdev -HCc adapter
name      status    location description
sa0       Available 01-S1    Standard I/O Serial Port
sa1       Available 01-S2    Standard I/O Serial Port
siokma0   Available 01-K1    Keyboard/Mouse Adapter
fda0      Available 01-D1    Standard I/O Diskette Adapter
scsi0     Available 10-60    Wide/Ultra-2 SCSI I/O Controller
scsi1     Available 10-61    Wide/Ultra-2 SCSI I/O Controller
son10     Available 20-58    GXT4000P Graphics Adapter
sioka0    Available 01-K1-00 Keyboard Adapter
siota0    Available 01-Q1    Tablet Adapter
ppa0      Available 01-R1    CHRP IEEE1284 (ECP) Parallel Port Adapter
paud0     Available 01-Q2    Ultimeidia Integrated Audio
ent0      Available 10-80    IBM 10/100 Mbps Ethernet PCI Adapter (23100020)
tok0      Available 10-88    IBM PCI Tokenring Adapter (14103e00)
```

```
sioma0  Available 01-K1-01 Mouse Adapter
ent1    Available          VLAN
```

Enter the following command to further set up a VLAN, then follow the examples in Figure 8-30 and Figure 8-31.

```
# smit chinnet
```

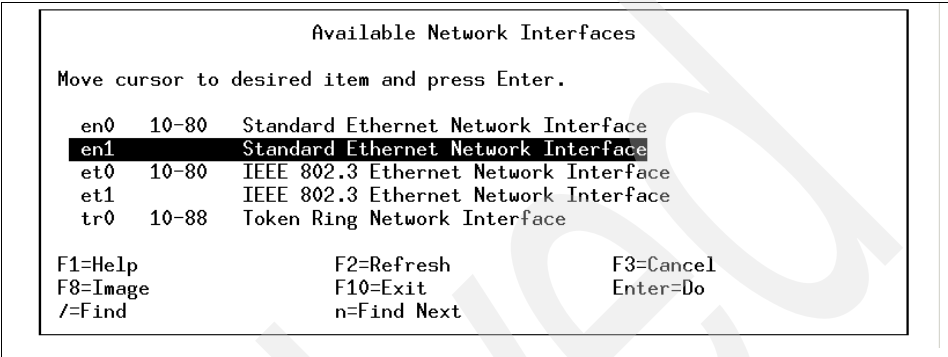


Figure 8-30 SMIT Available Network Interfaces panel

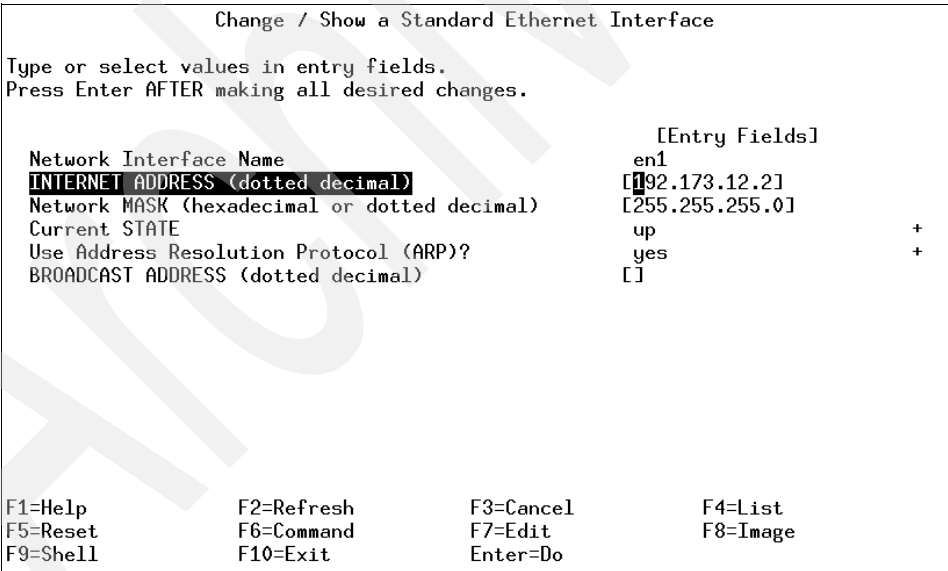


Figure 8-31 SMIT Change/Show a Standard Ethernet Interface panel

The **netstat** command reports the existence of the newly defined interface. Also, you will notice that the en0 and en1 have the same MAC address:

```
# netstat -in
```

Name	Mtu	Network	Address	Ipkts	Ierrs	Opkts	Oerrs	Coll
tr0	1492	link#2	0.60.94.8a.b0.77	250386	0	69264	0	0
tr0	1492	9.3.240	9.3.240.57	250386	0	69264	0	0
en0	1500	link#3	0.6.29.4.44.2	466302	0	1069552	0	0
en0	1500	192.1.1	192.1.1.3	466302	0	1069552	0	0
en1	1500	link#4	0.6.29.4.44.2	0	0	1	0	0
en1	1500	192.173.12	192.173.12.2	0	0	1	0	0
lo0	16896	link#1		20830	0	20867	0	0
lo0	16896	127	127.0.0.1	20830	0	20867	0	0
lo0	16896	::1		20830	0	20867	0	0

Remote dump is not supported over a VLAN. Also, VLAN logical devices cannot be used to create a Cisco Systems EtherChannel.

8.25 AIX Web browser support (5.2.0)

AIX 5L Version 5.2, has two supported versions of the Netscape Web browser, 4.79 and 7.0. Netscape Communicator Version 4.79 is available on the AIX 5L Expansion Pack and is normally pre-installed. AIX Netscape 7.0 is only available for download from the IBM AIX Web browsers home page at the following URL:

<http://www.ibm.com/servers/aix/browsers/index.html>

Netscape Communicator Version 4.79 is packaged into the following filesets and can be installed with **installp**, SMIT, or the Web-based System Manager. The Netscape.help and Netscape.msg fileset names are language specific. You must replace XX_XX with your locale (for example, ja_JP).

- ▶ Netscape.communicator.us
- ▶ Netscape.communicator.com
- ▶ Netscape.help.XX_XX.communicator.rte
- ▶ Netscape.msg.XX_XX.communicator.rte

AIX Netscape 7.0 is based on the open-source Mozilla 1.0.1 Web browser. It features new browser technology including the new Gecko layout engine, HTML 4.0, Extended Mark-up Language (XML) 1.0, XML-based User Interface Language (XUL), Cascading Style Sheets (CSS), Document Object Model (DOM), Resource Description Framework (RDF), JavaScript 1.5, and the Open JVM Integration (OJI) of AIX Java. AIX Netscape 7.0 does not include the AOL Instant Messaging Client.

AIX Netscape 7.0 is packaged into the following filesets and can be installed with **installp**, SMIT, or the Web-based System Manager. The Netscape.msg fileset names are language specific. You must replace XX_XX with your locale (for example, ja_JP).

- ▶ Netscape.base.rte

- Netscape.msg.XX_XX.base.rte

Netscape 7.0 has the following software prerequisites:

- Required LPPs (Licensed Product Packages): rpm.rte 3.0.5.20
- Required RPMs (Red Hat Package Manager):
 - glib-1.2.10-2
 - gtk+-1.2.10-3

The RPMs for glib and gtk+ can be downloaded from the AIX Toolbox for Linux Applications home page located at the following URL:

<http://www.ibm.com/servers/aix/products/aixos/linux/>

Install the glib and gtk+ RPM packages by running the following **rpm** commands:

```
# rpm -i glib-1.2.10-2.aix4.3.ppc.rpm gtkplus-1.2.10-3.aix4.3.ppc.rpm
# rpm -q glib gtk+
glib-1.2.10-2
gtk+-1.2.10-3
```

After the prerequisites are installed, install the Netscape.base.rte and language-specific filesets using the following commands, SMIT, or Web-base System Manager. You must specify the device or directory where the software LPPs are located in your environment. Replace the *LPPSOURCE* tag in the following commands with the correct location.

```
# installp -acYgXd LPPSOURCE Netscape.base.rte
```

To start the AIX Netscape 7.0 browser either run **/usr/bin/netcape7** or **/usr/netcape/base/netcape**. See Figure 8-32 on page 568 for an image of the AIX Netscape 7.0 browser.

To configure the Netscape Java plug-in or to enable your browser for foreign languages, refer to the readme or readme.html file in **/usr/netcape/base**.



Figure 8-32 AIX Netscape 7 Web browser

Security, authentication, and authorization

This chapter is dedicated to the latest security topics as they apply to AIX 5L. Topics include, but are not limited to:

- ▶ IBM Directory Server
- ▶ NIS and NIS+
- ▶ Public Key Infrastructure
- ▶ CAPP and EAL+
- ▶ Tivoli
- ▶ IP security
- ▶ Secure rcmds

9.1 Java security enhancements (5.1.0)

In AIX 5L Version 5.1, a Java security enhancement has been made, providing several new APIs. These APIs are used by the Tivoli Security Toolkit. The new APIs allow you to develop more secure Java applications and are provided with the following new Java enhancements:

- ▶ Certificate Management Protocol (CMP)
- ▶ Java Cryptography Extension (JCE)
- ▶ Java Secure Sockets Extension (JSSE)
- ▶ Public-Key Cryptography Standards (PKCS)

The Java enhancements are provided in 32-bit and 64-bit versions, as provided in Table 9-1 and discussed in the following sections.

Table 9-1 Java enhancements versus fileset

Java security enhancements	32-bit filesets	64-bit filesets
Certificate Management Protocol	Java130.cmp-us	Java130_64.cmp-us
Java Cryptography Extension	Java130.jce-us	Java130_64.jce-us
Java Secure Sockets Extension	Java130.jsse-us	Java130_64.jsse-us
Public-Key Cryptography Standards	Java130.pkcs-us	Java130_64.pkcs-us

9.1.1 Certificate Management Protocol

Certificate Management Protocol (CMP) provides support to online interactions between Public Key Infrastructure (PKI) components. For a full description of CMP, refer to RFC2510 and 2511 for CRMF. These RFCs are available at:

<http://www.ietf.org/rfc.html>

9.1.2 Java Cryptography Extension

Java Cryptography Extension (JCE) provides a framework and implementations for encryption and key handling. For more information about JCE, visit:

<http://java.sun.com/products/jce>

9.1.3 Java Secure Sockets Extension

Java Secure Sockets Extension (JSSE) enables secure Internet communications. It provides a Java version of Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols.

For more information about JSSE, visit:

<http://java.sun.com/products/jsse>

9.1.4 Public-Key Cryptography Standards

IBM Public-Key Cryptography Standards (PKCS) implementation supports the following RSA standards: PKCS #1, #3, #5, #6, #7, #8, #9, #10, and #12. For more information about PKCS, go to:

<http://www.rsasecurity.com/rsalabs/pkcs/index.html>

9.2 User and group integration

In previous AIX releases, DCE and NIS were supported as alternate authentication mechanisms. AIX Version 4.3.3 added LDAP support and the initial support for specifying a loadable module as an argument for the user/group managing commands, such as **mkuser**, **lsuser**, and **rmuser**. This was only generally documented in the `/usr/lpp/bos/README` file. AIX 5L now offers a general mechanism to separate the identification and authentication of users and groups, and defines an application programming interface (API) that specifies what function entry points a module has to make available to be able to work as an identification or authentication method. This allows for more sophisticated customized login methods beyond what is provided by the standard ones based on `/etc/passwd` or DCE.

9.2.1 Existing authentication methods

The standard AIX authentication method is a variant of the regular UNIX shadow password-based implementation, meaning that the information about groups and their members is stored in the `/etc/group` file, information about users is stored in the `/etc/passwd` file (with the exception of the encrypted passwords), and related information, which is stored in `/etc/security/passwd`. This standard method is only implicitly defined and is therefore referred to by the name files when you have to distinguish it from other methods. Other authentication methods have to be explicitly defined in configuration files, as explained in the following section.

The information stored in the `/etc/group` and `/etc/passwd` files is called the basic attributes, while the information in the files in the `/etc/security` directory is called the extended attributes. The files in the `/etc/security` directory are AIX-specific files, such as the `/etc/security/user.roles`, which defines which roles a user can take. All the regular AIX commands that create groups or users, change their settings, or remove them are working with this set of files. DCE, for instance, is an identification and authentication mechanism (in addition to the standard file

method supported in AIX). This allows DCE users to be locally authenticated on an AIX system by specifying their DCE identity and password. For user and group management, you have to use the DCE-specific commands; you cannot use the **mkuser** command, for example, to create a DCE user.

The setup for using this alternate authentication involves several steps. DCE uses a loadable binary module named `/usr/lib/security/DCE`. This module belongs to the `dce.client.core.rte.security` fileset. It handles the communication between user, local AIX commands, and the DCE servers. You can specify the full path to this module as a stanza with a freely chosen name as the value for the `program` attribute in the `/usr/lib/security/methods.cfg` file. If you choose the name `DCE`, the stanza appears as follows:

```
DCE:
    program = /usr/lib/security/DCE
```

Because there was no clear separation between user identification and authentication before AIX 5L, the name of this stanza is used for two different purposes:

- ▶ As a value for the `registry` attribute in the `/etc/security/user` file for either single specific users or in the default stanza. This informs AIX that this user is not locally managed, but managed by a remote mechanism.
- ▶ To enable authentication using DCE, override the value of the `SYSTEM` attribute, for example, with the following statement (use of the `auth1` and `auth2` attributes are no longer supported):

```
SYSTEM = "DCE OR DCE[UNAVAIL] AND compat"
```

When a user tries to log in to an AIX system with this setting for a user ID, the user ID and password are automatically handed over to the loadable module specified as the value of the `program` attribute of the `DCE` stanza in `/usr/lib/security/methods.cfg`. This module checks with the DCE servers to see if the user ID and password combination is valid. If it is, the user is authenticated locally in the AIX system and obtains DCE credentials. If this fails due to the unavailability of DCE, not because of a wrong password, the next step is to check if this user ID and password combination is a locally valid one. If it is, the user is authenticated locally, but has no DCE credentials. If it fails, the user receives the message that either a wrong user ID or a wrong password was used. There is a defined grammar that specifies the order of authentication modules to try, and what actions to take if one of them fails or is unavailable.

If you set the `registry` attribute to `DCE` to indicate that the DCE loadable module is responsible for managing the user IDs, and use the **lsuser** command to see the attributes for a specific user, you will miss some of the attributes, such as `unsuccessful_login_count` or `roles`. Some attributes are not even listed and some of them are listed but without their values. If you want to see or reset the value for

the unsuccessful_login_count of a user, you have to temporarily switch the registry attribute back to files. Starting with AIX Version 4.3.3, several user and group managing commands now support an optional -R flag, which specifies the loadable module used for accessing the user and group attributes.

The commands supporting the -R flag are:

- ▶ **chfn**
- ▶ **chgroup**
- ▶ **chgrpmem**
- ▶ **chsh**
- ▶ **chuser**
- ▶ **lsgroup**
- ▶ **lsuser**
- ▶ **mkgroup**
- ▶ **mkuser**
- ▶ **passwd**
- ▶ **rmgroup**
- ▶ **rmuser**

9.2.2 Identification and authentication architecture

In AIX 5L, support for loadable identification and authentication modules is now fully documented and enhanced, in comparison to the feature already available with AIX Version 4.3.3. The tasks of user identification and user authentication are now clearly separated and can be executed by two different loadable modules.

User identification comprises all the necessary information about what user IDs exist and what the attributes for these user IDs are. This information must be consistent, so some kind of database must be used. This database can be flat file based, such as the regular /etc/passwd mechanism, or it can be a relational database, such as DB2, as in the case of the IBM LDAP implementation.

User authentication, on the other hand, is a transitory process where a user claims to have a certain identity and the system has to check if this is true or not. For this process, the system requires a unique piece of information about this user (usually a password). When the user authenticates, the system challenges her by requesting that she type in her password. The user's response is then compared to the stored unique piece of information and, depending on the outcome of this comparison, the request is accepted or denied. This information,

which uniquely identifies a user, must also be stored permanently, but it does not necessarily have to be in the same database where the user identification is stored. With this separation of identification and authentication, and the definition of an API, the architecture in AIX exists to support authentication methods that are far more sophisticated than the usual password-based mechanism.

AIX 5L now supports loadable modules that are either responsible for identification, for authentication, or both (as already supported in the past). For a fully supported login process, you need both identification and authentication. You can use either one loadable module, which supports both (as in the past), or you can specify one loadable module, which is responsible for the identification part, and another that is responsible for authentication. Such a combination of two modules is called a compound module.

To support this new feature, the stanzas in the `/usr/lib/security/methods.cfg` file now accept the attributes `domain` and `option` in addition to the already supported `program` and `program_64` attributes. With the optional `domain` attribute, you can specify an arbitrary text string that is passed as is to the loadable module. The module can use this string for whatever purposes it likes, but usually it is used to distinguish between several supported domains. The `options` attribute also takes an arbitrary text string, consisting of comma-separated values or name/value pairs, which is then passed to the loadable module as is. There are some predefined values that are interpreted by the AIX system itself. You can specify either `authonly` or `dbonly` to indicate that this module is only responsible for the authentication or the identification part. To connect a single purpose module with a specific module for the complementary part of the identification and authentication process, you can use the `db=module` or `auth=module` options.

For example, suppose you want to configure a system to use LDAP for user identification and DCE for user authentication. You have to create, at minimum, two stanzas in the `/usr/lib/security/methods.cfg` file that specify these two programs:

```
DCE:
    program = /usr/lib/security/DCE
    options = authonly

LDAP:
    program = /usr/lib/security/LDAP
    options = auth=DCE
```

With this setting you can, for example, specify LDAP as the value for the registry attribute. For identification purposes, the LDAP load module would be used and as soon as authentication is needed, the module specified in the DCE stanza would be used. You can create the same effect with the following three stanzas:

```
DCE:
```

```
program = /usr/lib/security/DCE
options = authonly
```

LDAP:

```
program = /usr/lib/security/LDAP
```

LDAPDCE:

```
options = auth=DCE,db=LDAP
```

In this case, you would specify LDAPDCE as the value of the registry attribute. This would allow for other possible authentication modules to be used in conjunction with LDAP identification. Stanza names can only be used in other stanzas if they have been previously defined.

In AIX 5L, programming interfaces have been documented that describe what function calls a loadable module has to support if it wants to handle the identification part or the authentication part. There are also a couple of support and administrative function calls that handle the internal table that tracks pointers to all available authentication and identification modules that must be opened and closed.

If you are using user or group accounting commands, such as **lsuser** without using the **-R** flag, information from all defined identification load modules is displayed. Therefore, a user ID may be listed twice if it is defined for two modules. The displayed attributes can also be different, because not all attributes have to be supported by all modules. Values for attributes defined for more than one module are shown as set for the first loaded module (this is often the implicitly defined standard files module). To avoid confusion, we recommend that you always supply a name for a specific load module using the **-R** flag.

9.2.3 Native Kerberos Version 5 support

AIX 5L includes native Kerberos Version 5 support, which can be used as an authentication loadable module, as described in 9.2.2, “Identification and authentication architecture” on page 573. If you use the Kerberos Version 5 authentication method as the default login method, a user will automatically acquire appropriate credentials after a successful login. This support has to be installed separately and is provided in the following filesets:

```
# lsllpp -L "krb5*"
Fileset                                Level  State  Description
-----
krb5.client.rte                        1.1.0.0  C      Network Authentication Service
                                         Client
krb5.client.samples                    1.1.0.0  C      Network Authentication Service
                                         Samples
krb5.doc.en_US.html                    1.1.0.0  C      Network Auth Service HTML
```

krb5.doc.en_US.pdf	1.1.0.0	C	Documentation - U.S. English Network Auth Service PDF
krb5.msg.en_US.client.rte	1.1.0.0	C	Documentation - U.S. English Network Auth Service Client Msgs - U.S. English
krb5.server.rte	1.1.0.0	C	Network Authentication Service Server
krb5.toolkit.adt	1.1.0.0	C	Network Authentication Service App. Dev. Toolkit

The executables and documentation are installed in the `/usr/krb5` directory; configuration files, logs, and other changing files are in the `/etc/krb5` and `/var/krb5` directories. This avoids any mix-up with an already existing Kerberos installation (for example, from DCE).

The only exceptions are the files and links put into `/usr/sbin`, as shown in the following partial directory listing:

```
# ls -l /usr/sbin/*krb*
lrwxrwxrwx 1 root security 26 Sep 13 08:45 /usr/sbin/config.krb5
-> /usr/krb5/sbin/config.krb5
-r-x----- 1 root security 8119 Aug 23 12:33 /usr/sbin/mkkrb5clnt
-r-x----- 1 root security 8648 Aug 23 12:33 /usr/sbin/mkkrb5srv
-r-x----- 1 root security 13864 Aug 24 22:41 /usr/sbin/mkseckrb5
lrwxrwxrwx 1 root security 25 Sep 13 08:45 /usr/sbin/start.krb5
-> /usr/krb5/sbin/start.krb5
lrwxrwxrwx 1 root security 24 Sep 13 08:45 /usr/sbin/stop.krb5 ->
/usr/krb5/sbin/stop.krb5
lrwxrwxrwx 1 root security 28 Sep 13 08:45
/usr/sbin/unconfig.krb5 -> /usr/krb5/sbin/unconfig.krb5
```

The `configure`, `unconfigure`, `start`, and `stop` scripts are only here for convenience, so you do not have to type the complete path to these commands. The **mkkrb5srv** command sets up a Kerberos Version 5 server and the **mkkrb5clnt** command sets up a Kerberos Version 5 client. Finally, the **mkseckrb5** command migrates existing users from the default authentication method to the Kerberos Version 5 method.

To make this setup work, the **hostname** command should provide a full, qualified host name, as shown in the following line:

```
# hostname
server1.itsc.austin.ibm.com
```

Note: If your `hostname` command only outputs a short name without the domain name, the setup will not work because only a principal for the short name will be created. The request from the client, where a user wants to log in with the Kerberos method, coming over the network will always be the conjunction of the short host name and the domain name, and no principal exists for this situation.

The first step in this setup is to create a Kerberos server. To accomplish this task use the `mkkrb5srv` command, specifying the flags as shown in the following example:

```
# mkkrb5srv -r DG.itsc.austin.ibm.com -s server1.itsc.austin.ibm.com -d
itsc.austin.ibm.com -a admin/admin
```

The flags are used specify a realm with the `-r` flag (which is a free-form string), the server name with the `-s` flag, and a domain with the `-d` flag. If you do not specify an admin principal with the `-a` flag, the default is `admin/admin`. These commands create the `/etc/krb5/krb5.conf` file and some other configuration files in the `/var/krb5/krb5kdc` directory. If these configuration files already exist, they are not modified by this command. Several default principals that manage the Kerberos environment will also be created. The command will also add two entries to the `/etc/inittab` file, as shown in the following example output:

```
krb5kdc:2:once:/usr/krb5/sbin/krb5kdc
kadmind:2:once:/usr/krb5/sbin/kadmind
```

These two daemons are also started by the `mkkrb5srv` command. The `kadmind` daemon is the administration daemon and the `krb5kdc` is the actual Key Distribution Center (KDC) daemon, which is responsible for the creation of the secret keys. During the setup process, you are prompted to provide passwords for various principals. You should make note of them, because they are needed in further steps of this setup.

On any machine where you want to use the Kerberos authentication method, you have to run the `mkkrb5clnt` command with several flags. An example is shown in the following line:

```
# mkkrb5clnt -r DG.itsc.austin.ibm.com -c server1.itsc.austin.ibm.com -s
server1.itsc.austin.ibm.com -d itsc.austin.ibm.com -a admin/admin -A -i files
-K -T
```

The meanings of the `-r`, `-d`, and `-a` flags are the same as described previously for the `mkkrb5srv` command. The `-c` and `-s` flags specify the host where the `kadmind` and the KDC daemon are running. The `-i` flag with the `files` argument specifies the integrated login, and the `-K` flag makes Kerberos the default authentication method. The `-A` flag makes root an administrator for Kerberos on this machine.

Finally, the `-T` flag requests a Ticket-Granting Ticket (TGT) from the server. This creates a keytab file in the `/var/krb5/security/keytab` directory and the `/etc/krb5/krb5.conf` configuration file. The last step is omitted if you create the client on the same machine you created the server on, because this file already exists in this case. The command also creates the following two entries in the `/usr/lib/security/methods.cfg` file:

```
KRB5:
    program = /usr/lib/security/KRB5

KRB5files:
    options = db=BUILTIN,auth=KRB5
```

The last entry is used to modify the `SYSTEM` attribute of the default stanza in the `/etc/security/user` file to read:

```
default:
    SYSTEM = "KRB5files OR compat"
```

With this setting, Kerberos is tried, as a first step, as the authentication method; if this fails, the regular AIX method is tried.

After being authenticated with the `/usr/krb5/bin/kinit` command, root can create users residing in the `KRB5files` domain. The following example commands can be used to create a user `krb5user` and to set an initial password (it is recommended that you use a more secure password):

```
# mkuser -R KRB5files krb5user
# passwd -R KRB5files krb5user
```

The output of the `lsuser` command shows all the Kerberos attributes, beginning with `krb5_`, defined for this user in addition to the regular AIX user attributes:

```
# lsuser -R KRB5files krb5user
krb5user id=202 pgrp=staff groups=staff home=/home/krb5user shell=/usr/bin/ksh
login=true su=true rlogin=true daemon=true admin=false sugroups=ALL admgroups=
tpath=nosak ttys=ALL expires=0 auth1=SYSTEM auth2=NONE umask=22
registry=KRB5files SYSTEM=KRB5files or compat logintimes= loginretries=0
pwdwarntime=0 account_locked=false minage=0 maxage=0 maxexpired=-1 minalpha=0
minother=0 mindiff=0 maxrepeats=8 minlen=0 histexpire=0 histsize=0 pwdchecks=
dictionlist= fsize=2097151 cpu=-1 data=262144 stack=65536 core=2097151
rss=65536 nofiles=2000 time_last_login=0 time_last_unsuccessful_login=0
tty_last_login=/dev/pts/4 host_last_login=server1.itsc.austin.ibm.com
unsuccessful_login_count=0 roles=
krb5_principal=krb5user@DG.itsc.austin.ibm.com
krb5_principal_name=krb5user@DG.itsc.austin.ibm.com
krb5_realm=DG.itsc.austin.ibm.com maxage=0 expires=0
krb5_last_pwd_change=968878232 admchk=false krb5_attributes=requires_preauth
krb5_mod_name=krb5user@DG.itsc.austin.ibm.com krb5_mod_date=968878232
krb5_kvno=4 krb5_mkvno=0 krb5_max_renewable_life=604800 time_last_login=0
```



```
time_last_unsuccessful_login=0 unsuccessful_login_count=0
krb5_names=krb5user:server1.itsc.austin.ibm.com
```

The new user can **telnet** to the client machine and log in with the password just set up. After a successful login, the user environment has the following settings:

```
AUTHSTATE=KRB5files
KRB5CCNAME=FILE:/var/krb5/security/creds/krb5cc_krb5user@DG.itsc.austin.ibm.com
_202
```

These settings show that the user is authenticated using the KRB5files method and the path to the credentials file.

With the help of the **mkseckrb5** command, you can migrate a user existing in the files domain to the KRB5files domain. The following lines show an example session for a user **krb5eins**:

```
# mkseckrb5 krb5eins
Please enter the admin principal name: admin/admin
Enter password:
Importing krb5eins
Enter password for principal "krb5eins@DG.itsc.austin.ibm.com":
Re-enter password for principal "krb5eins@DG.itsc.austin.ibm.com":
```

If you do not want to enter the password twice for the migrated user, you can use the **-r** flag, which creates a random password for you. You can then use the **passwd** command to set a password for this user.

9.3 Concurrent groups enhancement (5.1.0)

In AIX 5L Version 5.1, the number of concurrent user groups has been enhanced to allow up to 64 groups per process. In previous versions of AIX, the system allowed a maximum of 32 concurrent group memberships. Applications must invoke the `sysconf(_SC_MAX_GROUPS)` call to determine the actual value. POSIX standards may enforce that `MAX_GROUPS` is smaller than the current system implementation; therefore, invoke `sysconf()` with the actual value the system is using.

9.4 IBM SecureWay Directory Version 3.2

Version 3.2 of the IBM SecureWay Directory implements the Lightweight Directory Access Protocol (LDAP) Version 3.2 and is offered with the AIX operating system product at no additional charge.

LDAP consists of two major functions, the client and the server.

9.4.1 LDAP overview

The IBM SecureWay Directory Version 3.2 consists of the following components:

- ▶ slapd: The server executable
- ▶ Command line import/export utilities
- ▶ A server administration tool with a Web-browser based interface for configuration and administration of the directory
- ▶ A Java-based directory content management tool and online user guide
- ▶ Online administration help
- ▶ Online LDAP programming references (C, server plug-ins, and Java/JNDI)
- ▶ SecureWay Directory Client Software Development Kit (SDK) that includes C runtime libraries and Java classes

The product includes a Lightweight Directory Access Protocol (LDAP) Version 3 server that supports IETF LDAPv3 (RFC2251) protocol, schema, RootDSE, UTF-8, referrals, Simple Authentication and Security Layer (SASL) authentication mechanism, and related specifications. In addition, it includes support for Secure Socket Layer (SSL), replication, access control, client certificate authentication, CRAM MD5 authentication, change log, password encryption, server plug-ins, enhanced search capability for compound Relative Distinguish Name (RDN), Web-based Server Administration, LDAPv3 schema definitions, IBM common schema definitions, schema migration, and performance improvements.

With over 18 major product enhancements, Version 3.2 of the IBM SecureWay Directory represents one of the most significant updates of the product to date. Some of the more significant enhancements and new functions and features include:

- ▶ Fine-grain access control - Attribute level ACLs

The IBM SecureWay Directory now allows the management of access down to the individual attribute level. A directory administrator may now control who may see individual attributes for each entry within the directory. This allows access to be managed on an individual attribute level, which gives a much finer control. Fine-grain access control is often used when specific attributes need to be managed by an entry owner and other entry attributes are managed by the directory administrator.

- ▶ Unlimited connections - Improved server threading model

The IBM SecureWay Directory has proven to be a performance leader. To sustain and further enhance the striking performance of the product, the threading model for the directory has been improved. The IBM SecureWay Directory will now utilize thread pools, thus reducing the number of threads

utilized when many clients connect to the server concurrently. This change will allow a much larger number of clients to connect to a server, which in turn reduces the number of servers required in a given LDAP environment.

- ▶ Support for Kerberos Version 5 (server and client, including C and JNDI) - GSSAPI

The IBM SecureWay Directory now supports authentication utilizing Kerberos Version 5. Kerberos Version 5 has become an important authentication method. Supporting Kerberos Version 5 authentication methods improves the ability of the directory to provide a single authentication method across the enterprise.

The SecureWay Directory Client SDK includes a Java-based Directory Management Tool, APIs to locate LDAP servers that are published in DNS, client-side caching for the Java-based JNDI interface, as well as other JNDI enhancements.

LDAP is a new technology that is rapidly evolving. IBM is committed to deliver the latest LDAP technology achievements in the robust high-performance LDAP server implementation of the IBM SecureWay Directory product. Version 3.2 of the IBM SecureWay Directory not only keeps pace with the industry, but provides many industry-leading innovations, as documented by the list of improvements given below:

- ▶ Performance improvements through Table Reduction (for Fast Server Startup)
- ▶ Componentization of install
- ▶ Integrated Install for selection of prerequisite software, separate server versus Client Install
- ▶ WebAdmin and Directory Management Tool (DMT) GUI
- ▶ Separation of Configuration versus Data Management Tasks
- ▶ Enhancements to Directory Management functions supported by DMT
- ▶ Improved panel helps, messages, error logging, and reporting
- ▶ Exploitation of Java 1.2
- ▶ Replication enhancements
- ▶ Event notification (server and client support)
- ▶ Security auditing
- ▶ Limited transaction support
- ▶ Automatic LDAP server selection for C and JNDI client
- ▶ Support for latest DB/2 releases - UDB 6.1 and UDB 7.1

- ▶ GSKit 4.0 exploitation
- ▶ Backup/restore support
- ▶ Sample Java beans illustrating JNDI usage

On AIX, the new IBM SecureWay Directory version translates messages for Group 1 national languages, including Brazilian Portuguese, French, German, Italian, Spanish, Japanese, Korean, Simplified Chinese, Traditional Chinese, Czech, Polish, Hungarian, Russian, Catalan, and Slovakian.

The directory provides scalability by storing information in the IBM DB2 Universal Database (UDB). DB2 is packaged with the directory product, but you may only use the DB2 component in association with your licensed use of the SecureWay Directory.

IBM SecureWay Directory is designed from the ground up to be a standards-based, reliable, secure, high-performing enterprise directory that can scale as your directory usage grows. For further information on the IBM SecureWay Directory, please refer to the URL:

<http://www-4.ibm.com/software/network/directory>

9.5 IBM Directory Server Version 4.1 (5.2.0)

AIX 5L Version 5.2 now includes the IBM Directory Server Version 4.1. IBM Directory Server Version 4.1 provides a powerful Lightweight Directory Access Protocol (LDAP) server that uses the IBM DB2 Universal Database Version 7.2 engine for reliability.

The IBM Directory Server is an integral part of the new directory enablement features announced in AIX 5L Version 5.2. AIX supports a Certificate Authentication Service with Public Key Infrastructure (PKI), that stores PKI certificates in LDAP. The AIX System V printing subsystem is now directory enabled, allowing printer and print queue configuration to be stored in LDAP. AIX supports LDAP authentication and storage of user and group security attributes into LDAP. Network information services (NIS) maps can now be stored and accessed in LDAP. For more information about the IBM Directory Server V4.1 integration with AIX 5L Version 5.2, refer to the AIX 5L Version 5.2 system documentation or the chapters in this document.

The IBM Directory Server without SSL support is packaged on the AIX 5L Version 5.2 product media. The IBM Directory Server with secure socket layer (SSL) support is included on the expansion pack media.

For more information about IBM Directory Server Version 4.1, refer to its documentation or the IBM Directory home page at the following URL:

<http://www-3.ibm.com/software/network/directory/server/>

9.5.1 LDAP 64-bit client and C API (5.2.0)

AIX 5L Version 5.2 includes a 64-bit LDAP client and C application programming interface (API). This release does not support SSL or the Network Authentication Services. NAS is the native Kerberos and GSSAPI library shipped with Version 5.2.

9.6 LDAP name resolution enhancement

The Lightweight Directory Access Protocol (LDAP) is an open industry standard that defines a method for accessing and updating information in a directory.

Prior to AIX 5L, the name resolver routines only resolve names using the Domain Name System (DNS) hierarchical naming function, through the Network Information Services (NIS and NIS+), or by the use of the local /etc/hosts file.

AIX 5L enhances the name resolver routines to optionally utilize the information stored in an LDAP server hosts database to accomplish name resolution.

In order to implement LDAP name resolution support in AIX 5L, some extensions to the LDAP server schema are indispensable. The relevant new object class and the related attributes are described in 9.6.1, “IBM SecureWay Directory schema for LDAP name resolution” on page 583. A new AIX command helps to migrate existing local /etc/hosts information to the LDAP server hosts database. More information about this command and the related LDAP Data Interchange Format file is given in 9.6.2, “LDIF file for LDAP host database” on page 585. Section 9.6.3, “LDAP configuration file for local resolver subroutines” on page 586, explains the integration of the LDAP name resolution support with the other, more traditional sources for name resolution in the AIX network subsystem environment. For a quick start and for experienced administrators, a brief outline of the procedures necessary to configure an LDAP-based name resolution is provided in 9.6.4, “LDAP-based name resolution configuration” on page 588. Finally, 9.6.5, “Performance and limitations” on page 589, covers performance aspects and limitations of the LDAP-based name resolution.

9.6.1 IBM SecureWay Directory schema for LDAP name resolution

An LDAP directory entry describes an object. An object class is a general description, sometimes called a template, of an object as opposed to the

description of a particular object. For instance, the object class person has a surname attribute, whereas the object describing John Smith has a surname attribute with the value Smith. The object classes that a directory server can store and the attributes they contain are described by schema. Schema define what object classes are allowed where in the directory, what attributes they must contain, what attributes are optional, and the syntax of each attribute. More generically, one can say that an LDAP schema defines the rules for ordering data within the directory structure.

In order to support LDAP name resolution, the new object class `ibm-HostTable` was introduced to the IBM SecureWay Directory schema. IBM SecureWay Directory designates IBM's implementation of the LDAP server and client functionality, and is included in the AIX operating system product at no additional charge. The new `ibm-HostTable` object class can be used to store the name-to-Internet address mapping information for every host on a given network.

The `ibm-HostTable` object class is defined as follows:

Object Class name: `ibm-HostTable`
Description: Host Table entry which has a collection of hostname to IP address mappings.
OID: TBD
RDN: `ipAddress`
Superior object class: `top`
Required Attributes: `host`, `ipAddress`
Optional Attributes: `ibm-hostAlias`, `ipAddressType`, `description`

The attribute definitions are:

Attribute Name: `ipAddress`
Description: IP Address of the hostname in the Host Table
OID: TBD
Syntax: `caseIgnoreString`
Length: 256
Single Valued: Yes
Attribute Name: `ibm-hostAlias`
Description: Alias of the hostname in the Host Table
OID: TBD
Syntax: `caseIgnoreString`
Length: 256
Single Valued: Multi-valued
Attribute Name: `ipAddressType`
Description: Address Family of the IP Address (1=IPv4, 2=IPv6)
OID: TBD
Syntax: `Integer`
Length: 11
Single Valued: Yes
Attribute Name: `host`
Description: The hostname of a computer system.

OID: 1.13.18.0.2.4.486
Syntax: caseIgnoreString
Length: 256
Single Valued: Multi-valued
Attribute Name: description
Description: Comments that provide a description of a directory object entry.
OID: 2.5.4.13
Syntax: caseIgnoreString
Length: 1024
Single Valued: Multi-valued

Please note that only the three attributes (ipAddress, ibm-hostAlias, and ipAddressType) are new to the IBM SecureWay Directory LDAP implementation. The attributes host and description were previously part of the IBM SecureWay Directory schema.

9.6.2 LDIF file for LDAP host database

When an LDAP directory is loaded for the first time or when many entries have to be changed at once, it is not very convenient to change every single entry on a one-by-one basis. For this purpose, LDAP supports the LDAP Data Interchange Format (LDIF), which can be seen as a convenient, yet necessary, data management mechanism.

The LDIF format is used to convey directory information or a description of a set of changes made to directory entries. An LDIF file consists of a series of records separated by line separators. A record consists of a sequence of lines describing a directory entry or a sequence of lines describing a set of changes to a single directory entry. An LDIF file specifies a set of directory entries or a set of changes to be applied to directory entries, but not both at the same time.

To support the implementation and configuration of LDAP-based name resolution, AIX 5L offers the new **hosts2ldif** command. The **hosts2ldif** command resides in the /usr/bin directory and creates an LDIF file from /etc/hosts or another file that has the same format. With no options, the /etc/hosts file is used to create the /tmp/hosts.ldif LDIF file using cn=hosts as the base distinguished name (base DN). The base DN specifies the starting point for the name resolution database within the directory information tree (DIT) structure of the LDAP server. The LDIF file can be used during the configuration process for the LDAP server to load any existing name resolution information that is stored in /etc/hosts files.

The listing below shows a sample LDAP data interchange format (LDIF) file that needs to be generated by the **hosts2ldif** command:

```
dn: cn=hosts
```

```
objectclass: top
objectclass: container
cn: hosts
dn: ipAddress=127.0.0.1, cn=hosts
host: loopback
ipAddress: 127.0.0.1
objectclass: ibm-HostTable
ipAddressType: 1
ibm-hostAlias: localhost
description: loopback (lo0) name/address
```

```
dn: ipAddress=1.1.1.1, cn=hosts
host: testaix5l
ipAddress: 1.1.1.1
objectclass: ibm-HostTable
ipAddressType: 1
ibm-hostAlias: e-testaix5l
ibm-hostAlias: testaix5l.austin.ibm.com
description: first ethernet interface
```

```
dn: ipAddress=fe80::dead, cn=hosts
host: testaix5l
ipAddress: fe80::dead
objectclass: ibm-HostTable
ipAddressType: 2
ibm-hostAlias: test-1l
ibm-hostAlias: test-1l.austin.ibm.com
description: v6 link level interface
```

The numbers in the value of the `ipAddressType` attribute are defined in RFC1700, where `ipAddressType 1` refers to IP Version 4 and `ipAddressType 2` designates the IP Version 6 protocol.

9.6.3 LDAP configuration file for local resolver subroutines

The process of obtaining an Internet address from a host name is known as **name resolution** and is done by the `gethostbyname` subroutine. The process of translating an Internet address into a host name is known as **reverse name resolution** and is done by the `gethostbyaddr` subroutine. These routines are essentially accessors into a library of name translation routines known as **resolvers**.

Resolver routines on hosts running TCP/IP normally attempt to resolve names using the following sources:

- ▶ BIND/DNS (named)
- ▶ Network Information Services (NIS and NIS+)

► Local /etc/hosts file

Traditionally, the ordering of name resolution services can be specified in the /etc/netsvc.conf file, the /etc/irs.conf file, or the NSORDER environment variable. The settings in the /etc/netsvc.conf configuration file override the settings in the /etc/irs.conf file. The NSORDER environment variable overrides the settings in the /etc/irs.conf and the /etc/netsvc.conf files.

Beginning with AIX 5L, the name resolver routines can optionally utilize the information of an LDAP server database to accomplish name resolution.

An entry in the /etc/irs.conf file is of the following format: map mechanism [option]. If the system administrator specifies hosts as the value for the map parameter, the given entry defines the mechanism for mapping host names to their IP addresses. AIX 5L allows you to configure LDAP as a new value for the mechanism parameter. The ldap parameter value prompts the resolver routines to query an LDAP server. For example, to use an LDAP server to resolve a host name that cannot be found in the /etc/hosts file, you would have to enter the following lines in the /etc/irs.conf file:

```
# Use LDAP server to resolve host names that cannot be found in the
# /etc/hosts file
hosts local continue
hosts ldap
```

The necessary information about the related LDAP server is supplied by the /etc/resolv.ldap file that must be configured for this mechanism to work.

The /etc/netsvc.conf configuration file format was similarly expanded to add support for LDAP-based name resolution. Within the /etc/netsvc.conf file, the ordering of the name resolution mechanism is specified by an entry of the following format: hosts = value [, value]. Beginning with AIX 5L, the keyword *hosts* accepts the new value *ldap*, in addition to the previously known values such as *bind*, *local*, *nis*, and *nis+*. In an analogy to the /etc/irs.conf file entries, the *ldap* value causes the network subsystem to use LDAP services for resolving names, and the necessary information about the related LDAP server is supplied by the /etc/resolv.ldap file, which must be configured to activate this mechanism. For example, to use the LDAP server for resolving names, indicate that it is authoritative, and to use the BIND service as an alternative, enter the following lines in the /etc/netsvc.conf file:

```
# Use LDAP server authoritative for resolving names, and use the BIND
# service if the resolver cannot contact the LDAP
hosts = ldap = auth , bind
```

Finally, the NSORDER environment variable accepts a new keyword (*ldap*) to refer to the LDAP-based name resolution. For example, if you want to

supplement the default name services ordering (bind, nis, or the local /etc/hosts file) with the additional support of an LDAP server, the NSORDER environment variable has to be defined as follows:

```
# export NSORDER=bind,nis,local,ldap
```

Whatever way is chosen to enable the network subsystem to benefit from an LDAP-based name resolution, the related /etc/resolv.ldap configuration file has to be present and appropriately configured. The /etc/resolv.ldap file defines the LDAP server information for local resolver subroutines. If the /etc/resolv.ldap file is not present, the system will rely on the default or alternative name resolution mechanisms defined by the /etc/netsvc.conf file, the /etc/irs.conf files, or the NSORDER environment variable.

The resolv.ldap file contains one ldapserver entry, which is required, and one searchbase entry, which is optional. The ldapserver entry specifies the Internet address of the LDAP server to the resolver subroutines. The entry must take the following format:

```
ldapserver address [ port ]
```

The address parameter specifies the dotted decimal address of the LDAP server. The port parameter is optional; it specifies the port number that the LDAP server is listening on. If you do not specify the port parameter, then it defaults to 389.

The searchbase optional entry specifies the base distinguished name (base DN) of the name resolution database on the LDAP server. This entry must take the following format:

```
searchbase baseDN
```

The baseDN parameter specifies the starting point for the name resolution database on the LDAP server. If you do not define this entry, then the searchbase entry defaults to cn=hosts. For example, to define an LDAP server with an IP address 192.9.201.1, which listens on the port 636, and has a searchbase of cn=hosttab, enter the following lines in the /etc/resolv.ldap file:

```
# LDAP server information for local resolver subroutines
ldapserver 192.9.201.1 636
searchbase cn=hosttab
```

9.6.4 LDAP-based name resolution configuration

Use the following procedure to configure the LDAP server to store name-to-Internet address mapping host information:

1. Add a suffix on the LDAP server. The suffix is the starting point of the hosts database. For example, "cn=hosts". This can be done using the Web-based IBM SecureWay Directory Server Administration tool.

2. Create an LDAP Data Interchange Format (LDIF) file. This can be done manually or with the **hosts2ldif** command, which creates an LDIF file from the `/etc/hosts` file.
3. Import the hosts directory data from the LDIF file on the LDAP server. This can be done with the **ldif2db** command or through the Web-based IBM SecureWay Directory Server Administration Tool.

To configure the client to access the hosts database on the LDAP server, use the following procedure:

1. Create the `/etc/resolv.ldap` file.
2. Change the default name resolution through the `NSORDER` environment variable, the `/etc/netsvc.conf` file, or the `/etc/irs.conf` file.

9.6.5 Performance and limitations

The AIX 5L enhancements of the resolver routines are designed and capable of supporting LDAP-based name resolution for either Version 2 or Version 3 of the Lightweight Directory Access Protocol. But in order to enable LDAP-based name resolution with an LDAP server that uses the protocol Version 2, it is necessary to manually create extensions to the LDAP schema. Refer to 9.6.1, “IBM SecureWay Directory schema for LDAP name resolution” on page 583, for more detailed information about the new and indispensable object class `ibm-HostTable` and the related attributes that were used to extend the LDAP schema of the IBM SecureWay Directory LDAP Version 3 implementation.

Since the resolver can possibly search through additional maps and the timeout for the LDAP search is 30 seconds, there could be some performance degradation in the amount of time it takes to resolve a name. However, if the LDAP server environment is properly designed and implemented to support LDAP-based name resolution, and if, on the client side, the appropriate configurations of the `/etc/netsvc.conf` file, the `/etc/irs.conf` file, or the `NSORDER` environment variable are established, the performance will be of the same order as for the DNS mechanism.

9.7 LDAP security audit plug-in (5.1.0)

Since the default audit function provided by the IBM SecureWay Directory may not be suited for the needs of the AIX security information management, an LDAP security plug-in has been added to AIX 5L Version 5.1.

The LDAP security audit plug-in provides auditing of the LDAP security information server under the framework of the AIX security audit subsystem. The

new LDAP plug-in works independently from the SecureWay Directory audit plug-in. You can decide to invoke either one of them or both of them at the same time.

9.7.1 Implementation

The LDAP security plug-in has been implemented as `/usr/ccs/lib/libsecldapaudit.a`. The result of the plug-in operation is either `AUDIT_OK` or `AUDIT_FAIL`. A logical diagram is shown in Figure 9-1.

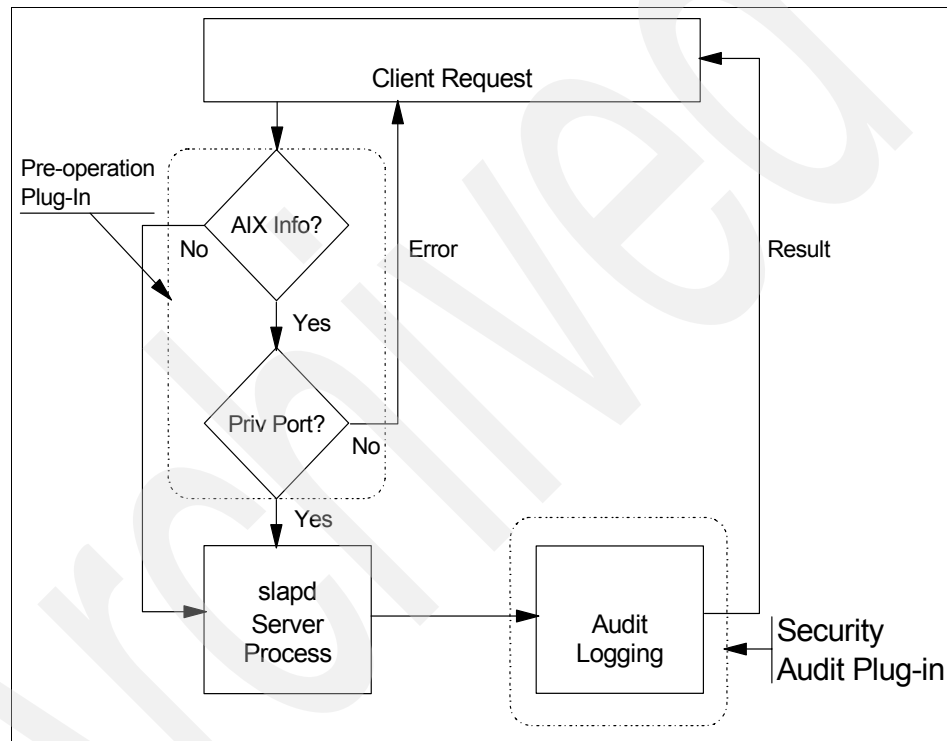


Figure 9-1 Implementation detail of the LDAP security audit plug-in

9.7.2 Configuration files

Due to the LDAP enhancements, the `/etc/security/audit/events` and `/etc/security/audit/config` files have been updated.

Audit events file

The following entries have been added to the `/etc/security/audit/events` file:

- * SecureWay Directory Server

```

*      LDAP_Bind
LDAP_Bind = printf "ConnectID: %d Host: %s Port: %d BindDN: %s"

*      LDAP_Unbind
LDAP_Unbind = printf "ConnectID: %d"

*      LDAP_Add
LDAP_Add = printf "ConnectID: %d Entry: %s"

*      LDAP_Delete
LDAP_Delete = printf "ConnectID: %d Entry: %s"

*      LDAP_Modify
LDAP_Modify = printf "ConnectID: %d Entry: %s"

*      LDAP_Modifydn
LDAP_Modifydn = printf "ConnectID: %d NewEntry: %s OldEntry: %s"

*      LDAP_Search
LDAP_Search = printf "ConnectID: %d Search: %s"

*      LDAP_Compare
LDAP_Compare = printf "ConnectID: %d Compare: %s"

```

Where:

Host	Host address
Port	Client port number
ConnectID	Connect session ID
BindDN	Distinguished name, for example, cn=admin,o=ibm,c=us
Entry	User/group name
Search	Search filter (criteria)
Compare	Object to be compared

Audit config file

The following class definition has been added to the /etc/security/audit/config file:

```

ldapserver = LDAP_Bind,LDAP_Unbind,LDAP_Add,LDAP_Delete,LDAP_Modify,LDAP
_Modifydn,LDAP_Search,LDAP_Compare

```

9.7.3 Audit information

If the audit service is started (**audit start**), you can check to see if the new LDAP security audit plug-in is active:

```
# audit query
auditing on
audit bin manager is process 9094
audit events:
ldapsrv -
LDAP_Bind,LDAP_Unbind,LDAP_Add,LDAP_Delete,LDAP_Modify,LDAP_Modifydn,LDAP_Search,LDAP_Compare
```

9.8 Overall AIX directory integration (5.2.0)

AIX 5L has several subsystems that can store information in an IBM LDAP Directory server. The directory-enabled subsystems are AIX user and group security, network information services (NIS), Public Key Infrastructure (PKI), and printing. In Version 5.2, the subsystem information has been brought together under a common subtree to simplify administration in a directory-enabled environment.

The AIX data subtree, also known as the AIX local data repository, is located at `cn=aixdata` by default. This subtree can be located at the top of the LDAP hierarchy or attached to an existing hierarchy. For example, the DN for an AIX local data repository for a particular department might use a distinguished name (DN) of `cn=aixdata,ou=mydept,o=mycompany.example,c=us`.

The LDAP hierarchy for `mycompany.example`'s AIX directory-enabled subsystems is illustrated in Figure 9-2 on page 593.

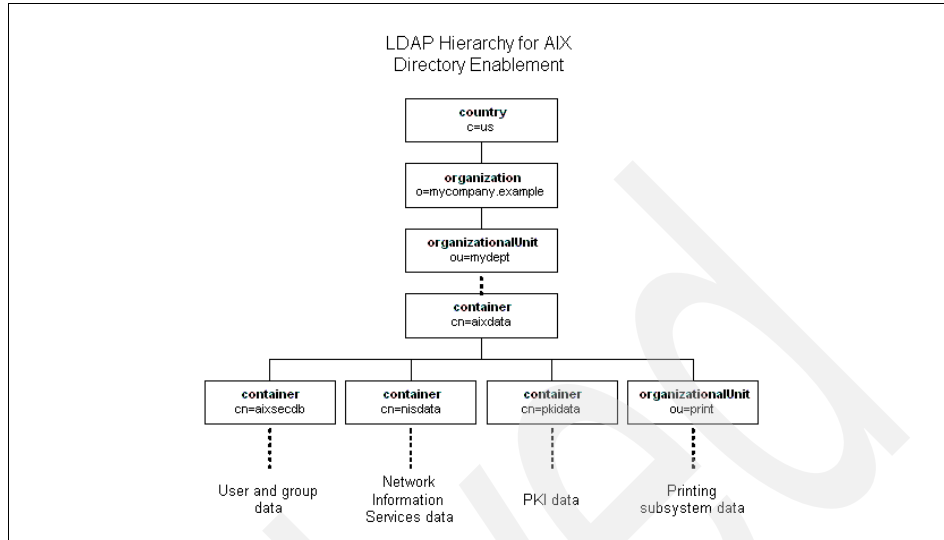


Figure 9-2 LDAP hierarchy for AIX directory-enabled subsystems

The directory enabled printing subsystem allows printer configuration to be stored in an LDAP server. The default location used by the **mkprtldap** command for the printer data is in the RDN `ou=print,cn=aixdata`. For more information on directory-enabled printing, refer to “Directory-enabled printing (5.2.0)” on page 594.

The Public Key Infrastructure (PKI) security subsystem stores certificates for AIX login in the LDAP server. The default RDN used by the **mksecpki** command for the AIX certificate data is `ou=pkidata,cn=aixdata`.

The NIS integration with LDAP allows NIS maps to be imported into an LDAP hierarchy using a schema defined by the experimental RFC2307 specification. After the NIS maps are migrated, AIX 5L Version 5.2 and other RFC2307-compliant platforms can use LDAP instead of NIS to access this data. The default RDN used by the **mksecldap** and **nistoldif** commands is `cn=nisdata,cn=aixdata`.

The AIX security subsystem allows user and group attributes to be stored in LDAP instead of a local file registry. Version 5.2 uses a RFC2307-compliant schema, which allows other platforms to access this data from LDAP. The default RDN used by the **mksecldap** and **sectoldif** commands is `cn=aixsecdb,cn=aixdata`. For more information on the NIS and AIX security integration, refer to 9.10, “AIX security LDAP integration (5.2.0)” on page 599.

9.9 Directory-enabled printing (5.2.0)

In Version 5.2, the AIX System V print subsystem supports storing its printers, print queue, and system information in an LDAP server. Printer configurations can now be maintained centrally for many machines. Several new commands were added to support administration of directory-enabled printers. The names and functions of commands are similar to their non-directory equivalents. The new commands and brief descriptions of their functions follow:

ds1paccept	Accept print queue requests for directory-enabled System V print systems.
ds1paccess	Allow or deny non-directory enabled users and systems access to a print queue for a System V print subsystem.
ds1padmin	Configure directory-enabled print service for a System V print subsystem.
ds1pdisable	Disable print queue requests for a System V print subsystem.
ds1penable	Enable print queue requests for a System V print subsystem.
ds1pprotocol	Configure the remote print protocol of print queue for a System V print subsystem.
ds1preject	Reject print queue requests for directory-enabled System V print systems.
ds1psearch	Search directory for print system objects on a System V print subsystem.

In order to use directory-enabled printing, you must install and enable the AIX System V print subsystem and the LDAP client. Use the following commands, SMIT, or Web-based System Manager to install the `bos.svprint` package. You must specify the device or directory where the AIX LPPs are located in your environment. Replace the *LPPSOURCE* tag in the following commands with the correct location.

```
# installp -acgXYd LPPSOURCE bos.svprint
```

After the System V print subsystem is installed, it must be enabled using the **switch.prt** command. The following example shows how to enable the AIX System V print subsystem using the **switch.prt** command.

```
# switch.prt -s SystemV
SystemV Print Subsystem Started
```

The following commands are for displaying the active print subsystem to verify the change:

```
# switch.prt -d
```



```
#printsubsystem  
SystemV
```

In order to use directory-enabled printing, you must either install a new LDAP server or use an existing server. This section will assume that you are using an existing IBM Directory Server. The directory-enabled printing client and server components are configured using the **mkprtdap** command with the **-c** and **-s** flags, respectively.

The following section describes how a department uses an existing LDAP server to support the directory-enabled print subsystem. The AIX printing information subtree contains all the entries for the directory-enabled printers, printer queues, and system entries. The default distinguished name (DN) for this subtree is **cn=print,cn=aixdata**. The printing information subtree would be for department-related printers and queues only, so it was decided to be located at the DN **cn=print,cn=aixdata,ou=mydept,o=mycompany.example,c=us**. The first level of the printing information tree contains the **ou=print** container. The second level contains the printer, print queue, and system subtrees. The printer subtree, located at **ou=printer,cn=print**, contains entries for each directory-enabled printer. The print queue subtree, located at **ou=printq,cn=print**, contains entries for each directory-enabled print queue. The system subtree, located at **ou=system,cn=print**, contains the printer network entities to allow printing to network printers.

The LDAP hierarchy for the AIX System V directory-enabled printing is illustrated in Figure 9-3 on page 596.

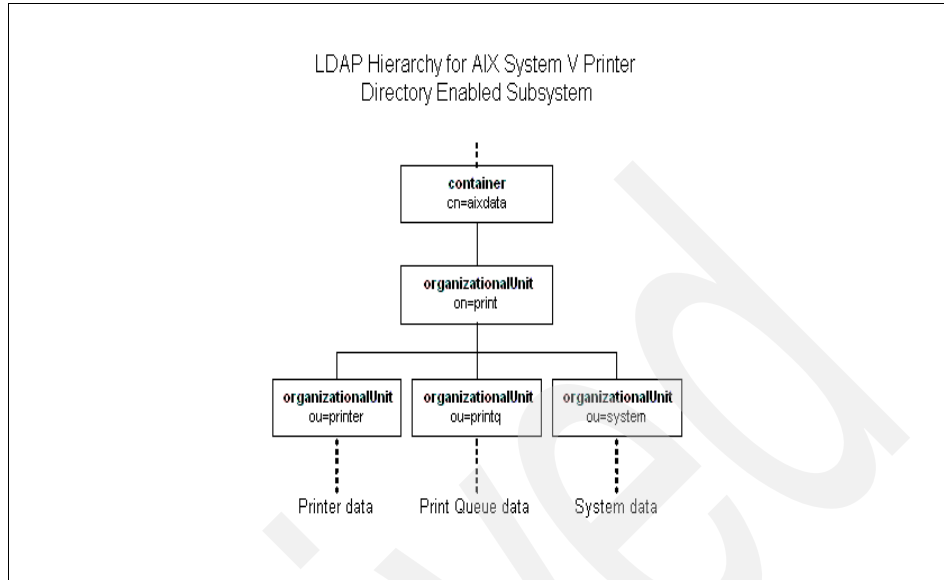


Figure 9-3 LDAP hierarchy for AIX System V directory-enabled printing

You must first run the **mkprtdap** command with the **-s** flag on the LDAP server machine to configure the server components of directory-enabled printing. If the LDAP server is installed but not configured, the **mkprtdap** command will set up the LDAP database and set the administrator's DN. It will create the printing subsystem and the AIX repository tree if necessary. The **mkprtdap** command can also be run on previously configured LDAP servers and it will perform any required configuration.

The following example uses the **mkprtdap** command to configure the LDAP server with the following options. The **-a** and **-p** flags specify the administrator's DN and password for LDAP server administration. The **-w** flag specifies that the password to protect the **ou=print, cn=aixdata** subtree. The **-d** flag specifies the base DN for the AIX local repository. This example will create the new printer repository in **ou=print,cn=aixdata,ou=mydept, o=mycompany.example,c=us**.

```
# mkprtdap -s -a "cn=admin,ou=mydept,o=mycompany.example,c=us" -p mysecret \
-w printsecret -d "cn=aixdata,ou=mydept,o=mycompany.example,c=us"
```

Checking version of IBM Directory

Starting the Server side configuration

Checking DB2 database and Administrator DN/Password configuration

Searching the Directory for existing AIX information subtrees(cn=aixdata objects)

Adding the required Print objects to the Print subtree on the Directory
Server side configuration successful

The Print Bind DN is `ou=print,cn=aixdata,ou=mydept,o=mycompany.example,c=us`. Use this Print Bind DN value when executing the `mkprtlldap` command to configure the client systems

After the server component is configured, the **mkprtlldap** command must be run with `-c` to configure the clients. The following example configures the directory enabled print subsystem with the following options. The `-h` flag specifies the address of the LDAP to connect to. The `-w` flag specifies the password to access the `ou=print,cn=aixdata` subtree. The `-d` flag specifies the print bind DN, which is displayed at the end of the **mkprtlldap** server setup.

```
# mkprtlldap -c -h ldap.mycompany.example -w printsecret \  
-d "ou=print,cn=aixdata,ou=mydept,o=mycompany.example,c=us"  
Starting the Client side configuration  
Checking version of IBM Directory  
Client side configuration successful
```

The client configuration of the **mkprtlldap** command generates two configuration files, `/etc/ldapsvc/server.print` and `/etc/ldapsvc/system.print`. The `server.print` file contains the host name and port of the LDAP server and the printer bind DN. The `system.print` file contains the password required to bind to the LDAP server. The following section shows the client configuration generated by the previous **mkprtlldap** command.

```
# cat /etc/ldapsvc/server.print  
PRINTSERVER=ldap.mycompany.example  
LDAPPORT=389  
PRINTBINDDN=ou=print,cn=aixdata,ou=mydept,o=mycompany.example,c=us  
# cat /etc/ldapsvc/system.print  
PRINTBINDPASSWD=printsecret
```

Now that the printing subsystem directory client is enabled, you can create directory-enabled queues and printers with the **dsldapadmin** command. The following examples create three printers with three different queues, named `printer1`, `printer2`, and `printer3`. The `-l` flag specifies the location of each printer.

```
# dsldapadmin -T "HP LaserJet 6L (Postscript)" -l "3rd floor" -m standard -A mail  
-q printer1 -P printer1 -s netprinter1 -a 9.3.4.10 -t BSD -F continue -I "PS"  
  
# dsldapadmin -T "HP Paint Jet" -l "1st floor" -D "color" -m standard -A mail -q  
printer2 -P printer2 -s netprinter2 -a 9.3.4.11 -t BSD -F continue -I "simple"  
  
# dsldapadmin -T "HP LaserJet 6L (Postscript)" -l "2nd floor" -m standard -A mail  
-q printer3 -P printer3 -s netprinter3 -a 9.3.4.12 -t BSD -F continue -I "PS"
```

You must then use the **dslopenable** command to enable the print queue to accept jobs. Use the **dsaccept** command to enable users or machines access to a printer queue. The following commands enables `printer1` for all users and machines.

```
# dslpenable printer1
# dslpaccept printer1
```

The **dslpsearch** command allows you to search for directory-enabled printers and queues in the LDAP directory. It also allows you to search for printers and queues with specific attributes. For example, you can search for a list of color printers at a specific location. The first example below displays all the print queues and printers defined in the LDAP directory. The second example displays the print queues and printers that are located on the first floor.

```
# dslpsearch -p

cn=printer1,ou=printq,ou=print,cn=aixdata,ou=mydept,o=mycompany.example,c=us =>
cn=printer1,ou=printer,ou=print,cn=aixdata,ou=mydept,o=mycompany.example,c=us
cn=printer2,ou=printq,ou=print,cn=aixdata,ou=mydept,o=mycompany.example,c=us =>
cn=printer2,ou=printer,ou=print,cn=aixdata,ou=mydept,o=mycompany.example,c=us
cn=printer3,ou=printq,ou=print,cn=aixdata,ou=mydept,o=mycompany.example,c=us =>
cn=printer3,ou=printer,ou=print,cn=aixdata,ou=mydept,o=mycompany.example,c=us
# dslpsearch -p -o 'location=1st*'
cn=printer2,ou=printq,ou=print,cn=aixdata,ou=mydept,o=mycompany.example,c=us =>
cn=printer2,ou=printer,ou=print,cn=aixdata,ou=mydept,o=mycompany.example,c=us
```

Web-based System Manager for directory-enabled printing

The Web-based System Manager has been enhanced to support directory-enabled printing. See Figure 9-4 on page 599 for the new printing, overview, and tasks page. There are now tasks to configure the printing directory client and server components and define local and directory printers.

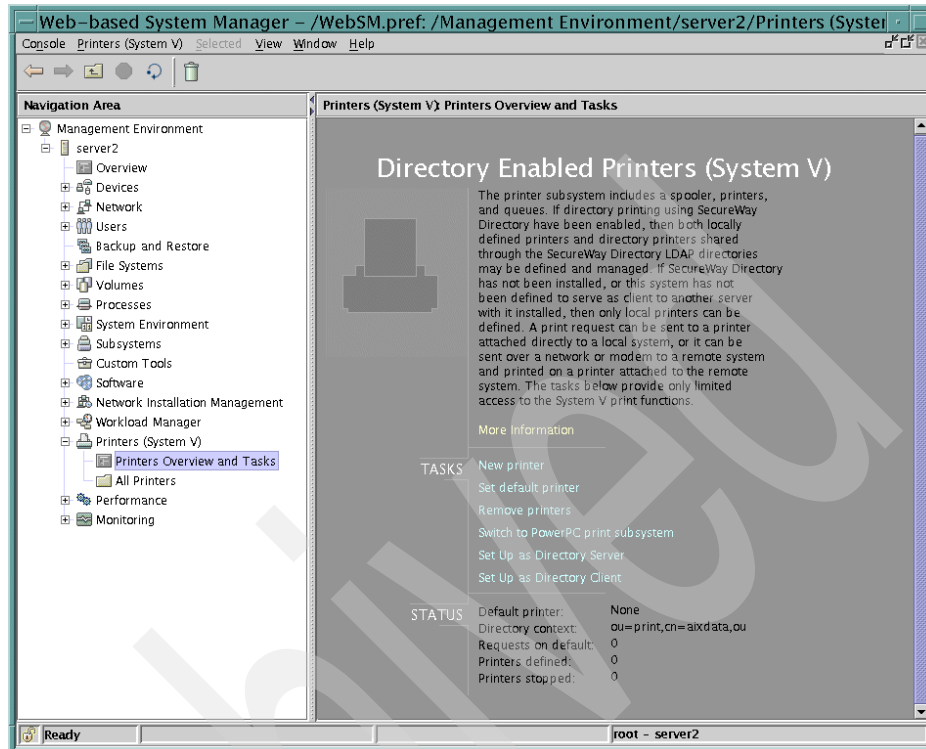


Figure 9-4 Web-based Systems Manager - Directory Enabled Printers

9.10 AIX security LDAP integration (5.2.0)

AIX 5L Version 5.2 now supports the authentication and storage of AIX user and group security attributes in LDAP. This allows centralized security authentication and access to user and group attributes, allowing consistency across clusters of machines.

This integration is implemented in an LDAP loadable authentication module, which is conceptually similar to the Kerberos 5, DCE, and NIS loadable authentication modules. Most of the high-level user and group administration commands, such as **mkuser** and **passwd**, can use the **-R** flag to select the authentication module. For example, to create a new LDAP user *beady*, use the following **mkuser** command:

```
# mkuser -R LDAP SYSTEM=LDAP beady
```

In Version 4.3 and Version 5.1, AIX used a proprietary schema to store the user and group security attributes. In Version 5.2, AIX now supports the following three schema: AIX, RFC2307, and RFC2307AIX.

- AIX** The AIX schema includes the `aixAccount` and `aixAccessGroup` object classes. This schema offers all the AIX user and group attributes. This schema is included to support legacy LDAP installations prior to Version 5.2.
- RFC2307** The RFC2307 schema includes the `posixAccount`, `posixGroup`, and other NIS-related object classes. This experimental RFC defines a schema that allows NIS maps to be imported into LDAP. RFC2307 only defines a subset of the AIX user and group attributes. This schema supports any RFC2307-compliant platforms and AIX 5L Version 5.2.
- RFC2307AIX** The RFC2307AIX schema includes the RFC2307 schema plus the AIX-specific object classes, `aixAuxAccount` and `aixAuxGroup`. The AIX-specific object classes provide attributes to store additional attributes not defined by the RFC2307 standard. The RFC2307AIX schema is the preferred schema for new installations as it supports RFC2307-compliant platforms and the extended attributes for AIX.

The following section describes how a department might set up the a new IBM Directory Server using the RFC2370AIX schema to support AIX and RFC2307-compliant authentication. The AIX local repository is located under the `ou=mydept,o=mycompany.example,c=us` subtree.

The first subtree is used for the AIX security database containing the user and group attributes. The default DN for this subtree is `cn=aixsecdb,cn=aixdata`. This AIX security subtree would be for department users only, so it was decided to locate it at DN

`cn=aixsecdb,cn=aixdata,ou=mydept,o=mycompany.example,c=us`. The first level of the AIX security subtree contains the `cn=aixsecdb` container. The second level contains the `aixuser`, `aixgroup`, and `system` subtrees. The `aixuser` subtree, located at `ou=aixuser,cn=aixsecdb`, contains entries for each user. The `aixgroup` subtree, located at `ou=aixuser,cn=aixsecdb`, contains entries for each group. The `system` subtree, located at `ou=system,cn=aixsecdb`, contains the auxiliary information about the AIX security database.

The second subtree is used to store the NIS maps. The default DN for this subtree is `cn=nisdata,cn=aixdata`. The original NIS maps were for an individual department originally, so it was decided to the locate the NIS data subtree at DN `cn=nisdata,cn=aixdata,ou=mydept,o=mycompany.example,c=us`. The first level of the NIS data subtree contains the `cn=nisdata` container. The second level

contains the hosts, netgroup, networks, protocols, rpc, and services subtrees. These subtrees contain all of the entries for each of the supported NIS maps.

The LDAP hierarchy for the AIX security database and the NIS maps is illustrated in Figure 9-5.

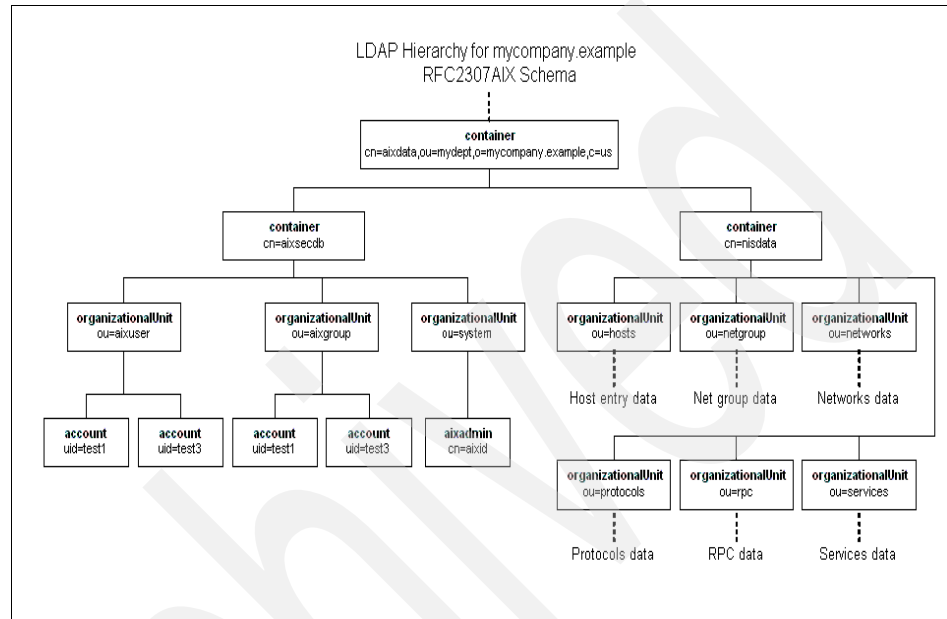


Figure 9-5 LDAP Hierarchy for AIX security database and NIS maps

AIX 5L Version 5.2 allows authentication using the subset of attributes defined by the RFC2307 schema. If the RFC2307 schema was used for Version 5.2 user authentication, certain information and capabilities would be lost. User limits and password rules could not be assigned to individual accounts and login information would not be available. The following list shows some of the AIX extended attributes that would not be supported with RFC2307 schema.

- ▶ User limits (ulimits)
 - coreSizeLimit
 - cPuSize
 - dataSegSize
 - fileSizeLimit
 - openFileLimit
- ▶ Password rules
 - passwordExpireTime
 - passwordHistSize
 - passwordMinDiffChars
 - passwordMinAlphaChars

- passwordMaxRepeatedChars
- ▶ Login information
 - maxFailedLogins
 - hostLastLogin
 - hostLastUnsuccessfulLogin
- ▶ Host login restrictions
 - hostsAllowedLogin
 - hostsDeniedLogin

When setting up an directory enabled authentication, the preferred schema is the RFC2307AIX schema. This will allow you the most flexibility when supporting AIX and RFC2307 compliant platforms.

RFC2307 also defines other object classes to contain the NIS map data. For more information about using LDAP for NIS data see Section 9.12, “NIS/NIS+ integration into LDAP (5.2.0)” on page 611.

For more information about RFC2307 - An Approach for Using LDAP as a Network Information Service, refer to the IETF Web site at the following URL.

<http://www.ietf.org>

IBM Directory Server configuration

In order to use LDAP for AIX authentication, you must either install a new LDAP server or use an existing server. The LDAP client and server security components are configured using the **mksecldap** command with the -c and -s flags, respectively.

You must install the IBM Directory Server Version 4.1 product to store the user, group, and NIS map attributes. Use the following commands, SMIT, or Web-based System Manager to install the following Licensed Product Packages (LPPs). IBM Directory Server uses DB2 as the backend datastore and will automatically install DB2. If you need more information about installation and configuration of this product, install the detailed documentation supplied with the product. The IBM Directory Server documentation is located in the ldap.html.en_US.* filesets. You must specify the device or directory where the software LPPs are located in your environment. Replace the *LPPSOURCE* tag in the following commands with the correct location:

```
installp -acgXd LPPSOURCE ldap.server ldap.client ldap.html.en_US
```

After the IBM Directory Server is installed, you can use the **mksecldap** command with the -s flag to configure the LDAP server to support authentication. If the LDAP server is installed but not configured, the **mksecldap** command will set up the LDAP database and set the administrator's DN. It will then make the required schema modifications to support the AIX, RFC2307, or RFC2307AIX schema.

Unless specifically disabled, **mksecldap** will load all the local user and group attributes into the LDAP security repository using the **sectoidif** and **ldif2db** commands. The **mksecldap** command can also be run on previously configured LDAP servers, and will perform any required configuration to support the LDAP schema.

The following example uses the **mksecldap** command to configure the LDAP server with the following options. The **-a** and **-p** flags specify the administrator's DN and password for LDAP server administration. The **-S** flag specifies that the server will be set up with the RFC2307AIX schema. The **-d** flag specifies the base DN for the AIX local repository. This example will create the new security repository in the **cn=aixsecdb,cn=admin,ou=mydept,o=mycompany.example,c=us** subtree. The **-u NONE** flag specifies that the user and group information should not be loaded into LDAP at this stage.

```
# mksecldap -s -a "cn=admin,ou=mydept,o=mycompany.example,c=us" -p mysecret -S
RFC2307AIX -d "cn=aixdata,ou=mydept,o=mycompany.example,c=us" -u NONE
Creating the directory DB2 default database.
This operation may take a few minutes.
```

Configuring the database.

Creating database instance: ldapdb2.

Created database instance: ldapdb2.

Starting database manager for instance: ldapdb2.

Started database manager for instance: ldapdb2.

Creating database: ldapdb2.

Created database: ldapdb2.

Updating configuration for database: ldapdb2.

Updated configuration for database: ldapdb2.

Completed configuration of the database.

IBM Directory Server Configuration complete.

Password for administrator DN **cn=admin,ou=mydept,o=mycompany.example,c=us** has been set.

IBM Directory Server Configuration complete.

Plugin of type EXTENDEDOP is successfully loaded from libevent.a.

Plugin of type EXTENDEDOP is successfully loaded from libtranext.a.

Plugin of type PREOPERATION is successfully loaded from libDSP.a.

Plugin of type EXTENDEDOP is successfully loaded from libevent.a.

Plugin of type EXTENDEDOP is successfully loaded from libtranext.a.

Plugin of type AUDIT is successfully loaded from /lib/libldapaudit.a.

Plugin of type AUDIT is successfully loaded from

/usr/ccs/lib/libsecldapaudit.a(shr.o).

Plugin of type EXTENDEDOP is successfully loaded from libevent.a.

Plugin of type EXTENDEDOP is successfully loaded from libtranext.a.

Plugin of type DATABASE is successfully loaded from /lib/libback-rdbm.a.

Non-SSL port initialized to 389.

```

Local UNIX socket name initialized to /tmp/s.slapd.
modifying entry cn=schema
...
modifying entry cn=schema
ldif2db: 2 entries have been successfully added out of 2 attempted.

```

Exporting local security repository into LDAP

After the LDAP server is configured, you must use the **sectoidif** command to export the local security repository to an LDIF file. The following example exports the local security repository into the file `allusers.ldif`. The `-d` flag specifies the base DN for the LDAP security repository. The `-S` flag specifies that the RFC2307AIX schema be used.

```

# sectoidif -d cn=aixsecdb,cn=aixdata,ou=mydept,o=mycompany.example,c=us \
-S RFC2307AIX >allusers.ldif

```

Note: At the time of writing, the `-u` flag for the **sectoidif** command allows you to export a specific user into the LDIF file. The `-u` flag will only export the account attributes and not the group attributes. The group attributes are required for successful login.

The following is an excerpt from the LDIF file created by the previous **sectoidif** command. The first entry is the user information and the second entry is the group information for the `ldapdb2` account.

```

dn:
uid=ldapdb2,ou=aixuser,cn=aixsecdb,cn=aixdata,ou=mydept,o=mycompany.example,c=u
s
uid: ldapdb2
objectClass: account
objectClass: posixAccount
objectClass: shadowAccount
objectClass: aixauxaccount
cn: ldapdb2
passwordchar: !
uidNumber: 400
gidNumber: 400
homeDirectory: /home/ldapdb2
loginShell: /usr/bin/ksh
authmethod1: SYSTEM
authmethod2: NONE
isadministrator: false
filepermmask: 22
userPassword: {crypt}cVIyvekXWsIqA
shadowLastChange: 1203755657
passwordflags: NOCHECK
ixtimelastlogin: 1032794759
hostlastlogin: server3

```

```
unsuccessfullogincount: 0
```

```
...
```

```
dn:  
cn=dbsysadm,ou=aixgroup,cn=aixsecdb,cn=aixdata,ou=mydept,o=mycompany.example,c=  
us  
cn: dbsysadm  
objectClass: posixGroup  
objectClass: aixauxgroup  
gidNumber: 400  
memberUid: ldapdb2  
isadministrator: false
```

```
...
```

After the local security DB is exported into an LDIF file, you must run the **ldif2db** command to import it into the LDAP directory. The following example imports the local LDAP server with the `allusers.ldif` file.

```
# ldif2db -i allusers.ldif  
ldif2db: 47 entries have been successfully added out of 47 attempted.
```

After using the **ldif2db** command imports the user and group data, you must restart the IBM Directory Server. To restart the server you need to kill the `slapd` process and then restart it. The procedure to restart the server is displayed below.

```
# ps -ef | grep slapd  
root 40650 58530 1 14:53:05 pts/7 0:00 grep slapd  
ldap 50440 1 4 14:15:22 - 0:52 /bin/slapd -f /etc/slapd32.conf  
# kill -9 50440  
# /bin/slapd -f /etc/slapd32.conf  
Plugin of type EXTENDEDOP is successfully loaded from libevent.a.  
Plugin of type EXTENDEDOP is successfully loaded from libtranext.a.  
Plugin of type PREOPERATION is successfully loaded from libDSP.a.  
Plugin of type EXTENDEDOP is successfully loaded from libevent.a.  
Plugin of type EXTENDEDOP is successfully loaded from libtranext.a.  
Plugin of type AUDIT is successfully loaded from /lib/libldapaudit.a.  
Plugin of type EXTENDEDOP is successfully loaded from libevent.a.  
Plugin of type EXTENDEDOP is successfully loaded from libtranext.a.  
Plugin of type DATABASE is successfully loaded from /lib/libback-rdbm.a.  
Non-SSL port initialized to 389.
```

The local UNIX socket name initialized to `/tmp/s.slapd`.

Configure AIX client for LDAP authentication

After the LDAP server is configured and loaded with user and group attributes, you must configure AIX to use the LDAP authentication load module. You must

run the **mksecldap** command with the **-c** flag to configure the client. The **-h** flag specifies the list of the host names of the LDAP servers to connect to. The **-a** and **-p** flags are the administrator's DN and password for access to the LDAP server. The **-d** flag is the base DN of the AIX data subtree. The **-u NONE** flag prevents any users from being migrated to LDAP.

```
# mksecldap -c -h ldap3.mycompany.example -a
"cn=admin,ou=mydept,o=mycompany.example,c=us"
-p mysecret -d "ou=mydept,o=mycompany.example,c=us" -u NONE
```

The **mksecldap** command enables the LDAP authentication load module by inserting the following stanza into the **/usr/lib/security/methods.cfg** file.

```
LDAP:
    program = /usr/lib/security/LDAP
    program_64 =/usr/lib/security/LDAP64
```

The **mksecldap** client setup also starts the **secldapclntd** daemon. The **secldapclntd** daemon manages connections and transactions from the LDAP authentication load module to the remote LDAP security information servers. The **secldapclntd** daemon caches LDAP queries in order to improve performance. It is configured using the **/etc/security/ldap/ldap.cfg** file. The following excerpt from the **ldap.cfg** file shows the client configuration generated from the previous **mksecldap** command.

```
...

# Comma separated list of ldap servers this client talks to
#ldapservers:myldapserver.ibm.com
ldapservers:ldap3.mycompany.example

# LDAP server bindDN
#ldapadmin:cn=admin
ldapadmin:cn=admin,ou=mydept,o=mycompany.example,c=us

# LDAP server bindDN password
#ldapadmpwd:secret
ldapadmpwd:mysecret

# Whether to use SSL to communicate with the LDAP server. Valid value
# is either "yes" or "no". Default is "no".
# Note: you need a SSL key and a password to the key to enable this.
#useSSL: no
useSSL:no

# SSL key file path and key password
#ldapsslkeyf:/tmp/key.kdb
#ldapsslkeypwd:mykeypwd
```

```

# AIX-LDAP attribute map path.
#userattrmappath:/etc/security/ldap/aixuser.map
userattrmappath:/etc/security/ldap/2307aixuser.map
#groupattrmappath:/etc/security/ldap/aixgroup.map
groupattrmappath:/etc/security/ldap/2307aixgroup.map
#idattrmappath:/etc/security/ldap/aixid.map
idattrmappath:/etc/security/ldap/aixid.map

# Base DN where the user and group data are stored in the LDAP server.
# e.g., if user foo's DN is: username=foo,ou=aixuser,cn=aixsecdb
# then the user base DN is: ou=aixuser,cn=aixsecdb
#userbasedn:ou=aixuser,cn=aixsecdb,cn=aixdata
userbasedn:ou=aixuser,cn=aixsecdb,cn=aixdata,ou=mydept,o=mycompany.example,c=us
#groupbasedn:ou=aixgroup,cn=aixsecdb,cn=aixdata
groupbasedn:ou=aixgroup,cn=aixsecdb,cn=aixdata,ou=mydept,o=mycompany.example,c=us
#idbasedn:cn=aixid,ou=system,cn=aixsecdb,cn=aixdata
idbasedn:cn=aixid,ou=system,cn=aixsecdb,cn=aixdata,ou=mydept,o=mycompany.example,c=us
#hostbasedn:ou=hosts,cn=nisdata,cn=aixdata
#servicebasedn:ou=services,cn=nisdata,cn=aixdata
#protocolbasedn:ou=protocols,cn=nisdata,cn=aixdata
#networkbasedn:ou=networks,cn=nisdata,cn=aixdata
#netgroupbasedn:ou=netgroup,cn=nisdata,cn=aixdata
#rpcbasedn:ou=rpc,cn=nisdata,cn=aixdata

# LDAP class definitions.
#userclasses:aixaccount,ibm-securityidentities
userclasses:account,posixaccount,shadowaccount,aixauxaccount
#groupclasses:aixaccessgroup
groupclasses:posixgroup,aixauxgroup

# LDAP server version. Valid values are 2 and 3. Default is 3.
#ldapversion:3

# LDAP server port. Default to 389 for non-SSL connection and
# 636 for SSL connection
#ldapport:389
ldapport:389
#ldapsport:636
...

```

The following entry is added to the `/etc/inittab` file to start the `secdapclntd` daemon during the system boot.

```
ldapclntd:2:once: /usr/sbin/secdapclntd > /dev/console 2>&1
```

Several commands were added to control and monitor the `secdapclntd` daemon. The **`flush-secdapclntd`** and **`ls-secdapclntd`** commands flush the LDAP client

cache and display LDAP client statistics. The **restart-secdapclntd**, **start-secdapclntd**, and **stop-secdapclntd** commands restart, start, and stop the secdapclntd daemon. The following section shows examples of these commands.

```
# start-secdapclntd
Starting the secdapclntd daemon.
The secdapclntd daemon started successfully.

# stop-secdapclntd
The secdapclntd daemon terminated successfully.

# restart-secdapclntd
The secdapclntd daemon terminated successfully.
Starting the secdapclntd daemon.
The secdapclntd daemon started successfully.

# ls-secdapclntd
ldapservers=ldap3.mycompany.example
ldapport=389
ldapversion=3
userbasedn=ou=aixuser,cn=aixsecdb,cn=aixdata,ou=mydept,o=mycompany.example,c=us
groupbasedn=ou=aixgroup,cn=aixsecdb,cn=aixdata,ou=mydept,o=mycompany.example,c=us
idbasedn=cn=aixid,ou=system,cn=aixsecdb,cn=aixdata,ou=mydept,o=mycompany.example,c=us
usercachesize=1000
usercacheused=0
groupcachesize=100
groupcacheused=0
cachetimeout=300
heartbeatT=300
numberofthread=10
alwaysmaster=no
userobjectclass=account,posixaccount,shadowaccount,aixauxaccount
groupobjectclass=posixgroup,aixauxgroup

# flush-secdapclntd
```

Note: You will not be able to configure the LDAP authentication client using the **mksecdap** command unless you have user and group entries defined to set up the client correctly.

User and group administrative commands using LDAP

After the LDAP authentication load module and the secdapclntd daemon is running, you can now use most of the AIX user and group administration

commands to administer LDAP users and groups. The following command creates an LDAP user test20, using the **mkuser** and **passwd** commands.

```
# mkuser -R LDAP SYSTEM=LDAP test20

# passwd test20
Changing password for "test20"
test20's New password: test20
Enter the new password again: test20

# lsuser -R LDAP test20
test20 id=219 pgrp=staff groups=staff home=/home/test20 shell=/usr/bin/ksh
login=true su=true rlogin=true telnet=true daemon=true admin=false sugroups=ALL
admgroups= tpath=nosak ttys=ALL expires=0 auth1=SYSTEM auth2=NONE umask=22
registry=LDAP SYSTEM=LDAP logintimes= loginretries=0 pldwarntime=0
account_locked=false minage=0 maxage=0 maxexpired=-1 minalpha=0 minother=0
mindiff=0 maxrepeats=8 minlen=0 histexpire=0 histsize=0 pwdchecks= dictionlist=
fsize=2097151 cpu=-1 data=262144 stack=65536 core=2097151 rss=65536
nofiles=2000 roles=
```

The following example shows how to create the LDAP group group20 and add the test20 user to that group, using the **mkgroup** command.

```
# mkgroup -R LDAP users=test20 group20

# lsgroup -R LDAP group20
group20 id=210 admin=false users=test20 registry=LDAP
```

Note: The LDAP server and client software configured in this example were not setup using SSL for secure LDAP communications. In order to maintain a secure environment, SSL should be configured on the server and client side.

In the legacy AIX and RFC2307AIX schema, the AIX user attribute `account_locked` is mapped to the LDAP attribute `isAccountEnabled`. The names of the two attributes portray opposite meanings. The correct way to interpret these attributes is using the `account_locked` attribute. If you use the AIX user administration utilities, the use of this attribute will appear to be consistent.

9.10.1 Host login restrictions for LDAP users

In Version 5.2, AIX now supports two new security attributes to restrict the machines that a user can log in to using an LDAP account. The new attributes are named `hostsallowedlogin` and `hostsdeniedlogin` and can be assigned to each user account. The default setting is that the `hostsallowedlogin` and `hostsdeniedlogin` attribute are not defined, allowing unrestricted access to all LDAP client machines. If the `hostsallowedlogin` and `hostsdeniedlogin` rules both

match the current system, the `hostsdeniedlogin` rule is preferred and user login is denied. These attributes can be a host name, IP address, network address, and subnet. These attributes are only available if the LDAP security information server is using the `RFC2307AIX` schema.

The following example allows the user `test20` to only log in on the machine named `server20`.

```
# chuser -R LDAP hostsallowedlogin=server20 test20
# lsuser -R LDAP -a hostsallowedlogin hostsdeniedlogin test20
test20 hostsallowedlogin=server20
```

The following example allows the user `test20` to only log in to any machines with IP addresses of `192.168.1.1` through `192.168.1.254`. The `192.168.1/24` parameter specifies a network address of `192.168.1`, the network ID of 24 bits, and the host ID of 8 bits.

```
# chuser -R LDAP hostsallowedlogin=192.168.1/24 test20
# lsuser -R LDAP -a hostsallowedlogin hostsdeniedlogin test20
test20 hostsallowedlogin=192.168.1/24
```

The following example allows the user `test20` to log in to any machine except the machines named `private1.mycompany.example`, `private2.mycompany.example`.

```
chuser -R LDAP
hostsdeniedlogin=private1.mycompany.example,private2.mycompany.example test20
# lsuser -R LDAP -a hostsallowedlogin hostsdeniedlogin test20
test20 hostsdeniedlogin=private1.mycompany.example,private2.mycompany.example
```

If you telnet into a machine that is denied access through these attributes you will receive the following message:

```
# telnet server3
Trying...
Connected to server3.mycompany.example.
Escape character is '^]'.

telnet (server3)
...
AIX Version 5
(C) Copyrights by IBM and by others 1982, 2002.
login: test20
test20's Password:
3004-339 You are not allowed to login to this system.
login:
```


9.11 Updating password maps in NIS (5.1.0)

In AIX 5L Version 5.1, the `yppasswdd` daemon directly updates the password maps and pushes the new maps to the slave servers when a password change request is processed. This results in a performance improvement when updating the NIS maps, compared to previous versions of AIX, where a rebuild of the maps occurred each time an update was made.

By default, this function is disabled, therefore a traditional mechanism, such as forking a command child process on the `/var/yp` directory is used. To use this function, you must issue the following command to add the `-r` option to the `yppasswdd` subsystem.

```
# chsys -s yppasswdd -a "/etc/passwd -r"
```

9.12 NIS/NIS+ integration into LDAP (5.2.0)

With Version 5.2, AIX supports LDAP for authentication, user and group attribute storage and schema for NIS data. Refer to 9.10, "AIX security LDAP integration (5.2.0)" on page 599, for information how to set up the LDAP server for the RFC2307 schema. The following section describes how to migrate NIS maps into the LDAP directory using the RFC2307 schema. After the NIS maps are migrated, the NIS client can be disabled, as the NIS maps can be accessed directly via LDAP.

The RFC2307 specification defines a schema to hold the data from the following NIS maps:

- ▶ `passwd`
- ▶ `group`
- ▶ `networks`
- ▶ `netgroups`
- ▶ `rpc`
- ▶ `hosts`
- ▶ `services`
- ▶ `protocols`

To migrate the data from your NIS maps you must run the `nistoldif` command to dump the maps into an LDIF file. The following example uses the `nistoldif` command to dump all the NIS MAP files into the LDIF file `nisdump.ldif`. The `-d` flag specifies the base DN where the AIX local repository resides.

```
# nistoldif -d cn=aixdata,ou=mydept,o=mycompany.example,c=us >nisdump.ldif
```

The following section is an excerpt of the `nisdump.ldif` file generated from the previous `nistoldif` command. The first LDIF entry is the loopback host entry in

the host NIS map. The second LDIF entry is the udp protocol entry in the protocols NIS map.

```
dn:
cn=loopback+ipHostNumber=127.0.0.1,ou=hosts,cn=nisdata,cn=aixdata,ou=mydept,o=
mycompany.example,c=us
objectClass: top
objectClass: ipHost
objectClass: device
ipHostNumber: 127.0.0.1
cn: loopback
cn: localhost

dn:
cn=udp,ou=protocols,cn=nisdata,cn=aixdata,ou=mydept,o=mycompany.example,c=us
cn: udp
cn: UDP
objectClass: top
objectClass: ipProtocol
ipProtocolNumber: 17
description: description
```

By default, the **nistoldif** command will export all the NIS maps into LDIF. Use the **-s** flag to specify the list of maps to export into LDIF. After the LDIF file is generated, you must use the **ldapadd** command to load the NIS maps into LDAP. The following command demonstrates this.

```
# ldapadd -c -a -D "cn=admin,ou=mydept,o=mycompany.example,c=us" -w mysecret -f
nisdump.ldif
```

The **nistoldif** command will not directly export NIS+ maps to LDIF files. You must use the **nisaddent** command to export the data from each table. After the data is exported to a LDIF file, you can import it using the **ldapadd** command. The following example shows the syntax of the **nisaddent** command.

```
# /usr/lib/nis/nisaddent -d -t table tabletype > filename
```

After the NIS maps are imported into the LDAP server you must configure the AIX LDAP security client using the **mksecldap** command with the **-c** flag. This must be done after the NIS maps are loaded, as **mksecldap** will search the LDAP directory and only enable the NIS maps it locates. The following example will configure the LDAP security client. The **-h** flag specifies the list of host names of the LDAP servers to connect to. The **-a** and **-p** flags are the administrator's DN and password for access to the LDAP server. The **-d** flag is the base DN of the AIX data subtree. The **-u NONE** flag prevents any users from being migrated to LDAP.

```
# mksecldap -c -h ldap3.mycompany.example -a
"cn=admin,ou=mydept,o=mycompany.example,c=us"
-p mysecret -d "ou=mydept,o=mycompany.example,c=us" -u NONE
```

The **mksecldap** command will modify the `/etc/security/ldap/ldap.cfg` configuration file. If any NIS maps have been located in the LDAP directory, it will also modify the `/etc/irs.conf` and `/etc/netsvc.conf` files.

The following excerpt from the `ldap.cfg` file shows the NIS map data to DN mapping generated by the previous **mksecldap** command. The `ldap.cfg` will only have configuration entries for NIS maps it was able to locate.

```
# Base DN where the user and group data are stored in the LDAP server.
# e.g., if user foo's DN is: username=foo,ou=aixuser,cn=aixsecdb
# then the user base DN is: ou=aixuser,cn=aixsecdb
#userbasedn:ou=aixuser,cn=aixsecdb,cn=aixdata
userbasedn:ou=aixuser,cn=aixsecdb,cn=aixdata,ou=mydept,o=mycompany.example,c=us
#groupbasedn:ou=aixgroup,cn=aixsecdb,cn=aixdata
groupbasedn:ou=aixgroup,cn=aixsecdb,cn=aixdata,ou=mydept,o=mycompany.example,c=us
#idbasedn:cn=aixid,ou=system,cn=aixsecdb,cn=aixdata
idbasedn:cn=aixid,ou=system,cn=aixsecdb,cn=aixdata,ou=mydept,o=mycompany.example,c=us
#hostbasedn:ou=hosts,cn=nisdata,cn=aixdata
hostbasedn:ou=hosts,cn=nisdata,cn=aixdata,ou=mydept,o=mycompany.example,c=us
#servicebasedn:ou=services,cn=nisdata,cn=aixdata
servicebasedn:ou=services,cn=nisdata,cn=aixdata,ou=mydept,o=mycompany.example,c=us
#protocolbasedn:ou=protocols,cn=nisdata,cn=aixdata
protocolbasedn:ou=protocols,cn=nisdata,cn=aixdata,ou=mydept,o=mycompany.example,c=us
#networkbasedn:ou=networks,cn=nisdata,cn=aixdata
networkbasedn:ou=networks,cn=nisdata,cn=aixdata,ou=mydept,o=mycompany.example,c=us
#netgroupbasedn:ou=netgroup,cn=nisdata,cn=aixdata
netgroupbasedn:ou=netgroup,cn=nisdata,cn=aixdata,ou=mydept,o=mycompany.example,c=us
#rpcbasedn:ou=rpc,cn=nisdata,cn=aixdata
rpcbasedn:ou=rpc,cn=nisdata,cn=aixdata,ou=mydept,o=mycompany.example,c=us
```

The **mksecldap** command will add `nis_ldap` to the host line in the `/etc/netsvc.conf` file. The `NSORDER` environment variable will also support the `nis_ldap` parameter. The following example will set the name resolution order to `nis_ldap`, `bind`, `NIS`, and then `local` `/etc/hosts`.

```
hosts = nis_ldap, bind, nis, local
```

If NIS maps are detected, the **mksecldap** command will also modify the `/etc/irs.conf` file. The `irs.conf` file specifies the resolution order for the NIS map files. The following example shows the `/etc/irs.conf` file. The lookup order for the services routines are `nis_ldap`, `nis`, and then `local`.

```
hosts nis_ldap continue
```

```
hosts dns continue
hosts nis continue
hosts local
services nis_ldap continue
services nis continue
services local
networks nis_ldap continue
networks dns continue
networks nis continue
networks local
netgroup nis_ldap continue
netgroup nis continue
netgroup local
protocols nis_ldap continue
protocols nis continue
protocols local
```

9.13 Pluggable Authentication Module support

Pluggable Authentication Mechanism (PAM) is a flexible mechanism for authenticating users.

9.13.1 PAM services (5.1.0)

The PAM support provides a way to develop programs that are independent of an authentication scheme. These programs need authentication modules to be attached to them at runtime in order to work. Which authentication module is to be attached is dependent on the local system setup.

Note: The PAM-related files are not included in AIX 5L Version 5.1 BOS CD-ROM media, but are included in the first shipped update CD as APAR IY19060. After applying this APAR, PAM-related files are included in bos.rte.security and bos.adt.includes fileset updates, both at the 5.1.0.1 level.

In AIX 5L Version 5.1, support for X/Open Single Sign-on Service (XSSO) and PAM has been added. For more information about XSSO, please visit:

<http://www.opennc.com/pubs/catalog/u039.htm>

9.13.2 PAM enhancements (5.2.0)

AIX 5L Version 5.2 security services has been integrated with the Pluggable Authentication Modules (PAM) framework. The PAM framework allows administrators to incorporate multiple authentication mechanisms into an existing

system through the use of pluggable modules. Applications written using the PAM framework do not need to be modified to support new authentication methods or modules.

In Version 5.1, the PAM libraries and include files were supplied but were not integrated into the AIX Security Services. In Version 5.2, applications that use the PAM framework could call AIX Security Services and applications that use the AIX security libraries could now call PAM modules.

AIX Security Services to PAM authentication

The AIX Security Services to PAM authentication is implemented using a PAM loadable authentication module (LAM), which is conceptually similar to the Kerberos 5, DCE, and NIS LAMs. The PAM LAM allows applications written to use the AIX Security Services to call PAM modules for authentication. Commands such as **passwd**, **su**, **telnetd**, **tftpd**, and **ftpd**, written to use the AIX Security Services can now use PAM modules to change passwords and authenticate users. See Figure 9-6 on page 616 for an illustration of the AIX Security Service to PAM module path. The **pam_krb**, **pam_ldap**, and **pam_dce** PAM modules are not supplied with AIX. They are only listed as examples of third-party solutions.

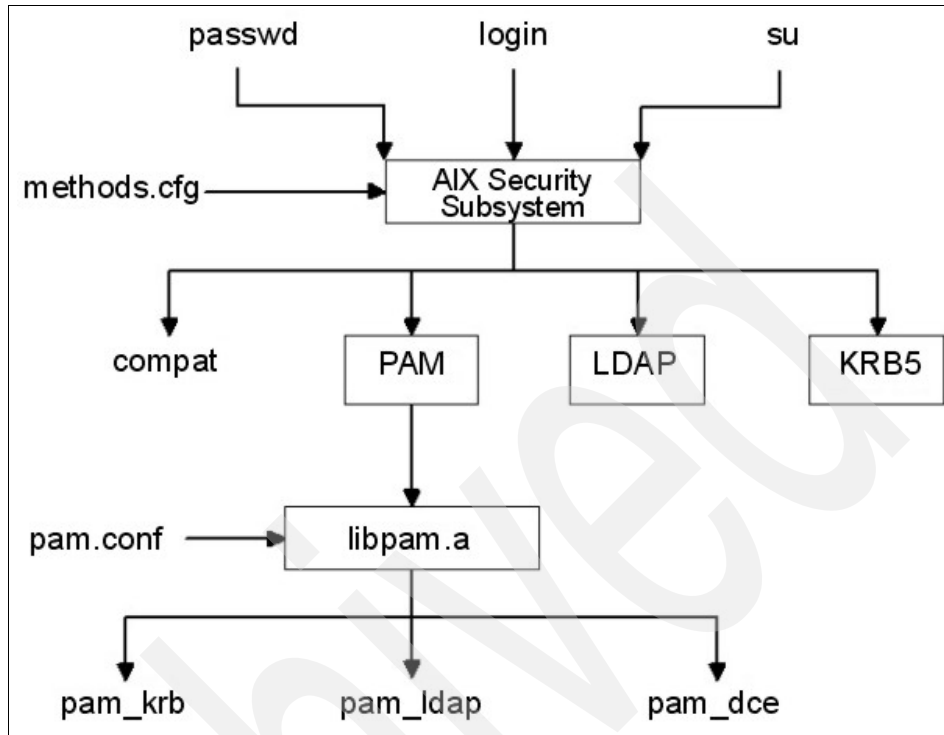


Figure 9-6 AIX Security Service to PAM module path

The PAM LAM can be enabled on a per-user or per-machine basis, using the per-user or default registry and SYSTEM attributes. Normally you would only want to use the PAM LAM on a per-user basis. To use the **mkuser** command to create a PAM-enabled user you must use the -R flag to select the PAM authentication module. For example, to create a new PAM user tommy, use the following **mkuser** command:

```
# mkuser -R PAMfiles registry=PAMfiles SYSTEM=PAMfiles tommy
```

The `/usr/lib/security/methods.cfg` file specifies the definitions of the authentication grammar used by the registry and SYSTEM attributes. The PAM stanza below specifies the LAM used for PAM authentication. The PAMfiles stanza specifies PAM to be used for authentication, and user attributes are to be stored in local files. Insert the following stanzas into your `methods.cfg` configuration file. If stanzas with the same names already exist, then carefully merge the following stanzas into your configuration.

```
PAM:
    program = /usr/lib/security/PAM
```

PAMfiles:

```
options = auth=PAM,db=BUILTIN
```

The `/etc/pam.conf` file specifies the order and names of the PAM modules to call when requests for PAM authentication are made. PAM modules can be stacked to allow a request to call multiple PAM modules, in order to service the authentication request. Entries in the file are composed of the following whitespace-delimited fields:

```
service_name module_type control_flag module_path module_options
```

Where:

service_name	Specifies the name of the service. The keyword <code>OTHER</code> is used to define the default module to use for applications not specified in an entry.
module_type	Specifies the module type for the service. Valid module types are <code>auth</code> , <code>account</code> , <code>session</code> , or <code>password</code> .
control_flag	Specifies the stacking behavior for the module. Supported control flags are <code>required</code> , <code>sufficient</code> , or <code>optional</code> .
module_path	Specifies the path name to a library object that implements the service functionality. Entries for <code>module_path</code> should start from the root (<code>/</code>) directory. If the entry does not begin with <code>/</code> , then <code>/usr/lib/security</code> will be prepended to the file name.
module_options	Specifies a list of options that can be passed to the service modules. Values for this field are dependent on the options supported by the module defined in the <code>module_path</code> field.

The following `pam.conf` file specifies that for the `telnet` and `login` services, requests for the `auth` and `account` PAM services are routed to the `/usr/lib/security/pam_unix` module. The required keyword specifies that the all required modules in the stack must pass for a successful result. The `passwd` service will use the `/usr/lib/security/pam_unix` module for password PAM service requests. For any other services not specifically mentioned, the `/usr/lib/security/pam_aix` module will service `auth`, `account`, `session`, and `password` service requests. The `pam_aix` module allows PAM applications to access the AIX Security Services. For more information refer to “PAM authentication to AIX Security Services” on page 618.

```
# Authentication Management
#
login    auth    required    /usr/lib/security/pam_unix
telnet   auth    required    /usr/lib/security/pam_unix
OTHER    auth    required    /usr/lib/security/pam_aix
```

```
# Account Management
```

```
#
login    account required    /usr/lib/security/pam_unix
telnet   account required    /usr/lib/security/pam_unix
OTHER    account required    /usr/lib/security/pam_aix

# Session Management
#
OTHER     session required    /usr/lib/security/pam_aix
# Password Management
#
passwd    password required    /usr/lib/security/pam_unix
OTHER     password required    /usr/lib/security/pam_aix
```

Note: AIX 5L Version 5.2 only ships with the PAM module pam_aix. To use AIX Security Services to PAM module authentication you must create your own PAM modules using the PAM framework or get PAM modules from a third-party, such as the Internet.

Table 9-2 lists the mapping of the AIX Security Services calls and the PAM API. This mapping is used for all authentication requests when the register and SYSTEM attributes are set to PAMfiles.

Table 9-2 Mapping of the AIX Security Services calls and the PAM API

AIX	PAM API
authenticate	pam_authenticate
chpass	pam_chauthtok
passwdexpired	pam_acct_mgmt
passwdrestrictions	No comparable mapping exists, success returned

PAM authentication to AIX Security Services

PAM authentication to AIX Security Services is implemented using the pam_aix PAM. The pam_aix PAM allows applications written using the PAM framework to call AIX Security Services for authentication. One such application developed to use the PAM framework for authentication is the OpenSSH daemon. OpenSSH is a free SSH/SecSH protocol suite providing encryption for network services like remote login or remote file transfers. IBM has made a PAM-enabled OpenSSH LPP available from the IBM developerWorks site. The following section uses the OpenSSH package to show how PAM applications call AIX Security Services. See Figure 9-7 on page 619 for an illustration of the PAM authentication to AIX Security Services path.

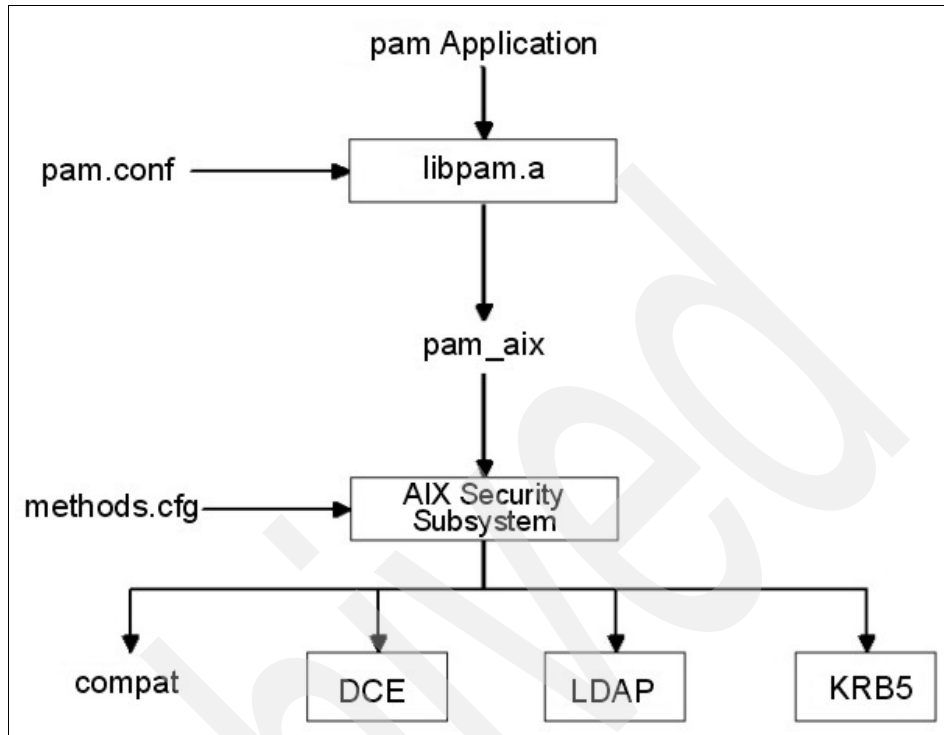


Figure 9-7 PAM Module to AIX Security Service Path

To install the OpenSSH you must download OpenSSH package for AIX 5L from the IBM developerWorks Web site at the following URL:

<http://oss.software.ibm.com/developerworks/projects/opensshi>

OpenSSH LPP requires the OpenSSL library to be installed. The OpenSSL RPM can be downloaded from the AIX Toolbox for Linux Applications home page located at the following URL:

<http://www.ibm.com/servers/aix/products/aixos/linux/>

Install the OpenSSL RPM packages by running the following **rpm** commands:

```
# rpm -i openssl-0.9.6e-2.aix4.3.ppc.rpm
# rpm -q openssl
openssl-0.9.6e-2
```

After OpenSSL RPM is installed, use the following commands, SMIT, or Web-based System Manager to install the OpenSSH package. You must specify the device or directory where the OpenSSH LPPs are located in your

environment. Replace the *LPPSOURCE* tag in the following commands with the correct location.

```
# installp -acgXYd LPPSOURCE openssh
...
# ls1pp -L "openssh*"
Fileset                                Level  State  Type  Description (Uninstaller)
-----
openssh.base.client                    3.4.0.5200  C    F    Open Secure Shell Commands
openssh.base.server                    3.4.0.5200  C    F    Open Secure Shell Server
openssh.license                        3.4.0.5200  C    F    Open Secure Shell License
openssh.man.en_US                      3.4.0.5200  C    F    Open Secure Shell
                                         Documentation - U.S. English
...
openssh.msg.zh_TW                      3.4.0.5200  C    F    Open Secure Shell Messages -
                                         Traditional Chinese
...
```

The following */etc/pam.conf* file specifies that for the *sshd* service, requests for the *auth*, *account*, *session*, and *password* PAM services are routed to the */usr/lib/security/pam_aix* module. The *pam_aix* module will then route those requests to the AIX Security Services libraries.

```
# Authentication Management
#
sshd  auth    required    /usr/lib/security/pam_aix
OTHER auth    required    /usr/lib/security/pam_aix

# Account Management
#
sshd  account required    /usr/lib/security/pam_aix
OTHER account required    /usr/lib/security/pam_aix

# Session Management
#
sshd  session required    /usr/lib/security/pam_aix
OTHER session required    /usr/lib/security/pam_aix

# Password Management
#
sshd  password required    /usr/lib/security/pam_aix
OTHER password required    /usr/lib/security/pam_aix
```

After the */etc/pam.conf* file is configured properly, the *sshd* daemon must be configured to use PAM. In the */etc/ssh/sshd_config* file, uncomment the following line. You must restart the *sshd* daemon for the configuration change to take effect, using the **stopsrc** and **startsrc** commands.

```
PAMAuthenticationViaKbdInt yes
```

```
# stopsrc -s sshd
0513-044 The sshd Subsystem was requested to stop.
# startsrc -s sshd
0513-059 The sshd Subsystem has been started. Subsystem PID is 507956.
```

The sshd daemon will now use the AIX Security Services for authentication. For more information on OpenSSH refer to the OpenSSH home page at the following URL:

<http://www.openssh.org>

Below we have listed the mapping of the PAM API calls to AIX Security Services. This mapping is used for all authentication requests when the pam_aix module is called to service a request.

pam_sm_authenticate	authenticate
pam_sm_chauthtok	passwdexpired, chpass Note: passwdexpired is only checked if the PAM_CHANGE_EXPIRED_AUTH Tok flag is passed in
pam_sm_acct_mgmt	loginrestrictions, passwdexpired
pam_sm_setcred	No comparable mapping exists, PAM_SUCCESS returned
pam_sm_open_session	No comparable mapping exists, PAM_SUCCESS returned
pam_sm_close_session	No comparable mapping exists, PAM_SUCCESS returned

9.14 Public Key Infrastructure enhancements (5.2.0)

AIX 5L Version 5.2 provides its own Certificate Authentication Service, with the ability to authenticate users using X.509 Public Key Infrastructure (PKI) certificates and to associate certificates with processes as proof of a user's identity. It provides this capability through the Loadable Authentication Module Framework (LAMF), the same extensible AIX mechanism used to provide DCE, Kerberos, and other authentication mechanisms.

This section is broken down into the following topics:

- Overview of PKI and Certificate Authentication Service
- LDAP server installation and configuration
- Certificate Authentication Service configuration
- Common user and administrator tasks using PKI

- Process Authentication Group (PAG) commands

9.14.1 Overview of PKI and Certificate Authentication Service

PKI is a comprehensive system of policies, processes, and technologies working together to allow users and applications to exchange information securely and confidentially. PKI uses pairs of asymmetric keys, provided by a trusted third party known as a CA, to encrypt and decrypt information. These digital signatures provide the following security services:

Entity authentication	The identity of a user can be positively validated by verifying that a certificate was actually generated by a trusted certificate authority. By checking the certificate revocation list (CRL), the current status of the certificate can be checked for revocation.
Data confidentiality	Allows data to be exchanged securely across an insecure medium, such as the Internet. Data can be encrypted so that only the intended recipient can decrypt the data. Data transmissions across insecure networks can also be protected by using digital signatures in a key exchange to build a secure tunnel.
Data integrity	Allows users and applications to ensure that stored or transmitted data has not been accidentally or maliciously altered. If the data's digital signature is valid, then the user can be quite certain the data is unaltered.
Non-repudiation	Prevents an individual or entity from denying having performed a particular action.
Privilege management	Since the identity of an entity can be verified using digital signatures, access policies can be assigned to specific entities. The policies can then be used to restrict access-sensitive information or resources.

The certificate authority (CA) is a trusted entity that is responsible for generating and assigning digital certificates. The CA is trusted by one or more users to ensure the owner's identity of certificates it has issued. A CA must verify the identity of the new certificate owner before assigning a certificate to them. When the certificate owner's identity is verified, a certificate is generated and it is signed by the CA. When a certificate is presented for identity verification, the user or application will verify that the certificate is signed by a trusted CA. This allows for detection of bogus certificates not generated by the trusted CA. When a certificate has been compromised or revoked, the CA must issue a certificate revocation and update the CRL. Normally a CA would be responsible for issuing certificates for an organization or enterprise.

For information about the current and upcoming PKI standards, refer to the PKIX working group documents on the Internet Engineering Task Force (IETF) home page at the following URL:

<http://www.ietf.org/html.charters/pkix-charter.html>

Installation of PKI and Certificate Authentication Service

In order for AIX to use PKI authentication, you must install a certificate authority and an LDAP server. The LDAP server will store all the user's public keys generated by the CA and the AIX PKI repository.

In order to easily describe the capabilities of the AIX PKI enhancements, the following sections will describe how to install and configure PKI for a fictitious company. The name of this US company is MyCompany and their internet domain name is mycompany.example. The company has defined an enterprise-wide LDAP hierarchy, with a top-level distinguished name (DN) of o=mycompany.example, c=us. LDAP allows the enterprise to divide its directory into sections reflecting its organizational structure. The company defined a subtree for a specific department, specified by the relative distinguished named (RDN) ou=mydept.

The AIX PKI requires two different subtrees in the LDAP directory. The first subtree is used to store all the public keys generated by the CA. The default DN for this subtree is ou=cert. Since a CA will normally generate certificates for an entire enterprise, it was decided that the user certificate subtree would be located at DN ou=cert,o=mycompany.example, c=us. The user certificate subtree contains an entry for every entity the CA generated a certificate for.

The second subtree is used to store all the information AIX needs for PKI authentication, including all the user's public keys and the one certificate used for authentication. The default DN for this subtree is ou=pkidata,cn=aixdata. It was decided that the organizationalUnit ou=mydept would have their own directory for AIX PKI data. The assigned DN for the departments's PKI data is ou=pkidata,cn=aixdata,ou=mydept, o=mycompany.example,c=us. The first level of the PKI data subtree contains an entry for every account enabled for AIX PKI authentication. The second level contains all the details of individual certificates that are candidates for AIX PKI authentication. Examples of the information stored in the second level are the URI representing the location of the keystore and the name that distinguishes one key from another. Only one certificate can be used for PKI authentication at a time.

The complete LDAP hierarchy for mycompany.example's PKI deployment is illustrated in Figure 9-8 on page 624.

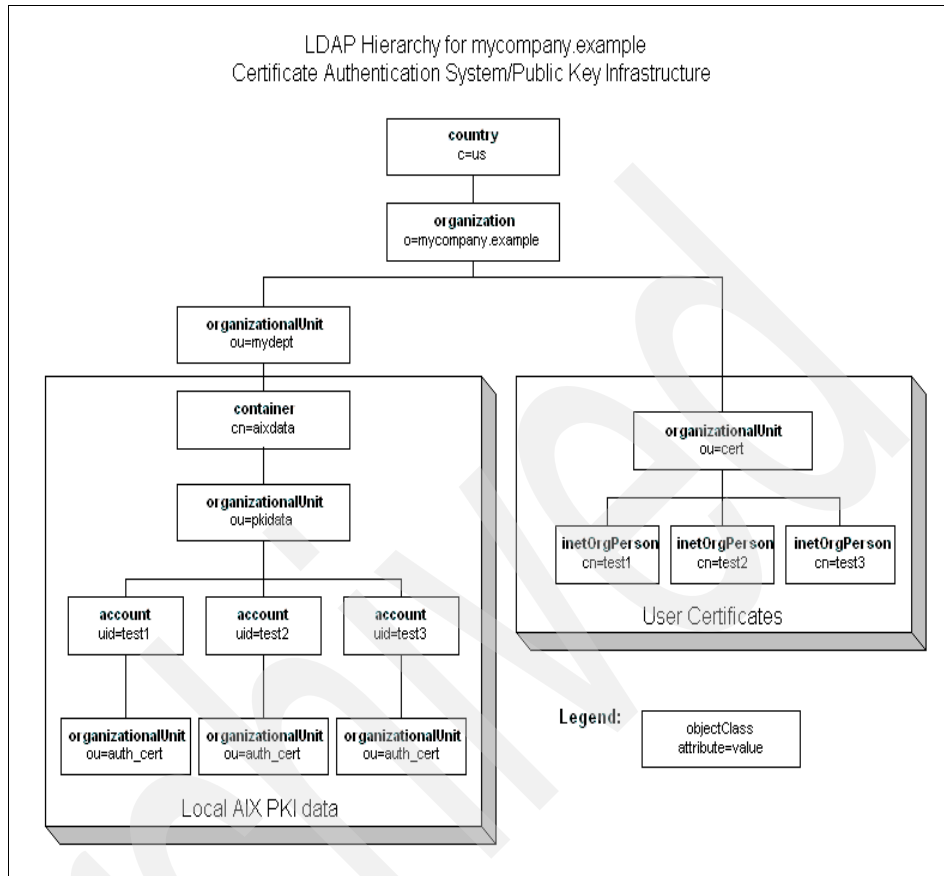


Figure 9-8 LDAP hierarchy for myexample.company PKI example

9.14.2 LDAP server installation and configuration

You must install the IBM Directory Server Version 4.1 product to store PKI user certificate data and the AIX local repository. Use the following commands, SMIT, or Web-based System Manager to install the following Licensed Product Packages (LPPs). IBM Directory Server uses DB2 as the backend datastore and will automatically install DB2. If you need more information about installation and configuration of this product, install the detailed documentation supplied with the product. The Directory server documentation is located in the `ldap.html.en_US.*` files. You must specify the device or directory where the software LPPs are located in your environment. Replace the `LPPSOURCE` tag in the following commands with the correct location:

```
# installp -acgXd LPPSOURCE ldap.server ldap.client ldap.html.en_US
```

To enable the directory server configuration GUI you need to install the IBM HTTP Server. Use the following commands, SMIT, Web-based System Manager to install the following Licensed Product Packages. You must specify the device or directory where the software LPPs are located in your environment. Replace the *LPPSOURCE* tag in the following commands with the correct location:

```
# installp -acgXYd LPPSOURCE http_server
```

Now that all the required software is installed, you must set the administrator DN and password for the directory server. The administrator DN has unrestricted access to the entire directory server, so it would be good practice to use a strong password. Run the following commands to set the administrator DN to `cn=admin,ou=mydept,o=mycompany.example,c=us` and the password to `mysecret`. The `-u` and `-p` flags of the **ldapcfcg** command specify the administrator DN and password of the directory server, respectively.

```
# ldapcfcg -u cn=admin,ou=mydept,o=mycompany.example,c=us -p mysecret
Password for administrator DN cn=admin,ou=mydept,o=mycompany.example,c=us has
been set.
IBM Directory Server Configuration complete.
```

After setting the administrator DN and password, you need to configure the Web server to enable the IBM Directory Server's configuration GUI. You must run the following commands to configure the configuration GUI and restart the server.

```
# ldapcfcg -s ibmhttp -f /usr/HTTPServer/conf/httpd.conf
IBM Directory Server Configuration complete.
# /usr/HTTPServer/bin/apachectl restart
/usr/HTTPServer/bin/apachectl restart: httpd restarted
```

The `-s` and `-f` flags of the **ldapcfcg** command specify the Web server type and the location of the Web server configuration file to modify, respectively. You must restart the Web server to have the changes take affect. After restarting the server, you can access the configuration GUI by accessing the following URL in a browser:

<http://ldap.mycompany.example/ldap>

The next step is to create the DB2 database used by the directory server. You must run the following commands to create the default DB2 database. The `-l` flag of the **ldapcfcg** command specifies the location of the DB2 database. You must ensure that there is at least 80 MB free in the file system in the specified location.

```
# ldapcfcg -l /home/ldapdb2
Creating the directory DB2 default database.
This operation may take a few minutes.
```

Configuring the database.
Creating database instance: ldapdb2.
Created database instance: ldapdb2.

```
Starting database manager for instance: ldapdb2.  
Started database manager for instance: ldapdb2.  
Creating database: ldapdb2.  
Created database: ldapdb2.  
Updating configuration for database: ldapdb2.  
Updated configuration for database: ldapdb2.  
Completed configuration of the database.
```

IBM Directory Server Configuration complete.

You must now configure the directory server with the suffixes needed for our LDAP hierarchy. A suffix is a DN that identifies the top entries in a locally held directory hierarchy. You can add suffixes through the directory server configuration GUI or by editing the `/usr/ldap/etc/slapd32.conf` file directly. In the example, the DN `o=mycompany.example,c=us` is our only locally held directory hierarchy. After the suffix is added, you must restart the directory for the changes to take effect.

To use the directory server configuration GUI, log in to the directory server configuration GUI by using the administrator DN and password set earlier. In this example, the administrator DN is `cn=admin,ou=mydept,o=mycompany.example,c=us` and the password is `mysecret`. After successfully logging in, locate the navigation bar on the left side and click through to the suffixes administration page (select **Settings** -> **Suffixes**).

Enter `o=mycompany.example,c=us` in the Suffix DN text box and then click the **Update** button. When this step is completed, your browser should resemble Figure 9-9 on page 627.

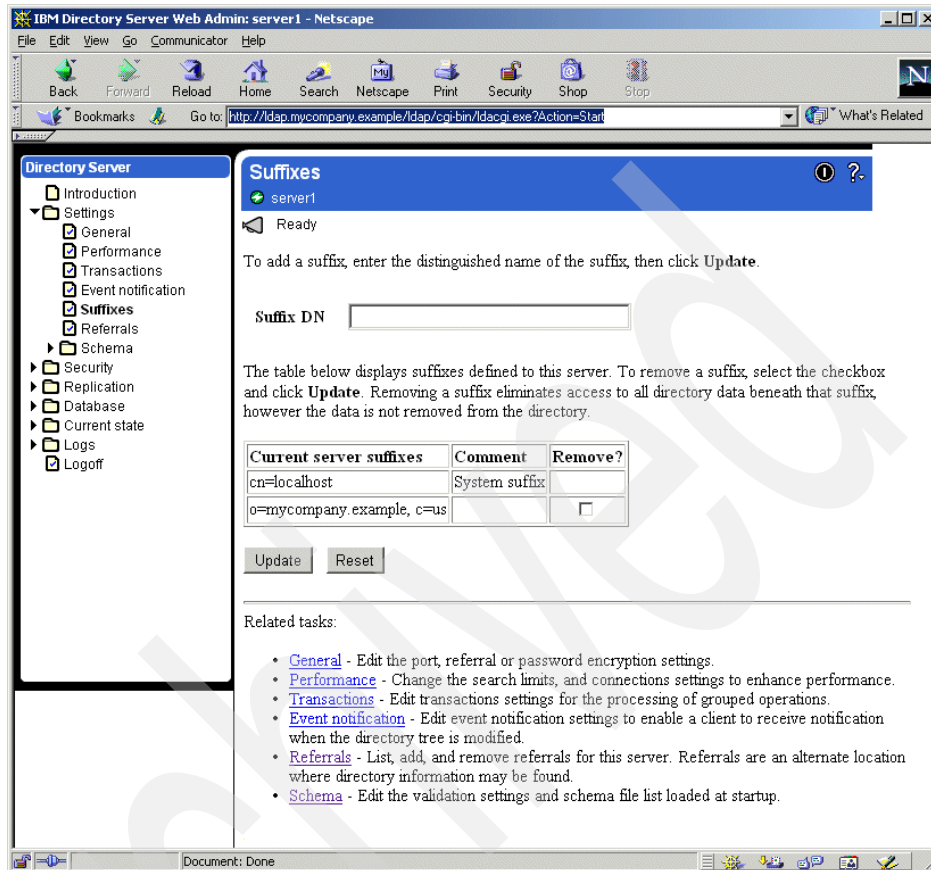


Figure 9-9 IBM directory administration GUI - suffixes

For this change to take effect, the directory server must be restarted. This can be done with the configuration GUI by selecting the restart icon in the upper-right corner.

To add suffixes using command line utilities, edit the `/usr/ldap/etc/slapd32.conf` file and make the following modifications. Find the following stanza and add the `o=mycompany.example,c=us` suffix after the `cn=localhost` line. The boldface text in the following stanza shows the required modification.

```
dn: cn=Directory, cn=RDBM Backends, cn=IBM SecureWay, cn=Schemas,
cn=Configuration
cn: Directory
ibm-slapdDbAlias: ldapdb2b
ibm-slapdDbConnections: 15
ibm-slapdDbInstance: ldapdb2
```

```

ibm-slapdDbLocation: /home/ldapdb2
ibm-slapdDbName: ldapdb2
ibm-slapdDbUserId: ldapdb2
ibm-slapdDbUserPW: <encryped passowrd>
ibm-slapdPlugin: database /lib/libback-rdbm.a rdbm_backend_init
ibm-slapdReadOnly: FALSE
ibm-slapdSuffix: cn=localhost
ibm-slapdSuffix: o=mycompany.example,c=us
objectclass: top
objectclass: ibm-slapdRdbmBackend

```

You must restart the directory server to have the changes take effect. To restart the server you need to kill the slapd process and then restart it. The procedure to restart the server is displayed below.

```

# ps -ef | grep slapd
  root 40650 58530   1 14:53:05 pts/7  0:00 grep slapd
  ldap 50440      1   4 14:15:22    -  0:52 /bin/slapd -f /etc/slapd32.conf
# kill -9 50440
# /bin/slapd -f /etc/slapd32.conf
Plugin of type EXTENDEDOP is successfully loaded from libevent.a.
Plugin of type EXTENDEDOP is successfully loaded from libtranext.a.
Plugin of type PREOPERATION is successfully loaded from libDSP.a.
Plugin of type EXTENDEDOP is successfully loaded from libevent.a.
Plugin of type EXTENDEDOP is successfully loaded from libtranext.a.
Plugin of type AUDIT is successfully loaded from /lib/libldapaudit.a.
Plugin of type EXTENDEDOP is successfully loaded from libevent.a.
Plugin of type EXTENDEDOP is successfully loaded from libtranext.a.
Plugin of type DATABASE is successfully loaded from /lib/libback-rdbm.a.
Non-SSL port initialized to 389.
Local UNIX socket name initialized to /tmp/s.slapd.

```

Now that the correct suffix has been added to the directory server configuration, you need to add entries to the directory to create the required LDAP hierarchy. To add entries to the directory you must create an LDAP Data Interchange Format (LDIF) file and then run the **ldapadd** command. Copy the stanza's below into a file called **mycompany.ldif**. The first stanza adds an organizational entry for the top level suffix **o=mycompany.example,c=us**, which we added in the previous step. The second stanza adds an organizationalUnit entry for the department-specific information.

```

dn: o=mycompany.example,c=us
objectclass: top
objectclass: organization
o: mycompany.example

dn: ou=mydept,o=mycompany.example,c=us
objectclass: organizationalUnit
ou: mydept

```

Run the following **ldapadd** command to add these entries into the directory. You will need to specify the directory administrator DN and password with the **-D** and **-w** flags, respectively. The **-f** flag specifies the name of the LDIF file to import.

```
# ldapadd -c -D cn=admin,ou=mydept,o=mycompany.example,c=us -w mysecret -f
mycompany.ldif
adding new entry o=mycompany.example,c=us
adding new entry ou=mydept,o=mycompany.example,c=us
```

9.14.3 Certificate Authentication Service configuration

To install the Certificate Authentication Service, you must install the Java security filesets and the Certificate Authentication Service filesets from the Expansion Pack CD. Use the following commands, SMIT, or Web-based System Manager to install the Java security filesets. You must specify the device or directory where the software LPPs are located in your environment. Replace the *LPPSOURCE* tag in the following commands with the correct location:

```
# installp -acgXd LPPSOURCE java131.ext.security
```

Note: At the time of writing, there is a conflict with the file `ibmjcaprovider.jar` located in `/usr/java131/jre/lib/ext`. This file must be moved for the Certificate Authentication Service to work properly. Perform the following commands:

```
# mkdir /usr/java131/jre/lib/ext/orig
# mv /usr/java131/jre/lib/ext/ibmjcaprovider.jar
/usr/java131/jre/lib/ext/orig/
```

Use the following commands, SMIT, or Web-based System Manager to install the Certificate Authentication Service filesets. The Certificate Authentication Service server requires the DB2 fileset `db2_07_01.jdbc` and will install it automatically. You must specify the device or directory where the software LPPs are located in your environment. Replace the *LPPSOURCE* tag in the following commands with the correct location:

```
# installp -acgXd LPPSOURCE cas.server cas.client
```

The next step is to create the LDAP hierarchy and access control list (ACL) for the local AIX repository and the user certificate tree. The `cas.server.rte` fileset includes template LDIF files for these steps in `/usr/cas/server/ldap`. You should only make modifications to copies of the supplied template. This allows you to go back to the default file if you have problems with modifying the files.

The file `pkiconfig.ldif` adds entries to create the local AIX repository and sets the ACLs. Copy the `pkiconfig.ldif` file to `pkiconfig_custom.ldif` and modify the copy to match the stanza below. The first stanza creates an entry for the `aixdata` tree. The second stanza creates the password-protected entry for all AIX-related PKI data storage and administration. The third stanza sets the `entryOwner` for the

pkidata entry to itself. The final stanza sets up the ACLs for the pkidata tree so only the pkidata administrator DN can access the pkidata directory tree. The password for the pkidata administrator DN is highlighted in boldface type. This password should be protected and difficult to guess.

```
dn: cn=aixdata,ou=mydept,o=mycompany.example,c=us
objectclass: container
cn: aixdata
```

```
dn: ou=pkidata,cn=aixdata,ou=mydept,o=mycompany.example,c=us
objectclass: organizationalUnit
ou: cert
userpassword: secret
```

```
dn: ou=pkidata,cn=aixdata,ou=mydept,o=mycompany.example,c=us
changetype: modify
add: entryOwner
entryOwner: access-id:ou=pkidata,cn=aixdata,ou=mydept,o=mycompany.example,c=us
ownerPropagate: true
```

```
dn: ou=pkidata,cn=aixdata,ou=mydept,o=mycompany.example,c=us
changetype: modify
add: aclEntry
aclEntry: group:cn=anybody:normal:grant:rsc:normal:deny:w
aclEntry: group:cn=anybody:sensitive:grant:rsc:sensitive:deny:w
aclEntry: group:cn=anybody:critical:grant:rsc:critical:deny:w
aclEntry: group:cn=anybody:object:deny:ad
aclPropagate: true
```

Run the following **ldapadd** command to add these entries into the directory and set the ACLs. Again you will need to specify the directory administrator DN and password with the **-D** and **-w** flags, respectively. The **-f** flag specifies the name of the LDIF file to import.

```
# ldapadd -c -D cn=admin,ou=mydept,o=mycompany.example,c=us -w mysecret -f
pkiconfig_custom.ldif
adding new entry cn=aixdata,ou=mydept,o=mycompany.example,c=us
adding new entry ou=pkidata,cn=aixdata,ou=mydept,o=mycompany.example,c=us
modifying entry ou=pkidata,cn=aixdata,ou=mydept,o=mycompany.example,c=us
modifying entry ou=pkidata,cn=aixdata,ou=mydept,o=mycompany.example,c=us
```

The file **setup.ldif** ensures that the LDAP server's schema has the appropriate **objectClasses** and attributes for the PKI enhancements. This file should not require any modification.

```
dn: cn=schema
changetype: modify
add: objectClasses
objectClasses: ( 2.5.6.21 NAME 'pkuser' DESC 'auxiliary class for non-CA
certificate owners' SUP top AUXILIARY MAY userCertificate )
```

```
dn: cn=schema
changetype: modify
add: objectClasses
objectClasses: ( 2.5.6.22 NAME 'pkiCA' DESC 'class for Certification
Authorities' SUP top AUXILIARY MAY ( authorityRevocationList $ caCertificate $
certificateRevocationList $ crossCertificatePair ) )
```

```
dn:cn=schema
changetype: modify
replace: attributetypes
attributetypes: ( 2.5.4.39 NAME ( 'certificateRevocationList'
'certificateRevocationList;binary' ) DESC ' ' SYNTAX
1.3.6.1.4.1.1466.115.121.1.5 SINGLE-VALUE )
-
replace:ibmattributetypes
ibmattributetypes:( 2.5.4.39 DBNAME ( 'certRevocationLst' 'certRevocationLst' )
ACCESS-CLASS NORMAL)
```

Run the following **ldapmodify** command to make the modifications to the LDAP directory's schema. Again you will need to specify the directory administrator DN and password with the **-D** and **-w** flags, respectively. The **-f** flag specifies the name of the LDIF file to import. If the schema already contains these objectClasses, the add operation will fail. These errors can be safely ignored.

```
# ldapmodify -c -D cn=admin,ou=mydept,o=mycompany.example,c=us -w mysecret -f
setup.ldif
modifying entry cn=schema
ldap_modify: Type or value exists
ldap_modify: additional info: object class '2.5.6.21' already exists, add
operation failed.
modifying entry cn=schema
ldap_modify: Type or value exists
ldap_modify: additional info: object class '2.5.6.22' already exists, add
operation failed.
modifying entry cn=schema
```

The file **addentries.ldif** adds the LDAP hierarchy needed for the certificates published by the Certificate Authentication Service. Copy the **addentries.ldif** file to **addentries_custom.ldif** and modify the copy to match the stanza below.

```
dn: ou=cert,o=mycompany.example,c=us
changetype: add
objectclass: organizationalUnit
objectclass: pkiCA
ou: cert
```

Run the following **ldapadd** command to add these entries into the directory. As previously mentioned, you will need to specify the directory administrator DN and

password with the -D and -w flags, respectively. The -f flag specifies the name of the LDIF file to import.

```
# ldapadd -c -D cn=admin,ou=mydept,o=mycompany.example,c=us -w mysecret -f
addentries_custom.ldif
adding new entry ou=cert,o=mycompany.example,c=us
```

The next step is to configure the Certificate Authority Service server. Before running the **mksecpki** command, you must create a reference file. The reference file contains one or more certificate-creation reference number and password pairs. In this example, the reference file is located at `/usr/cas/server/iafile` and contains the following information. The reference numbers and passphrase are sensitive information and should be kept private and be difficult to guess. If these numbers are compromised, someone could generate certificates without permission. The following reference numbers and passphrases are for examples only.

```
12345678
password1234
```

The **mksecpki** command requires many parameters to configure the Certificate Authentication Service server correctly. The -u flag specifies the user name that the Certificate Authentication Service server will run as. The -f flag specifies the location of the file that contains the reference number and passphrase that is used when creating certificates. The -p flag specifies the port number the Certificate Authentication Service server listens on for requests. The -H flag specifies the host name of the LDAP server that the certificates are published to. The -D and -w flags specify the administrator DN and password for the LDAP server specified in the -H flag. The -i flag specifies the location in the LDAP hierarchy to publish certificates to. If a certificate is created for the user test1, the certificate DN will be `cn=test1,ou=cert,o=mycompany.example,c=us`. You will have to supply a password and confirm it for the pkiuser user account that is created. **mksecpki** generates pages of output and will generate error messages. You can safely ignore those errors if **mksecpki** displays Configuration is completed.

```
# mksecpki -u pkiuser -f /usr/cas/server/iafile -p 1077 -H
ldap.mycompany.example \
-D cn=admin,ou=mydept,o=mycompany.example,c=us -w mysecret \
-i ou=cert,o=mycompany.example,c=us
Enter new Password: abc123
Enter the new password again: abc123
...
```

Please wait for the configuration to complete.
Configuration is completed.

```
# keytool -list -v -keystore /usr/lib/security/pki/trusted.pkcs12 -keyalg RSA
-storetype pkcs12ks
Enter keystore password: abc123
```

```
Keystore type: pkcs12ks
Keystore provider: IBMJCE
```

Your keystore contains 1 entry:

```
Alias name: trustedkey
Creation date: Wed Dec 31 18:00:00 CST 1969
Entry type: keyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=trusted key
Issuer: CN=trusted key
Serial number: 3d6ba75d
Valid from: Tue Aug 27 11:22:53 CDT 2002 until: Mon Nov 25 10:22:53 CST 2002
Certificate fingerprints:
    MD5:  6A:95:51:9C:AA:2F:B2:29:3A:30:A9:FD:CC:22:41:0C
    SHA1: 04:7D:C6:A7:7C:27:04:2B:1D:B5:CA:4C:F8:B6:D8:34:69:1E:36:7A
```

```
*****
*****
```

The Certificate Authentication Service server and client are configured by modifying the files `acct.cfg`, `ca.cfg`, and `policy.cfg`, located in `/usr/lib/security/pki`. You can configure the Certificate Authentication Service using SMIT or by editing the files directly.

The `acct.cfg` file contains private account information for the Certificate Authentication Service components. The file contains both LDAP and CA stanzas. The LDAP stanzas contain the host name of the LDAP server, the certificate directory tree DN, and the PKI administration DN and password, which are required to publish certificates into the LDAP directory tree. The CA stanzas contain the certificate creation reference number and password pairs, which are required to communicate with the CA to create certificates. The CA stanzas could optionally contain the label and password for the trusted signing key, used in certificate verification.

The `ca.cfg` contains public information for the Certificate Authentication Service components. The stanzas contain URI for the CA server, encryption algorithm type, key sizes, and signing hash types.

The `policy.cfg` file contains attributes about policies that the Certificate Authentication Service components enforce. The most commonly modified

stanza is the newuser stanza, which is used to customize the **mkuser** command. It contains initial user password, keystore location, validity period, and CA name.

To configure the ca.cfg portion of the Certificate Authentication Service server using SMIT do the following:

```
smitty pki
Select Change/Show a Certificate Authority
Enter local for the Certificate Authority Name
Modify all fields to match Figure 9-10.
Press Enter to commit changes
```

Change / Show a Certificate Authority			
Type or select values in entry fields. Press Enter AFTER making all desired changes.			
* Certificate Authority Name	[Entry Fields]		
Service Module Name	local		
Pathname of CA's Certificate	[usr/lib/security/pki/JSM1.sm1]		+/
Pathname of CA's Trusted Key	[usr/lib/security/pki/CERTFILE_NAME.der]		/
URI of the Certificate Authority Server	[file:usr/lib/security/pki/trusted.pkcs12]		/
Certificate Distribution Point	[cmp://ca.mycompany.example:1077]		
Certificate Revocation List (CRL) URI	[test]		
Default Certificate Distinguished Name	[ldap://ldap.mycompany.example/ou=cert,o=mycompany.example,c=us]		
Default Certificate Subject Alternate Name URI	[ou=cert,o=mycompany.example,c=us]		
Public Key Algorithm	[http://www.mycompany.example/]		+
Public Key Size (in bits)	[RSA]		++
MAX. Communications Retries	[1024]		#
Signing Hash Algorithm	[5]		+
	[MD5]		
F1=Help	F2=Refresh	F3=Cancel	F4=List
F5=Reset	F6=Command	F7=Edit	F8=Image
F9=Shell	F10=Exit	Enter=Do	

Figure 9-10 SMIT screen of PKI - Change/Show a Certificate Authority

To configure the CA stanzas in the acct.cfg file using SMIT do the following:

1. smitty pki
2. Select **Change/Show a CA Account**.
3. Enter the local for the certificate authority name.
4. Modify all fields to match Figure 9-11 on page 635.
5. Press Enter to commit changes.

Change / Show a CA Account

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

* Certificate Authority Name	[Entry Fields]	
Certificate Creation Reference Number	[12345678]	+
Certificate Creation Password	[password1234]	
Certificate Revocation Reference Number	[89347389]	
Certificate Revocation Password	[notpassword123]	
Trusted Key Label	[trustedkey]	
Trusted Key Password	[abc123]	

F1=Help
F5=Reset
F9=Shell

F2=Refresh
F6=Command
F10=Exit

F3=Cancel
F7=Edit
Enter=Do

F4=List
F8=Image

Figure 9-11 SMIT screen of PKI - Change/Show a CA Account

To configure the LDAP stanzas in the acct.cfg file using SMIT do the following:

1. smitty pki
2. Select **Add/Change/Show an LDAP Account**.
3. Enter the local for the certificate authority name.
4. Modify all fields to match Figure 9-12.
5. Press Enter to commit changes.

Add / Change / Show an LDAP Account

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

Administrative User Name	[Entry Fields]
Administrative Password	[ou=pkidata,cn=aixdata,ou=mydept,o=mycompany.example,c=us]
Server Name	[secret]
Suffix	[ldap.mycompany.example]
	[ou=pkidata,cn=aixdata,ou=mydept,o=mycompany.example,c=us]

F1=Help
F5=Reset
F9=Shell

F2=Refresh
F6=Command
F10=Exit

F3=Cancel
F7=Edit
Enter=Do

F4=List
F8=Image

Figure 9-12 SMIT screen of PKI - Add/Change/Show an LDAP Account

To configure the policy.cfg file using SMIT do the following:

1. smitty pki
2. Select **Add/Change/Show the Policy**.
3. Modify all fields to match Figure 9-13.
4. Press Enter to commit changes.

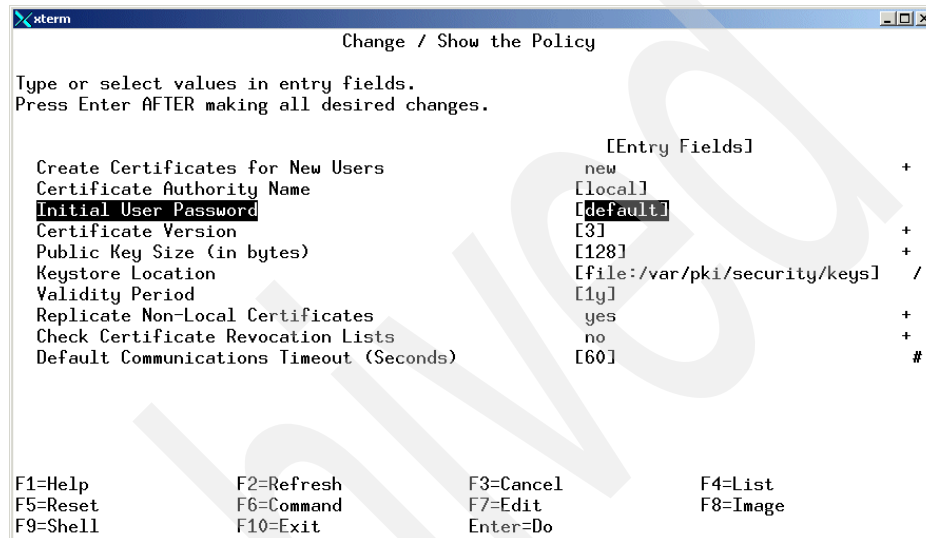


Figure 9-13 SMIT screen of PKI - Change/Show the Policy

To configure the Certificate Authentication Service using command line utilities, you must edit the acct.cfg, ca.cfg, and policy.cfg configuration files located in /usr/lib/security/pki. Insert the stanzas in the following sections into the appropriate configuration files. If a stanza with the same name already exists, then replace the existing stanza with the ones below.

Configuration file /usr/lib/security/pki/acct.cfg

The following is a sample configuration file.

```
ldap:
    ldappkiadmin      =
    "ou=pkidata,cn=aixdata,ou=mydept,o=mycompany.example,c=us"
    ldappkiadmpwd     = "secret"
    ldapservers       = "ldap.mycompany.example"
    ldapsuffix        =
    "ou=pkidata,cn=aixdata,ou=mydept,o=mycompany.example,c=us"

local:
```

```

carefnum      = 12345678
capasswd      = "password1234"
rvrefnum      = 89347389
rvpasswd      = "notpassword123"
keylabel      = "trustedkey"
keypasswd     = "abc123"

```

Configuration file /usr/lib/security/pki/ca.cfg

The following is a sample configuration file.

```

local:
  program      = /usr/lib/security/pki/JSML.sm1
  certfile     = /usr/lib/security/pki/CERTFILE_NAME.der
  trustedkey    = file:/usr/lib/security/pki/trusted.pkcs12
  server       = "cmp://ca.mycompany.example:1077"
  cdp          = test
  crl          =
"ldap://ldap.mycompany.example/ou=cert,o=mycompany.example,c=us"
  dn           = "ou=cert,o=mycompany.example,c=us"
  url          = "http://www.mycompany.example/"
  algorithm     = RSA
  keysize      = 1024
  retries      = 5
  signinghash  = MD5

```

Configuration file /usr/lib/security/pki/policy.cfg

The following is a sample configuration file.

```

newuser:
  cert      = new
  ca        = local
  passwd    = default
  version   = 3
  keysize   = 128
  keystore  = file:/var/pki/security/keys
  validity  = 1y

storage:
  replicate = yes

crl:
  check     = no

comm:
  timeout   = 60

```

The methods.cfg file specifies the definitions of the authentication grammar used by the registry and SYSTEM attributes. The PKI stanza below defines the

method to be used for PKI authentication. The PKIfiles stanza defines PKI for authentication, and user attributes are stored in local files. Insert the following stanzas into your /usr/lib/security/methods.cfg configuration file. If stanzas with the same names already exist, then carefully merge the following stanzas into your configuration.

```
PKI:
    program = /usr/lib/security/PKI
    options = authonly

PKIfiles:
    options = auth=PKI,db=BUILTIN
```

9.14.4 Common user and administrator tasks using PKI

Now that Certificate Authentication Service is configured, the most common administration task is user management. When adding users to the PKI, you will either be creating users from scratch or migrating existing users. The sections below describe how to create PKI-enabled users for each scenario.

To migrate an existing user to PKI authentication, there are four steps that you must perform:

1. Use the **certcreate** command to request a new certificate from the certificate authority (CA). The CA returns a DER-encoded certificate and publishes the certificate into the CA repository. The CA repository for this example is `ou=cert,o=mycompany.example,c=us`.
2. Use the **certadd** command to publish the certificate into LDAP, in the local AIX repository. The local AIX repository for this example is located in `ou=pkidata,cn=aixdata,ou=mydept,o=mycompany.example,c=us`.
3. Use the **certverify** command to verify that the invoker is in possession of the private key for the certificate. Until a certificate is verified, AIX will consider that certificate untrusted. Use the **certlist** command to determine the state of the verified attribute.
4. Use the **chuser** command to modify the user's SYSTEM and registry attributes to PKIfiles. Use the **chuser** command to set the user's `auth_cert` attribute to the tag of the certificate to log in to AIX.

It is also possible to have the non-root user run the steps 1–3 and then the administrator would run step 4 as root. See the section below for an example of migrating a user named test3 to PKI authentication.

```
# certcreate -f test3.der -l defaultLabel cn=test3 test3
Enter password for the keystore : test3
Re-enter password for the keystore : test3
```

```

# certlist -f ALL test3
test3:
    auth_cert=
    distinguished_name=c=us,o=mycompany.example,ou=cert,cn=test3
    alternate_name=
    validafter=0830103702
    validuntil=0827152403
    issuer=c=us,o=mycompany.example,ou=cert
    tag=tag1
    verified=false
    label=defaultLabel
    keystore=file:/var/pki/security/keys/test3
    serialnumber=0D

# certverify tag1 test3
Enter password for the keystore : test3
# certlist -f ALL test3
test3:
    auth_cert=
    distinguished_name=c=us,o=mycompany.example,ou=cert,cn=test3
    alternate_name=
    validafter=0830103702
    validuntil=0827152403
    issuer=c=us,o=mycompany.example,ou=cert
    tag=tag1
    verified=true
    label=defaultLabel
    keystore=file:/var/pki/security/keys/test3
    serialnumber=0D

# chuser SYSTEM="PKIfiles" registry=PKIfiles test3

# chuser -R PKIfiles auth_cert=tag1 test3
# lsuser -R PKIfiles test3
test3 id=209 pgrp=staff groups=staff home=/home/test3 shell=/usr/bin/ksh
login=true su=true rlogin=true daemon=true admin=false sugroups=ALL admgroups=
tpath=nosak ttys=ALL expires=0 auth1=SYSTEM auth2=NONE umask=22
registry=PKIfiles SYSTEM=PKIfiles logintimes= loginretries=0 pldwarntime=0
account_locked=false minage=0 maxage=0 maxexpired=-1 minalpha=0 minother=0
mindiff=0 maxrepeats=8 minlen=0 histexpire=0 histsize=0 pldchecks= dictionlist=
fsize=3097151 cpu=-1 data=262144 stack=65536 core=2097151 rss=65536
nofiles=2000 roles= auth_cert=tag1
subject_DN=c=us,o=mycompany.example,ou=cert,cn=test3
subject_altname=c=us,o=mycompany.example,ou=cert,cn=test3 valid_after=20020830
valid_until=20030827 issuer=c=us,o=mycompany.example,ou=cert

```

To create a PKI-authenticated user account from scratch, you just need to run the **mkuser** command. The **mkuser** command gets the default values for certificate

validity dates, initial keystore password, CA to request certificate from, and location of the keystore from the newuser stanza in the policy.cfg file.

```
# mkuser -R PKIfiles SYSTEM=PKIfiles registry=PKIfiles test1

# certlist -f ALL test1
test1:
    auth_cert=auth_cert
    distinguished_name=c=us,o=mycompany.example,ou=cert,cn=test1
    alternate_name=email=test1@itsc.austin.ibm.com
    validafter=0830091302
    validuntil=0827152403
    issuer=c=us,o=mycompany.example,ou=cert
    tag=auth_cert
    verified=true
    label=DefaultLabel
    keystore=file:/var/pki/security/keys/test1
    serialnumber=07

# lsuser -R PKIfiles test1
test1 id=205 pgrp=staff groups=staff home=/home/test1 shell=/usr/bin/ksh
login=true su=true rlogin=true daemon=true admin=false sugroups=ALL admgroups=
tpath=nosak ttys=ALL expires=0 auth1=SYSTEM auth2=NONE umask=22
registry=PKIfiles SYSTEM=compat logintimes= loginretries=0 pldwarntime=0
account_locked=false minage=0 maxage=0 maxexpired=-1 minalpha=0 minother=0
mindiff=0 maxrepeats=8 minlen=0 histexpire=0 histsize=0 pwdchecks= dictionlist=
fsize=3097151 cpu=-1 data=262144 stack=65536 core=2097151 rss=65536
nofiles=2000 roles= auth_cert=auth_cert
subject_DN=c=us,o=mycompany.example,ou=cert,cn=test1
subject_altname=email=test1@itsc.austin.ibm.com valid_after=20020830
valid_until=20030827 issuer=c=us,o=mycompany.example,ou=cert
```

Note: Accounts created by the **mkuser** command are immediately available for login using the default password specified in the policy.cfg file. It is good security practice to immediately change the user's local keystore password using the **keypasswd** command. See below for an example of using **keypasswd**.

```
# keypasswd -k test1
Old password: default
New password: test1
Re-enter password for the keystore : test1
```

9.14.5 Process authentication group commands

Version 5.2 now supports several new process authentication group (PAG) commands: **paginit**, **pagdel**, and **paglist**. The PAG is a data structure that associates user-authentication data with processes. If the PAG mechanism is enabled and you are using the Certificate Authentication Service, the user's

authentication certificate is associated with the user's login shell. When the shell spawns new processes, the PAG information is propagated to each child. By default the PAG mechanism is not enabled. The Certificate Authentication Service does not require the PAG mechanism to work, but will exploit it, if enabled. To enable the PAG mechanism you must start the `certdaemon` daemon. The following example shows how to use the `mkinitab` command to add the `certdaemon` to the `/etc/inittab` file, so the `certdaemon` daemon will restart upon reboot.

```
# mkinitab "certdaemon:2:wait:/usr/sbin/certdaemon"
# lsinitab certdaemon
certdaemon:2:wait:/usr/sbin/certdaemon
```

The `paglist` command allows you to display the PAG associated with the current process. The following example shows the PAG for the PKI user `test3`.

```
$ who am i
test3      pts/14      Aug 30 15:20      (9.3.4.144)
$ paglist
PAG_DATA=308202c730820230a00302010202010d300d06092a864886f70d01010505003038310b
3009060355040613027573311a3018060355040a13116d79636f6d70616e792e6578616d706c653
10d300b060355040b130463657274301e170d3032303833303135333735365a170d303330383237
3230323435335a3048310b3009060355040613027573311a3018060355040a1311...
1d7a532cf7f8f3d47b3f417f053f85745e07722b9314dc9462e358aefc46b9c0c2d4ee125e70e3d
5dff70abc4fec306deae2444c95049d52d565a24e1ee77736e23bfadce15af728273264b74cb6c9
289cf9ddf23fe086e6437ef5350f1f6a74873b175955fda2f28a53726d1db921b648a
```

The `paginit` command allows you to authenticate the current user and to create a PAG association. This is often used when you use the `su` command to become another user. The following example shows when you might need the `paginit` command. When the root user used the `su` command to become the PKI user `test3`, a password was never entered. Since a password was never entered, the `test3` user was never authenticated with PKI. The `paginit` command can now be used to authenticate after the fact.

```
# su - test3
$ paglist
PAG_DATA=
$ paginit
test3's Password:
$ paglist
PAG_DATA=308202c730820230a00302010202010d300d06092a864886f70d01010505003038310b
3009060355040613027573311a3018060355040a13116d79636f6d70616e792e6578616d706c653
10d300b060355040b130463657274301e170d3032303833303135333735365a170d303330383237
3230323435335a3048310b3009060355040613027573311a3018060355040a1311
...
fda2f28a53726d1db921b648a
```

The **pagdel** command will delete the current PAG associated with the current process. The following example shows how to use the **pagdel** command:

```
$ paglist
PAG_DATA=308202c730820230a00302010202010d300d06092a864886f70d01010505003038310b
3009060355040613027573311a3018060355040a13116d79636f6d70616e792e6578616d706c653
10d300b060355040b130463657274301e170d3032303833303135333735365a170d303330383237
3230323435335a3048310b3009060355040613027573311a3018060355040a1311...
44c95049d52d565a24e1ee77736e23bfadce15af728273264b74cb6c9289cf9ddf23fe086e6437e
f5350f1f6a74873b175955fda2f28a53726d1db921b648a
$ pagdel
$ paglist
PAG_DATA=
```

Known limitations

There are some limitations with the AIX 5L Certificate Authentication Service/PKI components as of this writing. The certificate authority cannot generate certificates with a distinguished name (DN) with multiple object identifiers of the same type. For example, you can generate a certificate with a DN of `cn=test3,ou=cert,o=mycompany.example,c=us`, but generating a certificate with a DN of `cn=test2,ou=cert,ou=mydept,o=mycompany.example,c=us` will fail. This restriction will be removed in the next release of Certificate Authentication Services.

Currently only the certificate authority supplied with AIX 5L is supported. Third-party certificate authorities that use certificate management protocol (CMP) should work, but this has not been tested or supported. Non-file keystores such as smart cards or LDAP are currently not supported.

9.15 CAPP and EAL4+ security install (5.2.0)

Version 5.2 allows controlled access protection profile and evaluation assurance level 4+ to be specified at system install time. This is the replacement for the C2 security install with previous versions. It is only possible to install this software with a new and complete overwrite install.

9.15.1 Packaging summary

Prior to Version 5.2, it was necessary to install common criteria security code from the special order security CDs that replaced the normal AIX product CDs. Version 5.2 allows controlled access protection profile and evaluation assurance level 4+ (CAPP/EAL4+) to be selected in the More options screen on the Install menu. The code is now located on the base operating system install CD-ROMs.

This option is available for new and complete overwrite install only and is only available for 64-bit systems. If CAPP/EAL4+ is selected, then TCB, Enable 64bit Kernel, and create JFS2 File Systems are all set to yes. The only desktop choices are CDE or none, and Enable System Backups to install on any system (install all devices and kernels) is set to no. The language the system will be installed with must be either English or C. The Install More Software option will not be offered.

9.15.2 Installation steps

The machine needs to be booted into the system maintenance screen. Ensure that Version 5.2, CD1 is in the drive and either the bootlist is set to read the CD before either a disk or network boot, or the boot process is interrupted with the 5 or F5 key sequence.

Select the terminal as the system console and press Enter, then select the language of your choice for the install (default is English).

This will go into the following screen, where option 2, Change/Show Installation Settings should be selected, as shown in Figure 9-14.

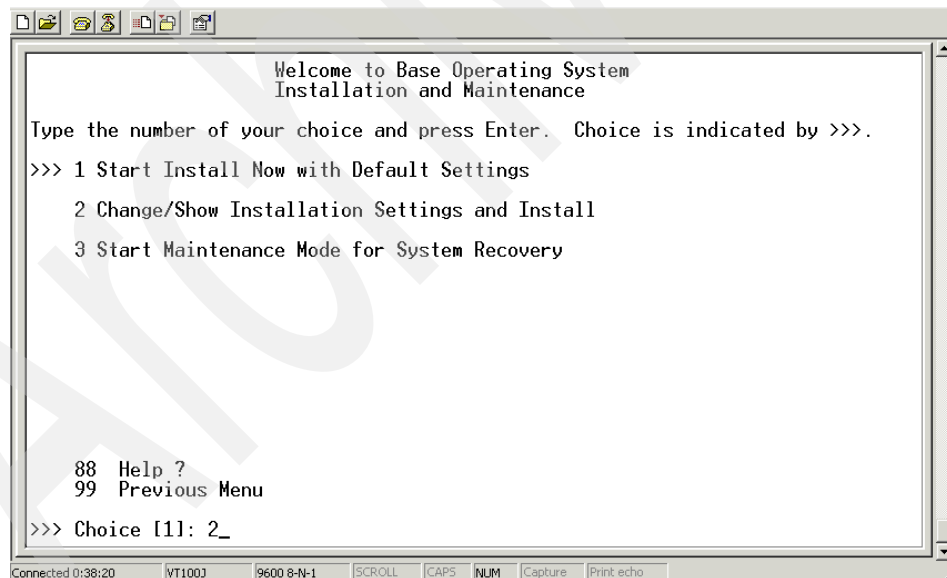


Figure 9-14 BOS Installation and Maintenance screen

From the Installation and Settings screen (Figure 9-15 on page 644), select option 1 and change the method of installation to a new and complete overwrite.

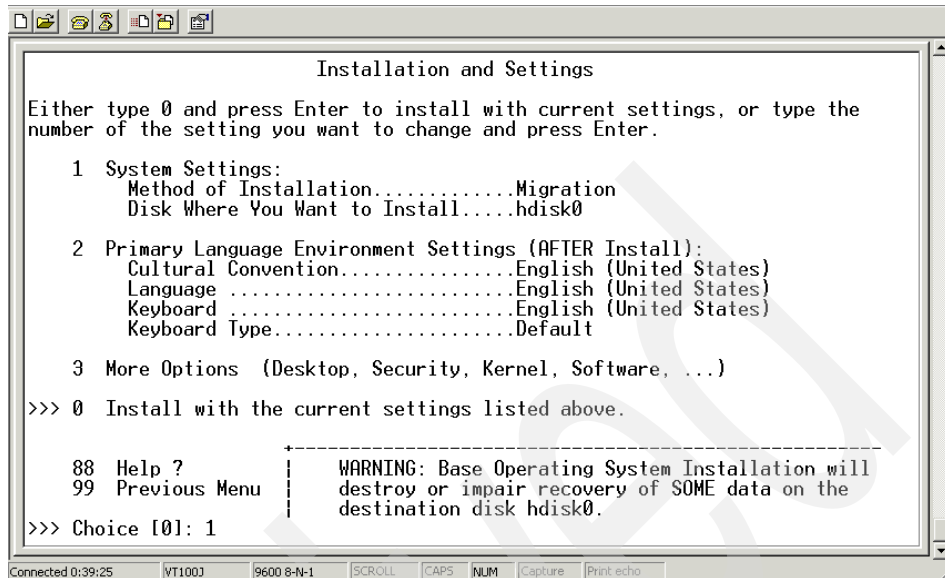


Figure 9-15 Installation and Settings screen

By selecting option 1, you are taken into the Change Method of Installation screen, and here option 1, New and Complete Overwrite should be selected, as shown in Figure 9-16.

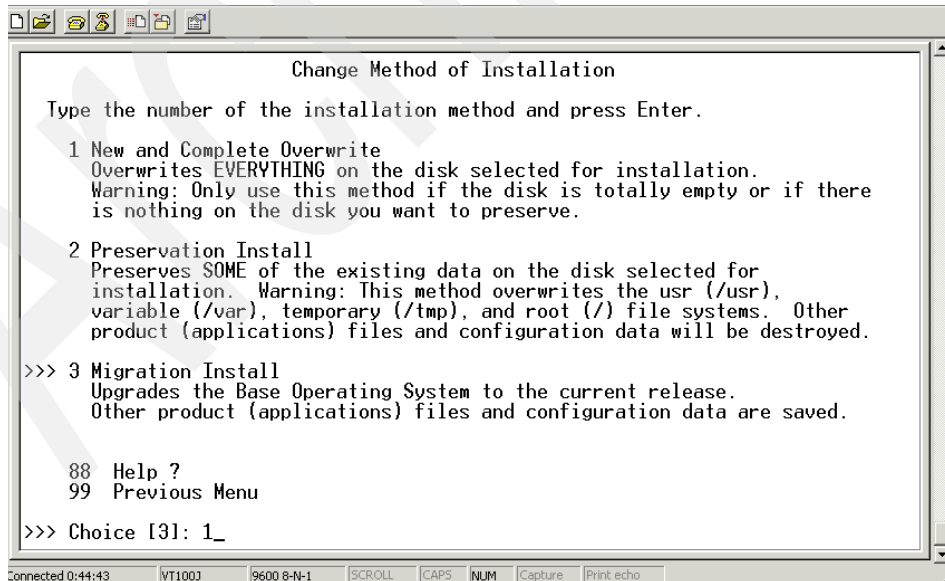


Figure 9-16 Change method of installation to new and complete overwrite

After selecting option 1, the user is taken into the Change Disk(s) screen automatically, where it is possible to select the disks for rootvg (Figure 9-17).

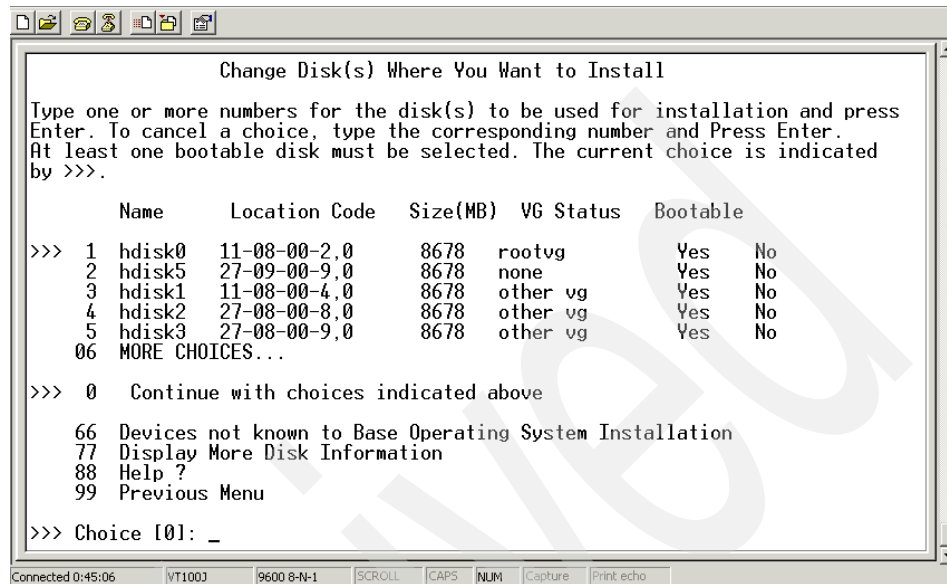


Figure 9-17 Change disks to BOS install

Select option 0, or in this case press Enter (as option 0 is already selected), The user is then returned to the Installation and Settings menu, but with New and Complete Overwrite Install selected. From this screen select option 3, More Options, as shown in Figure 9-18 on page 646.

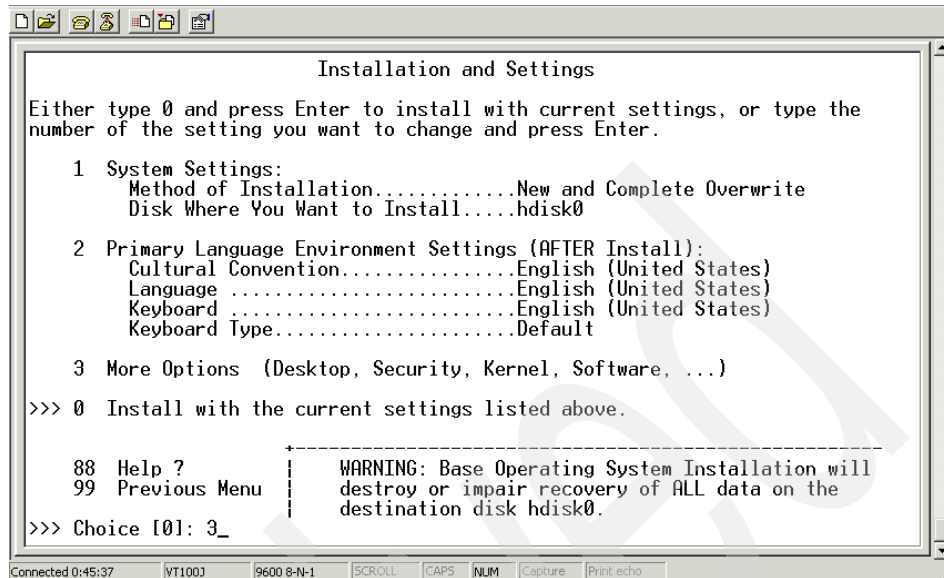


Figure 9-18 Installation and Settings screen, selecting option 3, More Options

This goes into the further options screen where it is possible to install CAPP/ EAL4+. The screen initially looks like Figure 9-19.

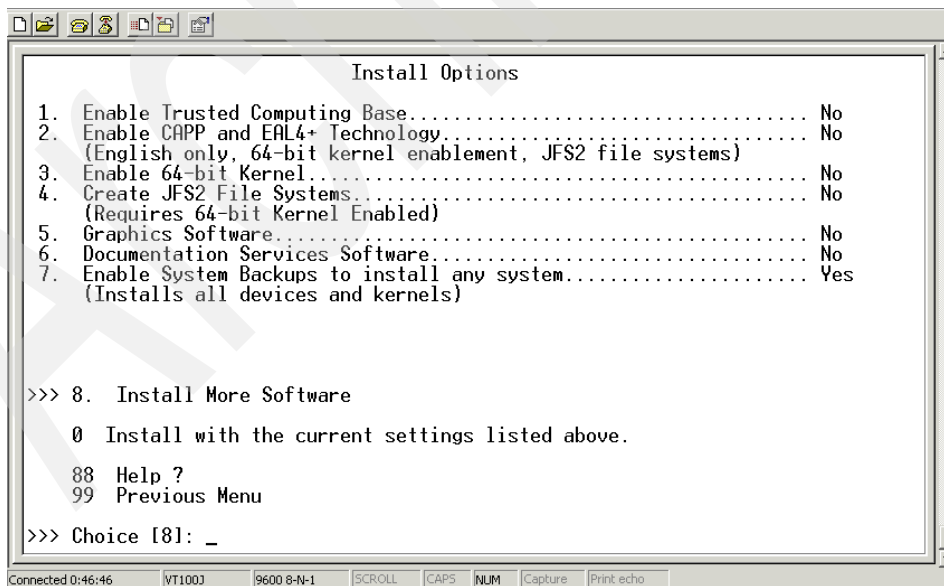


Figure 9-19 Install Options screen

By selecting option 2, Enable CAPP/EAL4+ Technology, the following options are automatically selected:

- ▶ Trusted computing base (option 1)
- ▶ 64-bit kernel (option 3)
- ▶ JFS2 file systems (option 4)

Prior to selecting CAPP/EAL4+ install, it is possible to enable system backups to install any system (option 7). Also, there is an option to install more software, option 8. Once CAPP/EAL4+ is selected, option 7 will be set to no and option 8 will disappear altogether. In Figure 9-20, only option 2 (CAPP and EAL4+) was selected. This caused the other three options to be automatically selected.

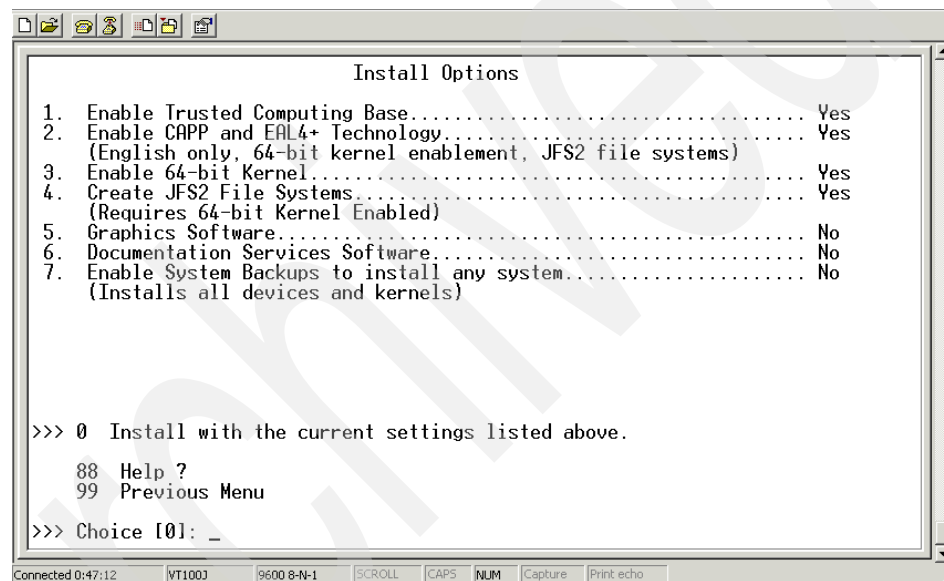


Figure 9-20 Selecting CAPP and EAL4+

CAPP/EAL4+ has prerequisites of TCB, the 64-bit kernel, and JFS2; for this reason it can only be installed on 64-bit systems. It is not possible to deselect any of these options and still install CAPP/EAL4+. If either option is deselected, the 64-bit CAPP and EAL4+ are automatically deselected.

Also note that the other two others changes mentioned previously have occurred, just by selecting the install of CAPP/EAL4+. Namely that "Enable System Backups to install any system" has been set to no and Install More Software (option 8) has gone. If "Enable System Backups to install any system" was set to yes, then Enable CAPP and EAL4+ Technology will be set to no.

By selecting option 0, the user is presented with a summary screen of what is to be installed, and can start the installation from that screen. This is the only place that shows that by selecting CAPP and EAL4+ Technology the language convention has been changed back to C or En_US (the only language conventions that are compatible with CAPP/EAL4+).

9.16 Tivoli readiness

AIX 5L for the POWER architecture is compliant with the specifications that the *Tivoli Ready* mark requires for operating systems.

The difference from AIX Version 4.3 is that the Tivoli Management Agent (TMA) is now part of the base CDs, and is installed automatically with a normal AIX installation.

The following lines are a list of filesets installed for Tivoli readiness:

```
# lsllpp -L "Tivoli*"
Fileset                                Level  State  Description
-----
Tivoli_Management_Agent.client.rte    3.2.0.0  C      Management Agent runtime"
```

9.17 TCB integration with Tivoli Risk Manager (5.2.0)

Version 5.2 allows Trusted Computing Base to interface with Tivoli Risk Manager.

The Trusted Computing Base (TCB) can only be enabled at BOS installation and can be selected from the Advanced Options. TCB allows the administrator to access the trusted shell, trusted processes, the Secure Attention Key (SAK), and system integrity checking (**tcbck** command), which also runs at system boot time.

Version 5.2 allows Tivoli Risk Manager to report on configured security exceptions as identified by the **tcbck** command. The command must be run with the -o option in order to output to the syslog.

The Tivoli logfile adapter for Risk Manager can then be configured to read the syslog file and report on any exceptions found by the **tcbck** command. There are two AIX-level configuration files that Risk Manager uses that do not need to be changed; otherwise all configuration is done at the Tivoli level.

The two configuration files are:

- ▶ /usr/lib/security/risk-manager/tcb.baroc
- ▶ /usr/lib/security/risk-manager/tcb.fmt

Most of the configuration is needed Tivoli Risk Manager, the logfile adapter is an enablement feature at the AIX level.

9.18 Enterprise Identity Mapping (5.2.0)

The Enterprise Identity Mapping (EIM) infrastructure has two primary objectives:

- ▶ Enable the creation of heterogeneous cross-platform operating system functions and applications that do not force administrators to manage additional user registries and security semantics.
- ▶ Enable SWG/Tivoli and business partners to build a single-point-of-management enterprise user management application. To accomplish these objectives, we provide two sets of EIM APIs:
 - A set that handles creating, changing, retrieving, and removing identity mapping information.
 - A set of APIs that provides the function needed to create, change, and remove local user identities residing in IBM-defined user registries.

Both sets of APIs rely on infrastructure built on top of LDAP, LDAP protocol, and legacy interfaces to each platform's user registry function (user profile SPIs and APIs for AS/400, RACF interfaces for OS/390, and user registry interfaces for AIX).

For example, John Smith's ID may be JSmith on hostname1 and JohnSmith on hostname2. EIM enables John to be treated as a single user on both machines, even though his IDs have not changed on either machine. The EIM APIs are in the library libeim.a, which is part of the bos.eim fileset. This API is provided so application programmers can make use of this function.

9.19 Enhanced login privacy (5.2.0)

AIX 5L Version 5.2 now supports enhanced security options regarding the user's interface. On the default AIX's login screen, the user name is visible when entered and the password line also includes the user name. In some security environments, displaying the user name on the screen is considered a security exposure. In Version 5.2, the administrator has the option to change the login

password prompt and to hide the user name from login and system messages. These settings can be configured as the system default or on a per-port basis.

See the following example for the default behavior for logging in with **telnet**. The user is logging in as test9 and the user name test9 is displayed twice. The **/usr/bin/su** command also echoes the user name test8 in the password prompt.

```
telnet (server1)
```

```
AIX Version 5
(C) Copyrights by IBM and by others 1982, 2000.
login: test9
test9's Password:
...
$ su - test8
test8's Password:
$
```

The new attributes for login privacy are located in **/etc/security/login.cfg**. The **pwdprompt** attribute defines the password prompt message when asking for the password during login. The **usernameecho** attribute is a boolean value that determines if the user name is displayed during log in and security-related messages. If **usernameecho** is false, the user name will be hidden during log in and security-related messages. If **usernameecho** is true (the default), user names are displayed as normal. To set these attributes on a per-port basis, you must create a new stanza, if necessary for that port (for example, **/dev/lft0**) and add the attributes to that port. If you want to make these attributes system wide, add them to the default stanza. Attributes in a port-specific stanza, will override attributes in the default stanza.

The following example shows the result of changing the system-wide password prompt to Password:

```
# chsec -f /etc/security/login.cfg -s default -a pwdprompt="Password:"
```

```
telnet (server1)
```

```
AIX Version 5
(C) Copyrights by IBM and by others 1982, 2000.
login: root
Password:
```

In the following example, the password prompt is reset to default and the **usernameecho** is set to false. The output for the **telnet** session is below. Notice the user names for the **/usr/bin/su** and **/usr/bin/passwd** commands are hidden.

```
# chsec -f /etc/security/login.cfg -s default -a pwdprompt=
# chsec -f /etc/security/login.cfg -s default -a usernameecho=false
```



```
telnet (server1)

AIX Version 5
(C) Copyrights by IBM and by others 1982, 2000.
login:
****'s Password:

...
$ passwd
Changing password for "****"
****'s Old password:
****'s New password:
Enter the new password again:

$ su - test8
3004-500 User "****" does not exist.

$ su - test4
****'s Password:
```

The following example shows how to specify the `usernameecho` attribute for a specific port (for example, `/dev/lft0`). Attributes specified in per-port stanzas override the default stanza.

```
chsec -f /etc/security/login.cfg -s /dev/lft0 -a usernameecho=false
```

With the password prompt attribute `pwdprompt` set, the specified string is used by the `su` command when invoked by a non-root user, but the string will not be used by the `passwd` command to change the existing user password.

9.20 Cryptographically secure pseudo-random numbers

AIX 5L Version 5.2 now supports a cryptographically secure pseudo-random number generator (PRNG). Random numbers are extremely important for any sort of cryptographic application. Random numbers are used to generate session keys, salts used for hashed passwords, and initializing public key certificates. If the generated random numbers are easily predictable, any application using those insecure numbers is also insecure. No algorithms or protocol can fix problems with random number generation.

The PRNG on Version 5.2 is based on the Yarrow engine and collects entropy from the running system and feeds an entropy pool to seed a PRNG. The entropy gathering process selects three hardware devices upon startup such as, SSA, Ethernet, and SCSI adapters. The entropy-gathering daemon detects hardware interrupts or network packets and determines the times between two events. These timings are then put into the entropy pool.

The API for accessing the PRNG is quite simple. An application just has to open the `/dev/random` or `/dev/urandom` file and read the required number of bytes of the special device. The `/dev/random` and `/dev/urandom` have different behaviors when the pool of entropy is exhausted or requires reseeding. The `/dev/random` device will have the reading application block until more entropy is gathered. The `/dev/urandom` device will behave the same as `/dev/random`, but when entropy is exhausted it will fall back and generate entropy using a software algorithm. The level of randomness of the numbers generated by the software algorithm is not as high as the entropy gathered from the running system.

The PRNG automatically keeps the entropy pools replenished and reseeds it occasionally. When the entropy pool is half empty, the entropy gatherer will intercept the hardware interrupts and network packets until the entropy is replenished. There is a slight performance penalty while entropy is being gathered. When the pools are full, the entropy-gathering process goes idle and no longer effects machine performance.

For more information on the Yarrow engine, refer to the Counterpane Labs home page at the following URL:

<http://www.counterpane.com/yarrow.html>

9.21 IP security enhancements (5.2.0)

The following are the security enhancements pertaining to IP in AIX 5L Version 5.2.

9.21.1 IKE components using `/dev/random`

In Version 5.2, the AIX Internet Key Exchange (IKE) components now use the system-wide pseudo-random number generator (PRNG) as the random number source. For more information about the AIX random number generator, refer to 9.20, “Cryptographically secure pseudo-random numbers” on page 651. The `ikentropy` daemon introduced in Version 5.0 to generate entropy was removed in Version 5.2.

9.21.2 Diffie-Hellman group 5 supported

The AIX IKE has now been enhanced to support Diffie-Hellman (DH) group 5. Prior releases of the AIX only supported DH groups 1 and 2. Diffie-Hellman key exchange is a public key cryptosystem where public values are exchanged to arrive at a symmetric key among the end entities. The OAKLEY Key Determination Protocol defines five well-known DH groups. Each DH group defines a prime and a generator function to create symmetric key. DH groups 1,

2, and 5 are all modular exponentiation group primes (MODP) with 768, 1024, and 1536 bits, respectively. Since DH group 5 has greater entropy than DH groups 1 and 2, symmetric keys generated from DH group 5 will be more secure but require more processing time.

The Document Type Definition (DTD) of the IKE database configuration file has been extended to support DH group 5. The following is an excerpt of the `ikedb` command of the modified IKETransform and IPSecProtection elements.

```
# ipsec -o
...
<!-- ===== IKETransform =====
      IKETransform. A list of these will be used for Phase 1 SA
      Negotiations.
-->
<!ELEMENT IKETransform EMPTY>
<!-- ATTLIST IKETransform
      IKE_AuthenticationMethod (Preshared_key | RSA_signatures)
                                "Preshared_key"
      IKE_Encryption (DES-CBC | 3DES-CBC) "3DES-CBC"
      IKE_Hash (SHA | MD5) "SHA"
      IKE_DHGroup (1 | 2 | 5) "2"
      IKE_KeyRefreshMinutes CDATA "480"
>
...
<!-- ===== IPSecProtection =====
      IPSecProtection.
-->
<!ELEMENT IPSecProtection EMPTY>
<!-- ATTLIST IPSecProtection
      IPSec_ProtectionName ID #REQUIRED
      IPSec_ProposalRefs IDREFS #REQUIRED
      IPSec_Role (Initiator|Responder|Both|Neither) "Both"
      IPSec_KeyOverlap CDATA "5"
      IPSec_Flags_UseCommitBit (Yes | No) "No"
      IPSec_Flags_UseLifeSize (Yes | No) "No"
      IPSec_InitiatorDHGroup (0 | 1 | 2 | 5) "0"
      IPSec_ResponderDHGroup CDATA "NO_PFS GROUP_1 GROUP_2 GROUP_5"
      IPSec_ResponderKeyRefreshMaxMinutes CDATA "120"
      IPSec_ResponderKeyRefreshMinMinutes CDATA "1"
      IPSec_ResponderKeyRefreshMaxKB CDATA #IMPLIED
      IPSec_ResponderKeyRefreshMinKB CDATA #IMPLIED
>
...

```

The following example is an IKEtransform element with the `IKE_DHGroup` attribute specifying IKE to use Diffie-Hellman group 5 for the key management tunnel (phase one).

```

<IKETransform
    IKE_Hash="MD5"
    IKE_DHGroup="5"
    IKE_Encryption="DES-CBC"
    IKE_KeyRefreshMinutes="480"
    IKE_AuthenticationMethod="Preshared_key"/>

```

The following example shows an IPsecProtection element specifying the attributes to create the data management tunnel (phase two). The IPsec_InitiatorDHGroup attribute specifies using DH group 5 if this machine is initiating a tunnel. The IPsec_ResponderDHGroup attribute specifies allowing either no perfect forwarding secrecy (PFS) or PFS using DH group 1, 2, or 5 when this machine is responding to a tunnel request.

```

<IPsecProtection
    IPsec_Role="Both"
    IPsec_KeyOverlap="15"
    IPsec_ProposalRefs="server3_toprivenet_PROPOSAL "
    IPsec_ProtectionName="server3_toprivenet_POLICY"
    IPsec_InitiatorDHGroup="5"
    IPsec_ResponderDHGroup="NO_PFS GROUP_1 GROUP_2 GROUP_5"
    IPsec_Flags_UseLifeSize="No"
    IPsec_Flags_UseCommitBit="No"
    IPsec_ResponderKeyRefreshMaxKB="1000000"
    IPsec_ResponderKeyRefreshMinKB="50"
    IPsec_ResponderKeyRefreshMaxMinutes="60"
    IPsec_ResponderKeyRefreshMinMinutes="2"/>

```

9.21.3 Generic data management tunnel support

The AIX IKE now supports the creation of a generic data management tunnel, also known as a phase 2 tunnel. This feature is used mainly when an IPSEC endpoint is using dynamic host configuration protocol (DHCP) to assign IP addresses. Normally data management tunnels are identified by their IP address; with DHCP the endpoint address is dynamic. The generic data management tunnel will be used if a request was authenticated by phase 1 and an IP address is not specifically configured in the database.

The generic data management tunnel is not a real tunnel, but a tunnel definition that is used when an incoming data management message does not match any defined data management tunnels. Defining a generic data management tunnel is optional and there can only be one generic data management tunnel per key management tunnel definition. It can only be used in the case where the AIX system is the responder.

To define a generic data management tunnel, you must first define an IPsecProtection element that you would like to use as default tunnel definition.

The `IPSec_ProtectionName` attribute of the default `IPSecProtection` element must start with `_defIPSProt_`.

You must then choose the `IKEProtection` element that would like to use this default `IPSecProtection`. You must specify values for the `IKE_IPSecDefaultProtectionRef` and `IKE_IPSecDefaultAllowedTypes` attributes. The `IKE_IPSecDefaultProtectionRef` attribute refers to the default `IPSecProtection` element that should be used if no other matching tunnel is found. The `IKE_IPSecDefaultAllowedTypes` attribute must contain at least one local and one remote ID type. The possible values for the initiator's local and remote ID types are as follows:

- ▶ Initiator's local ID types
 - `Local_IPV4_Address`
 - `Local_IPV6_Address`
 - `Local_IPV4_Subnet`
 - `Local_IPV6_Subnet`
 - `Local_IPV4_Address_Range`
 - `Local_IPV6_Address_Range`
- ▶ Initiator's remote ID types
 - `Remote_IPV4_Address`
 - `Remote_IPV6_Address`
 - `Remote_IPV4_Subnet`
 - `Remote_IPV6_Subnet`
 - `Remote_IPV4_Address_Range`
 - `Remote_IPV6_Address_Range`

The following example is a skeleton showing the relationship of different components of the IKE XML configuration components when defining a generic data management tunnel. The key management tunnel named `myTunnel` is assigned to the IKE protection policy named `myIKEProtection`. `myIKEProtection` defines a generic data management tunnel. The default `IPSecProtection` for the generic tunnel is assigned to `_defIPSProt_myIPSECPProtection` with the local and remote ID types of `Local_IPV4_Subnet` and `Remote_IPV4_Subnet`. The `IPSecProtection` element named `_defIPSProt_myIPSECPProtection` assigns the default `IPSecProposal` to `IPSECPProposal`.

```
<IKETunnel
    IKE_TunnelName="myTunnel"
    IKE_ProtectionRef="myIKEProtection"
    ...
</IKETunnel>

<IKEProtection
    IKE_ProtectionName="myIKEProtection"
    ...
    IKE_IPSecDefaultProtectionRef="_defIPSProt_myIPSECPProtection"
```

```

        IKE_IPSecDefaultAllowedTypes="Local_IPV4_Subnet Remote_IPV4_Subnet"
    ...
</IKEProtection>
<IPSecProtection
    ...
        IPSec_ProposalRefs="IPSECProposal"
        IPSec_ProtectionName="_defIPSProt_myIPSECProtection"
    ...
/>

<IPSecProposal
    IPSec_ProposalName="IPSECProposal">
    ...
</IPSecProposal>

```

A sample configuration file for generic data management tunnel definition can be found in the file /usr/samples/ipsec/default_p2_policy.xml.

9.21.4 SMIT IKE support (5.2.0)

Management of IKE tunnels just became easier with a series of SMIT dialogs to guide you through the configuration tasks. There are two areas of enhancement as follows:

- ▶ **smitty ipsec4 -> Basic IP Security Configuration -> Use Internet Key Exchange Refresh Model** (as shown in Figure 9-21 on page 657)
- ▶ **smitty ipsec4 -> Advanced IP Security Configuration** (as shown in Figure 9-22 on page 657)

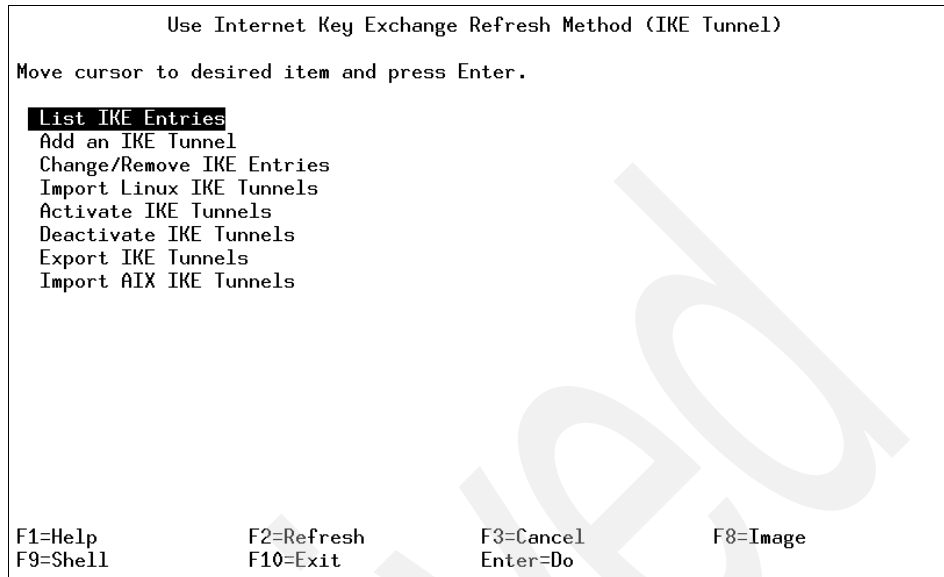


Figure 9-21 SMIT Use Internet Key Exchange Refresh Method dialog

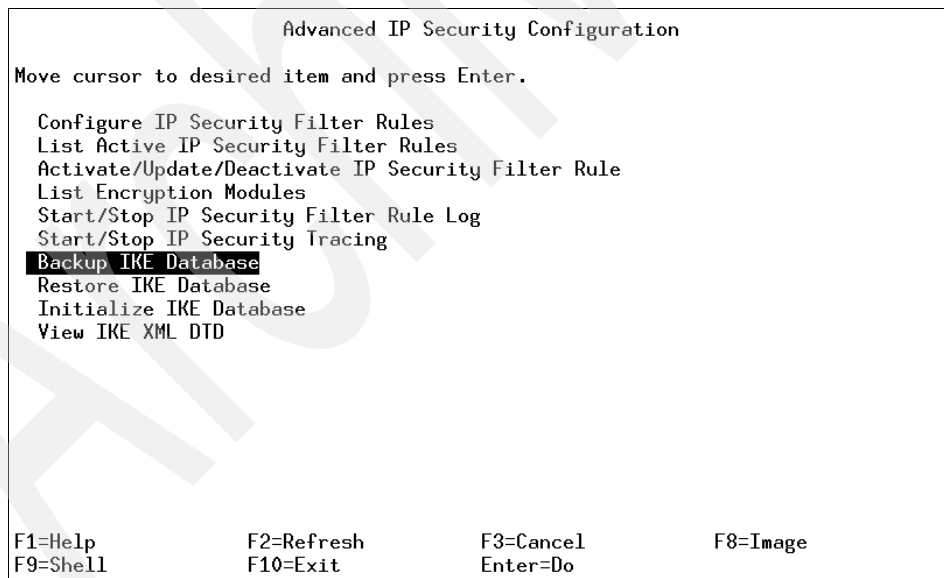


Figure 9-22 SMIT Advanced IP Security Configuration IKE enhancements

9.21.5 Web-based System Manager for IP security enhancements

The Web-based System Manager IKE plug-ins have been rewritten to enhance its performance. Support for Diffie-Hellman group 5 has been enabled in the appropriate pull-downs and checkboxes. Figure 9-23 shows the Overview and Tasks page of the IP Security management plug-in.

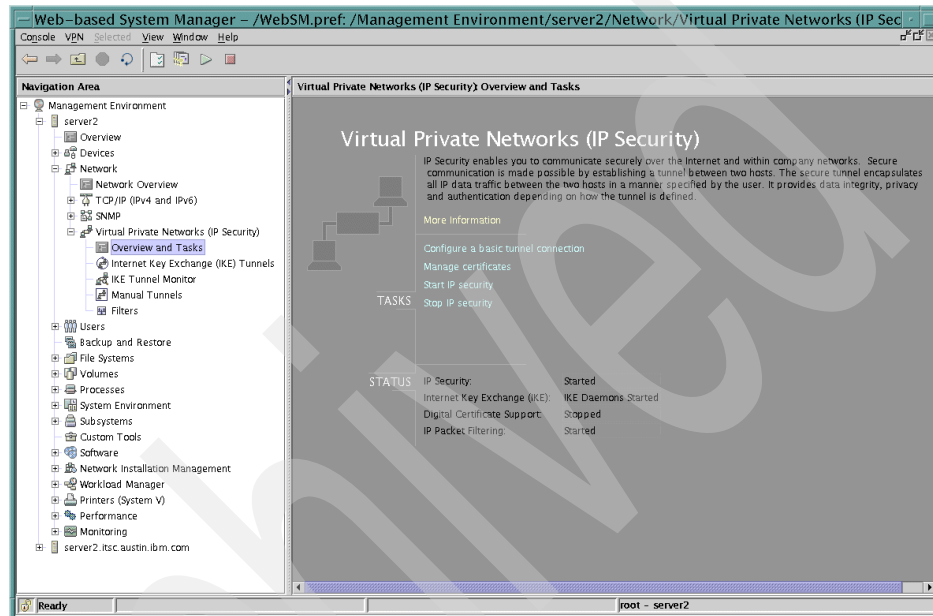


Figure 9-23 IP security Overview and Tasks dialog

Figure 9-24 on page 659 shows the first panel of the basic tunnel connection wizard. The wizard allows you to set up a basic tunnel with minimal effort.

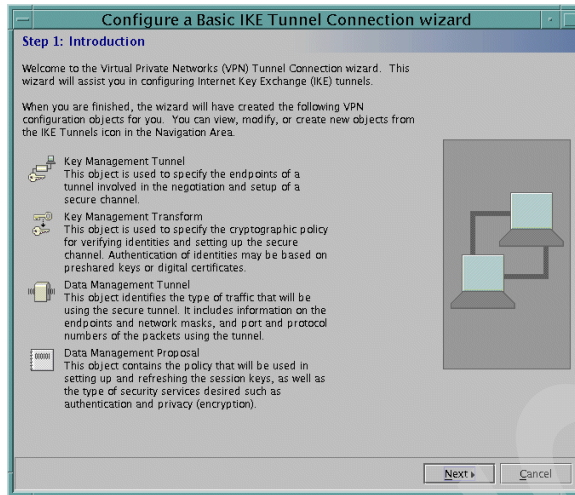


Figure 9-24 IP Security Basic IKE Tunnel Connection wizard

9.21.6 IP Security static filter description

An optional Description field has been added to the static filter rules, allowing the filter rules to be annotated by the administrator. The **genfilt** and **chfilt** commands have a new flag, **-D**, to specify the description.

The following example shows how to use the **genfilt** command to generate a filter rule to deny all SMTP requests to 192.168.1.6.

```
# genfilt -v 4 -n 2 -a D -s 0.0.0.0 -m 0.0.0.0 -d 192.168.1.6 \
-M 255.255.255.255 -O eq -P 25 -I Y \
-D "deny/log incoming sendmail"
```

Filter rule 2 for IPv4 has been added successfully.

The **lsfilt** command has been modified to display the filter rules description. If the **-a** flag is used to display the active filter list in the kernel, then the description will not be displayed. The following examples show the output of the **lsfilt** command with and without the **-a** flag.

```
# lsfilt -v4 -n 2
Rule 2:
Rule action      : deny
Source Address   : 0.0.0.0
Source Mask      : 0.0.0.0
Destination Address : 192.168.1.6
Destination Mask  : 255.255.255.255
Source Routing   : yes
Protocol         : all
```

```

Source Port      : any 0
Destination Port : eq 25
Scope           : both
Direction       : both
Logging control  : yes
Fragment control : all packets
Tunnel ID number : 0
Interface       : all
Auto-Generated   : no
Description      : deny/log incoming sendmail

```

```

# lsfilt -v4 -a
Beginning of IPv4 filter rules.
Rule 1:
Rule action      : deny
Source Address   : 0.0.0.0
Source Mask      : 0.0.0.0
Destination Address : 192.168.1.6
Destination Mask  : 255.255.255.255
Source Routing   : yes
Protocol         : all
Source Port      : any 0
Destination Port : eq 25
Scope           : both
Direction       : both
Logging control  : yes
Fragment control : all packets
Tunnel ID number : 0
Interface       : all
Auto-Generated   : no
Description      :
...

```

The **impfilt** and **expfilt** commands do not support importing or exporting the Description field. The Description field could be misleading or incorrect if imported into another machine.

9.21.7 Cryptographic Library

AIX 5L Version 5.2 now includes Cryptographic Library Version 5.2. The new library contains up-to-date cryptographic functions, including the new NIST Advanced Encryption Standard (named Rijndael), cryptosecure hash generation functions, and the header files needed for use with the library. Included in the library is an API for developers and programmers to access the library to perform the necessary cryptographic functions for their applications that require encryption and decryption or cryptosecure hash generation. Table 9-3 on page 661 provides a list of algorithms included and their key lengths.

Table 9-3 Cryptographic Library algorithms and key lengths

Algorithms	Key length
Rijndael (128-bit block cipher)	28, 192, 256 bits
SEAL (stream cipher)	160 bits
Mars (128-bit block cipher)	128, 192, 256 bits
Twofish (128-bit block cipher)	128, 192, 256 bits
MD5 (cryptographic hash generator)	128 bits
SHA-1(cryptographic hash generator)	160 bits

The licensed product packages (LPP) for Cryptographic Library Version 5.2 are included in the AIX Expansion Pack. The cryptographic library is packaged into the following filesets:

modcrypt.base.includes Contains the xcrypt.h header file
modcrypt.base.lib Contains the libmodcrypt.a library file

9.22 Secure rcmds enhancements (5.2.0)

The **rlogin**, **rcp**, **rsh**, **telnet**, and **ftp** commands, collectively known as the secure rcmds, have been updated to support the native Kerberos and GSSAPI libraries. The secure rcmds are no longer statically linked to the distributed computing environment (DCE) libraries. They are now dynamically linked to the NAS library, which removes the requirement of having DCE installed when you want to use native Kerberos. The secure rcmds can now authenticate against DCE Kerberos 5, Kerberos 4, and native Kerberos 5.

To enable the secure rcmds to use native Kerberos, you must install and configure the NAS client. The NAS client is packaged in the krb5.client.rte fileset and can be installed with **installp**, SMIT, or the Web-based Systems Manager. To configure the client, refer to the *IBM NAS Administrator's and User's Guide* in the krb5.doc.XX_XX.html filesets for further information, where XX_XX is the character string representing your language code (for example, U.S. English is en_US).

You must then use **chauthent** to enable Kerberos as the authentication methods for the secure rcmds. The following example shows how to set the system authentication methods to Kerberos 5 and standard AIX using the **chauthent** command:

```
# chauthent -k5 -std
# lsauthent
```

Kerberos 5
Standard Aix

If you attempt to change to the Kerberos authentication method and you do not have the NAS client installed, you will see the following message:

```
# chauthent -k5 -std
Kerberos 4 permitted on SP system only.
Kerberos 5_DCE requires DCE version 3.2 or greater.
Kerberos 4, Kerberos 5_DCE and Kerberos 5 require krb5.client.rte version 1.3.
```

If DCE is installed, the **lsauthent** command will display Kerberos 5_DCE instead of just Kerberos 5.

```
# lsauthent
Kerberos 5_DCE
Standard Aix
```

Note: To use the secure rcmds with DCE, DCE Version 2.2 or later must be installed. The only supported version of DCE for use with the secure rcmds, is Version 3.2 or later.

System V affinity

AIX 5L Version 5.2 includes several new features to allow for further affinity with System V UNIX-based systems. This aligns Version 5.2 with many new areas of System V and builds on enhancements made in previous releases of AIX 5L.

The following enhancements have been made to AIX 5L Version 5.2:

- ▶ Weak symbol support
- ▶ Affinity commands
- ▶ The /proc file system
- ▶ Tools enhancements: /proc, pTools, and **truss**
- ▶ User API for Sun threaded applications
- ▶ System V printing subsystem for AIX

10.1 Weak symbol support (5.2.0)

Weak symbol support is provided for both 64-bit and 32-bit object files and modules. This is mainly applicable to C++ applications and enhances portability from System V platforms to AIX 5L Version 5.2 (hereafter referred to as Version 5.2). Weak symbols allow the link editor to ignore multiple definitions without producing warnings for them.

10.1.1 AIX C++ compiler

Version 5.2 provides the capability to suppress warnings when weak symbols are used. The compiler must generate weak symbols for this support.

It is an error to have the same name for a non-inline function with external linkage and an inline function with external linkage. Different definitions of the same inline function in two compilation units is also an error. However, there is no requirement under the standard to detect these errors.

10.1.2 GNU C++ compiler and templates

The compiler can generate a function instance from a template or an explicit instance can be defined, which would require it to be used everywhere. Duplicate symbols can occur when the same instance is required in multiple compilation units.

The AIX C++ compiler use of *munch* to manage symbol resolution for template functions is no longer needed. The AIX compiler uses the functionality provided in Version 3 of the GNU C++ compiler that generates code requiring support for weak symbols.

Weak symbol support safely allows the linker to ignore multiple definitions. The Version 5.2 assembler supports the definition of weak symbols in assembler files. Features of weak semantics include:

- ▶ Weak symbols are marked with a storage class of `C_WEAKEXT` and have the same visibility as global symbols. A global symbol preempts a weak symbol with the same name.
- ▶ Weak symbols may have multiple definitions, the linker will use the first symbol and ignore all others without warnings.
- ▶ A global definition takes priority over a weak one, even if the weak definition is seen first. Common symbols also take precedence over weak symbols.
- ▶ During run time, weak symbols use the first-round definition, the symbol that is first processed.

Version 5.2 behaves differently from System V in the following ways:

- ▶ Unresolved weak references in executable objects will result in an error. AIX does not set undefined weak symbols to zero value as System V does.
- ▶ In System V, archive members are not actively searched to find a definition of weak reference. AIX searches for definitions for referenced symbols from all objects and archives and chooses which to retain.

10.1.3 Differences between weak and global links

The differences between weak and global links are discussed in the following section. In general:

- ▶ If a defined global symbol exists, the coexistence of a weak symbol will not cause a linking error. The global symbol is used and the weak symbol is ignored.
- ▶ When the link editor processes archive libraries, it retains archive member csects that contain definitions of both global and weak symbols.

Weak symbols are supported in both the XCOFF symbol table (identified by the storage class, `C_WEAKEXT`; a new assembler pseudo-op, `.weak`, has been created to enable a symbol to be marked as `C_WEAKEXT`), and the loader section (identified by the loader flag, `L_WEAK`). Weak data in the TOC is supported. The TOC entries continue to have the `C_HIDEXT` storage class for both text and data symbols.

Common symbols (mapping type `XTY_CM`) may also be marked as weak, although a weak common symbol will not be used if a regular common symbol exists. Therefore, a common symbol takes precedence over a weak common symbol. mport symbols may only have the weak export attribute. An imported symbol from another module will have all references to the symbol rebounded.

Import files can specify the `weak` keyword as an import symbol attribute to enable the linker to identify weak symbols for linking with a shared library.

The `weak` keyword is also valid for export files by associating the symbols mapping type with `L_WEAK` in the loader section, which allows symbol marking without any compiler support. The weak attribute may be used in combination with any other export attribute.

10.2 System V commands (5.2.0)

System V functionality has been enabled for a number of commands with Version 5.2, although some can be found in maintenance releases of Version

5.1. This could either mean the inclusion of a complete new command, a System V version of it, or the addition of System V flags to an existing AIX command. The commands described in the following sections are affected as a result of these changes (for full documentation refer to the relevant man pages):

10.2.1 atrm

Removes jobs spooled by the **at** command but not yet executed. Only root user has permission to execute this command. The enhancement of this command is the **-a** flag. The syntax of **atrm** is as follows, and the most common flags are provided in Table 10-1:

```
atrm [-f] [-i] [-a | - ] [Job# | User.....]
```

Table 10-1 Most common flags for atrm

Flag	Description
-	Removes all jobs belonging to the invoking user
-a	Removes all jobs belonging to the invoking user (provided for System V Affinity)

10.2.2 cpio

A new version of the **cpio** command has been introduced. This System V command is in **/usr/sysv/bin** and not **/usr/bin** as with the standard AIX **cpio** command, although the standard **cpio** command still exists in **/usr/bin**.

There is support for further header types in addition to the ASCII support offered in previous versions (the **-c** option, which is equivalent to **-Hodc** on other UNIX variants).

The flag to specify new header types is **-H hdr**. Valid options for the **hdr** value are shown in Table 10-2.

Table 10-2 Most common flags for cpio

Flag	Description
-Hcrc	Same as CRC, ASCII header with per-file checksum. CRC handles files greater than 2 GB.
-Hustar	Same as USTAR - IEEE/P1003 Data Interchange Standard head and format.
-Htar	Same as TAR, tar header compatibility.
-Hodc	ASCII header with small fundamental types.

10.2.3 date

The **date** command now has support for the -a option. This option can only be run by the root user and allows the date to be slowly adjusted by *sss.fff* (where *fff* is fractions of a second). The change can be either negative or positive, and either slows down or speeds up the system clock to enable the change. Syntax of the -a option is as follows:

```
date [-a] [[+|-]sss.fff]
```

10.2.4 df

The **df** command in System V reports on the number of free blocks and files in a file system, displayed in 512-byte blocks. Note that the System V command is in */usr/sysv/bin* and not */usr/bin* as with the standard AIX **df** command. The command syntax is as follows, and the most common flags are provided in Table 10-3:

```
df [-al] [ [-egn] | [-iv | -t] ] [file system ...] [file ...]
```

This differs from the standard AIX **df** syntax, which is shown below:

```
df [-P] | [-IMitv] [-gkm] [-s] [file system] [file]
```

Table 10-3 Most common flags for */usr/sysv/bin/df*

Flag	Description
-a	Prints mount points, device name, number of free blocks, and number of used inodes.
-e	Prints only the number of free files.
-g	Prints the entire statvfs structure and overrides all other options. The numbers for available, total, and free blocks are reported in 512-byte blocks.
-i	Prints number of inodes, free inodes, used inodes, and percentage of inodes in use.
-l	Reports on local file systems only.
-n	Prints the type of file system.
-t	Prints the total allocated block figures.
-v	Prints total blocks, blocks used, and blocks free.

An example of the **df** command is as follows:

```
# /usr/sysv/bin/df -i
Mount Dir file system      iused   ifree   itotal  %iused
```

```

/          /dev/hd4      1439    6753    8192    18%
/usr       /dev/hd2      23751   164665  188416  13%
/var       /dev/hd9var    443     3653    4096    11%
/tmp       /dev/hd3       26     8166    8192    1%
/home      /dev/hd1       18     4078    4096    1%
/proc      /proc         0        0        0        0
/opt       /dev/hd10opt   278    7914    8192    4%

# /usr/sysv/bin/df -g
/          (          /dev/hd4):  4096 block size
4096 frag size
   32768 total blocks      11824 free blocks      11824 available
8192 total files
   6753 free files      655364 filesys id
   jfs fstype          0x0000 flag          255 filename length
/usr       (          /dev/hd2):  4096 block size
4096 frag size
  1507328 total blocks      3672 free blocks      3672 available
188416 total files
  164665 free files      655365 filesys id
   jfs fstype          0x0000 flag          255 filename length
.....

```

10.2.5 dfshares

The **dfshares** command lists available file systems from both remote and local systems. The syntax is as follows, although the only file system type that is supported is NFS. The **-h** flag suppresses the header information:

```
dfshares [-F file systemType] [-h] [Server....]
```

Examples of the **dfshares** command are as follows:

```

root@server2:/ #dfmounts -h server1
-          server1 /lpp_source
server2.itsc.austin.ibm.com
-          server1 /spot
server2.itsc.austin.ibm.com

root@server2:/ #dfmount -h -F nfs server1
server1:/home          server1    - -
server1:/spot          server1    - -
server1:/lpp_source    server1    - -

```

10.2.6 dfmounts

The **dfmounts** command displays mounted system resources. Flags are the same as for the **dfshares** command and NFS is the only supported file system type. Examples of the **dfmounts** command are as follows:

```
root@server2:/ #dfmounts -h server1
-          server1 /lpp_source          server2.itsc.austin.ibm.com
-          server1 /spot                server2.itsc.austin.ibm.com

root@server2:/ #dfmounts -F nfs server1
RESOURCE   SERVER PATHNAME          CLIENTS
-          server1 /lpp_source          server2.itsc.austin.ibm.com
-          server1 /spot                server2.itsc.austin.ibm.com
```

10.2.7 dircmp

The **dircmp** command compares the contents of common files in two directories (files that exist in both directories). The **-n num** flag has been introduced. The command syntax is as follows, and the most commonly used flags are provided in Table 10-4:

```
dircmp [-d ] [-s] [-w num] Directory1 Directory2
```

Table 10-4 Most common flags for dircmp

Flag	Description
-d	Prints display in same format as diff command for differing files
-s	Does not list names of identical files
-n num	Changes the width of the output to <i>num</i> number of characters

The following example compares the content of **/etc** and a backup of some key files from **/etc/** (**/tmp/etcbk**) that were made prior to some changes on the system, although the full output is not provided in this excerpt:

```
# dircmp -ds /etc/ /tmp/etcbk
Fri Aug 23 12:04:21 CDT 2002 Comparison of /etc/ and /tmp/etcbk Page 1
Fri Aug 23 12:05:56 CDT 2002 Comparison of /etc/ and /tmp/etcbk Page 1
```

```
different      ./hosts
different      ./rc.tcpip
different      ./resolv.conf
```

```
Fri Aug 23 12:05:56 CDT 2002 diff of ./hosts in /etc/ and /tmp/etcbk Page 1
```

```
54,55d53
< 9.3.4.98      server2
< 9.3.4.99      server3
```

Fri Aug 23 12:05:56 CDT 2002 diff of ./rc.tcpip in /etc/ and /tmp/etcbk Page 1

```
201,205d200
<
< # Set no options, from default
< no -o tcp_sendspace=32768
< no -o tcp_recvspace=32768
< no -o rfc1323=1
```

Fri Aug 23 12:05:56 CDT 2002 diff of ./resolv.conf in /etc/ and /tmp/etcbk Page 1

```
1,2c1,2
< nameserver 9.3.4.29
< search mycompany.example itsc.austin.ibm.com
---
> #nameserver 9.3.4.29
> #search mycompany.example itsc.austin.ibm.com
```

From this example it is possible to tell that two host entries have been made to the /etc/hosts file, some network options have been changed in /etc/rc.tcpip, and DNS has been enabled in /etc/resolv.conf.

10.2.8 dispgid

The **dispgid** command displays all valid groups on the system. There are no options with this command.

10.2.9 dispuid

The **dispuid** command displays all valid user IDs on the system. There are no options with this command.

10.2.10 getconf

The **getconf** command writes system configuration variables to standard out. To display all variables use the -a flag. This is a new flag.

```
getconf -a
```

From the output of this command, there is a new feature to use variable names to uniquely specify specific values (wild carding is not supported), and the syntax is as follows:

```
getconf [-v specification] [SystemwideConfiguration] [PathConfiguration  
Pathname] [DeviceVariable DeviceName]
```

Examples of the **getconf** command follow:

```
# getconf KERNEL_BITMODE  
32  
  
# getconf HARDWARE_BITMODE  
32  
  
# getconf REAL_MEMORY  
524288  
  
# getconf MP_CAPABLE  
1  
  
# getconf PIPE_BUF /usr  
32768  
  
# getconf NAME_MAX /usr  
255  
  
# getconf DISK_SIZE /dev/hdisk0  
8678  
  
# getconf DISK_PARTITION /dev/hdisk0  
16
```

10.2.11 getdev

The **getdev** command lists devices, and has the following syntax:

```
getdev [-ae] [criteria,.....][devicelist,.....]
```

Where **-a** is a logical *and*, which will include all devices that match all the criteria in the command. The **-e** option does the opposite and excludes devices listed in the command. Useful criteria are alias (its name) and type (field PdDvLn, as found in the CuDv ODM file). The various criteria types can be found with the following command:

```
odmget CuDv | grep PdDvLn | uniq | awk '{print $3}' | awk -F / '{print $3}' | sed  
s/\\/\\/g
```

Example outputs include the following:

```
# getdev type=proc_rspc
proc0
proc1
proc2
proc3
```

Using the same command but with `!=`, would return all devices excluding those belonging to `proc_rspc`.

10.2.12 getdgrp

The **getdgrp** command lists device classes. The `-a` flag lists groups that match all search criteria, and the `-e` flag excludes groups that match the search criteria. The `-l` flag lists all device classes that are subject to the `-e` flag. The type is the same as defined for the **getdev** command. The syntax is:

```
getdgrp [-a] [-e] [-l] [Criteria] [DeviceClassList]
```

An example output of the **getdgrp** command is as follows:

```
# getdgrp
adapter
aio
bus
cdrom
container
disk
diskette
gxme
if
keyboard
lft
logical_volume
lvm
memory
mouse
planar
posix_aio
processor
pty
rcm
sys
tape
tcPIP
```

For group tcpip, the type is inet, as found by using the following command, noting that the command semantics are the same as **getdev** for the criteria and DeviceClassList:

```
# odmget CuDv|grep tcpip
PdDvLn = "tcpip/TCP/IP/inet"
```

10.2.13 groups

The **groups** command displays groups for either the current user or the specified user(s). Multiple users are allowed in the command string, which is an enhancement from previous versions. The command syntax is as follows:

```
groups [Users ...]
```

Executed as root user, the output would be similar to:

```
# groups bin root
bin : bin sys adm
root : system bin sys security cron audit lp dbsysadm
```

10.2.14 last

The **last** command displays information about previous user logins. Version 5.2 supports the use of the *-Number* flag, which restricts the output of the command to the number of entries specified by the *Number* parameter. This has been introduced for System V affinity and is equivalent to the *-n Number* flag. The **last** command also provides support for host names greater than 16 characters. The new *-t* flag enables the last command to report logins at a given time. The *Time* variable is specified in decimal form as follows:

```
last -t [[CC]YY]MMDDhhmm[.SS]
```

Where the arguments have the following definitions:

CC	Specifies the first two digits of the year
YY	Specifies the last two digits of the year
MM	Specifies the month of the year (01 to 12)
DD	Specifies the day of the month (01 to 31)
hh	Specifies the hour of the day (00 to 23)
mm	Specifies the minute of the hour (00 to 59)
SS	Specifies the second of the minute (00 to 59)

The syntax of the **last** command is as follows:

```
last [ -f FileName ] [ -t Time ] [ -n Number | -Number ] [ Name ... ]  
[ Terminal ...]
```

An example illustrating long host names is shown as follows:

```
# last -n 5  
root      pts/6          9.3.4.145          Aug 26 11:59 - 11:59 (00:00)  
root      pts/4          3d052-2.itsc.austin.ibm.com Aug 26 09:54 - 09:54  
(00:00)  
root      pts/1          3d052-2.itsc.austin.ibm.com Aug 26 09:53 - 09:54  
(00:00)  
root      pts/1          3d052-2.itsc.austin.ibm.com Aug 26 09:53 - 09:53  
(00:00)  
root      dtremote      3d052-1.itsc.austin.ibm.com:0 Aug 26 09:05 - 09:13  
(00:07)  
  
# last -t 200209190700  
root      pts/2          9.182.18.103       Sep 19 05:49 still logged in.  
root      pts/1          9.182.18.107       Sep 17 07:10 still logged in.  
root      pts/0          chocate.austin.ibm.com Sep 10 10:20 still logged  
in.
```

10.2.15 ldd

The **ldd** command lists dynamic dependencies, such as full path names of shared objects that would be loaded as a result of executing a file. Only one file at a time can be specified and it must be an executable. The syntax is as follows:

```
ldd <exe>
```

An example of the **ldd** command follows:

```
# ldd arp  
arp needs:  
    /usr/lib/libc.a(shr.o)  
    /unix  
    /usr/lib/libcrypt.a(shr.o)
```

10.2.16 listdgrp

The **listdgrp** command displays devices in a device class. An object must be specified as defined in Customized Devices in the Device Configuration database. This command uses the **lsdev -Cc device** command. The syntax is as follows:

```
listdgrp dgroup
```


An example of the **listdgrp** command is as follows:

```
# listdgrp disk
hdisk0
hdisk1
```

10.2.17 ln

The file linking command, under Version 5.2, now supports the use of the **-n** flag, which ensures that a link is not overwritten if the file already exists. The **-f** flag still has the functionality to overwrite the target file if it exists.

10.2.18 logins

The **logins** command is new to Version 5.2 and lists user and system login information. The most common flags are provided in Table 10-5.

Table 10-5 Most common flags for the logins command

Flag	Description
-a	In addition to the default output, the -a flag adds two password expiration fields to the display. These fields show how many days a password can remain unused before it automatically becomes inactive and the date that the password will expire.
-g Groups	Displays all users belonging to group, sorted by user ID. Multiple groups can be specified as a comma-separated list. Groups must specify valid group names on the system. Comma separate names when specifying more than one group.
-l Logins	Displays the requested login. Multiple logins can be specified as a comma-separated list. Logins must specify valid user names on the system.
-m	Displays multiple group membership information.
-o	Formats output into one line of colon-separated fields
-p	Displays users without passwords.
-s	Displays all system logins.
-t	Sorts output by user name instead of by user ID.
-u	Displays all user logins.

Flag	Description
-x	<p>Prints an extended set of information about each selected user. Information for each user is printed on a separate line containing the home directory, login shell, and password aging information. The extended information includes the following:</p> <ul style="list-style-type: none">▶ The password status▶ The date on which the password was last changed▶ The number of days required between changes▶ The number of days allowed before a change is needed▶ The number of days that the user will receive a password expiration warning message before the password expires <p>The password status is displayed in an abbreviated form as PS for logins with password, NP for no password, or LK for locked.</p>

Output is sorted by user ID, first by system and then user logins. An example for the **logins** command follows:

```
# logins
root      0      system      0
daemon    1      staff        1
bin        2      bin          2
sys        3      sys          3
adm        4      adm          4
uucp       5      uucp         5
nuucp      6      uucp         5      uucp login user
lpd        9      nobody       -2
lp         11     lp           11
guest     100     usr          100
imnadm    188     imnadm       188
invscout  200     staff         1
snapp     201     snapp        12      snapp login user
nobody    -2      nobody       -2
```

10.2.19 mach

The **mach** command displays the processor architecture of the machine, for example:

```
# mach
powerpc
```

10.2.20 ps

There is now a System V **ps** command in `/usr/sysv/bin` to support all the System V options. The most common flags are provided in Table 10-6.

Table 10-6 *Flags not found in AIX for ps*

Flags	Description
-L	Prints status of active threads in a process
-j	Prints session ID and process group ID
-s <i>sidlist</i>	Prints all processes whose session leader IDs are specified in <i>sidlist</i> , where <i>sidlist</i> is a list of PIDs (as referred to in AIX)
-y	If combined with -l, prints RSS and SZ fields in KB and does not print F and ADDR fields

An example of the **ps** command follows:

```
#/usr/sysv/bin/ps -L
  PID  LWP  TTY  LTIME CMD
 22060 39347 pts/2  0:00 ksh

# /usr/sysv/bin/ps -j
  PID PGID  SID  TTY  TIME CMD
 22060 22060 22060 pts/2  0:03 ksh

# /usr/sysv/bin/ps -l
      F S UID  PID  PPID  C PRI NI      ADDR  SZ      WCHAN  TTY
TIME CMD
 240001 A  0 22060 20246  0  60 20      71ce  174      pts/2
0:03 ksh

# /usr/sysv/bin/ps -yl
  S UID  PID  PPID  C PRI NI      RSS  SZ      WCHAN  TTY  TIME CMD
  A  0 22060 20246  1  60 20      728  696      pts/2  0:03 ksh
```

10.2.21 pwck

The **pwck** command scans the password information to verify local authentication methods. Essentially this command calls `pwdck` with the `-n` and `ALL` options specified. This means that the command reports on all users but does not fix any issues. There are no flags to specify with this command.

10.2.22 quot

The **quot** command provides a summary of file system ownership by displaying the number of 512-byte blocks owned by each user. If no file system is specified, then all file systems of type jfs as defined in /etc/file systems are used. The syntax of this command is as follows, and the most common flags are provided in Table 10-7:

```
quot [-cfhmv] [filesystem ...]
quot -a [-cfhmv]
```

Table 10-7 Most common flags for quot

Flags	Description
-a	A full report of all mounted JFSs
-c	Generates a three-column report (file size in 512-byte blocks, number of files of that size, and cumulative of files of that size or smaller)
-f	Prints the total number of blocks and files for users in JFS
-v	Additional to default (and -a), displays three columns with blocks not accessed for 30, 60, and 90 days

An example of the **quot** command follows:

```
# quot -c /tmp
/tmp:
0      6      0
8      163    1304
16     24     1688
24     12     1976
32     4      2104
40     4      2264
48     8      2648
64     8      3160
72     2      3304
88     2      3480
96     4      3864
104    4      4280
208    2      4696
256    1      4952
499    2      6728
# quot -f /tmp
/tmp:
6720    247   root
8        1    bin
```

10.2.23 settime

The **settime** command, by default, will update the files specified with the current access and modification times. Dates beyond 2038 are not valid. The syntax of the command is as follows:

```
settime [ [MMddhhmm[yy] ] | [-f ReferenceFile]] File....
```

Where *File* would contain the name of a file or space-separated list of files. An example of the **settime** command is as follows:

```
# ls -l file*
-rw-r--r--  1 root      system          0 Aug 26 15:30 file1
-rw-r--r--  1 root      system          0 Aug 26 15:31 file2

# settime 0203093501 file1 file2

# ls -l file*
-rw-r--r--  1 root      system          0 Feb 03 2001  file1
-rw-r--r--  1 root      system          0 Feb 03 2001  file2
```

10.2.24 setuname

The **setuname** command is new with Version 5.2 and is used to set the node name of the system. Only the root user can execute this command. The syntax is as follows:

```
setuname [-t] -n node
```

Where the **-t** option is a temporary change and calls the **hostname** command. The node name will be set as before the command after a reboot. If the **-t** flag is not specified, the name is changed in the ODM with a **chdev** command and is permanent. An example of the **setuname** command follows:

```
# hostname
ausprod1

# setuname -t -n austest1

# hostname
austest1
```

10.2.25 swap

The **swap** command displays paging characteristics and enables the allocation and deallocation of paging devices. It has the same function as the AIX commands **lspv**, **swapon**, and **swapoff**, where *device* is in the format */dev/dev_name*.

The syntax of this command is as follows, and the most common flags are provided in Table 10-8:

```
swap [-l | -s] | [-d device] | [-a device]
```

Table 10-8 Most common flags for the swap command

Flags	Description
-a	Activates device
-d	Deactivates device
-l	Prints device, major and minor numbers, and total and free space
-s	Prints allocated blocks, used blocks, and free blocks as a total of all swap space

10.2.26 umountall

This **umountall** command unmounts all mounted file systems except /, /usr, /var, and /proc. The **umountall** command calls the **umount** AIX command. The most common flags are provided in Table 10-9, and the syntax of the command is as follows:

```
umountall [-ks] [-f FStype] [-l | -r]
```

Table 10-9 Most common flags for umountall

Flags	Description
-F FStype	Limits the umountall by FStype
-l / -r	Limits action to local/remote file systems
-k	Runs a SIGKILL to each process on the mount point before unmounting (using fuser -k)

10.2.27 wall

The **wall** command is used to send logged-on users messages, and has been enhanced for Version 5.2 with the addition of the -a and -g flags.

The -a option broadcasts to the console and pseudo terminals. This is normally the default behavior of the AIX **wall** command, but this flag has been incorporated for System V affinity.

The -g command allows broadcasting to a particular group specified with the flag, as defined in /etc/group.

10.2.28 whodo

The **whodo** command is new to Version 5.2 and reports the list of processes and their child processes belonging to users on the system. The syntax of the command is:

```
whodo [-h] [-l] [user]
```

Where the **-h** flag suppresses the heading and the **-l** flag provides a long listing:

```
# whodo
Fri Sep 20 14:36:52 2002
aixcomm

pts/0  root    10:10
      pts/0    22326    0:00    ksh
      pts/0    22204    0:02    ksh
      pts/0    13214    0:00    server
      ?        21170    0:00
      pts/0    21334    0:00    mail

pts/1  root    17:48
      pts/1    19234    0:00    ksh

pts/2  root    14:54
      pts/2    9468     0:00    ksh

pts/3  root    12:13
      pts/3    24728    0:00    ksh
      pts/3    24890    0:00    whodo

pts/5  root    18:40
      pts/5    25234    0:01    ksh

pts/7  root    12:32
      pts/7    6618     0:00    ksh
      pts/7    18808    0:00    vi
```

10.2.29 zdump

The **zdump** command displays the current time in each time zone specified. Standard zone information is stored in the `/usr/share/lib/zoneinfo` directory. Some are in the format of the country name, others are abbreviations. It is advisable to check this directory for specific requirements. The syntax of this command is as follows, and the most common flags are provided in Table 10-10 on page 682.

```
zdump [-v] [-c cutoffyear] zonename
```

Table 10-10 Most common flags for zdump

Flags	Description
-c	Cuts the verbose output up to the year specified.
-v	Prints current time, time at lowest possible time value, time one day after the lowest value, times both one second before and exactly at each time for computing local time change, time at highest possible time value and the time at one day less than the highest possible time value.

An example for Australia is as follows:

```
# zdump -v -c 1999 Australia
Australia Fri Aug 23 17:39:27 2002 Australia
Australia Fri Dec 13 20:45:52 1901 GMT = Fri Dec 13 20:45:52 1901 Australia
isdst=0 gmtoff = 0
Australia Sat Dec 14 20:45:52 1901 GMT = Sat Dec 14 20:45:52 1901 Australia
isdst=0 gmtoff = 0
Australia Wed Dec 31 23:59:59 1969 GMT = Wed Dec 31 23:59:59 1969 Australia
isdst=0 gmtoff = 0
Australia Thu Jan 1 00:00:00 1970 GMT = Thu Jan 1 00:00:00 1970 Australia
isdst=0 gmtoff = 0
Australia Sun Jan 25 16:50:57 1987 GMT = Sun Jan 25 16:50:57 1987 Australia
isdst=0 gmtoff = 0
Australia Sun Jan 25 16:50:58 1987 GMT = Sun Jan 25 16:50:58 1987 Australia
isdst=0 gmtoff = 0
Australia Mon Jan 18 03:14:07 2038 GMT = Mon Jan 18 03:14:07 2038 Australia
isdst=0 gmtoff = 0
Australia Tue Jan 19 03:14:07 2038 GMT = Tue Jan 19 03:14:07 2038 Australia
isdst=0 gmtoff = 0
```

10.2.30 zic

The **zic** command is a time zone compiler. Text is processed from a file specified on the command line and creates the time conversion. If the file name is -, standard input is assumed. The default directory for conversion files is /usr/share/lib/timezone, although with the -d flag an alternative directory can be specified. The syntax of this command is as follows, and the most common flags are provided in Table 10-11.

```
zic [-v] [-d directory] [-l localtime] [-y yearistype] [filename.....]
```

Table 10-11 Most common flags for zic

Flag	Description
-l	Uses local time as the time zone

Flag	Description
-y YearIsType	Uses the given YearIsType rather than /usr/sbin/yearistype

The output of the **zic** command looks as follows:

```
# pwd
/usr/share/lib/zoneinfo

# pg timezone.infile

#Zone  NAME      GMTTOFF RULES/SAVE  FORMAT  [UNTIL]
Zone   Singapore  8:00    -          SST
Zone   India      -1:00   India     IST

#Rule  NAME      FROM  TO      TYPE  IN      ON      AT      SAVE
LETTER/S
Rule   India   2030  max     -      Mar    lastSun 2:00  1:00  D
Rule   India   2030  max     -      Sep    Sun>=15 2:00 -1:00  S
```

There are two zones, Singapore (which is plus eight hours of GMT for standard time in this zone) and India (which is -1 hour from GMT for standard time in this zone).

The rules (which the India zone references) state:

- ▶ From the year 2030 to max (which in Version 5.2 is 2038) at 2:00 a.m. on the last Sunday of March, add one hour to local standard time. The D character represents EDT.
- ▶ From the year 2030 to max (which again is 2038) at 2:00 a.m. on the Sunday on or after the 15th in September subtract one hour from local standard time. The S character is EST.

```
# ls -lt | head -4
total 504
-rw-r--r--  1 root    system      140 Aug 27 10:21 India
-rw-r--r--  1 bin     bin         54 Aug 27 10:21 Singapore
-rw-r--r--  1 root    system     238 Aug 27 10:20 timezone.infile

# file India Singapore
India:      data or International Language text
Singapore:  data or International Language text
```

10.3 The /proc file system

AIX 5L provides support of the /proc file system. This pseudo-file system maps processes and kernel data structures to corresponding files. The output of the **mount** and **df** commands showing /proc is provided in the following examples:

```
# mount
node      mounted      mounted over    vfs      date      options
-----
/dev/hd4   /dev/hd4         /               jfs      Sep 11 16:52 rw,log=/dev/hd8
/dev/hd2   /dev/hd2         /usr            jfs      Sep 11 16:52 rw,log=/dev/hd8
/dev/hd9var /dev/hd9var      /var            jfs      Sep 11 16:52 rw,log=/dev/hd8
/dev/hd3   /dev/hd3         /tmp            jfs      Sep 11 16:52 rw,log=/dev/hd8
/dev/hd1   /dev/hd1         /home           jfs      Sep 11 16:53 rw,log=/dev/hd8
/proc      /proc            /proc           procfs   Sep 11 16:53 rw
```

```
# df
Filesystem    512-blocks      Free %Used    Iused %Iused Mounted on
/dev/hd4       65536         27760   58%      2239    14% /
/dev/hd2      1507328        242872   84%     22437   12% /usr
/dev/hd9var    32768         16432   50%       448    11% /var
/dev/hd3       557056        538008    4%       103     1% /tmp
/dev/hd1       32768         31608    4%        47     2% /home
/proc          -              -         -         -       - /proc
```

The entry in the /etc/vfs file appears as follows:

```
# lsvfs procfs
procfs 6      none  none
```

Each process is assigned a directory entry in the /proc file system with a name identical to its process ID. In this directory, several files and subdirectories are created corresponding to internal process control data structures. Most of these files are read-only, but some of them can also be written to and be used for process control purposes. The interfaces to these files are the standard C language subroutines `open()`, `read()`, `write()`, and `close()`. It is possible to have several concurrent readers, but for reliability reasons, the first write access should use the exclusive flag so that subsequent opens for write access fail. The description of the data structures used can be found in `/usr/include/sys/procfs.h`. The ownership of the files in the /proc file system is the same as for the processes they represent. Therefore, regular users can only access /proc files that belong to their own processes.

A simple example illustrates this further. Suppose a process is waiting for standard input (the information in the process data structures is basically static). If you look at an active process, a lot of the information would constantly change:

```
# ls -l /proc/19082/
total 0
```

```

dr-xr-xr-x  1 root    system      0 Sep 15 15:12 .
dr-xr-xr-x  1 root    system      0 Sep 15 15:12 ..
-rw-----  1 root    system      0 Sep 15 15:12 as
-r-----  1 root    system    128 Sep 15 15:12 cred
--w-----  1 root    system      0 Sep 15 15:12 ctl
dr-xr-xr-x  1 root    system      0 Sep 15 15:12 lwp
-r-----  1 root    system      0 Sep 15 15:12 map
dr-x-----  1 root    system      0 Sep 15 15:12 object
-r--r--r--  1 root    system    448 Sep 15 15:12 psinfo
-r-----  1 root    system   1024 Sep 15 15:12 sigact
-r-----  1 root    system   1520 Sep 15 15:12 status
-r--r--r--  1 root    system      0 Sep 15 15:12 sysent

```

Table 10-12 provides the function of the pseudo files listed in the previous output.

Table 10-12 Function of pseudo files in /proc/<pid> directory

Pseudo file name	Function
as	Read/write access to address space
cred	Credentials
ctl	Write access to control process; for example, stop or resume
lwp directory	Kernel thread information
map	Virtual address map
object directory	Map file names
psinfo	Information for the <code>ps</code> command; readable by everyone
sigact	Signal status
status	Process state information, such as address, size of heap or stack
sysent	Information about system calls

The pseudo file named *as* allows you to access the address space of the process, and as it can be seen by the *rw* (read/write) access flags, you can read and write to the memory belonging to the process.

It should be understood that only the user regions of the process' address can be written to under */proc*. Also, a copy of the address space of the process is made while tracing under */proc*. This is the address space that can be modified. This is done so when the *as* file is closed; the original address space is unmodified.

The *cred* file provides information about the credentials associated with this process. Writing to the *ctl* file allows you to control the process; for example, to stop or to resume it. The *map* file allows access to the virtual address map of the

process. Information usually shown by the **ps** command can be found in the **psinfo** file, which is readable for all system users. The current status of all signals associated with this process is recorded in the **sigact** file. State information for this process, such as the address and size of the process heap and stack (among others), can be found in the **status** file. Finally, the **sysent** file allows you to check for the system calls available to this process.

The object directory contains files with names as they appear in the **map** file. These files correspond to files mapped in the address space of the process. For example, the content of this directory appears as follows:

```
# ls -l /proc/19082/object
total 13192
dr-x----- 1 root    system      0 Sep 15 15:09 .
dr-xr-xr-x  1 root    system      0 Sep 15 15:09 ..
-r-xr-xr-x  1 bin     bin        6264 Aug 24 21:16 a.out
-rwxr-xr-x  1 bin     bin       14342 Aug 22 22:37 jfs.10.5.10592
-r-xr-xr-x  2 bin     bin     6209308 Aug 24 13:03 jfs.10.5.2066
-r--r--r--  1 bin     bin     118267 Aug 24 15:06 jfs.10.5.2076
-r-xr-xr-x  1 bin     bin     11009 Aug 24 14:59 jfs.10.5.4129
-r--r--r--  1 bin     bin    377400 Aug 24 15:05 jfs.10.5.4161
-r-xr-xr-x  1 bin     bin     6264 Aug 24 21:16 jfs.10.5.6371
```

The **a.out** file always represents the executable binary file for the program running in the process itself. Because the example program is written in C and must use the C runtime library, it can be concluded from the size of the entry named **jfs.10.5.2066** that this corresponds to the **/usr/ccs/lib/libc.a** file. Checking this file reveals that the numbers in the file name are the major and minor device numbers, and the inode number, respectively. This can be seen in the following output, where **/usr** corresponds to **/dev/hd2** and the **ncheck** command is used to find a file belonging to an inode in a specific file system:

```
# ls -l /dev/hd2
brw-rw---- 1 root    system    10,  5 Sep 20 16:09 /dev/hd2

# ncheck -i 2066 /dev/hd2
/dev/hd2:
2066    /ccs/lib/libc.a
```

The **lwp** directory has subdirectory entries for each kernel thread running in the process. The term **lwp** stands for lightweight process and is the same as the term *thread* used in the AIX documentation. It is used in the context of the **/proc** file system to keep a common terminology with the **/proc** implementation of other operating systems. The names of the subdirectories are the thread IDs. The test program has only one thread with the ID 54891, as shown in the output of the **ps** command. Therefore, only the content of this one thread directory is shown:

```
# ps -mo THREAD -p 19082
USER  PID  PPID   TID ST  CP PRI SC   WCHAN      F   TT BND COMMAND
```

```

root 19082 20678      - A    0 83 1 700e6244 200001 pts/3 - wc
-      -      - 54891 S    0 83 1 700e6244 10400 - - -

# ls -l /proc/19082/lwp/54891
total 0
dr-xr-xr-x 1 root system 0 Sep 15 15:03 .
dr-xr-xr-x 1 root system 0 Sep 15 15:03 ..
--w----- 1 root system 0 Sep 15 15:03 lwpctl
-r--r--r-- 1 root system 120 Sep 15 15:03 lwpsinfo
-r----- 1 root system 1200 Sep 15 15:03 lwpstatus

```

The `lwpctl`, `lwpsinfo`, and `lwpstatus` files contain thread-specific information to control this thread, for the `ps` command, and about the state, similar to the corresponding files in the `/proc/pid` directory.

As an example of what can be obtained from reading these files, the following lines show the content of the `cred` file (after the use of the `od` command):

```

# ls -l /proc/19082/cred
-r----- 1 root system 128 Sep 15 15:07 /proc/19082/cred

# od -x /proc/19082/cred
0000000 0000 0000 0000 0000 0000 0000 0000 0000
*
0000160 0000 0000 0000 0007 0000 0000 0000 0000
0000200 0000 0000 0000 0002 0000 0000 0000 0003
0000220 0000 0000 0000 0007 0000 0000 0000 0008
0000240 0000 0000 0000 000a 0000 0000 0000 000b
0000260

```

The output in the leftmost column shows the byte offset of the file in octal representation. The remainder of the lines are the actual content of the file in hexadecimal notation. Even if the directory listing shows the size of the file to be 128 bytes or 0200 bytes in octal, the actual output is 0260 or 176 bytes in size. This is due to the dynamic behavior of the last field in the corresponding structure. The digit 7 in the line with the number 0160 specifies the number of groups the user ID running this process belongs to. Because every user ID is at least part of its primary group, but belongs possibly to a number of other groups that cannot be known in advance, only space for the primary group is reserved in the `cred` data structure. In this case, the primary group ID is zero because the user ID running this process is `root`. Reading the complete content of the file, nevertheless, reveals all the other group IDs the user currently belongs to. The group IDs in this case (2, 3, 7, 8, 0xa (10), and 0xb (11)) map to the groups `bin`, `sys`, `security`, `cron`, `audit`, and `lp`. This is exactly the set of groups the user ID `root` belongs to by default.

10.3.1 The /proc file system enhancements (5.2.0)

The /proc file system has been enhanced in Version 5.2 to provide access to additional process information using the new tools **procwdx** and **procfiles**.

Two new directories (/proc/pid#/cwd and /proc/pid#/fd) were created and are the subject of the following discussion.

Examples in this section use the sendmail process. On the running system the PID was 4448.

10.3.2 /proc/pid#/cwd

The /proc/pid#/cwd directory provides access to the current working directory of the process. The link has permissions 555.

An example of the directory structure is shown in the following:

```
# ls -l /proc/4448/cwd
lr-x----- 2 root      system      0 Aug 20 11:31 /proc/4448/cwd ->
/var/spool/mqueue/
```

10.3.3 /proc/pid#/fd

The /proc/pid#/fd directory contains files for all the open file descriptors of the process. As seen in the example, each entry is a decimal number that corresponds to an open file descriptor in the process. Any directories are displayed as links. The following **ls** command output shows the directory layout for sendmail:

```
# ls -l /proc/4448/fd
total 112
c----- 1 root      system      2, 1 Aug 22 17:36 5
-r--r--r-- 1 root      system      54587 Aug 20 00:24 7
```

These enhancements to the /proc file system running under Version 5.2 have enabled the use of the **procwdx** and the **procfiles** commands. Their use is detailed in the following section together with further process control commands, commonly referred to as proctools.

10.4 New proctools (5.2.0)

The /proc-based tools commonly found on System V systems are now include in Version 5.2. They include: **procwdx**, **procfiles**, **procflags**, **proccred**, **procmap**, **procldd**, **procsig**, **procstack**, **proctop**, **procrun**, **procwait**, and **proctree**. These commands are covered in more detail in this section.

10.4.1 procwdx

The **procwdx** command prints the current working directory of a process. The **-F** flag forces **procwdx** to take control of the target process even if another process has control of it, as shown in the following example:

```
# procwdx 4448
4448: /var/spool/mqueue/
```

10.4.2 procfiles

The **procfiles** command prints information about all file descriptors opened by the processes. The **-n** flag names the files referred to by descriptors, and the **-F** flag is the force option, as with the **procwdx** command, as shown in the following example:

```
# procfiles -n 12924
12924 : /usr/sbin/getty /dev/console
Current rlimit: 2000 file descriptors
0: S_IFCHR mode:00 dev:10,4 ino:4463 uid:0 gid:0 rdev:22,0
   O_RDWR name:/dev/lft0
1: S_IFCHR mode:00 dev:10,4 ino:4463 uid:0 gid:0 rdev:22,0
   O_RDWR name:/dev/lft0
2: S_IFCHR mode:00 dev:10,4 ino:4463 uid:0 gid:0 rdev:22,0
   O_RDWR name:/dev/lft0
3: S_IFREG mode:0644 dev:10,5 ino:12340 uid:0 gid:0 rdev:2,104
   O_RDWR size:483328 name:/usr/lib/objrepos/PdAt
4: S_IFREG mode:0644 dev:10,4 ino:47 uid:0 gid:0 rdev:0,315
   O_RDWR size:12288 name:/etc/objrepos/CuDv
5: S_IFREG mode:0644 dev:10,5 ino:12341 uid:0 gid:0 rdev:0,50131
   O_RDWR size:139264 name:/usr/lib/objrepos/PdAt.vc
```

10.4.3 procflags

The **procflags** command prints the **/proc** tracing flags, with the pending and held signals, as shown in the following example:

```
# procflags 4448
4448 : sendmail: accepting connections
data model = _ILP32 flags = PR_FORK
/12913: flags = PR_ASLEEP | PR_NOREGS
```

10.4.4 proccred

The **proccred** command prints effective, real, saved user, and group IDs of processes, as shown in the following example:

```
# proccred 4448
```

```
4448: e/r/suid=0 e/r/sgid=0
```

10.4.5 procmap

The **procmap** command prints address space map of processes, as shown in the following example:

```
# procmap 4448
4448 : sendmail: accepting connections
10000000      1005K  read/exec      sendmail
200003f0      241K  read/write     sendmail
d007f100      79K   read/exec      /usr/lib/libiconv.a
20252bf0      41K   read/write     /usr/lib/libiconv.a
d0076100      33K   read/exec      /usr/lib/libi18n.a
20250190      4K    read/write     /usr/lib/libi18n.a
d0073000      11K   read/exec      /usr/lib/nls/loc/en_US
2024d130      8K    read/write     /usr/lib/nls/loc/en_US
d0093100      71K   read/exec      /usr/lib/libodm.a
f0139220      21K   read/write     /usr/lib/libodm.a
d00be100      67K   read/exec      /usr/lib/libsrc.a
d01cde0      1941K  read/exec      /usr/lib/libc.a
.....
Total          5507K
```

10.4.6 procldd

The **procldd** command lists dynamic libraries loaded, as shown in the following example:

```
# procldd 4448
4448 : sendmail: accepting connections
/usr/lib/libiconv.a
/usr/lib/libi18n.a
/usr/lib/nls/loc/en_US
/usr/lib/libodm.a
/usr/lib/libsrc.a
/usr/lib/libc.a
```

10.4.7 procsig

The **procsig** command lists signal actions of processes, as shown in the following example:

```
# procsig 4448
4448 : sendmail: accepting connections
HUP      caught  RESTART | SIGINFO
INT      caught  RESTART | SIGINFO
QUIT     default RESTART
```


ILL	default	RESTART	
TRAP	default	RESTART	
ABRT	default	RESTART	
EMT	default	RESTART	
FPE	default	RESTART	
KILL	default		
BUS	default	RESTART	
SEGV	default	RESTART	
SYS	default	RESTART	
PIPE	ignored	RESTART	SIGINFO
ALRM	caught	RESTART	SIGINFO
TERM	caught	RESTART	SIGINFO
.....			
UVTALRM	default		
MIGRATE	default		
PRE	default	RESTART	
VIRT	default		
ALRM1	default		
WAITING	default	RESTART	
CPUFAIL	default		
KAP	default		
RETRACT	default		
SOUND	default		
SAK	default		

10.4.8 procstack

The **procstack** command prints a hexadecimal address and symbolic names for each stack frames of the current thread in process, as shown in the following example:

```
# procstack 4448
4448 : sendmail: accepting connections
d024fdf0 select  (? , ? , ? , ? , ? ) + 90
1000ec24 getrequests  (?) + 714
1000051c main  (? , ? , ? ) + 29a8
10000100 __start  () + 8c
```

10.4.9 procstop

The **procstop** command stops processes using the /proc interface on the PR_REQUESTED event, as shown in the following example:

```
# procstop 4448
```

10.4.10 procrun

The **procrun** command starts processes stopped by the previous command, **procstop**, as shown in the following example:

```
# procrun 4448
```

10.4.11 procwait

The **procwait** command waits for all specified processes to stop. -v is the verbose option, as shown in the following example:

```
# procwait -v 4448
```

10.4.12 proctree

The **proctree** command prints a process tree containing the specified process IDs or users, by either specifying the PID or the user ID, as shown in the following example:

```
# proctree 4448
11452  /usr/sbin/srcmstr
      4448  sendmail: accepting connections

# proctree pki
50404  /usr/java131/bin/java -Dcom.tivoli.pki.main.javaPki.remote=true
-verbose -class
20322  db2wdog
      33400  db2sysc
            24670  db2ipccm
                  50092  db2agent (PKIUSER)
                  47268  db2agent (PKIUSER)
                  46346  db2agent (idle)
                  41368  db2agent (PKIUSER)
                  41164  db2agent (PKIUSER)
                  38770  db2agent (PKIUSER)
                  36674  db2agent (PKIUSER)
            36572  db2gds
                  46944  db2pfchr
                  42170  db2pfchr
                  40222  db2loggr
                  39896  db2pfchr
                  39084  db2spmlw
                  37742  db2srvtst
                  37058  db2dlock
                  32708  db2pclnr
                  33226  db2resyn
            38292  db2spmrm
36220  db2tcpcm
```

```

16590    db2tcpcm
11452    /usr/sbin/srcmstr
4880     /usr/sbin/inetd
21176    telnetd -a
19884    -ksh
35838    -ksh
49392    vi PkMessage.log

```

10.5 Process system call tracing with truss

AIX 5L now supports the **truss** command, which allows you to trace system calls executed by a process as well as record the received signals and the occurrence of machine faults.

The application to trace is either specified on the command line of the **truss** command or **truss** can be attached to one or more already running processes by using the **-p** flag with a list of process IDs. The complete list of flags supported by the **truss** command is:

```

# truss
Usage: [ -f ] [ -c ] [ -a ] [ -e ] [ -i ] [ - [ tx ] [ ! ] ]
syscall [ ,syscall ] [ -s [ ! ] signal [ ,signal ] ] [ -m [ ! ] ]
fault [ ,fault ] [ -[ rw ] [ ! ] fd [ ,fd ] ] [ -o outfile ] { command | -p
pid [ . . . ] }

```

If the **-o** flag that redirects the output of **truss** to a file is not used, the **truss** output goes to standard out and can be mixed with the output of the command **truss** is tracing. Before describing the other flags, the following lines show an example of running the **date** command under **truss**:

```

# truss -e -o truss.out date
Thu Sep 14 15:28:20 CDT 2000

# cat truss.out
execve("/usr/bin/date", 0x2FF22C44, 0x2FF22C4C)  argc: 1
envp: _=/usr/bin/truss LANG=en_US LOGIN=root
NLSPATH=/usr/lib/nls/msg/%L/%N:/usr/lib/nls/msg/%L/%N.cat
PATH=/usr/bin:/etc:/usr/sbin:/usr/ucb:/usr/bin/X11:/sbin
LC_FASTMSG=true WINDOWID=4194317
CGI_DIRECTORY=/var/docsearch/cgi-bin LOGNAME=root
MAIL=/usr/spool/mail/root LOCPATH=/usr/lib/nls/loc USER=root
DOCUMENT_SERVER_MACHINE_NAME=localhost AUTHSTATE=compat
DISPLAY=9.3.240.103:0.0 SHELL=/usr/bin/ksh ODMDIR=/etc/objrepos
DOCUMENT_SERVER_PORT=49213 HOME=/ TERM=xterm
MAILMSG=[YOU HAVE NEW MAIL] ITECONFIGSRV=/etc/IMNSearch PWD=/
DOCUMENT_DIRECTORY=/usr/docsearch/html TZ=CST6CDT
ITECONFIGCL=/etc/IMNSearch/clients ITE_DOC_SEARCH_INSTANCE=search

```

```

A__z=! LOGNAME
sbrk(0x00000000) = 0x20001C50
brk(0x20011C50) = 0
getuidx(4) = 0x00000000
getuidx(2) = 0x00000000
getuidx(1) = 0x00000000
getgidx(4) = 0
getgidx(2) = 0
getgidx(1) = 0
__loadx(0x01000080, 0x2FF1E8E0, 0x00003E80, 0x2FF22870, 0x00000000, 0x00000000,
0x80000000, 0x7F7F7F7F) = 0xD0072130
__loadx(0x01000180, 0x2FF1E8D0, 0x00003E80, 0xF0133E10, 0xF0133D40, 0x00000000,
0xFFFFFFFF, 0xD0074388) = 0xF02885B8
__loadx(0x07080000, 0xF0133DE0, 0xFFFFFFFF, 0xF02885B8, 0x00000000, 0x6000C018,
0x600078AF, 0x00000000) = 0xF02892BC
__loadx(0x07080000, 0xF0133D20, 0xFFFFFFFF, 0xF02885B8, 0x00000000, 0x6000C018,
0x600078AF, 0x00000000) = 0xF02892C8
__loadx(0x07080000, 0xF0133DF0, 0xFFFFFFFF, 0xF02885B8, 0x00000000, 0x6000C018,
0x600078AF, 0x00000000) = 0xF02892F8
__loadx(0x07080000, 0xF0133D30, 0xFFFFFFFF, 0xF02885B8, 0x00000000, 0x6000C018,
0x600078AF, 0x00000000) = 0xF0289304
__loadx(0x07080000, 0xF0133DB0, 0xFFFFFFFF, 0xF02885B8, 0x00000000, 0x6000C018,
0x600078AF, 0x00000000) = 0xF02892D4
__loadx(0x07080000, 0xF0133D60, 0xFFFFFFFF, 0xF02885B8, 0x00000000, 0x6000C018,
0x600078AF, 0x00000000) = 0xF02892EC
__loadx(0x07080000, 0xF0133DC0, 0xFFFFFFFF, 0xF02885B8, 0x00000000, 0x6000C018,
0x600078AF, 0x00000000) = 0xF0289310
__loadx(0x07080000, 0xF0133DD0, 0xFFFFFFFF, 0xF02885B8, 0x00000000, 0x6000C018,
0x600078AF, 0x00000000) = 0xF0289340
__loadx(0x07080000, 0xF0133D50, 0xFFFFFFFF, 0xF02885B8, 0x00000000, 0x6000C018,
0x600078AF, 0x00000000) = 0xF0289328
__loadx(0x07080000, 0xF0133D70, 0xFFFFFFFF, 0xF02885B8, 0x00000000, 0x6000C018,
0x600078AF, 0x00000000) = 0xF02892F8
__loadx(0x07080000, 0xF0133D30, 0xFFFFFFFF, 0xF02885B8, 0x00000000, 0x6000C018,
0x600078AF, 0x00000000) = 0xF0289304
__loadx(0x07080000, 0xF0133DB0, 0xFFFFFFFF, 0xF02885B8, 0x00000000, 0x6000C018,
0x600078AF, 0x00000000) = 0xF02892D4
__loadx(0x07080000, 0xF0133D60, 0xFFFFFFFF, 0xF02885B8, 0x00000000, 0x6000C018,
0x600078AF, 0x00000000) = 0xF02892EC
__loadx(0x07080000, 0xF0133DC0, 0xFFFFFFFF, 0xF02885B8, 0x00000000, 0x6000C018,
0x600078AF, 0x00000000) = 0xF0289310
__loadx(0x07080000, 0xF0133DD0, 0xFFFFFFFF, 0xF02885B8, 0x00000000, 0x6000C018,
0x600078AF, 0x00000000) = 0xF0289340
__loadx(0x07080000, 0xF0133D50, 0xFFFFFFFF, 0xF02885B8, 0x00000000, 0x6000C018,
0x600078AF, 0x00000000) = 0xF0289328
__loadx(0x07080000, 0xF0133D70, 0xFFFFFFFF, 0xF02885B8, 0x00000000, 0x6000C018,
0x600078AF, 0x00000000) = 0xF028934C
access("/usr/lib/nls/msg/en_US/date.cat", 0) = 0
_getpid() = 19528

```

```

kiocntl(1, 22528, 0x00000000, 0x00000000)      = 0
kwrite(1, 0xF018ABD8, 29)                       = 29
kfcntl(1, F_GETFL, 0xF0170918)                  = 2
kfcntl(2, F_GETFL, 0xF0170918)                  = 2
_exit(0)

```

The **-e** flag is responsible for the display of the environment content in the **truss** output file. By default, **truss** does not trace forked processes; the **-f** flag will force **truss** to go into forked processes. Interruptible sleeping system calls are displayed once on completion if the **-i** flag is used. The **-c** flag generates a summary file instead of the detailed report shown previously. The **-c** flag also gives a count for how often a specific system call was executed and the overall time spent in total in it.

The other flags allow the inclusion (or exclusion, if the exclamation point is used) by name of specific system calls, signals, machine faults, or the data read from or written to specific file descriptors. By default, **truss** displays symbolic constants from the appropriate system header files as the arguments of the system calls. This can be forced to always display hexadecimal values by using the **-x** flag. These four flags accept the symbol *all* to include all possible system calls, signals, and so forth. The return value of the system call is shown on the right-hand side of the equal sign.

For this simple **date** command (shown in the previous output), the **truss** output file is already about 10 KB. You need to reduce the number of system calls you are tracing, or attach **truss** to a running process only for a limited amount of time, to keep the size of the **truss** output file within a manageable range.

10.5.1 Truss enhancements (5.2.0)

The **truss** command has been enhanced to optionally add timestamps on each output file, and to be able to trace library calls. For each call, it prints parameters and return code values. A subset of libraries and/or routines can be selected or excluded from tracing.

To display timestamps along with the standard output using the new **-d** flag:

```

# truss -d ifconfig -a > /tmp/out
0.0006:      execve("/etc/ifconfig", 0x2FF22BF4, 0x2FF22C00)  argc: 2
0.0247:      sbrk(0x00000000)                                = 0x2000220C
0.0254:      sbrk(0x00000004)                                = 0x2000220C
0.0262:      sbrk(0x00010010)                                = 0x20002210
0.0268:      getuidx(4)                                       = 0
0.0273:      getuidx(2)                                       = 0
0.0280:      getuidx(1)                                       = 0
0.0286:      getgidx(4)                                       = 0
0.0291:      getgidx(2)                                       = 0

```

```

0.0296:      getgid(1)                                = 0
0.0303:      __loadx(0x01000080, 0x2FF1E760, 0x00003E80, 0x2FF226F0,
0x00000000) = 0x20013130
0.0312:      __loadx(0x01000180, 0x2FF1E750, 0x00003E80, 0xF0365734,
0xF0365664) = 0x20016190
0.0330:      __loadx(0x07080000, 0xF0365704, 0xFFFFFFFF, 0x20016190,
0x00000000) = 0x20016F7C
0.0335:      __loadx(0x07080000, 0xF0365644, 0xFFFFFFFF, 0x20016190,
0x00000000) = 0x20016F88
0.0339:      __loadx(0x07080000, 0xF0365714, 0xFFFFFFFF, 0x20016190,
0x00000000) = 0x20016FB8
0.0343:      __loadx(0x07080000, 0xF0365654, 0xFFFFFFFF, 0x20016190,
0x00000000) = 0x20016FC4
0.0347:      __loadx(0x07080000, 0xF03656D4, 0xFFFFFFFF, 0x20016190,
0x00000000) = 0x20016F94
0.0351:      __loadx(0x07080000, 0xF0365684, 0xFFFFFFFF, 0x20016190,
.....

```

The output for **truss** commands can become very large. For full documentation on this particular command, refer to the Online Documentation and man pages.

10.6 User API for Sun threaded applications (5.2.0)

The new user thread library provides for source compatibility with Solaris thread routines. This allows applications that are run on Solaris machines to be recompiled without change to their application source code so that they can run under Version 5.2.

The API for Sun threaded applications for Version 5.2 is designed to be compatible with Solaris Version 8 of the thread library. Versions prior to Version 8 are not supported.

The Sun user thread library does not alter the pthread library, so compatibility with POSIX and X/Open standards for pthreads are maintained. The Sun user threads have been put on top of the POSIX threads so as to not affect POSIX performance.

There is, however, no binary compatibility with applications compiled under Solaris. All source code is required to be recompiled under Version 5.2.

10.6.1 Application binary interface (ABI)

The design of the existing ABI of the pthread library is not altered with respect to:

- Exported function names

- ▶ Exported function signatures
- ▶ Exported data structures
- ▶ Exported data structures used in file formats

10.6.2 AIX LPP packaging

The filesets listed in Table 10-13 contain the AIX files needed for the user API for Sun threaded applications.

Table 10-13 Filesets for Sun user thread library

File	Fileset
/usr/ccs/lib/libthread.a	bos.adt.lib
/usr/include/thread.h	bos.adt.include
/usr/include/synch.h	bos.adt.include

There are no user interfaces required for either command line, SMIT, or Web-based System Manager. All applications need to be recompiled.

10.7 System V Release 4 print subsystem

On AIX 5L:

- ▶ Both the AIX and the System V Release 4 print subsystems are available.
- ▶ The AIX print subsystem is the default.

When the AIX print subsystem was created, it was designed to combine the features of the System V and Berkeley Software Distribution (BSD) printing standard, along with some unique features found only in AIX. This design had some distinct advantages in the past:

- ▶ Easy transition to AIX

To provide an easy transition from another operating system to AIX, many of the commands traditionally used for printing were provided. For example, BSD users could still print using the same **lpr** command they had become accustomed to. Also, scripts that were used to print did not necessarily need to be changed.

- ▶ Powerful and versatile print drivers

The print drivers used to drive specific printers were designed in such a way that most printing options available on the printer could be used by selecting one or more of the many flags known to the backend. In addition, the print

data stream could easily be modified with user- and system-defined filters and formatters.

► Limits fields

Limits fields that gave users a valid range of choices for each option would prohibit a user from using an incorrect value, and would send a message to the user stating the reason for the resulting print job rejection.

However, the same features that gave AIX printing an advantage over other UNIX operating systems also served to make the AIX print subsystem less compliant to widely used standards.

The System V Release 4 (SVR4) print subsystem was added to AIX 5L with the long-term goal of making it the default print solution for AIX. Section 10.7.1, “Understanding the System V print service” on page 698, provides a brief overview of the print request processing of the newly implemented System V print subsystem in AIX 5L, and 10.7.3, “System V print subsystem management” on page 711, describes the commands that are available to manage the System V printer services. System administrators who prefer to use graphical system management tools will find useful information in 10.7.5, “User interface for AIX and System V print subsystems” on page 715.

If the code for both print subsystems is installed, the base operating system of the current AIX 5L release uses the traditional AIX print subsystem by default and the System V print subsystem is not active. Section 10.7.2, “Packaging and installation” on page 701, covers the details about fileset packaging and the installation of the System V print subsystem support in AIX 5L.

AIX 5L provides a command menu, a SMIT menu, and a Web-based System Manager menu, which allows the system administrator to switch between the AIX and the System V print subsystems, but will not allow both print subsystems to be active at the same time. Section 10.7.7, “Switching between AIX and System V print subsystems” on page 723, gives in-depth information about the switching process and the related commands.

Supplemental information about the user interface specification, the terminfo database, and the supported printers can be found in 10.7.4, “User interface specifications” on page 713, and 10.7.6, “Terminfo and supported printers” on page 720.

10.7.1 Understanding the System V print service

The System V print subsystem was ported from SCO's UnixWare 7 to AIX 5L. The print subsystem, as such, supports local printing (parallel and serial), remote printing using BSD's lpd protocol (RFC1179), and network printing using

Hewlett-Packard's (HP) JetDirect. The code was internationalized to conform to and to comply with AIX international standards and requirements.

The System V print service is a collection of utilities that assists you, as system administrator (or printer administrator), to configure, monitor, and control the printers on your system.

The print service:

- ▶ Receives files users want to print
- ▶ Filters the files (if needed), so they can print correctly
- ▶ Schedules the work of one or more printers
- ▶ Starts programs that interface with the printers
- ▶ Keeps track of the status of jobs
- ▶ Alerts you to printer problems
- ▶ Keeps track of mounting forms and filters
- ▶ Issues error messages when problems arise

Figure 10-1 on page 700 shows an overview of the processing of a print request, illustrates the following explanations, and helps to understand the overall concept.

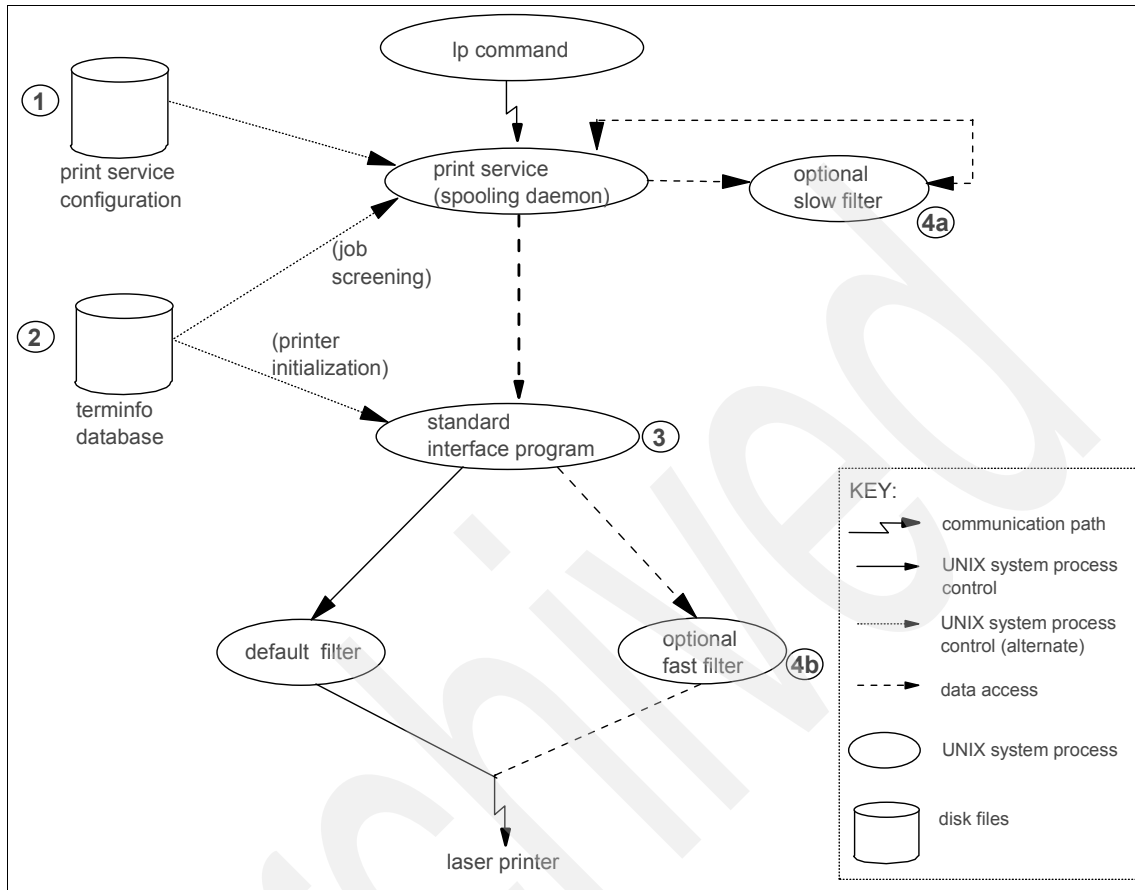


Figure 10-1 Overview of print request processing

When a user sends a file to a printer, the print service assigns a unique name, the request ID, to the request (print job).

The request ID consists of the name of the printer on which the file is to be printed and a unique number identifying the file. Use this request ID to find out the status of the print job or to cancel the print job. The print service keeps track of all the print requests in an associated request log.

The print job is spooled, or lined up, with other print jobs to be sent to a printer. Each print job is processed and waits its turn in line to be printed. This line of pending print jobs is called a print queue.

Each printer has its own queue; you can hold jobs in the queue, move jobs up in a queue, or transfer jobs to another queue.

Each print request is sent to a spooling daemon, **lp sched**, that keeps track of all the jobs. The daemon is created when you start the print service. The spooling daemon is also responsible for keeping track of the status of the printers and slow filters; when a printer finishes printing a job, the daemon starts printing another job if one is queued.

You can customize the print service by adjusting or replacing some of the items shown in Figure 10-1 on page 700. The following numbers are explanations of the keys used in the diagram:

1. For most printers, you need only to change the printer configuration stored on disk. For further details, refer to the **lpadmin** command documentation for adding or modifying a local printer.
2. The print service relies on the standard interface script and the terminfo database to initialize each printer and set up a selected page size, character pitch, line pitch, and character set. For printers that are not represented in the terminfo database, you can add a new entry that describes the capabilities of the printer. The print service uses the terminfo database in two parallel capacities: Screening print requests to ensure that those requests can be handled by the desired printer, and setting the printer so it is ready to print the requests. For example, if the terminfo database does not show a printer capable of setting a page length requested by a user, the spooling daemon rejects the request. However, if it does show it to be capable, then the interface program uses the same information to initialize the printer.
3. If you have a particularly complicated printer or if you want to use features not provided by the print service, you can change the interface script. This script is responsible for managing the printer: It prints the banner page, initializes the printer, and invokes a filter to send copies of the user's files to the printer.
4. To provide a link between the applications used on your system and the printers, you can add slow and fast filters. Each type of filter can convert a file into another form (for example, mapping one set of escape sequences into another), and can provide a special setup by interpreting print modes requested by a user. Slow filters are run separately by the spooling daemon to avoid slow queues. Fast filters are run so their output goes directly to the printer; thus, they can exert control over the printer.

10.7.2 Packaging and installation

The AIX and System V print subsystems are both packaged with the base operating system, but which filesets are installed during the initial base installation depends on the hardware configuration of your system. The option chosen for the Installation Configuration (default/minimal) under the Advanced Options menu during the base system installation process does not have any impact on the selection and installation of the print subsystem filesets.

The filesets given below provide the core function of the AIX print subsystem:

bos.rte.printers	Frontend printer support
printers.rte	Printer backend
printers.msg.xx_XX.rte	Printer backend messages for the system-specific locale indicated by xx_XX in the fileset name

The frontend printer support, **bos.rte.printers**, is part of the **bos.rte** file package, and therefore is always installed on the system. This fileset provides frontend print commands, such as **qprt**, **lpr**, **enq**, **mkque**, and **rmque**, that allow a user or the system administrator to interact with the **qdaemon**'s spooler queues. For compatibility and usability reasons, the traditional AIX print subsystem maps several System V and BSD print commands to the AIX-specific print commands. For example, the **lp** command used to be nothing more than a program that translates the System V **lp** flags to their counterparts of the **enq** AIX command, and after all the command line arguments were processed, the translated list of flags is finally used to call the **enq** command. As far as the frontend is concerned, the System V commands affected are **cancel**, **lp**, and **lpstat**. For BSD, the relevant frontend commands are **lpq**, **lpr**, and **lprm**.

In AIX 5L, the System V and BSD frontend print commands are still in the **/usr/bin** directory, but, by default, they are now linked to the traditional AIX print command wrappers in the **/usr/aix/bin** directory:

```
# ls -l /usr/bin | grep aix
lrwxrwxrwx 1 root system      19 Sep 06 15:46 cancel ->
/usr/aix/bin/cancel
lrwxrwxrwx 1 root system      15 Sep 06 15:46 lp -> /usr/aix/bin/lp
lrwxrwxrwx 1 root system      16 Sep 06 15:46 lpq -> /usr/aix/bin/lpq
lrwxrwxrwx 1 root system      16 Sep 06 15:46 lpr -> /usr/aix/bin/lpr
lrwxrwxrwx 1 root system      17 Sep 06 15:46 lprm -> /usr/aix/bin/lprm
lrwxrwxrwx 1 root system      19 Sep 06 15:46 lpstat ->
/usr/aix/bin/lpstat
```

The AIX printer backend is a collection of programs called by the spooler's **qdaemon** command to manage a print job that is queued for printing. The printer backend performs the following functions:

- ▶ Receives a list of one or more files to be printed from the **qdaemon** command
- ▶ Uses printer and formatting attribute values from the database; overridden by flags entered on the command line
- ▶ Initializes the printer before printing a file
- ▶ Runs filters as necessary to convert the print data stream to a format supported by the printer
- ▶ Provides filters for simple formatting of ASCII documents

- ▶ Provides support for printing national language characters
- ▶ Passes the filtered print data stream to the printer device driver
- ▶ Generates header and trailer pages
- ▶ Generates multiple copies
- ▶ Reports paper out, intervention required, and printer error conditions
- ▶ Reports problems detected by the filters
- ▶ Cleans up after a print job is canceled
- ▶ Provides a print environment that a system administrator can customize to address specific printing needs

The AIX printer backend files `printers.rte` belongs to several of the default system bundles that are located in the `/usr/sys/inst.data/sys_bundle` directory. These bundles include:

App-Dev.bnd	Application development bundle: A collection of software products for developing application programs
Client.bnd	Client bundle: A collection of software products for single user systems running in a stand-alone or networked client environment
Pers-Prod.bnd	Personal productivity bundle: A collection of software products for graphical desktop systems running AIX and PC applications
Server.bnd	Server bundle: A collection of software products for multi-user systems running in a stand-alone or networked environment

The fact that the bundles listed belong to the default system bundle category does not imply that any of these bundles are installed by default. They are predefined and supplied for your convenience, but the system administrator would have to intentionally initiate the installation of any of the bundles.

Furthermore, the `printers.rte` fileset is not listed in any of the default system bundles, which are used during the base installation process:

ASCII.autoi	An ASCII terminal system bundle file that lists filesets to install if the console is not a low function terminal (LFT)
BOS.autoi	A system bundle file that lists the group of packages and filesets that will always be installed when the Default Installation Configuration under the Advanced Options menu (during the base system installation process) was specified

MIN_BOS.autoi	A system bundle file that lists the group of packages and filesets that will always be installed when the Minimal Installation Configuration under the Advanced Options menu (during the base system installation process) was specified
GOS.autoi	A graphics system bundle file that lists filesets to install if the console is an LFT and when the Default Installation Configuration was chosen (during the base system installation process)
MIN_GOS.autoi	A graphics system bundle file that lists filesets to install if the console is an LFT and when the Minimal Installation Configuration was chosen (during the base system installation process)

Since printers.rte is not explicitly included in any of the bundle files with the autoi extension, the requisite for printers.rte of other filesets determines whether the backend support for the AIX print subsystem is installed. The fileset dependencies are defined by the multi-volume .toc file in the /usr/sys/mvCD directory of the installation media, and at the time of publication, four fileset dependencies designated printers.rte as a required fileset for installation. These fileset dependencies include:

bos.txt.tfs	Text formatting services commands
printers.ibmNetPrinter.attach	en_US IBM Network Printer attachment
printers.ibmNetColor.attach	en_US IBM Network Color Printer attachment
printers.hpJetDirect.attach	en_US Hewlett-Packard JetDirect Network Printer

The most significant fileset of the ones listed is bos.txt.tfs. The text formatting services are included in GOS.autoi and MIN_GOS.autoi and are also directly required by the X11.Dt.rte fileset for the AIX Common Desktop Environment (CDE) support.

Table 10-14 summarizes the different combinations for the AIX print subsystem backend support. These combinations' parts include the HW configuration, installation configuration, and system administrators intervention.

Table 10-14 AIX print subsystem backend support

Hardware graphics support	Installation configuration	Installation initiation and process	AIX print backend support
No	Minimal	NA	No

Hardware graphics support	Installation configuration	Installation initiation and process	AIX print backend support
No	Default	NA	No
Yes	Minimal	BOS installation: MIN_GOS.autoi	Yes
Yes	Default	BOS installation: GOS.autoi	Yes
No	Minimal/ default	Manual Installation: printers.rte	Yes
No	Minimal/ default	Manual Installation: App-Dev.bnd Cleint.bnd Pers.Prod.bnd Server.bnd	Yes

As mentioned before, the traditional AIX print subsystem maps several System V and BSD print commands to the AIX-specific print commands. As far as the backend print support is concerned, the only two System V commands affected are **disable** and **enable**. In AIX 5L, these specific System V backend print commands are still in the /usr/bin directory, but by default they are now linked to the traditional AIX print command wrappers in the /usr/aix/bin directory:

```
# ls -l /usr/bin | grep -E "\/enable|disable"
lrwxrwxrwx 1 root system      20 Sep 05 13:46 disable ->
/usr/aix/bin/disable
lrwxrwxrwx 1 root system      19 Sep 05 13:46 enable ->
/usr/aix/bin/enable
```

In addition to the AIX print command wrappers for System V and BSD print commands in the /usr/aix/bin directory, a new lock file `_AIX_print_subsystem` is installed under the /usr/aix directory. The existence of the lock file indicates that the AIX print subsystem is active. For reference, a full listing of the /usr/aix directory is provided in the following:

```
# ls -lR /usr/aix
total 8
-rw-rw-r-- 1 root system      0 Sep 01 18:02 _AIX_print_subsystem
drwxr-xr-x 2 bin bin        512 Sep 05 13:46 bin
/usr/aix/bin:
total 576
-r-xr-xr-x 1 bin bin        33648 Aug 24 21:22 cancel
-r-xr-x--- 1 root printq    33488 Aug 24 21:22 disable
-r-xr-x--- 1 root printq    33376 Aug 24 21:22 enable
-r-xr-xr-x 1 bin bin        34228 Aug 24 21:22 lp
```

-r-xr-xr-x	1	bin	bin	33916	Aug 24 21:22	lpq
-r-xr-xr-x	1	bin	bin	35236	Aug 24 21:22	lpr
-r-xr-xr-x	1	bin	bin	34312	Aug 24 21:22	lprm
-r-xr-xr-x	1	bin	bin	35368	Aug 24 21:22	lpstat

The package of the System V print subsystem is named `bos.svprint` and consists of four filesets:

bos.svprint.fonts	System V print fonts
bos.svprint.hpnp	System V Hewlett-Packard JetDirect
bos.svprint.ps	System V print postscript
bos.svprint.rte	System V print subsystem

These filesets are supplemented by the locale-specific message support and the System V printer terminal definitions:

bos.msg.xx_XX.svprint	System V print subsystem messages for the system-specific locale (indicated by <code>xx_XX</code> in the fileset name)
bos.terminfo.svprint.data	System V printer terminal definitions

The filesets `bos.svprint.*` and `bos.terminfo.svprint.data` are included in the `BOS.autoi` system bundle and will be installed by default on all AIX 5L systems. The main script that handles the system installation tasks, `/usr/lpp/bosinst/bi_main`, also ensures that the locale-specific message support is available through `bos.msg.xx_XX.svprint`.

All System V and BSD commands that are mapped by the executables in the `/usr/aix/bin` directory to the AIX print subsystem-specific commands have their native System V or BSD counterpart in the `/usr/sysv/bin` directory. During a switch from the AIX to the System V print subsystem, the respective duplicate commands will be handled by removing the inactive print subsystem's command symbolic links and adding new symbolic links for the active commands. The following directory listing reflects this configuration on a system where the initially active AIX print subsystem was deactivated and switched to the System V print subsystem by the use of the newly introduced **`switch.prt`** command:

```
ls -l /usr/bin | grep sysv
lrwxrwxrwx 1 root system      20 Sep 12 18:58 cancel ->
/usr/sysv/bin/cancel
lrwxrwxrwx 1 root system      21 Sep 12 18:58 disable ->
/usr/sysv/bin/disable
lrwxrwxrwx 1 root system      20 Sep 12 18:58 enable ->
/usr/sysv/bin/enable
lrwxrwxrwx 1 root system      16 Sep 12 18:58 lp -> /usr/sysv/bin/lp
lrwxrwxrwx 1 root system      17 Sep 12 18:58 lpq -> /usr/sysv/bin/lpq
lrwxrwxrwx 1 root system      17 Sep 12 18:58 lpr -> /usr/sysv/bin/lpr
```



```

lrwxrwxrwx 1 root system      18 Sep 12 18:58 lprm -> /usr/sysv/bin/lprm
lrwxrwxrwx 1 root system      20 Sep 12 18:58 lpstat ->
/usr/sysv/bin/lpstat

```

Once the System V print subsystem is active, the new lock file `_SYS5_print_subsystem` will be present in the `/usr/sysv` directory and the AIX print subsystem lock file `/usr/aix/_AIX_print_subsystem` will no longer exist. You will find the recursive listing for the `/usr/sysv` directory in the following example (note the differences in user and group ownership in comparison to the executables in the `/usr/aix/bin` directory):

```

# ls -lR /usr/sysv
total 8
-r--r--r-- 1 root      system0 Sep 12 16:13 _SYS5_print_subsystem
drwxr-xr-x 2 bin      bin 512 Dec 31 1969 bin
/usr/sysv/bin:
total 2136
---x--x--x 1 lp      lp 112506 Aug 24 21:21 cancel
---s--x--- 1 root    lp 113034 Aug 24 21:22 disable
---s--x--- 1 root    lp 113034 Aug 24 21:22 enable
---x--x--x 1 lp      lp 137338 Aug 24 21:21 lp
-r-sr-xr-x 1 lp      lp 166690 Aug 24 21:22 lpq
-r-xr-xr-x 1 bin     bin 27182 Aug 24 21:22 lpr
-r-xr-xr-x 1 bin     bin 116930 Aug 24 21:22 lprm
---x--x--x 1 lp      lp 189442 Aug 24 21:21 lpstat

```

AIX 5L introduces a new user named `lp` and a related group named the same. The user `lp` is added to the `/etc/passwd` file for ownership of a majority of the files, which belong to the `bos.svprint` package. The entry in the `/etc/passwd` file is similar to the following example:

```
lp:*:11:11::/var/spool/lp:/bin/false
```

The group `lp` is added to the `/etc/group` file for group ownership of a majority of the files, which belong to the `bos.svprint` package. The entry in the `/etc/group` file is similar to the following example:

```
lp:!11:root,lp,printq
```

Furthermore, the `lp` user is added to the formerly existing `printq` group. The entry in the `/etc/group` file is similar to the following example:

```
printq:!9:lp
```

The `lp` user and a user who belongs to the `lp` group can administer the System V print subsystem, while root user and a user who belongs to the `printq` group (the newly added `lp` user is also a member of the `printq` group) can administer the AIX print subsystem. The root user can administer both print subsystems, since the root user belongs to both `printq` and `lp` groups.

The AIX print subsystem is active by default. For both print subsystems, the active frontend commands are located and accessible as always through links in the /usr/bin directory. The commands for the frontend that are not active are not located in the directories, which are normally accessible to users through the standard definition of the PATH environment variable. To use the inactive frontend, it must be switched using a command or, preferably, by the use of the System Management Interface Tool (SMIT), or by the Web-based System Management tool. More details about switching between the different print subsystems are given in 10.7.7, “Switching between AIX and System V print subsystems” on page 723. Only one frontend can be active at any moment.

The remainder of this section provides a set of comprehensive listings of files, directories, user and administrative commands, and internal programs that are installed or created on your system in order to support System V printing. For each entity, the file mode, ownership, group ownership, and the fully qualified path name is given. Separate listings account for the differences, which depend on the type of the active print subsystem, and some comments are given for further explanation.

Changes and additions, which were applied to the bos.rte.printers fileset, are as follows:

File Mode	Owner	Group	Pathname	
=====	=====	=====		
drwxr-xr-x	bin	bin	/usr/aix/bin	(AIX)
-rwxr-xr-x	bin	bin	/usr/aix/bin/cancel	(AIX)
-rwxr-xr-x	bin	bin	/usr/aix/bin/lp	(AIX)
-rwxr-xr-x	bin	bin	/usr/aix/bin/lpq	(AIX)
-rwxr-xr-x	bin	bin	/usr/aix/bin/lpr	(AIX)
-rwxr-xr-x	bin	bin	/usr/aix/bin/lprm	(AIX)
-rwxr-xr-x	bin	bin	/usr/aix/bin/lpstat	(AIX)
-r-sr-x---	root	system	/usr/sbin/switch.prt	(AIX)
-rwx-----	root	system	/usr/sbin/switch.prt.subsystem	(AIX)

During the installation of AIX 5L, the bos.rte.printers fileset and the newly introduced directory /usr/aix/bin are created. They hold the AIX print subsystem BSD compatibility executables. The switch.prt executable and switch.prt.subsystem script allow switching to the System V print subsystem.

Links and the lock file that were created during the base operating system installation process are as follows:

File Mode	Owner	Group	Pathname
=====	=====	=====	
lrwxrwxrwx	root	system	/usr/bin/cancel -> /usr/aix/bin/cancel
lrwxrwxrwx	root	system	/usr/bin/lp -> /usr/aix/bin/lp

```

lrwxrwxrwx  root    system    /usr/bin/lpq -> /usr/aix/bin/lpq
lrwxrwxrwx  root    system    /usr/bin/lpr -> /usr/aix/bin/lpr
lrwxrwxrwx  root    system    /usr/bin/lprm -> /usr/aix/bin/lprm
lrwxrwxrwx  root    system    /usr/bin/lpstat -> /usr/aix/bin/lpstat
lrwxrwxrwx  root    system    /usr/bin/disable -> /usr/aix/bin/disable
lrwxrwxrwx  root    system    /usr/bin/enable -> /usr/aix/bin/enable
-rwxrwx---  root    system    /usr/aix/_AIX_print_subsystem      (AIX)

```

The listed links and the lock file are only present when the traditional AIX print subsystem is active, and they are created during the BOS installation process by the function `Add_Printer_Links` of the `bi_main` script. For your reference, an excerpt of the relevant section in the `bi_main` script is provided in the following example:

```

...
# Add_Printer_Links
# Adds links and touches a file, to support
# the repackaging of printer filesets.
# This is only called for product installs ($PT=yes).
#
function Add_Printer_Links
{
...

    ln -s /usr/aix/bin/cancel /usr/bin/cancel
    ln -s /usr/aix/bin/lp /usr/bin/lp
    ln -s /usr/aix/bin/lpstat /usr/bin/lpstat
    ln -s /usr/aix/bin/lpq /usr/bin/lpq
    ln -s /usr/aix/bin/lpr /usr/bin/lpr
    ln -s /usr/aix/bin/lprm /usr/bin/lprm

    touch /usr/aix/_AIX_print_subsystem
    return 0
}
...

```

Changes and additions, which were applied to the `printers.rte` fileset, appear as follows:

File Mode	Owner	Group	Pathname
=====	=====	=====	
-r-xr-x---	root	printq	/usr/aix/bin/disable (AIX)
-r-xr-x---	root	printq	/usr/aix/bin/enable (AIX)
lrwxrwxrwx	root	system	/usr/bin/disable -> /usr/aix/bin/disable (AIX)
lrwxrwxrwx	root	system	/usr/bin/enable -> /usr/aix/bin/enable (AIX)

The links `/usr/bin/disable` and `/usr/bin/enable` are created during the `printers.rte` post-installation phase.

A list of all files and directories in bos.svprint.rte are as follows:

File Mode	Owner	Group	Pathname
=====	=====	=====	
drwxrwxr-x	lp	lp	/usr/lib/lp
drwxrwxr-x	lp	lp	/usr/lib/lp/bin
drwxrwxr-x	lp	lp	/usr/lib/lp/model
drwxrwxr-x	root	system	/usr/lib/lp/objrepos
drwxr-xr-x	bin	bin	/usr/sysv
drwxr-xr-x	bin	bin	/usr/sysv/bin
-r-xr-xr-x	bin	bin	/usr/bin/lpc
-r--r--r--	lp	lp	/usr/lib/lp/bin/alert.proto
---x--x--x	lp	lp	/usr/lib/lp/bin/drain.output
---x--x--x	lp	lp	/usr/lib/lp/bin/lp.cat
---x--x--x	lp	lp	/usr/lib/lp/bin/lp.lvlproc
---x--x--x	lp	lp	/usr/lib/lp/bin/lp.pr
---x--x--x	lp	lp	/usr/lib/lp/bin/lp.set
---x--x--x	lp	lp	/usr/lib/lp/bin/lp.tell
-r-xr-xr-x	lp	lp	/usr/lib/lp/bin/slow.filter
---s--x---	root	lp	/usr/lib/lp/lpsched
---s--x---	root	lp	/usr/lib/lp/lpNet
--x--x--x-	lp	lp	/usr/lib/lp/model/B2
-r-xr-xr-x	lp	lp	/usr/lib/lp/model/B2.banntrail
-r-xr-xr-x	lp	lp	/usr/lib/lp/model/B2.job
-rwxrwxr-x	lp	lp	/usr/lib/lp/model/PS
-rwxr-xr-x	lp	lp	/usr/lib/lp/model/standard
---s--x---	root	lp	/usr/sbin/accept
---s--x---	root	lp	/usr/sbin/lpadmin
---s--x---	root	lp	/usr/sbin/lpfilter
---s--x---	root	lp	/usr/sbin/lpforms
---s--x---	root	lp	/usr/sbin/lpmove
---s--x---	root	lp	/usr/sbin/lpshut
---s--x---	root	lp	/usr/sbin/lpsystem
---s--x---	root	lp	/usr/sbin/lpusers
---s--x---	root	lp	/usr/sbin/reject
---x--x--x	lp	lp	/usr/sysv/bin/cancel
---s--x---	root	lp	/usr/sysv/bin/disable
---s--x---	root	lp	/usr/sysv/bin/enable
---x--x--x	lp	lp	/usr/sysv/bin/lp
-r-sr-xr-x	lp	lp	/usr/sysv/bin/lpq
-r-xr-xr-x	bin	bin	/usr/sysv/bin/lpr
-r-xr-xr-x	bin	bin	/usr/sysv/bin/lprm
---x--x--x	lp	lp	/usr/sysv/bin/lpstat

Links and files that are exclusively present when the System V print subsystem is active are as follows:

File Mode	Owner	Group	Pathname
=====	=====	=====	
=====			
lrwxrwxrwx	root	system	/usr/bin/cancel -> /usr/sysv/bin/cancel
lrwxrwxrwx	root	system	/usr/bin/lp -> /usr/sysv/bin/lp
lrwxrwxrwx	root	system	/usr/bin/lpq -> /usr/sysv/bin/lpq
lrwxrwxrwx	root	system	/usr/bin/lpr -> /usr/sysv/bin/lpr
lrwxrwxrwx	root	system	/usr/bin/lprm -> /usr/sysv/bin/lprm
lrwxrwxrwx	root	system	/usr/bin/lpstat -> /usr/sysv/bin/lpstat
lrwxrwxrwx	root	system	/usr/bin/disable -> /usr/sysv/bin/disable
lrwxrwxrwx	root	system	/usr/bin/enable -> /usr/sysv/bin/enable
[Created on the fly when switching to System V print subsystem]			
-rwxrwx---	root	lp	/usr/sysv/_SYS5_print_subsystem

10.7.3 System V print subsystem management

In general, print administrators should use the Web-based System Manager to manage the System V print service. For further details about the Web-based System Manager support for the System V print service management, refer to 10.7.5, “User interface for AIX and System V print subsystems” on page 715. If you need to manage your print service from the command line, the remainder of this section provides a brief summary of the System V print service command line interface.

Table 10-15 lists the print service commands available to all users. All commands are located in the /usr/bin directory.

Table 10-15 Print service commands available to all users

Command	Description
cancel	The cancel command allows users to cancel print requests previously sent with the lp command. This command permits cancellation of requests based on their request-ID or based on the login ID of their owner.
lp	The lp command arranges for the named files and associated information (collectively called a request) to be printed. If file names are not specified on the command line, the standard input is assumed. Alternatively, the lp command is used to change the options for a request submitted previously. The print request identified by the request ID is changed according to the print options specified with this command.

Command	Description
lpstat	The lpstat command displays information about the current status of the print service. If no options are given, lpstat displays the status of all print requests made by the user.

The administrator can give users the ability to disable and enable a printer so that, when a printer is malfunctioning, the user can turn the printer off without having to call the administrator. (However, in your printing environment, it might not be reasonable to allow regular users to disable a printer.)

Table 10-16 provides a summary of the print service commands available only to the system or print administrator. To use the administrative commands, you must have root user authority or be a member of either the **printq** or the **lp** group. All of the administrative print service commands listed in Table 10-16 are located in the **/usr/sbin** directory with two exceptions: The **lpsched** program resides in the **/usr/lib/lp** directory, and the **enable** and **disable** commands are found in the **/usr/bin** directory.

Table 10-16 Administrative print service commands

Command	Description
accept reject	accept allows the queuing of print requests for the named destinations. A destination can be either a printer or a class of printers. reject prevents queuing of print requests for the named destinations.
enable disable	The enable command activates the named printers, enabling them to print requests submitted by the lp command. If the printer is remote, the command will only enable the transfer of requests to the remote system. The disable command deactivates the named printers, disabling them from printing requests submitted by lp .
lpadmin	lpadmin configures the lp print service by defining printers and devices. It is used to add and change printers, to remove printers from service, to set or change the system default destination, to define alerts for printer faults, to mount print wheels, and to define printers for remote printing services.
lpfilter	The lpfilter command is used to add, change, delete, and list a filter used with the lp print service. These filters are used to convert the content type of a file to a content type acceptable to a printer.
lpforms	The lpforms command is used to administer the use of preprinted forms, such as company letterhead paper, with the System V print service.

Command	Description
lpmove	lpmove moves requests that were queued by lp between destinations (printers or classes of printers).
lpsched	lpsched allows you to start the System V print service.
lpshut	lpshut shuts down the print service. All printers that are printing at the time lpshut is invoked will stop printing.
lpssystem	The lpssystem command is used to define parameters for the LP print service, with respect to communication (using a high-speed network like TCP/IP) with remote systems.
lpusers	The lpusers command is used to set limits to the queue priority level that can be assigned to jobs submitted by users of the System V print service.

The administrative print service commands listed in Table 10-16 on page 712 are supplemented by three default printer filters used by interface programs, which are located in the /usr/lib/lp/bin directory: lp.cat, lp.set, and lp.tell. The lp.cat program reads the file to be printed on its standard input and writes it to the device to be printed on. Interface programs may call lp.set to set the character pitch, line pitch, page width, page length, and character set on the printer. Also, interface programs can use lp.tell to forward descriptions of printer faults to the print service. lp.tell sends everything that it reads on its standard input to the print service. The print service forwards the message as an alert to the print administrator

Finally, the four BSD compatibility commands (**lpc**, **lpr**, **lpq**, and **lprm**) are available in the /usr/bin directory for users and administrators.

A comprehensive listing of the file modes, ownership, group ownership, and the fully qualified path name for each of the commands mentioned in this section are given in 10.7.4, “User interface specifications” on page 713.

10.7.4 User interface specifications

The user interface specifications for the System V print subsystem are documented in the man pages for the printing and associated commands. Table 10-17 on page 714 provides an overview of the available commands for the System V print subsystem. BSD system compatibility commands are also included in the list and noted accordingly.

In previous AIX releases, some System V and BSD print commands were mapped to AIX print subsystem commands to enhance compatibility and usability of the AIX print services. The executables of these commands were

nothing more than wrappers, which called the AIX print subsystem-specific **enq** command after all command line arguments had been translated to a list of **enq** specific flags. Since AIX 5L offers the possibility to use the System V print subsystem as an alternative to the traditional AIX print subsystem, the relevant commands have to be supplied in two different versions. The traditional AIX print subsystem command wrappers for the System V and BSD print executables are kept in the `/usr/aix/bin` directory, while the native System V print subsystem counterparts are collectively located in the `/usr/sysv/bin` directory. The relevant commands are referenced by symbolic links in the `/usr/bin` directory. The symbolic links always point to the version of the executable related to the type of the active print subsystem. The duplicate commands are marked below with an asterisk (*), but as far as the user interface specification for the System V print subsystem is concerned, only the native BSD compatibility executables in the `/usr/sysv/bin` directory are relevant.

Table 10-17 System V printing: User and administrative commands

accept	cancel *	disable *	enable *
lp *	lp.cat	lp.set	lp.tell
lpadmin	lpc (BSD)	lpfilter	lpforms
lpmove	lpq * (BSD)	lpr * (BSD)	lprm* (BSD)
lpsched	lpshut	lpstat *	lpssystem
lpusers	reject		

For more detailed information about specific commands, refer to 10.7.3, “System V print subsystem management” on page 711.

At the end of this section, a set of comprehensive listings of properties that are associated with the user interface commands and their related directories is provided. For each entity, the file mode, ownership, group ownership, and the fully qualified path name is given.

Properties of System V user interface commands and related directories appear as follows:

File Mode	Owner	Group	Pathname
=====	=====	=====	
drwxrwxr-x	lp	lp	/usr/lib/lp
drwxrwxr-x	lp	lp	/usr/lib/lp/bin
drwxr-xr-x	bin	bin	/usr/sysv
drwxr-xr-x	bin	bin	/usr/sysv/bin
-r-xr-xr-x	bin	bin	/usr/bin/lpc


```

---x--x--x lp lp /usr/lib/lp/bin/lp.cat
---x--x--x lp lp /usr/lib/lp/bin/lp.set
---x--x--x lp lp /usr/lib/lp/bin/lp.tell
---s--x--- root lp /usr/lib/lp/lpsched
---s--x--- root lp /usr/sbin/accept
---s--x--- root lp /usr/sbin/lpadmin
---s--x--- root lp /usr/sbin/lpfilter
---s--x--- root lp /usr/sbin/lpforms
---s--x--- root lp /usr/sbin/lpmove
---s--x--- root lp /usr/sbin/lpshut
---s--x--- root lp /usr/sbin/lpsystem
---s--x--- root lp /usr/sbin/lpusers
---s--x--- root lp /usr/sbin/reject
-r-sr-x--- root system /usr/sbin/switch.prt
-rwx----- root system /usr/sbin/switch.prt.subsystem
---x--x--x lp lp /usr/sysv/bin/cancel
---s--x--- root lp /usr/sysv/bin/disable
---s--x--- root lp /usr/sysv/bin/enable
---x--x--x lp lp /usr/sysv/bin/lp
-r-sr-xr-x lp lp /usr/sysv/bin/lpq
-r-xr-xr-x bin bin /usr/sysv/bin/lpr
-r-xr-xr-x bin bin /usr/sysv/bin/lprm
---x--x--x lp lp /usr/sysv/bin/lpstat

```

Links and files, which are only present when the System V print subsystem is active, appear as follows:

File Mode	Owner	Group	Pathname
lrwxrwxrwx	root	system	/usr/bin/cancel -> /usr/sysv/bin/cancel
lrwxrwxrwx	root	system	/usr/bin/lp -> /usr/sysv/bin/lp
lrwxrwxrwx	root	system	/usr/bin/lpq -> /usr/sysv/bin/lpq
lrwxrwxrwx	root	system	/usr/bin/lpr -> /usr/sysv/bin/lpr
lrwxrwxrwx	root	system	/usr/bin/lprm -> /usr/sysv/bin/lprm
lrwxrwxrwx	root	system	/usr/bin/lpstat -> /usr/sysv/bin/lpstat
lrwxrwxrwx	root	system	/usr/bin/disable -> /usr/sysv/bin/disable
lrwxrwxrwx	root	system	/usr/bin/enable -> /usr/sysv/bin/enable
[Created on the fly when switching to System V print subsystem]			
-rwxrwx---	root	lp	/usr/sysv/_SYS5_print_subsystem (AIX S5 mode)

10.7.5 User interface for AIX and System V print subsystems

In the current release of AIX 5L, the Web-based System Manager provides the graphical user interface that will be used for the most common functions of the System V print subsystem. For more advanced functions, or to use less common features, users and administrators have to rely on the command line interfaces.

The System V print subsystem management tasks to be performed by the Web-based System Manager application include:

- ▶ Adding new printers or classes (parallel, serial, remote, and network)
- ▶ Setting the default printer
- ▶ Removing printers or classes of printers
- ▶ Switching to AIX print subsystem

The status information to be displayed by the Web-based System Manager application includes:

- ▶ Showing the default printer
- ▶ Displaying the requests on the default printer
- ▶ Displaying the printers defined on the system
- ▶ Displaying the stopped printers on the system
- ▶ Showing the printers that currently have problems

Before you can use the Web-based System Manager environment that supports System V printing, you have to switch from the AIX to the System V print subsystem. You can either utilize the **switch.prt -s SystemV** command, as described in 10.7.7, “Switching between AIX and System V print subsystems” on page 723, or use the following sequence of menu selections and operations with the Web-based System Manager tool: Select **Printers -> Overview and Tasks**. Select the **Switch to System V print subsystem task**.

After the task has been completed, the Printer container icon is replaced by the Printers (System V) container icon. The Web-based System Manager environment for System V printing is now accessible through the following sequence of menu selections on the Web-based System Manager console: Select **Printers (System V) -> Directory Disabled Overview and Tasks**.

Figure 10-2 on page 717 shows the Web-based System Manager menu for System V print subsystem management tasks.

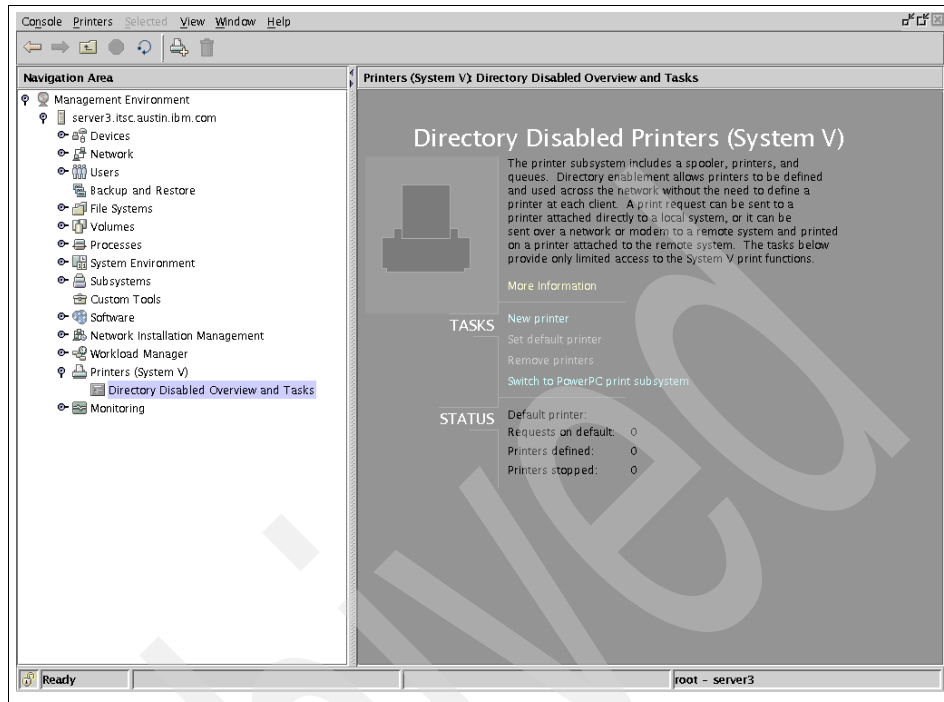


Figure 10-2 Web-based System Manager menu for System V print subsystem

If, for example, you would like to define a local print queue named prop24p for your predefined IBM Proprinter 24 P print device /dev/lp0, select the **New printer** task and follow the instructions of the Add New Printer wizard. Figure 10-3 on page 718 shows the Step 4 of 4: Verify Settings and Add New Printer panel, which is displayed by the Add New Printer wizard before you have the option to complete the task by clicking **Finish**. Note that the device support for the printer must be installed on the system and that the configuration for lp0 must be completed before you engage in the System V print queue configuration. The printer type can be selected from the pull-down menu next to the field What is the printer type? in the Step 3 of 4: Specify Printer Options wizard menu.

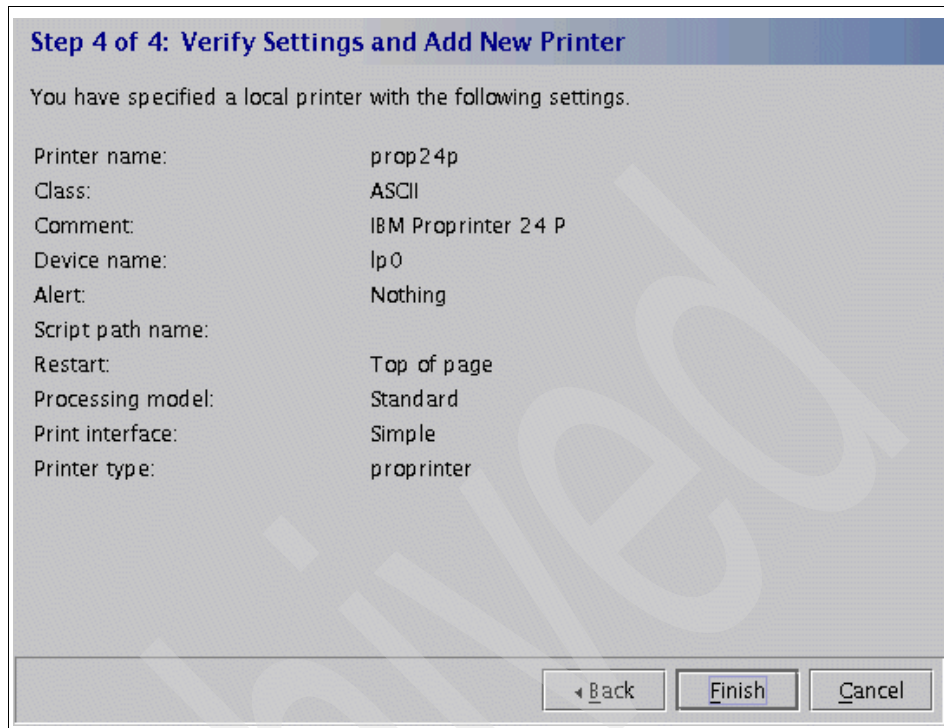


Figure 10-3 Add new printer Web-based System Manager wizard: Step 4 of 4

If the user-defined printer class ASCII does not already exist, it will be created during the final command execution of the Web-based System Manager wizard. Also, the final commands executed by the Web-based System Manager Add New Printer wizard allow the newly configured prop24p printer to accept (**accept** command) queuing requests and enable (**enable** command) the printer to print requests submitted by the **lp** command. The printer will not be defined as the system default print destination. If the user-defined class did not exist before, the wizard creates the class, but will not allow queueing of requests to the class as the print destination.

System administrators who prefer the command line interface to the System V print subsystem can configure the same print queue using the following command sequence:

```
# lpadmin -p prop24p -v /dev/lp0 -D "IBM Proprinter 24P" -c ASCII -I simple -m
standard
-T proprinter
# accept prop24p
# enable prop24p
```

The new printer can optionally be defined as the system default print destination and the `/etc/hosts` file may be submitted as the first test for the System V local print queue:

```
# lpadmin -d prop24p
# lp /etc/hosts
```

The **lpstat -t** command, entered immediately after the submission of the print request, gives comprehensive status information about the System V print subsystem:

```
# lpstat -t
scheduler is running
system default destination: prop24p
members of class ASCII:
    prop24p
device for prop24p: /dev/lp0
ASCII not accepting requests since Mon Sep 25 20:02:47 2000 -
    new destination
prop24p accepting requests since Mon Sep 25 20:03:08 2000
printer prop24p now printing prop24p-9. enabled since Mon Sep 25 20:03:15
2000.available.
prop24p-9                root                1439    Mon Sep 25 20:09:18 2000 on
prop24p
```

It was previously mentioned that the System V print subsystem management tasks are currently not supported through the SMIT tool. However, some changes and additions have been made to account for the introduction of the System V print subsystem feature.

The Print Spooling menu of the SMIT tool was changed to show that most of the menu choices that now exist are only valid for the AIX print subsystem. The AIX print subsystem menu items will still be displayed if the System V print subsystem is active, but they will not work properly, because most of the underlying AIX print subsystem commands and daemons are turned off or disabled in some manner by the `switch.prt.subsystem` script during the switch from the AIX to the System V print subsystem. In addition, one new menu item has been added at the bottom of the Print Spooling menu; it is valid for AIX and System V printing. The name of this item is Change/Show Current Print Subsystem and can be used for either displaying the current running print subsystem or for changing from one to the other. Figure 10-4 on page 720 shows the new Print Spooling menu of SMIT.

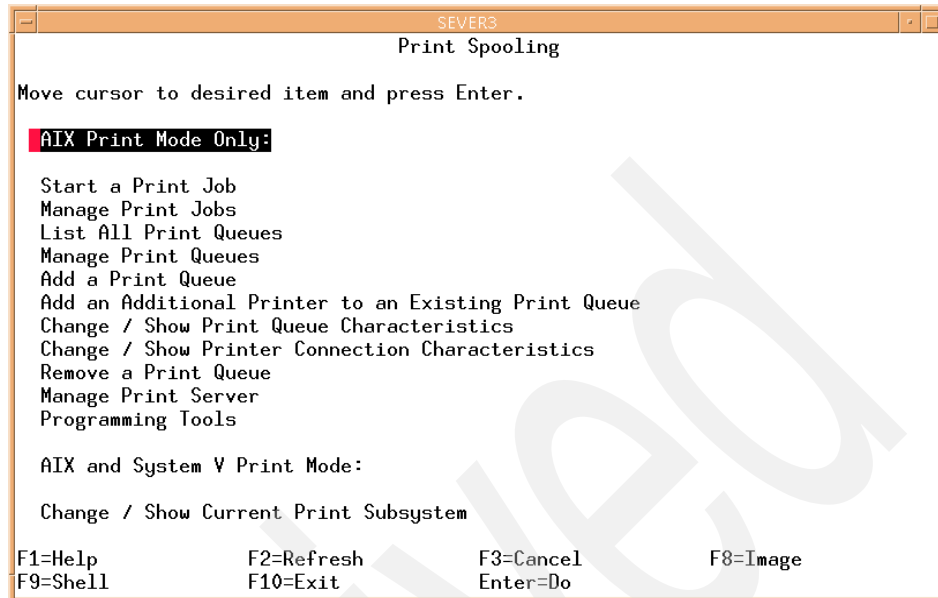


Figure 10-4 Print Spooling menu of SMIT

10.7.6 Terminfo and supported printers

Since System V printing depends heavily on extracting information from the terminfo database to configure and initialize printers, one file has been added that contains the terminfo definitions for all of the printers supported by this subsystem. The name of the file is `svprint.ti`, and it is located in the `/usr/lib/terminfo` directory. The file is compiled and stored in the respective terminfo directories at install time. The printers supported in the terminfo database are listed in Table 10-18.

Table 10-18 Supported printers in the terminfo database

AP1337-e	AP1337-i	AP1339-e	AP1339-i
AP1357-e	AP1357-i	AP1359-e	AP1359-i
AP1371-e	AP1371-i	AP9210-i	AP9210-lj
AP9210-ljplt	AP9215-d		AP9215-e
AP9215-i	AP9215-lj	AP9310-lj	AP9312-lj
AP9316-lj	AP9415-lj	PS	PS-b
PS-br	PS-r	bj-10ex	bj-130e

AP1337-e	AP1337-i	AP1339-e	AP1339-i
bj-200		bj-300	bj-330
lq-870	oki-320	oki-390	oki-ol400
oki-ol800	deskjet	dfx-5000	dfx-8000
epl-7500	fx-1050	fx-850	hplaserjet
	kx-p1124	kx-p1180	kx-p1624
kx-p1695	lq-1170	lq-570	paintjet
proprinter	unknown		

Since many printers can be supported by the same terminfo file, the list of printers that are officially supported by System V printing is much larger. In addition, many printer manufacturers support their own printers for System V and send the support out with the printers. This greatly increases the total number. The list of manufacturers includes, but is not limited to, the IBM Printer Division and Lexmark International. In later releases, more printers will be supported and shipped with AIX. The current list of supported printers is given in Table 10-19.

Table 10-19 Printer support by the System V print subsystem in AIX 5L

Canon Bubble Jet 10ex	Canon Bubble Jet 130e	Canon Bubble Jet 200	Canon Bubble Jet 300
Canon Bubble Jet 330	Epson FX 850	Epson FX 1050	Epson DFX 5000
Epson DFX 8000	Epson LQ 570	Epson LQ 870	Epson LQ 1170
Epson EPL 7500	HP LaserJet (PCL)	HP LaserJet (Postscript)	HP LaserJet II (PCL)
HP LaserJet II (Postscript)	HP LaserJet III (PCL)	HP LaserJet III (Postscript)	HP LaserJet IIIsi (PCL)
HP LaserJet IIIsi (Postscript)	HP LaserJet 4 (PCL)	HP LaserJet 4 (Postscript)	HP LaserJet 4L/4ML (PCL)
HP LaserJet 4L/4ML (Postscript)	HP LaserJet 4P/4MP (PCL)	HP LaserJet 4P/4MP (Postscript)	HP LaserJet 4M/4M (PCL)
HP LaserJet 4M/4M (Postscript)	HP LaserJet 4Si/4Si MX (PCL)	HP LaserJet 4Si/4Si MX (Postscript)	HP LaserJet 4 Plus/4M Plus (PCL)

Canon Bubble Jet 10ex	Canon Bubble Jet 130e	Canon Bubble Jet 200	Canon Bubble Jet 300
HP LaserJet 4 Plus/4M Plus (Postscript)	HP LaserJet 4V/4MV (PCL)	HP LaserJet 4V/4MV (Postscript)	HP LaserJet 5 (PCL)
HP LaserJet 5 (Postscript)	HP LaserJet 5L/5ML (PCL)	HP LaserJet 5L/5ML (Postscript)	HP LaserJet 5P/5MP (PCL)
HP LaserJet 5P/5MP (Postscript)	HP LaserJet 5Si/5Si MX (PCL)	HP LaserJet 5Si/5Si MX (Postscript)	HP LaserJet 5Si Mopier (PCL)
HP LaserJet 5Si Mopier (Postscript)	HP LaserJet 6P (PCL)	HP LaserJet 6P (Postscript)	HP LaserJet 6L (PCL)
HP LaserJet 6L (Postscript)	HP DeskJet 500	HP DeskJet 1200C/1200CPS	HP DeskJet 1600C/1600CM
HP Paint Jet	IBM ProPrinter	Oki 320	Oki 390
Oki OL 400	Oki OL 800	Panasonic KX-P1180	Panasonic KX-P1695
Panasonic KX-P1124	Panasonic KX-P1624	PostScript (Serial)	PostScript (Parallel)
PostScript (Serial w/ page reversal)	PostScript (Parallel w/ page reversal)	Unisys AP1337 - Epson emulation	Unisys AP1337 - IBM emulation
Unisys AP1339 - Epson emulation	Unisys AP1339 - IBM emulation	Unisys AP1357 - Epson emulation	Unisys AP1357 - IBM emulation
Unisys AP1359 - Epson emulation	Unisys AP1359 - IBM emulation	Unisys AP1371 - Epson emulation	Unisys AP1371 - IBM emulation
Unisys AP9205 - IBM emulation	Unisys AP9205 - HP Laserjet emulation	Unisys AP9205 - HP Laserjet Plotter emulation	Unisys AP9210 - IBM emulation
Unisys AP9210 - HP Laserjet emulation	Unisys AP9210 - HP Laserjet Plotter emulation	Unisys AP9215 - Epson emulation	Unisys AP9215 - Diablo emulation
Unisys AP9215 - IBM emulation	Unisys AP9215 - HP Laserjet emulation	Unisys AP9310 - HP Laserjet	emulation

Canon Bubble Jet 10ex	Canon Bubble Jet 130e	Canon Bubble Jet 200	Canon Bubble Jet 300
Unisys AP9312 - HP Laserjet emulation	Unisys AP9316 - HP Laserjet emulation	Unisys AP9415 - HP Laserjet emulation	Other

10.7.7 Switching between AIX and System V print subsystems

The current default print subsystem on AIX is the traditional AIX print subsystem. The System V print subsystem is offered as an alternate method of printing. At install time, the AIX print subsystem will always be set as the active one, and System V will always be set as the inactive one. They cannot both be set to the active state at the same time using the normal procedures. However, there is nothing to prevent an administrator from overriding this manually (at his own risk).

AIX provides a command, accessible through SMIT and the Web-based System Manager, which will allow a system administrator to display the current active print subsystem, and to switch between the active and inactive one. The command is intended to be executed only by the Web-based System Manager or SMIT, but will work from the command line with the proper permissions. That command, located in `/usr/sbin`, is **switch.prt** [**-s print_subsystem**] [**-d**]. The valid values for the `print_subsystem` keyword are AIX and SystemV. Running the command with the `-d` flag will display the current print subsystem; if you do not specify any flag, a brief help message is displayed on the screen:

```
# switch.prt
Usage: [-s AIX | SystemV] [-d]
-s switches to AIX print system or SystemV print system.
-d displays current subsystem.
```

For security reasons, the **switch.prt** command serves as a frontend to the script `/usr/sbin/switch.prt.subsystem`, which actually does the real work.

The basic logic of the script for switching from the traditional AIX to the System V print subsystem is outlined in the following example. The tasks that have to be performed by switching to the reverse direction (from the System V to the traditional AIX print subsystem) are similar, and you are encouraged to examine the code of the original script.

```
# Switch from AIX to System V

# sflag indicates the print subsystem to be switch to
# and the internal variable PRINTSUBSYSTEM refers to
# the type of the currently active print subsystem
```

```

else if sflag = SystemV && PRINTSUBSYSTEM = AIX
then if (active print jobs)
then echo "All print jobs must be terminated
before you can switch to $PRINTSUBSYSTEM"
exit 1
else
Stop qdaemon
Stop writesrv
Stop lpd

Change the action field of the inittab entries for
qdaemon, writesrv, lpd, and piobe to prevent the unwanted
start of this subsystems at system boot.

# The following disables the smit menus as much as
# possible
mv /usr/lib/lpd/pio/etc/*.attach files to *.attach.AIX

# Change the lock files from AIX to System V
rm /usr/aix/_AIX_print_subsystem
touch /usr/sysv/_SYS5_print_subsystem

#force System V links over the existing AIX links for the
#duplicate commands between them

ln -sf /usr/bin/cancel -> /usr/sysv/bin/cancel
ln -sf /usr/bin/enable -> /usr/sysv/bin/enable
ln -sf /usr/bin/disable -> /usr/sysv/bin/disable
ln -sf /usr/bin/lp -> /usr/sysv/bin/lp
ln -sf /usr/bin/lpstat -> /usr/sysv/bin/lpstat
ln -sf /usr/bin/lpq -> /usr/sysv/bin/lpq
ln -sf /usr/bin/lpr -> /usr/sysv/bin/lpr
ln -sf /usr/bin/lprm -> /usr/sysv/bin/lprm

#remove symbolic links from the tcbck database
tcbck -d /usr/bin/cancel
tcbck -d /usr/bin/enable
tcbck -d /usr/bin/disable
tcbck -d /usr/bin/lp
tcbck -d /usr/bin/lpstat
tcbck -d /usr/bin/lpq
tcbck -d /usr/bin/lpr
tcbck -d /usr/bin/lprm

#add the new symbolic links to the tcbck database
tcbck -a /usr/bin/cancel symlinks=/usr/sysv/bin/cancel
tcbck -a /usr/bin/enable symlinks=/usr/sysv/bin/enable
tcbck -a /usr/bin/disable symlinks=/usr/sysv/bin/disable
tcbck -a /usr/bin/lp symlinks=/usr/sysv/bin/lp

```

```

tcback -a /usr/bin/lpstat symlinks=/usr/sysv/bin/lpstat
tcback -a /usr/bin/lpq symlinks=/usr/sysv/bin/lpq
tcback -a /usr/bin/lpr symlinks=/usr/sysv/bin/lpr
tcback -a /usr/bin/lprm symlinks=/usr/sysv/bin/lprm

#start lpsched
/usr/lib/lp/lpsched
echo System V Print Subsystem Started

#Update the inittab to start the System V Print Subsystem at system
boot

exit 0

```

A closer examination of the `switch.prt.subsystem` script reveals that the `/var/spool/lpd/qdir` is probed for files with file names beginning with the letter `n` or `r`, which indicate the existence of pending print jobs. If the search yields a positive result, the script is terminated with an appropriate error message. Consequently, the method provided to switch from one print subsystem to the other does not migrate any pending print jobs.

If no pending print jobs could be identified, the system resource controller command `stopsrc` is used to stop the `qdaemon`, `writesrv`, and `lpd` daemons, which control the AIX print subsystem. After that, the Action field for the related inittab entries is changed by the `chitab` command from wait to off and the respective inittab entry for the `pio` print subsystem backend process is treated in the same fashion.

For the time being, there are no SMIT menus provided to assist users and system administrators with performing System V print subsystem related tasks. Therefore, the AIX print subsystem SMIT menus are not replaced by System V-specific entities, but merely hidden by appending the AIX suffix to the menu definition files in the `/usr/lib/lpd/pio/etc` directory.

Because the operating system determines (by the name of the relevant lock file) the type of the active print subsystem, the script replaces the lock file `/usr/aix/_AIX_print_subsystem` (of the traditional AIX print subsystem) with the lock file `/usr/sysv/_SYS5_print_subsystem` (of the System V print subsystem).

In AIX 5L, the System V and BSD print commands are still in the `/usr/bin` directory, but are now either linked to the traditional AIX print command wrappers in the `/usr/aix/bin` directory or to the appropriate executables in `/usr/sysv/bin` (if the System V print subsystem is active). Consequently, `switch.prt.subsystem` forces the System V links to take precedence over the AIX links when the system administrator switches from the AIX to the System V print subsystem.

If the Trusted Computing Base (TCB) feature is installed on the system, additional measures have to be taken in order to preserve the integrity of the `/etc/security/sysck.cfg` TCB file definition database. The **`tcbck -d`** command is used to remove the current symbolic links from the configuration during a switch, and the **`tcbck -a`** command adds the new symbolic link, including the proper user and group ownership attributes, to the file definition database. If the **`tcbck`** command audits the security state of the system by checking the installation of the files defined in `/etc/security/sysck.cfg`, no mismatch between the file attributes in the trusted computing base and the actual system configuration will be reported.

Finally, if the **`lpsched`** daemon is started, and if an entry for **`lpsched`** exists in `inittab`, then the related action state is changed from off to wait; otherwise, a new entry will be added after the cron entry.

10.7.8 Enable debugging for qdaemon

qdaemon has been enhanced in AIX 5L Version 5.1 so that debugging can be turned on by a system administrator. Debug information useful to diagnosing failures will be recorded in a file that can be examined by support or service personnel.

To enable debugging, qdaemon must be restarted by specifying the `-D` flag to **`startsrc`**, as in the following example.

```
# stopsrc -s qdaemon
# startsrc -s qdaemon -a "-D /tmp/qdaemon.log"
```

Note: Enabling the qdaemon debugging has the potential to adversely affect the performance of the AIX printing subsystem. The high level of disk I/O can slow down printing in a moderate to high volume printing installation. Turning on debugging will output information to a file on disk. It will be the responsibility of the system administrator to ensure that there is enough disk space, as this file could potentially get very large, very quickly in a high-volume printing environment.

10.7.9 Enable debugging for JetDirect backend

The JetDirect backend (`piohpnf`) has been modified to enhance the level of information that is reported to qdaemon when a failure occurs.

Traditionally, when the JetDirect backend (`piohpnf`) abends, the user only gets a very cursory message from qdaemon indicating that the backend has had a fatal exit. To get further information, the system administrator has to turn on logging capability for `piohpnf`. This generates a file on disk that contains more

specific information. However, in moderate to large size installations, it is often impractical to enable logging for piohnpnf (as it logs everything, not just failures). Hence, the need arises for more detailed messages to be sent back using the console or e-mail in case of failure.

To enable the debugging option on piohnpnf, modify the piojetd script so piohnpnf is invoked with the -D flag. You can find the piojetd file in the /usr/lib/lpd/pio/etc directory. Open the file and go to the line (34 on the test system):

```
/usr/lib/lpd/piobe "$@" | /usr/lib/lpd/pio/etc/piohnpnf -x $hostname -p $port
```

Add the -D flag for enabling the debug option:

```
/usr/lib/lpd/piobe "$@" | /usr/lib/lpd/pio/etc/piohnpnf -D -x $hostname -p $port
```

Note: The debugging should not be carelessly turned on. Some customers do not want to have messages e-mailed to them or shown on the console.

10.8 SMIT System V print (5.2.0)

SMIT functionality has now been added for System V printing, a feature itself that was introduced in AIX 5L Version 5.1.

Version 5.2 introduces SMIT screens for all aspects of System V Release 4 print management.

10.8.1 Installation

To install System V printing in Version 5.2, these filesets are installed as part of the New and Complete Overwrite Install. In the case of a Migration Install it is necessary to install the filesets post-migration. The filesets required include the following:

bos.msg.en_US.svprint	5.2.0.0	COMMITTED	System V Print Subsystem Msgs
bos.svprint.dir_enabled	5.2.0.0	COMMITTED	System V Directory-enabled
bos.svprint.fonts	5.2.0.0	COMMITTED	System V Print Fonts
bos.svprint.hpnp	5.2.0.0	COMMITTED	System V Hewlett-Packard
bos.svprint.ps	5.2.0.0	COMMITTED	System V Print Postscript
bos.svprint.rte	5.2.0.0	COMMITTED	System V Print Subsystem
bos.svprint.trans	5.2.0.0	COMMITTED	System V Print Translation
bos.svprint.ps	5.2.0.0	COMMITTED	System V Print Postscript
bos.terminfo.svprint.data	5.2.0.0	COMMITTED	System V Printer Terminal

10.8.2 SMIT integration

The SMIT integration builds on the enhancements brought in with AIX 5L Version 5.1, where SMIT provides the functionality to toggle between the AIX and System V Release 4 print subsystems.

On the command line, ensure that the System V print subsystem is active on the machine (the actual switching may take a minute to complete):

```
# switch.prt -d
#printsubsystem
AIX
# switch.prt -s SystemV
SystemV Print Subsystem Started
# switch.prt -d
#printsubsystem
SystemV
```

This can also be achieved through the SMIT menus, from the initial screen. The process and other relevant screen shots showing the SMIT frontend to the System V print functionality introduced in AIX 5L Version 5.1 are shown in Figure 10-5.

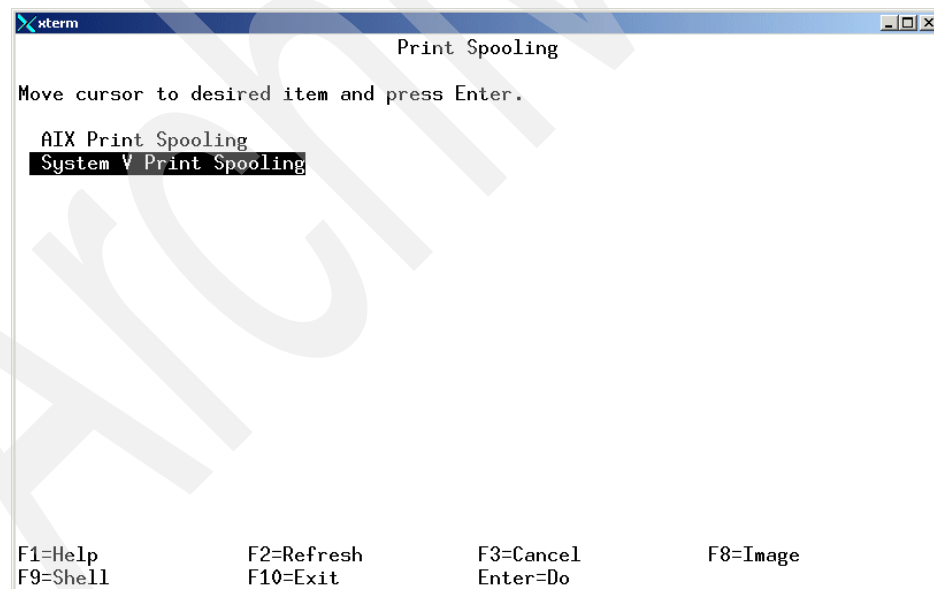


Figure 10-5 Selecting System V print spooling menus

From this menu, there are a number of print handling options to choose. The bottom option allows the user to toggle between print subsystems, as shown in Figure 10-6.

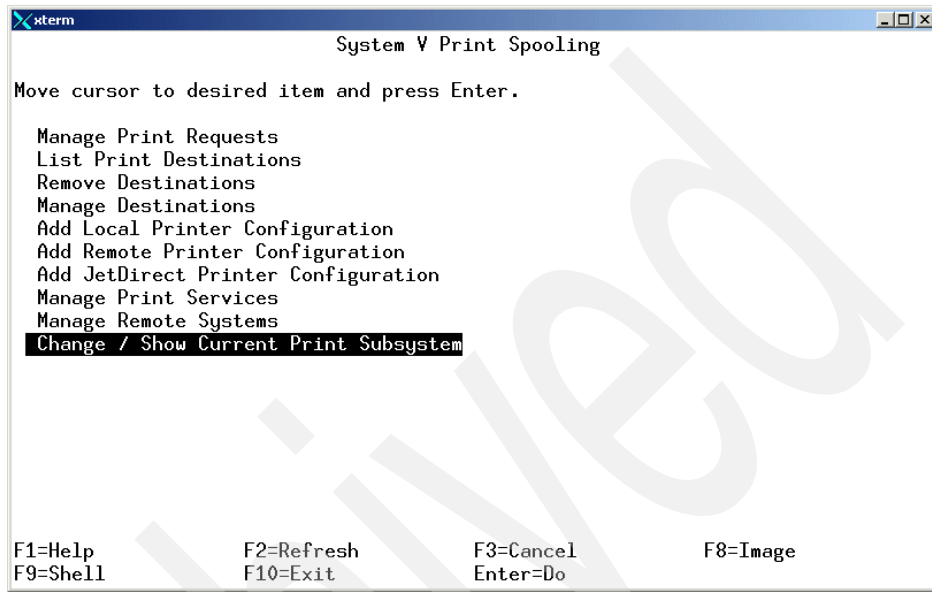


Figure 10-6 System V print spooling options

The Manage print requests screen gives a number of useful options for print management using the System V print subsystem, as shown in Figure 10-7 on page 730.

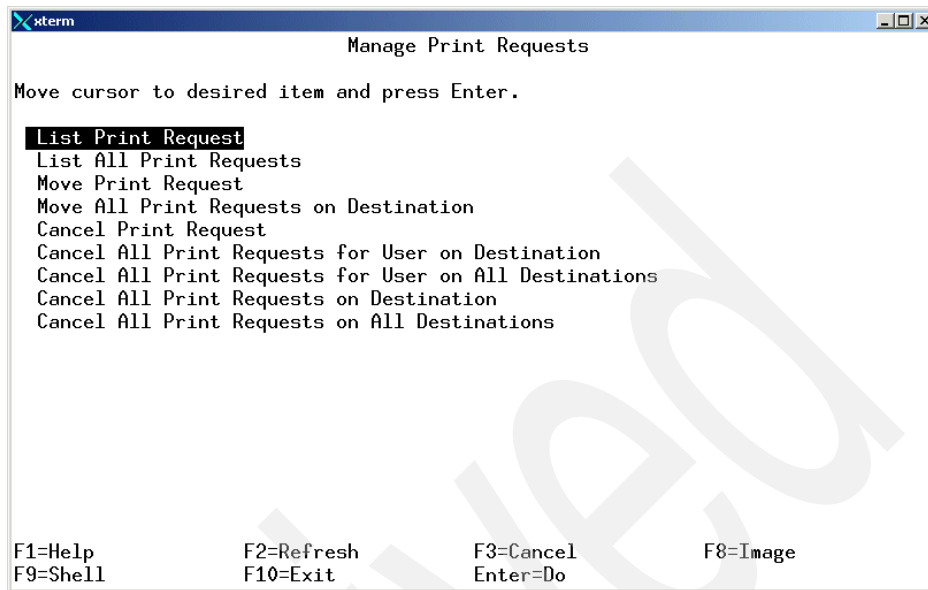


Figure 10-7 System V print request management screen

The destination management screen also has a number of System V options to providing a SMIT frontend to the System V commands introduced in AIX 5L Version 5.1. These are shown in Figure 10-8 on page 731.

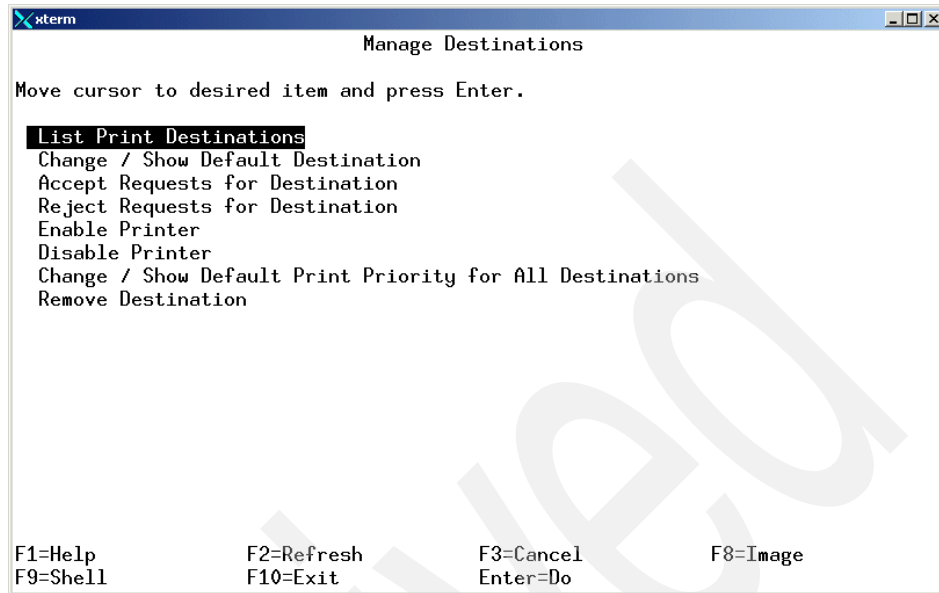


Figure 10-8 System V destination management screen

Linux affinity

AIX 5L incorporates a strong Linux affinity through the AIX Toolbox for Linux Applications and the integration of the Linux development environment into AIX libraries. This makes it possible to compile and run Linux applications on AIX, providing the ideal background to support this fast growing and competitive market. Countless developers around the world are focused on developing applications for Linux systems, and now you can easily port these applications and run them directly on AIX, taking advantage of all the features and benefits this operating system offers.

A dedicated publication on this topic is *Running Linux Applications on AIX*, SG24-6033.

11.1 The **geninstall** command (5.1.0)

AIX 5L Version 5.1 introduces a new install command named **geninstall**. The **geninstall** command allows the installation of software packaged in different formats other than **installp**. These include InstallShield Multi-Platform (ISMP), the Red Hat Package Manager (RPM) installer, and Uniform Device Interface (UDI).

The **geninstall** command accepts all current **installp** flags and passes them on to **installp**. This allows programs (such as NIM) to continue to always send in **installp** flags to **geninstall**, but only the flags that make sense are used.

The syntax of the **geninstall** command is:

```
Usage geninstall: Install software from device.  
geninstall -d Media  
[ -I installpFlags ] [ -R ResponseFile ] [ -E ResponseFile ] [ -N ] [ -Y ] [ -Z  
] -f file | install_list... | all  
Usage geninstall: Uninstall software.  
geninstall -u -f file | uninstall_list...  
Usage geninstall: List installable software on device  
geninstall -L -d media
```

Table 11-1 displays the flags that can be used with the **geninstall** command.

Table 11-1 The *geninstall* command flags

Flag	Description
-d <i>device media or directory</i>	Specifies the device or directory containing the images to install.
-E	Not supported in AIX 5L Version 5.1.
-f <i>file</i>	Specifies the file containing a list of entries to install. Each entry in the file must be preceded by a format type prefix. Currently, geninstall accepts the following prefixes: I:bos.net (Installp) J:Websphere (ISMP) R:mtools (RPM) U:devices.pci.8602912 (UDI) This information is given in the geninstall -L output.

Flag	Description
-l <i>installpflags</i>	Specifies the installp flags to use when calling the installp command. The flags that are used during an install operation for installp are the a, b, c, D, e, E, F, g, I, J, M, N, O, p, Q, q, S, t, v, V, w, and X flags. The installp flags that should not be used during install are the C, i, r, S, z, A, and I flags. The installp command should be called directly to perform these functions. The -u, -d, -L, and -f flags should be given outside the -l flag.
-L	Lists the contents of the media. The output format is the same as the installp -Lc format, with additional fields at the end for ISMP, RPM, and UDI formatted products.
-N	Not supported in AIX 5L Version 5.1.
-R <i>ResponseFile</i>	Takes the full path name of the ResponseFile to send to the ISMP installer program.
-u	Performs an uninstall of the specified software. For ISMP products, the uninstaller listed in the vendor database is called, prefixed by a "J:".
-Y	Agrees to required software license agreements for software to be installed. This flag is also accepted as an installp flag with the -l option.
-Z	Tells geninstall to invoke the installation in silent mode.

Note: If you are using **geninstall** for more than one package format, you have to split the packages into separate directories. Make sure that each directory contains only one package format. For example, make a subdirectory called rpm for all Linux RPM packages and an installp directory for all AIX LPPs.

11.1.1 Install RPM packages

Instead of using the **rpm** installer, you can use **geninstall** to install Linux RPM packages.

The following output shows a directory with RPM packages only:

```
# ls /tmp/geninstall/RPM
bash2-2.04-3.aix4.3.ppc.rpm      zlib-devel-1.1.3-7.aix4.3.ppc.rpm
info-4.0-5.aix4.3.ppc.rpm       zoo-2.10-4.aix4.3.ppc.rpm
```

```
zip-2.3-1.aix4.3.ppc.rpm      zsh-3.0.8-1.aix4.3.ppc.rpm
zlib-1.1.3-7.aix4.3.ppc.rpm  zsh-3.0.8-2.aix4.3.ppc.rpm
```

To install all RPM packages in the /tmp/geninstall/RPM directory at once, use the following command:

```
# geninstall -d /tmp/geninstall/RPM *
bash2-2.04-3
info-4.0-5
zip-2.3-1
zlib-devel-1.1.3-7
zoo-2.10-4
```

Use the **rpm** command to check if all packages have been installed successfully:

```
# rpm -qa
zlib-1.1.3-7
mtools-3.9.7-3
cdrecord-1.9-1
mkisofs-1.9-1
AIX-rpm-5.1.0.0-2
bash2-2.04-3
info-4.0-5
zip-2.3-1
zlib-devel-1.1.3-7
zoo-2.10-4
```

11.1.2 Install AIX LPPs

Using **geninstall** is also a way to install AIX LPP packages. The **geninstall** calls the **installp** command to install additional AIX LPP packages.

The directory in the following example output shows AIX LPP packages only:

```
# ls -l /tmp/geninstall/installp
total 5784
-rw-r--r--  1 root    system      2070528 Mar 29 18:10
IMNSearch.bld.2.3.1.0.I
-rw-r--r--  1 root    system      882688 Mar 29 18:11 bos.INed.5.1.0.0.I
```

To install the bos.INed LPP package, use the following **geninstall** syntax:

```
# geninstall -d /tmp/geninstall/installp bos.INed
+-----+
Pre-installation Verification...
+-----+
Verifying selections...done
Verifying requisites...done
Results...
```

SUCCESES

Filesets listed in this section passed pre-installation verification and will be installed.

Selected Filesets

bos.INed 5.1.0.0 # INed Editor

<< End of Success Section >>

FILESET STATISTICS

- 1 Selected to be installed, of which:
 - 1 Passed pre-installation verification

1 Total to be installed

+-----+
Installing Software...
+-----+

installp: APPLYING software for:
bos.INed 5.1.0.0

. << Copyright notice for bos.INed >>
Licensed Materials - Property of IBM

5765E6100
(C) Copyright International Business Machines Corp. 1985, 2001.
(C) Copyright INTERACTIVE Systems Corporation 1983, 1988.

All rights reserved.
US Government Users Restricted Rights - Use, duplication or disclosure
restricted by GSA ADP Schedule Contract with IBM Corp.

. << End of copyright notice for bos.INed >>.

Finished processing all filesets. (Total time: 7 secs).

+-----+
Summaries:
+-----+

Installation Summary

Name	Level	Part	Event	Result
bos.INed	5.1.0.0	USR	APPLY	SUCCESS

Note: Do not specify the Version, Release, Modification, or Fix level of the fileset; otherwise, the installation will fail with an error similar to this:

Pre-installation Failure/Warning Summary		
Name	Level	Pre-installation Failure/Warning

bos.INed.5.1.0.0		Not found on the installation
media		

11.2 The gencopy command (5.1.0)

AIX 5L Version 5.1 introduces a new install command named **gencopy**. The **gencopy** command allows a user to copy different package formats. It determines what images must be copied and calls the appropriate command.

In AIX 5L Version 5.1, the **gencopy** and **bffcreate** commands create subdirectories in the default or user-specified target directory that correspond to the package format type.

The syntax of the **gencopy** command is:

Usage gencopy: Copy software from media.
gencopy -d *media* [-t *target_location*] [-D] [-X]
[-b "*bffcreate_flags*"] -f *file* | *copy_list...* | all

-t Defaults to /usr/sys/inst.images

Usage gencopy: List software products and packages on media.
gencopy -L -d *media*

The commonly used flags are listed in Table 11-2.

Table 11-2 The gencopy command flags

Flag	Description
-b <i>bffcreate_flags</i>	The following flags are valid: l, q, v, w, and S.
-d <i>device media</i> or <i>directory</i>	The device or directory where the install images exist. Media can be a device (/dev/cd0, /dev/rmt0) or directory.

Flag	Description
<code>-f file</code>	File containing a list of entries to copy to the target location. Each entry in the file must be preceded by a format type prefix. Currently, gencopy accepts the following prefixes: I:bos.net -> Installp (BFF) J:WebSphere -> ISMP R:mtools -> RPM U:devices.pci.86802912 -> UDI This information is given in the gencopy -L output.
<code>-D</code>	Calls bffcreate with the <code>-D</code> option, instructing it to remove images after the copy. This flag is not valid with non- installp images.
<code>-L</code>	Lists the contents of the media. The output format is the same as the bffcreate -Lc format, with additional fields at the end for ISMP, RPM, and UDI formatted products.
<code>-t target_location</code>	Specifies the directory where the installation image files are to be stored. If the <code>-t</code> flag is not specified, the files are saved in the <code>/usr/sys/inst.images</code> directory.
<code>-X</code>	Automatically extends the file system if space is needed.

11.2.1 Examples

The following are examples of these commands:

- ▶ To copy all of the images from CD media (`/dev/cd0`) to an LPP_SOURCE (`/export/lpp_source/510_lppsource`):
- ▶ `gencopy -d/dev/cd0 -t /export/lpp_source/510_lppsource all`
- ▶ To copy several images from CD media to the default directory:
- ▶ `gencopy -d/dev/cd0 I:bos.games R:mtools J:WebSphere`
- ▶ To copy packages in a file:

```
gencopy -d/dev/cd0 -f/tmp/mixed_packages.txt
```

Where `/tmp/mixed_packages.txt` contains the following packages:

```
I:bos.games
R:mtools
J:WebSphere
```

- ▶ To list the contents of the CD media:

```
geninstall -Ld /dev/cd0
```

This listing is colon separated, and contains the following information:

```
file_name:package_name:fileset:V.R.M.F:type:platform:Description
bos.sysmgt:bos.sysmgt:bos.sysmgt.nim.client:4.3.4.0:I:R:Network Install
Manager - Client Tools
bos.sysmgt:bos.sysmgt:bos.sysmgt.smit:4.3.4.0:I:R:System Management
Interface Tool (SMIT)
```

When we copied the install images to the target directory, in this case the /usr/sys/inst.images directory, the **gencopy** and **bffcreate** command created two new subdirectories for the images:

```
# pwd
/usr/sys/inst.images
# ls
RPMS      installp
## find . -print
.
./installp
./installp/ppc
./installp/ppc/bos.perf.5.1.0.0.I
./installp/ppc/bos.msg.en_US.5.1.0.0.I
./installp/ppc/.toc
./installp/ppc/bos.docsearch.5.1.0.0.I
./installp/ppc/bos.mp.5.1.0.0.I
./RPMS
./RPMS/ppc
./RPMS/ppc/mtools-3.9.3-7.aix43.ppc.rpm
./RPMS/ppc/cdrecord-4.7.1-2.aix43.ppc.rpm
```

11.3 Install Wizard for applications (5.1.0)

A new installation method can be used by the **geninstall** command instead of the **installp** command.

The **geninstall** command allows the installation of software packaged in different formats other than **installp**. These include InstallShield Multi-Platform (ISMP), Red Hat Package Manager (RPM) installer, and Uniform Device Interface (UDI) formats. The **install_wizard** is contained in the **sysmgt.websm.apps** package.

There are three separate paths to the wizard: Standalone, NIM Client, and NIM master.

It is very similar to the Install Base Operating System wizard in that respect.

- | | |
|-------------------|--|
| Standalone | The user is installing from a locally attached device or directory. |
| NIM Client | The user is a configured NIM Client and is initiating the install from the client side. |
| NIM master | The user is a configured NIM master and is installing one or more NIM machines or a NIM machine group. |

The wizard does not support installing software on multiple NIM machine groups or NIM SPOT resources.

11.3.1 Invoking the Wizard

The Install Wizard can be invoked in many different ways:

- ▶ Using the Web-based System Manager Software Overview plug-in Install Software.
- ▶ From the command line using `/usr/sbin/install_wizard -d device_name/lpp_source`.
- ▶ From the Installed Software plug-in wizard Method. The current Install Additional Software dialog is invoked by the Advanced method menu item.
- ▶ From the NIM and NIM Overview plug-in's Install Software menu.
- ▶ From the NIM Overview plug-in Install Software on a Network Installation Client Tasks item.
- ▶ From the NIM Machines and Groups plug-in wizard Method menu item.

11.3.2 Example of the Install Wizard

The wizard is invoked from the command line using `/usr/sbin/install_wizard -d device_name/lpp_source`, as shown in Figure 11-1 on page 742.

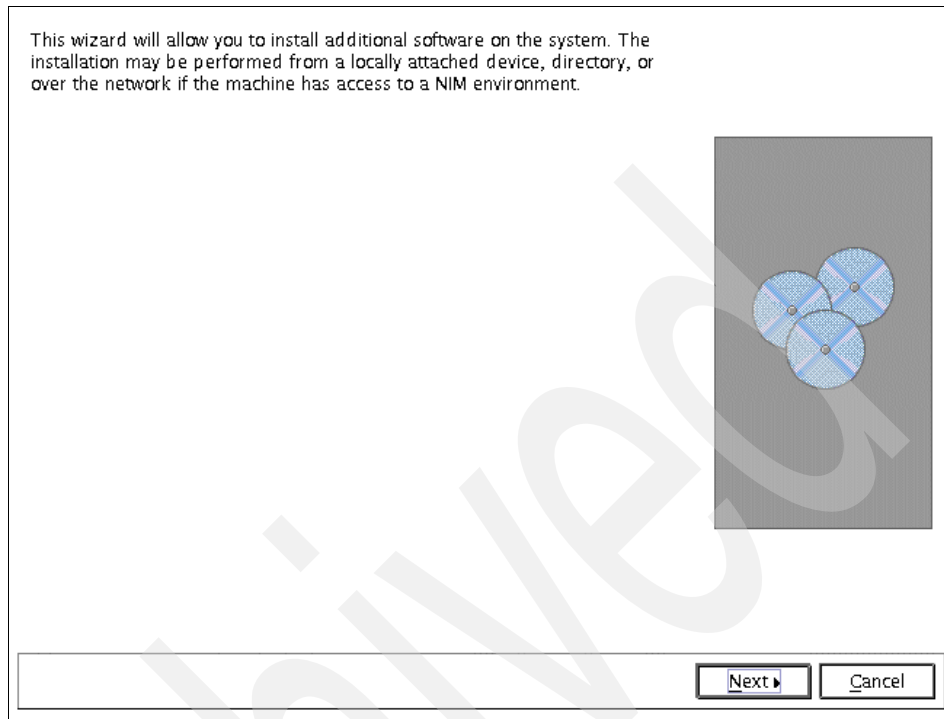


Figure 11-1 Installation Wizard invoked by the command line

Once the wizard is invoked, you can select the source of the installation image, which can be a device or a directory containing the image, as shown in Figure 11-2 on page 743.

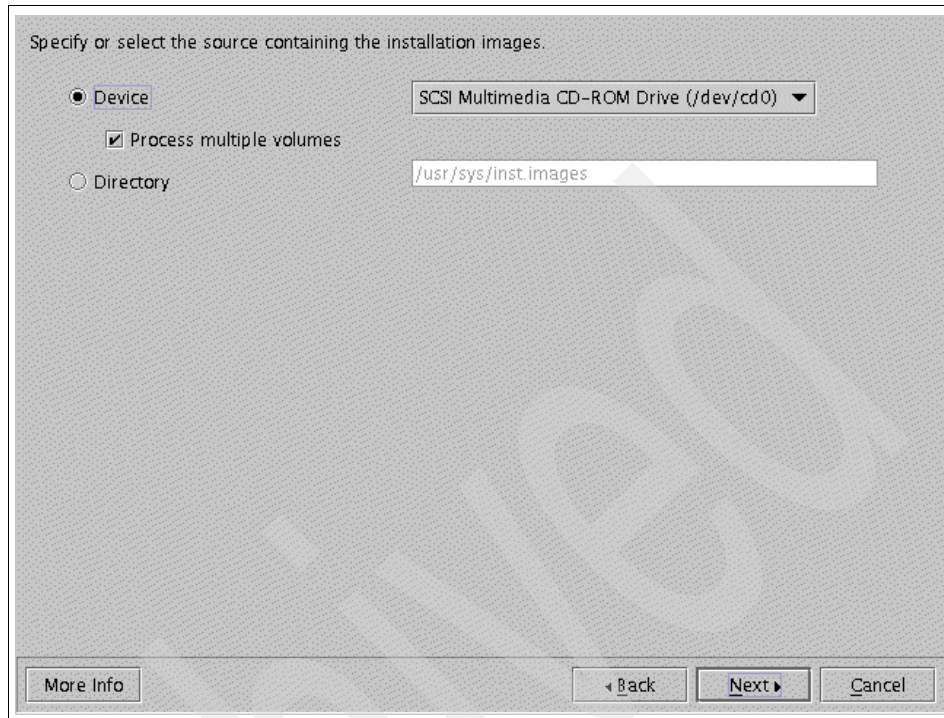


Figure 11-2 Installation Wizard for selecting source of installation

The wizard will guide you through the installation. Figure 11-3 on page 744 shows two ways of installation: You can select a full installation or select the software from a product to install.

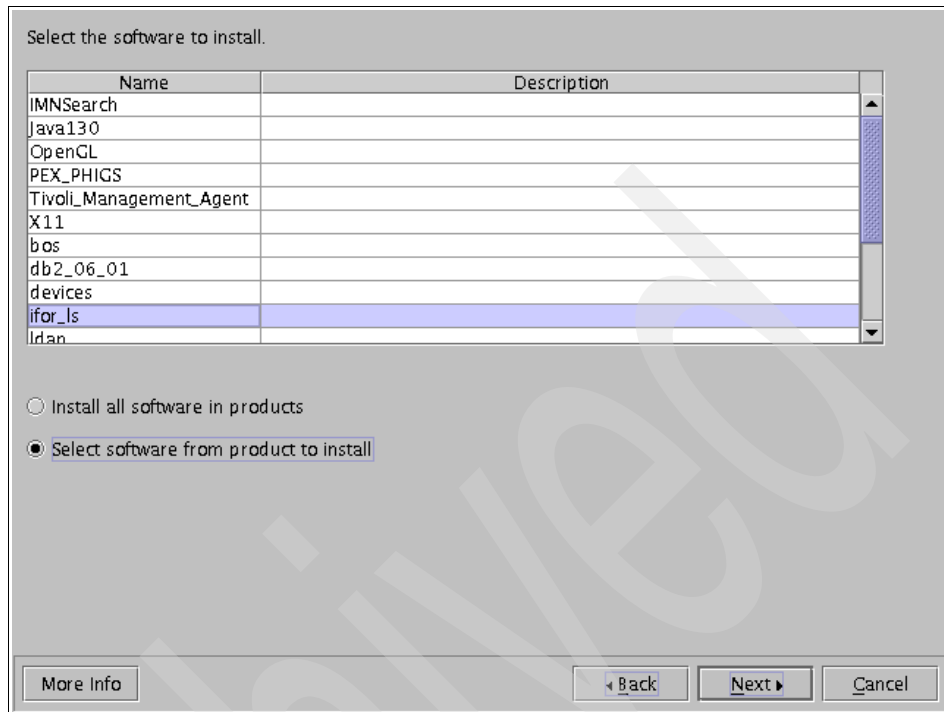


Figure 11-3 Installation Wizard for selecting the software to install

You can select the product you want to install; the next screen will list the software you can install (see Figure 11-4 on page 745).

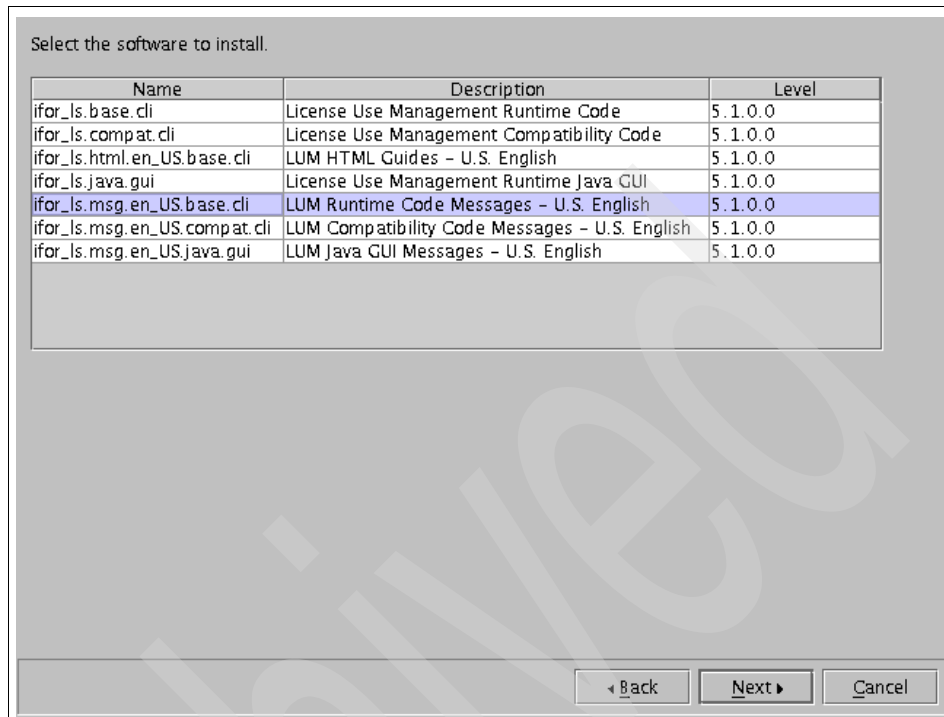


Figure 11-4 Installation Wizard for selecting software from product

Once you have your software selected, you can verify your settings or start the installation, as shown in Figure 11-5 on page 746.

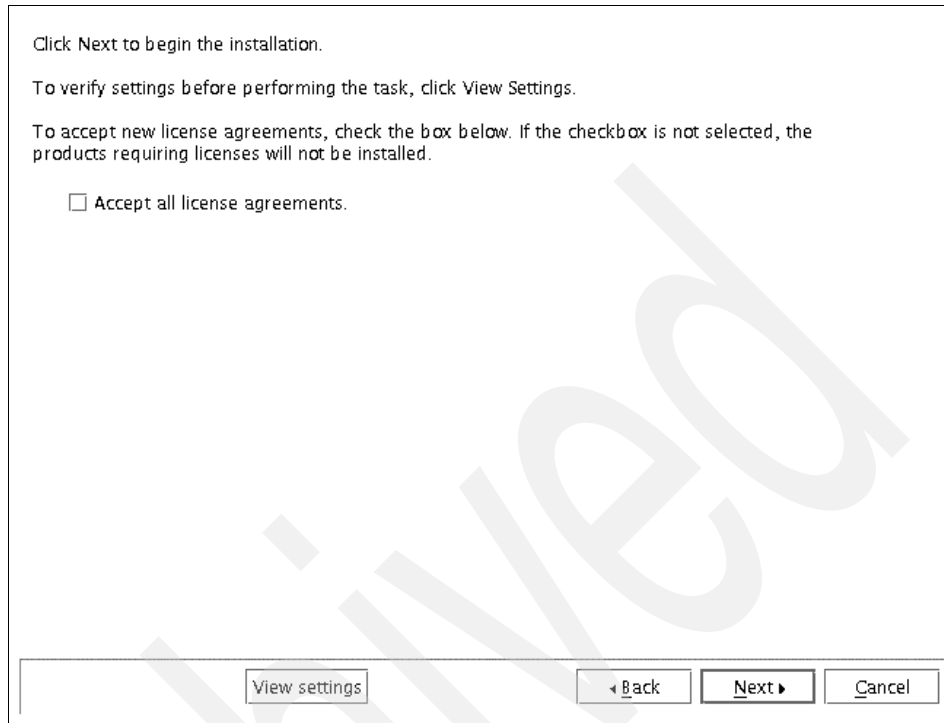


Figure 11-5 Installation Wizard to begin installation

The installation can be followed or stopped on the display (see Figure 11-6 on page 747).

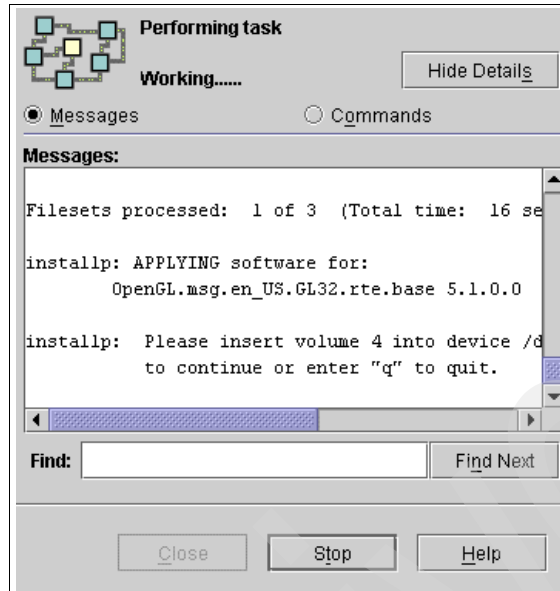


Figure 11-6 Installation Wizard task panel

11.4 The devinstall command enhancement (5.1.0)

The new **devinstall** command can be used to install different packages for devices. It is called by **cfgmgr** or BOS install.

Originally, **devinstall** called **installp** to install software required by devices; now it calls **geninstall** to add support for UDI-formatted device drivers.

The **geninstall** command is a wrapper program for **installp**, Install Shield Multi-Platform (ISMP), Red Hat Package Manager (RPM), and **udisetaup**. It accepts all current **installp** flags and passes them on to **installp**.

11.4.1 The previous structure of devinstall

The previous version of **devinstall** consists of three parts.

In the first part, the **devinstall** command does the initialization work, including parsing the input from the command line and setting up certain variables, such as package file (pkgfile) from the **-f** flag, and the device name used to install the required packages (instdev) from the **-d** flag. It then builds a package list based on the packages in the package file. Packages are listed only once in the package list.

Each entry in the list has the following structure:

```
struct pkgname {
    char    name[FNAME_SIZE];
    int     status;
    struct pkgname *next;
}
```

The fields used in this code are explained as follows:

name	The package name, for example, devices.pci.xxxxxxxx.
status	One of the following:
OLD_NAME	The package has already been processed.
DEL_NAME	The package failed to install during the current installation.
NEW_NAME	The first time this package will be processed. This is the initial value.
next	The pointer pointing to next entry

In the second part, **devinstall** calls **installp** by using **odm_run_method**:

```
odm_run_method(INSTALLP_CMD, argsbuf, NULL, NULL);
```

Where the parameters are defined as follows:

- ▶ **INSTALLP_CMD** is defined as **/usr/sbin/installp**.
- ▶ **argsbuf** is defined as **-axqNXQg -e /var/adm/ras/devinstall.log -d instdev -f pkgfile**.

In the third part, **devinstall** checks the summary file **/var/adm/sw/installp.summary**, which is generated by the **installp** command, for the results of each package install attempt and, based on this information, creates or updates the following two files:

- ▶ **/var/adm/dev_pkg.fail**
Lists the packages that failed to install (if any).
- ▶ **/usr/sys/inst.data/sys_bundles/Hdwr-Diag.def**
Lists all packages that have installed successfully.

11.4.2 Structure of the new version of devinstall

The first part stays the same as the previous version except the entry structure in the package list.

The new structure is:

```
struct pkgname {
    char packagename[256]; like devices.pci.xxxxxxxx
```

```

int inst_status;The package is installed or uninstalled, initialized as
uninstalled.
int pkg_status;it could be 0 or old_name. 0 means it is a new package name and
old_name:it is a existing package in dev_pkg.fail file or bundle file. It is
initialized as 0 (new package).
struct pkgname *next;The pointer pointing to next package.
};

```

The main changes are in the second and third parts. After setting up variables, it calls **geninstall** instead of **installp**:

```
odm_run_method(GENINSTALL_CMD, argsbuf, NULL, NULL);
```

Where the parameters are defined as follows:

- ▶ GENINSTALL_CMD is defined as /usr/sbin/geninstall.
- ▶ argsbuf is defined as -l "axqNXQge /var/adm/ras/devinstall.log" -d instdev -f pkgfile.

geninstall determines how to install the required packages by using the options following the -l flag.

In the third part, **devinstall** checks the summary file (/var/adm/sw/geninstall.summary) generated by **geninstall** for the results of each package install attempt and, based on this information, creates or updates the following two files:

- ▶ /var/adm/dev_pkg.fail
Lists the packages that failed to install (if any).
- ▶ /usr/sys/inst.data/sys_bundles/devices.bnd
Lists all packages that have installed successfully.

The geninstall.summary file has the same format as installp.summary, but it includes the results of **udisetaup**.

After installation is done, **devinstall** goes through the geninstall.summary file to find which packages are installed. If a package is installed successfully or is already installed, it will be marked in the package list as installed (inst_status = INSTALLED). Otherwise, it will stay in uninstalled state (inst_status = UNINSTALLED). Then **devinstall** will update the /usr/sys/inst.data/sys_bundles/devices.bnd file or /var/adm/dev_pkg.fail file. Before any packages are written to a file, **devinstall** checks if they are already in the file (usr/sys/inst.data/sys_bundle/devices.bnd or /var/adm/dev_pkg.fail). If a package is already in the file, it will be marked in the package list as old_name (pkg_status = OLD_NAME) and will not be written to the file. Only the packages that are installed successfully and are not in the bundle file will be written to

/usr/sys/inst.data/sys_bundles/devices.bnd. Similarly, only the packages that failed to install and are not in the /var/adm/dev_pkg.fail will be written to it.

11.5 BOS installation allows different desktops (5.1.0)

During a BOS installation, you can choose between different desktops:

CDE	The Common Desktop Environment
GNOME	The GNOME desktop
KDE	The K Desktop Environment
NONE	No desktop

CDE is the standard desktop for AIX. KDE and GNOME are part of the AIX Toolbox for Linux Applications.

If you want use KDE or GNOME as your primary desktop, the installation of the AIX Toolbox for Linux Applications is also required. For more information about KDE and GNOME, see 11.6.4, “Graphical framework” on page 758.

Note: The KDE and GNOME desktops and their utilities are not translated into the same languages as AIX.

The desktop option is only available if you use an LFT console when installing the system.

11.5.1 Using a TTY console

If you are using a TTY console when installing the system, you will not get the option to choose a different desktop (Figure 11-7 on page 751). Note that the 64-bit kernel option is only available if the hardware supports the 64-bit kernel.

```
Advanced Options

Either type 0 and press Enter to install with current settings, or type the
number of the setting you want to change and press Enter.

1  Installation Package Set..... Default
2  Enable Trusted Computing Base..... no
3  Enable 64-bit Kernel and JFS2..... no

>>> 0  Install with the settings listed above.

88  Help ?
```

Figure 11-7 BOS installation while using a TTY console

11.5.2 Using a LFT console

Using a LFT console (Figure 11-8) to install the system, you will get the option to choose between different desktops. The 64-bit kernel option is only available if the hardware is 64-bit enabled.

```
Advanced Options

Either type 0 and press Enter to install with current settings, or type the
number of the setting you want to change and press Enter.

1  Desktop..... GNOME
2  Enable Trusted Computing Base..... no
3  Enable 64-bit Kernel and JFS2..... no

>>> 0  Install with the settings listed above.

88  Help ?
```

Figure 11-8 BOS installation menu while using a LFT console

Since the AIX Toolbox for Linux Applications is not a part of the AIX BOS CDs, you need the Toolbox for Linux Applications CD. Therefore, a warning message is displayed on the console (Figure 11-9 on page 752).

WARNING: The desktop you have selected (GNOME or KDE) is not part of the operating system and is installed from the "Toolbox for Linux Applications" media. You will be prompted for the media later in the install process. If you do not have the "Toolbox for Linux Applications" media available, return to the Advanced options menu to select another desktop. To continue, type 1 and press Enter. You will have another opportunity to change your desktop selection after you insert the "Toolbox for Linux Applications" media.

1 Continue with Install
2 Return to the Advanced Options screen

Figure 11-9 Warning messages during desktop install

11.5.3 Using NIM for BOS installation

For a NIM Install, all additional filesets must be available in lpp_source. If it is a LFT CONSOLE, the DESKTOP field in the control_flow stanza of the bosinst.data file can be set to the desired desktop (CDE, NONE, GNOME, or KDE). If the CONSOLE is not a LFT, the DESKTOP field is ignored.

The following is an extract of the bosinst.data file, showing the Desktop variable set to GNOME:

```
control_flow:
  CONSOLE = /dev/lft0
  INSTALL_METHOD = overwrite
  PROMPT = no
  EXISTING_SYSTEM_OVERWRITE = yes
  INSTALL_X_IF_ADAPTER = yes
  RUN_STARTUP = yes
  RM_INST_ROOTS = no
  ERROR_EXIT =
  CUSTOMIZATION_FILE =
  TCB = no
  INSTALL_TYPE =
  BUNDLES =
  SWITCH_TO_PRODUCT_TAPE =
  RECOVER_DEVICES = yes
  BOSINST_DEBUG = no
  ACCEPT_LICENSES = no
  INSTALL_64BIT_KERNEL = no
  INSTALL_CONFIGURATION = Default
```

11.6 AIX Toolbox for Linux Applications

The AIX Toolbox for Linux Applications provides the tools to port Linux applications to AIX, as well as the tools to work on those applications. Additionally, the toolbox contains several applications that have already been recompiled for use with AIX

The AIX Toolbox for Linux Applications contains a wide variety of software, including, but not limited to:

Application Development	gcc, g++, gdb, rpm, cvs, automake, autoconf, libtool, bison, flex, and gettext
Desktop Environments	GNOME and KDE
GNU base utilities	gawk, m4, indent, sed, tar, diffutils, fileutils, findutils, textutils, grep, and sh-utils
Programming Languages	guile, python, tcl/tk, and rep-gtk
System Utilities	emacs, vim, bzip2, gzip, git, elm, ncftp, rsync, wget, lsof, less, samba, zip, unzip, and zoo
Graphics Applications	ImageMagick, transfig, xfig, xpdf, ghostscript, gv, and mpage
Libraries	ncurses, readline, libtiff, libpng, libjpeg, slang, fnlib, db, gtk+, and qt
System Shells	bash2, tcsh, and zsh
Window Managers	enlightenment and sawfish

For a complete and updated list of all the tools contained in the Toolbox and to check the availability of software for a specific platform, see:

<http://www.ibm.com/servers/aix/products/aixos/linux/index.html>

A version of the AIX Toolbox for Linux Applications is shipped with all AIX media. It can be ordered individually using the form numbers provided in Table 11-3.

Table 11-3 Form number for AIX Toolbox for Linux Applications CD

Form number	Product
LCD4-1077-00	AIX Toolbox for Linux Applications

11.6.1 Basic Linux commands

The basic Linux commands, such as **tar**, **gzip**, **gunzip**, **bzip2**, and so forth, are installed in the `/opt/freeware/bin` directory. To use those commands, you have to specify either the whole path or set the `PATH` variable.

Using a Linux command instead of an AIX command may be practical. For example, the Linux **tar** command offers options to directly compress and uncompress a tar file:

```
# /opt/freeware/bin/tar --help
GNU `tar' saves many files together into a single tape or disk archive, and
can restore individual files from the archive.
... skipping some output ...
Usage: /opt/freeware/bin/tar [OPTION]... [FILE]...
Archive format selection:
  -V, --label=NAME           create archive with volume name NAME
                             at list/extract time, a globbing PATTERN
                             PATTERN
  -o, --old-archive, --portability write a V7 format archive
  --posix                    write a POSIX conformant archive
  -z, --gzip, --ungzip       filter the archive through gzip
  -Z, --compress, --uncompress filter the archive through compress
  --use-compress-program=PROG filter through PROG (must accept -d)
```

Note: Because all AIX system management utilities are expecting to call the native AIX commands to manage the system, the use of Linux commands might cause unexpected results when the `PATH` variable is used to run Linux commands before AIX commands.

11.6.2 System management tools

Since AIX offers SMIT and Web-based System Manager to administer and manage the system, there is no need for Linux system configuration tools. However, there are a few management tools available that you can experiment with.

Note: In general, always use the native AIX tools, such as Web-based System Manager, to administer or manage an AIX system.

User administration

The **kuser** command, as shown in Figure 11-10 on page 755, allows easy user administration. The **kuser** command is provided by the KDE package.

Restriction: Any modification of the AIX flat files by a non-AIX program using non-AIX APIs has the potential to seriously corrupt the AIX files. It is recommended that the use of this command be restricted to non-production test systems that have a full system backup only.

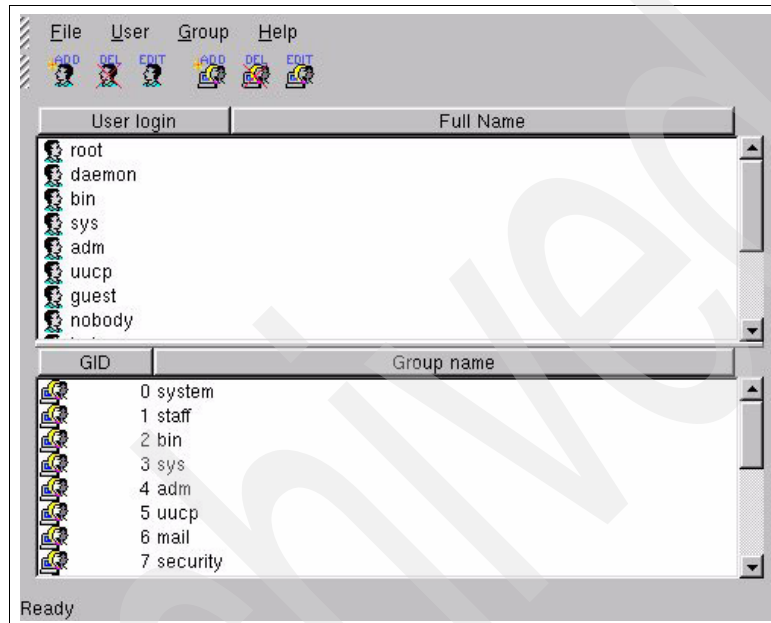


Figure 11-10 User administration provided by KDE

System V init editor

The **ksysv** command, provided by the KDE package, is an available tool to manage the System V initialization structure (/etc/rc.d). Figure 11-11 on page 756 shows the **ksysv** utility.

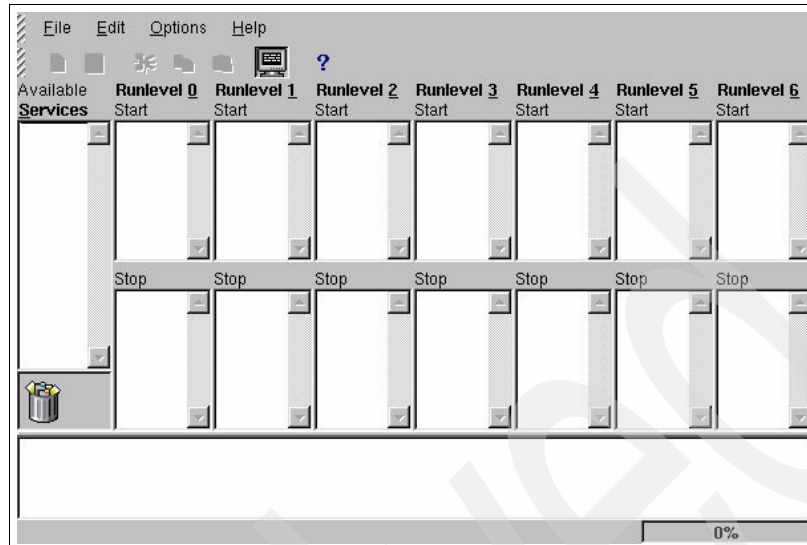


Figure 11-11 System V init editor provided by KDE

11.6.3 Red Hat Package Manager

The Red Hat Package Manager (RPM) is part of the AIX Toolbox for Linux Applications. It facilitates installation and maintenance of Linux applications.

The **rpm** command is available as an AIX LPP fileset on the AIX 5L Version 5.1 base CD. If you want use **rpm** to install additional Linux packages, make sure the corresponding fileset (**rpm.rte**) is installed, as shown in the following example:

```
# lspp -l rpm.rte
Fileset                                Level  State      Description
-----
Path: /usr/lib/objrepos
rpm.rte                                3.0.5.17  COMMITTED  RPM Package Manager
```

The RPM database, which holds information about the installed RPM packages, is located in **/var/opt/freeware/lib/rpm**, with a symbolic link created in **/var/lib**, so you can also access it at **/var/lib/rpm**.

rpm command

The **rpm** command is used to install, upgrade, query, and delete Linux RPM packages. The tool is also used to maintain the RPM package database. The following example provides a look at all the possible uses:

```
# rpm
usage: rpm [--help]
```

```

rpm {--version}
rpm {--initdb} [--dbpath <dir>]
rpm {--install -i} [-v] [--hash -h] [--percent] [--force] [--test]
    [--replacepkgs] [--replacefiles] [--root <dir>]
    [--excludedocs] [--includedocs] [--noscripts]
    [--rcfile <file>] [--ignorearch] [--dbpath <dir>]
    [--prefix <dir>] [--ignoreeos] [--nodeps] [--allfiles]
    [--ftp proxy <host>] [--ftpport <port>] [--justdb]
    [--httpproxy <host>] [--httpport <port>]
    [--noorder] [--relocate oldpath=newpath]
    [--badreloc] [--notriggers] [--excludepath <path>]
    [--ignoresize] file1.rpm ... fileN.rpm
rpm {--upgrade -U} [-v] [--hash -h] [--percent] [--force] [--test]
    [--oldpackage] [--root <dir>] [--noscripts]
    [--excludedocs] [--includedocs] [--rcfile <file>]
    [--ignorearch] [--dbpath <dir>] [--prefix <dir>]
    [--ftp proxy <host>] [--ftpport <port>]
    [--httpproxy <host>] [--httpport <port>]
    [--ignoreeos] [--nodeps] [--allfiles] [--justdb]
    [--noorder] [--relocate oldpath=newpath]
    [--badreloc] [--excludepath <path>] [--ignoresize]
    file1.rpm ... fileN.rpm
rpm {--query -q} [-afpg] [-i] [-l] [-s] [-d] [-c] [-v] [-R]
    [--scripts] [--root <dir>] [--rcfile <file>]
    [--whatprovides] [--whatrequires] [--requires]
    [--triggeredby] [--ftpport] [--ftp proxy <host>]
    [--httpproxy <host>] [--httpport <port>]
    [--ftpport <port>] [--provides] [--triggers] [--dump]
    [--changelog] [--dbpath <dir>] [targets]
rpm {--verify -V} -y [-afpg] [--root <dir>] [--rcfile <file>]
    [--dbpath <dir>] [--nodeps] [--nofiles] [--noscripts]
    [--nomd5] [targets]
rpm {--setperms} [-afpg] [target]
rpm {--setugids} [-afpg] [target]
rpm {--freshen -F} file1.rpm ... fileN.rpm
rpm {--erase -e} [--root <dir>] [--noscripts] [--rcfile <file>]
    [--dbpath <dir>] [--nodeps] [--allmatches]
    [--justdb] [--notriggers] rpackage1 ... packageN
rpm {-b|t}[plciba] [-v] [--short-circuit] [--clean] [--rcfile <file>]
    [--sign] [--nobuild] [--timecheck <s>] ]
    [--target=platform1[,platform2...]]
    [--rmsource] [--rmspec] specfile
rpm {--rmsource} [--rcfile <file>] [-v] specfile
rpm {--rebuild} [--rcfile <file>] [-v] source1.rpm ... sourceN.rpm
rpm {--recompile} [--rcfile <file>] [-v] source1.rpm ... sourceN.rpm
rpm {--resign} [--rcfile <file>] package1 package2 ... packageN
rpm {--addsign} [--rcfile <file>] package1 package2 ... packageN
rpm {--checksig -K} [--nogpg] [--nogpg] [--nomd5] [--rcfile <file>]
    package1 ... packageN

```

```
rpm [--rebuilddb] [--rcfile <file>] [--dbpath <dir>]
rpm [--querytags]
```

Install RPM packages

The following example shows the installation of the Linux xscreensaver **rpm** package:

```
# rpm -i xscreensaver-3.25-2.aix4.3.ppc.rpm
```

Trying to install an RPM package that is already installed on the system will fail, and a message similar to the following will appear:

```
# rpm -iv
package AfterStep-1.8.0-1 is already installed
```

Note: Before installing any RPM packages, make sure there is enough space left in the /opt file system. Since Linux applications are installed in the /opt/freeware directory and **rpm** does not automatically extend the file system, it has to be done manually.

Query the RPM database

To get an overview of all or just a particular RPM package installed on the system, use the **-q** flag with the **rpm** command, as shown in the following example:

```
# rpm -qa
bash2-doc-2.04-3
mttools-3.9.7-3
cpio-2.4.2-17
qt-2.2.4-1
AIX-rpm-5.1.0.0-2
a2ps-4.12-1
automake-1.4-3
bash2-2.04-3
bison-1.28-3
bzip2-1.0.1-3
cdda2wav-1.9-3
cdrecord-devel-1.9-3
info-4.0-6
less-358-2
libhttp-1.0.6-2
```

11.6.4 Graphical framework

The graphical desktops available in the AIX Toolbox for Linux Applications are composed of different elements that provide a specific graphical development framework. This framework depends upon the desktop you decide to use.

Figure 11-12 shows the interaction of the graphical libraries and the different desktops.

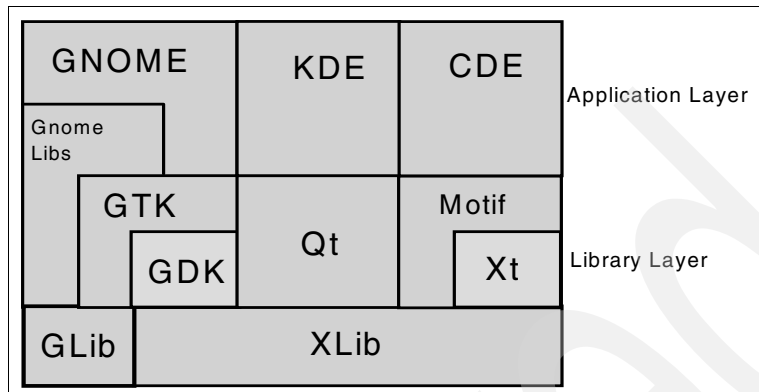


Figure 11-12 AIX Toolbox for Linux Applications graphical framework

GNOME desktop

GNOME, a very popular desktop environment (Figure 11-13 on page 760) on Linux platforms, is also part of the AIX Toolbox for Linux Applications. Once installed, you can use GNOME as your primary desktop. GNOME can be installed at BOS installation time (see 11.5, “BOS installation allows different desktops (5.1.0)” on page 750) or at any later time.

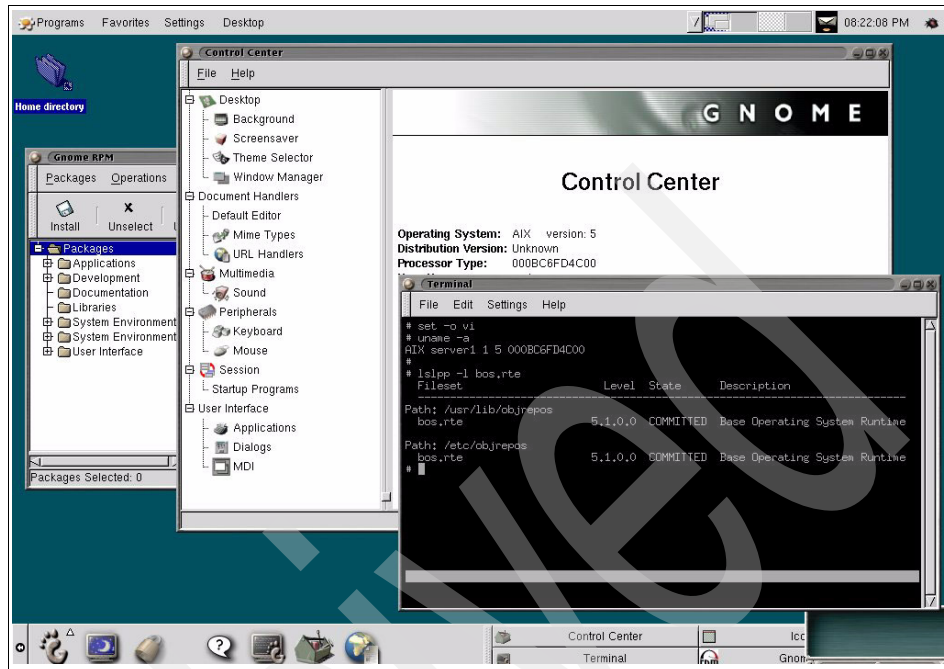


Figure 11-13 Gnome Desktop running on AIX 5L Version 5.1

KDE desktop

KDE is another well-known desktop for Linux. KDE2 has been recompiled on AIX 5L Version 5.1 and is part of the AIX Toolbox for Linux Applications. At the time of this writing, KDE 1.1.2 is available (shown in Figure 11-14 on page 761). Similar to the GNOME desktop, KDE can be installed at any time or while installing the base AIX operating system. For further details, see 11.5, “BOS installation allows different desktops (5.1.0)” on page 750.

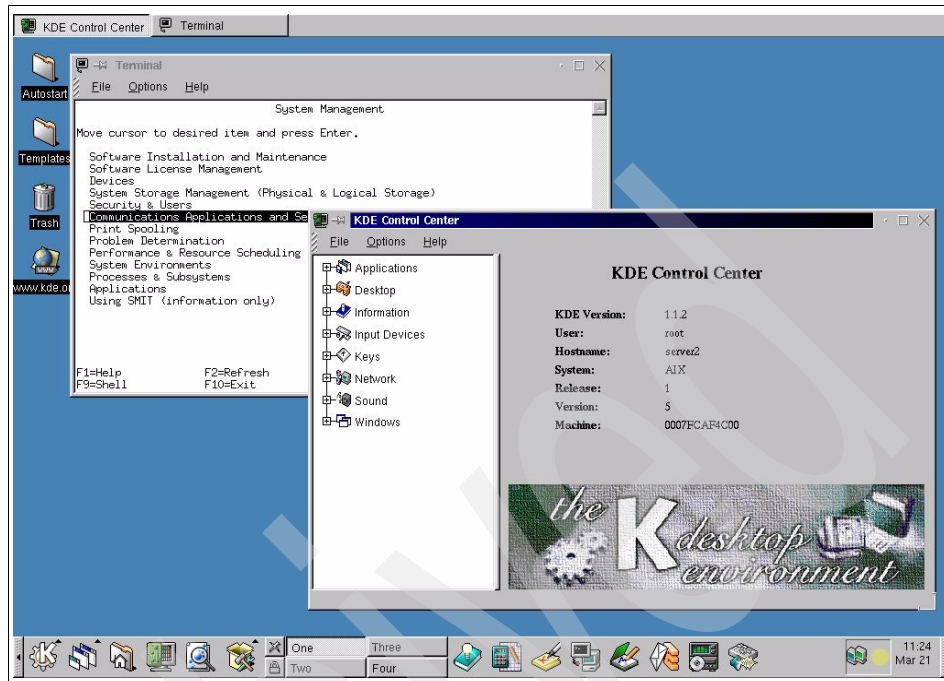


Figure 11-14 KDE 1.1.2 desktop running on AIX 5L Version 5.1

GTK+ user interface builder (Glade)

Glade (Figure 11-15 on page 762) is a free user interface builder for GTK+ and GNOME. It is released under the GNU General Public License (GPL).

Glade can produce C source code itself. C++, Ada95, Python, and Perl support are also available, using external tools that process the XML interface description files output by Glade.

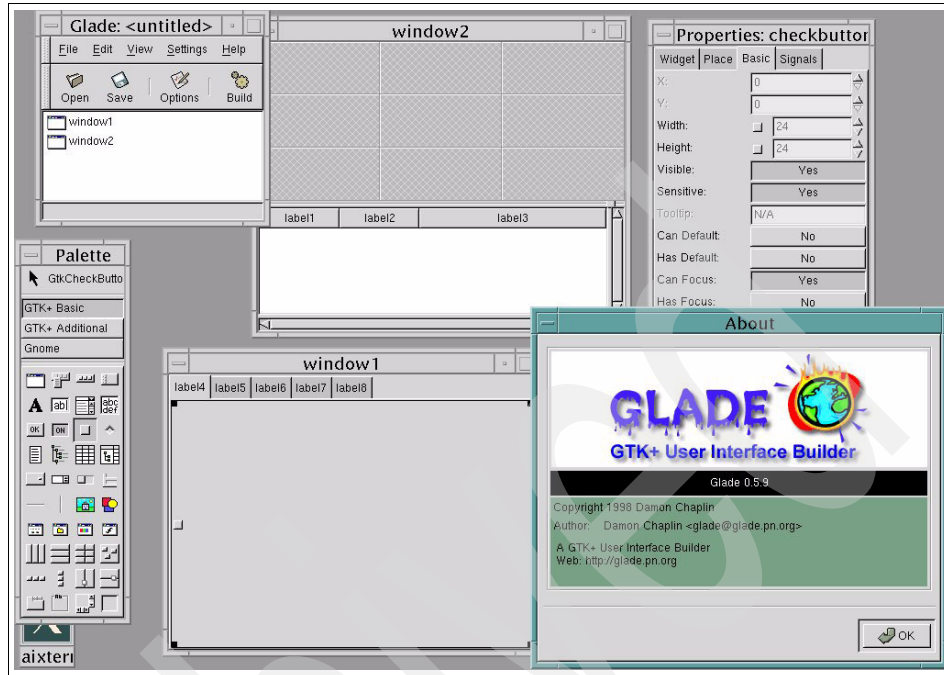


Figure 11-15 Glade running on AIX 5L Version 5.1

11.7 AIX source affinity for Linux applications (5.1.0)

Since AIX and Linux do not use the same APIs and system calls, several modifications have been made to provide more source level compatibility in AIX 5L Version 5.1.

The following example shows the changes for the reboot system call. Both the Linux and AIX reboot API are available in AIX 5L Version 5.1. The reboot API is just one example of a dual-semantic function. The list of dual-semantic functions is still increasing.

The Linux prototype is similar to the following:

```
#include <unistd.h>
#include <sys/reboot.h>
int reboot (int flag);
#ifdef _H_REBOOT
#define _H_REBOOT
```


The AIX Version 4.3.3 prototype is similar to the following:

```
#define RB_SOFTIPL      0
#define RB_HALT         1
#define RB_POWIPL      2
#define RB_HARDIPL      3
#define RB_HALT_POWERED 4
#define RB_UPDATE_FLASH 5

typedef struct {
    caddr_t uf_strt_ptr;          /* Pointer to start of image */
    ulong   uf_img_len;          /* Length of image */
    void     *uf_xmem;           /* Pointer to cross mem desc */
} update_flash_t;

#endif /* _H_REBOOT */
```

In AIX 5L Version 5.1 the prototype has been enhanced to be compatible with Linux. The new prototype is similar to the following:

```
#ifndef _H_REBOOT
#define _H_REBOOT

#define RB_SOFTIPL      0
#define RB_HALT         1
#define RB_POWIPL      2
#define RB_HARDIPL      3
#define RB_HALT_POWERED 4
#define RB_UPDATE_FLASH 5

typedef struct {
    caddr_t uf_strt_ptr;          /* Pointer to start of image */
    ulong   uf_img_len;          /* Length of image */
    void     *uf_xmem;           /* Pointer to cross mem desc */
} update_flash_t;

#ifdef _LINUX_SOURCE_COMPAT
extern int __linux_reboot(int);
#define reboot(a) __linux_reboot((a))

#define LINUX_REBOOT_CMD_RESTART RB_SOFTIPL
#define LINUX_REBOOT_CMD_HALT RB_HALT_POWERED
#define LINUX_REBOOT_CMD_POWER_OFF RB_HALT
#define LINUX_REBOOT_CMD_RESTART2 RB_POWIPL
#define LINUX_REBOOT_CMD_CAD_ON 90 /* AIX does not offer CAD reboot */
#define LINUX_REBOOT_CMD_CAD_OFF 91
#endif

#endif /* _H_REBOOT */
```

11.7.1 Compiling open source software

This short section describes how to compile and install open source software without using the RPM utility. Basically, by using the utilities provided by the toolbox, this can be done as usual for those packages. As an example, use the fvwm2 window manager. Download the sources, starting at <http://fvwm.org> or <http://xwinman.org>, and unpack under the directory /opt/freeware/src:

```
# cd /opt/freeware/src
# tar -xzf fvwm-2.2.4.tar.gz
```

Change to the newly created fvwm-2.2.4 directory and follow the instructions in the INSTALL and README files. During the final *make* install, the software will be installed in subdirectories (like bin, lib, man, and so on) of the directory given as the --prefix option to configure. Remember to set the environment appropriately to be able to execute the binaries and find the executables later on:

```
# ./configure --prefix=/opt/freeware
[...skipping some output...]
```

Configuration:

FVWM Version:	2.2.4
Build extra modules?	no
Have ReadLine support?	no
Have RPlay support?	no
Have XPM support?	no: Xpm library or header not found!

```
# make 2>&1 | tee make.log
[...skipping some output...]
# make install 2>&1 | tee makeinstall.log
[...skipping some output...]
```

The previously described installation procedure is generic for applications developed according to the GNU coding standards, as described at http://www.gnu.org/prep/standards_toc.html. In general, developing applications according to these standards will ensure easy portability to various UNIX-based platforms, including Linux.

However, if a Linux application does not compile on AIX, then you should add -D_LINUX_SOURCE_COMPAT to the compiler flags and try again. In general, the flag is not needed, but a few functions require it. It is always safe to use the flag when compiling Linux applications.



Hardware support

This chapter discusses enhancements to AIX 5L in the areas of device support, hardware-related behaviors, and commands that will assist you in determining the hardware configuration.

12.1 AIX 5L 64-bit kernel overview

AIX 5L provides a new, scalable, 64-bit kernel that:

- ▶ Provides simplified data and I/O device sharing for multiple applications on the same system
- ▶ Provides more scalable kernel extensions and device drivers that make full use of the kernel's system resources and capabilities
- ▶ Allows for future hardware development that will provide even larger single image systems ideal for server consolidation or workload scalability

The following sections provide a general understanding of the new 64-bit kernel.

12.1.1 Why a 64-bit kernel is needed

There are a combination of factors that drive the requirement for a 64-bit kernel. The primary factor is the trend in system design towards massive amounts of system resources, terabytes of memory, hundreds of processors, and thousands of I/O slots. A resulting factor is that customers see these massive single systems as an opportunity for server consolidation, migrating all of the workloads that used to be across a number of individual servers onto a single massive server. The kernel is responsible for managing the physical resources as well as the process workload, all of which are growing exponentially.

Similar to the need for a database program to move from a 32-bit environment to a 64-bit environment in order to take advantage of the vast address space to efficiently manage more data in memory, the kernel also needs to move from the constrained 32-bit environment to a 64-bit environment to efficiently support and manage the ever-expanding resources and workload. Some specific examples include:

- ▶ Increasing the size of Virtual Memory Manager (VMM) data structures in order to support the larger memory configurations
- ▶ The increased number and size of data structures in the global kernel address space required to support the possibility of thousands of physical and logical devices and their device drivers
- ▶ The ability to scale kernel data types to more easily support greater than 32-bit addressability in areas of 64-bit user address space, large files, number of inodes, device numbering, thread IDs, and so on

12.1.2 64-bit kernel considerations

There are some points for consideration for this new 64-bit kernel.

- ▶ Both 32-bit and 64-bit kernels are available.
- ▶ Only 64-bit CHRP-compliant PowerPC machines are supported for the 64-bit kernel.
- ▶ Only 64-bit kernel extensions are supported; this means that no existing 32-bit kernel extensions can be reused for the 64-bit kernel.
- ▶ Kernel extensions and device drivers must be compiled in 64-bit mode to be loaded into the 64-bit kernel.
- ▶ The 32-bit and 64-bit application environments are available on all 64-bit platforms.

12.1.3 External page table scaling for 64-bit kernel (5.2.0)

Prior to AIX 5L Version 5.2, the number of processes an application creates using `fork()` is limited to the remaining space in a PTA segment. This was also a restriction to the segments ability to create more virtual pages in expanding itself. This limitation has been removed from the Version 5.2 64-bit kernel using a dynamic allocation and creation of PTA segments at a tailend as opposed to the frontend.

12.2 Interrupt saturation avoidance (5.2.0)

The device drivers the following Ethernet adapters have been enhanced to prevent interrupt saturation. Interrupt saturation is the condition where a network adapter is generating interrupts at a rate that prevents the adapter's interrupt handler from exiting. This feature is supported on the following Ethernet adapters.

- ▶ FC 2968 - IBM 10/100 Mbps Ethernet PCI Adapter
- ▶ FC 4961 - IBM Universal 4-Port 10/100 Ethernet Adapter
- ▶ FC 4962 - 10/100 Mbps Ethernet PCI Adapter II
- ▶ FC 2969 - Gigabit Ethernet-SX PCI Adapter
- ▶ FC 2975 - 10/100/1000 Base-T Ethernet PCI Adapter
- ▶ FC 5700 - IBM Gigabit Ethernet-SX PCI-X Adapter
- ▶ FC 5701 - IBM 10/100/1000 Base-TX Ethernet PCI-X Adapter

To prevent interrupt saturation, a counter was added to the device driver to prevent the interrupt handler from running endlessly. If the counter hits a certain

number of iterations, the interrupt handler will be forced to exit. These limits are configurable in the device attributes. The attribute names are `slih_hog` and `rx_hog`. These enhancements were made to several other adapters since AIX Version 4.3 (specifically FC 2969 and 2975), with the exception of the device attributes, are named `slih_hog` and `rxdesc_count`.

The `slih_hog` (second level interrupt handler) attribute indicates the maximum number of iterations to be performed by the device driver's interrupt handler before returning to the system first level interrupt handler (FLIH). Allowed values range from 1 to 1000000. The default value is 10. This attribute prevents the device driver's interrupt handler from running endlessly while the adapter is busy transmitting or receiving data.

The `rx_hog` attribute indicates the maximum number of receive descriptors to be processed by the device driver's receive handler routine. Allowed values range from 1 to 1000000. The default value is 1000. This attribute prevents the device driver's receive handler from running forever while the adapter is busy receiving data.

To change these attributes you must use the `chdev` command. These attributes are not found on a SMIT panel. The following example shows how to change the `slih_hog` to 20 and the `rx_hog` to 1100.

```
# lsattr -E -l ent1 -a slih_hog -a rx_hog
slih_hog      10          Interrupt events processed per interrupt True
rx_hog        1000        RX buffers processed per RX interrupt   True
# chdev -l ent1 -a slih_hog=20 -a rx_hog=1100
ent1 changed
# lsattr -E -l ent1 -a slih_hog -a rx_hog
slih_hog      20          Interrupt events processed per interrupt True
rx_hog        1100        RX buffers processed per RX interrupt   True
```

12.3 Hardware Multithreading enabling (5.1.0)

Hardware Multithreading (HMT) has been enabled in AIX 5L Version 5.1. Currently, HMT is supported by the RS/6000 Enterprise Server M80, IBM @server pSeries 620 6F1, IBM @server pSeries 660 6H1, and IBM @server pSeries 680 series. See `/usr/lpp/bos/README.HMT` in your system for more information.

The basic technique of HMT is that the processor holds the state of N threads. In the current processor implementation, N=2. For example, when a cache miss occurs (L1 or L2), which would normally delay the processor for many cycles, the processor switches to another state and executes instructions from that thread. This will help eliminate memory access delays, keep the CPU more fully utilized, and potentially improve the processor throughput.

If the HMT feature is enabled, looking on the system (by using, for example, **bindprocessor -q**) will show you twice as many processors as are physically installed. In some cases, there are significant performance improvements (15 to 20 percent), as reflected in the TPC-C benchmark. You must test your own workload and decide if any gain in performance and potential loss of Dynamic Processor Deallocation (RAS) is justified.

To enable the HMT feature, change the **bosdebug** mode and reboot the system:

```
# bosdebug -H on
```

If you want to disable the HMT feature, set the **bosdebug** mode back and reboot the system again:

```
# bosdebug -H off
```

If you try to enable on a non-supported hardware, you will receive output similar to the following:

```
# bosdebug -H on
HMT not supported on this system.
```

12.4 DVD-ROM support (5.2.0)

AIX 5L Version 5.2 supports the IDE DVD-ROM Drive (FC 2634). This device is also supported with AIX 5100-03.

12.5 Kernel scalability for SMP machines (5.1.0)

In AIX 5L Version 5.1, changes in the kernel services for process/thread event handling have been made to improve scalability on SMP machines. The contention on the `kernel_lock` has been reduced by introducing a new service that uses a complex lock for serialization instead of the global `kernel_lock`. This reduces contention for the global `kernel_lock` and allows multiple event callouts to be made simultaneously.

12.5.1 Proch callouts implementation

Proch callouts are a service that allows a kernel extension to register a callout handler to be called when threads or processes are created and destroyed.

In AIX 5L Version 5.0 and earlier, these handlers are registered using the `prochadd()`, and unregistered using the `prochdel()` kernel service.

In AIX 5L Version 5.1 new kernel services have been added to register and unregister callouts. In the new implementation, callouts are registered through `prochr_reg()` and unregistered using `prochr_unreg()`.

The new callouts handle exactly the same potential set of events at exactly the same points with respect to kernel operation. The kernel extension specifies which event callouts' desired version is being used when the handler is registered by passing a mask (`prochr_mask`) of the desired callout events.

When the handler is called, it is passed the address of its `prochr` structure, the event type (for example, `PROCHR_TERMINATE`), and the thread or process ID identifying the thread or process for which event the callout is being made.

The following additions have been made to the `proc.h` file:

```
struct prochr
{
    struct prochr *prochr_next; /* next pointer */
    void (*prochr_handler)(); /* function to be called */
    uint prochr_mask; /* conditions under which to call */
    int pad; /* padding for structure */
};
#define PROCHR_INITIALIZE (1UL<<PROCH_INITIALIZE)
#define PROCHR_TERMINATE (1UL<<PROCH_TERMINATE)
#define PROCHR_EXEC (1UL<<PROCH_EXEC)
#define PROCHR_THREADINIT (1UL<<THREAD_INITIALIZE)
#define PROCHR_THREADTERM (1UL<<THREAD_TERMINATE)

extern int prochr_reg(struct prochr *);
extern int prochr_unreg(struct prochr *);
```

12.6 Audio support for the 64-bit kernel (5.1.0)

Audio drivers have been added to support the 64-bit kernel on POWER workstations that have audio hardware. The audio drivers are comprised of the following filesets:

- ▶ `devices.isa_sio.baud.rte`
- ▶ `devices.isa_sio.IBM0017.rte`
- ▶ `devices.isa_sio.IBM0017.diag`

12.7 The millicode functions (5.2.0)

The performance of many heavily used memory operations in the `libc` library can be substantially improved if optimized code is used for the specific architecture of

the machine on which it is run. These new functions provide optimization for accessing code tuned for the functions `memmove()`, `bzero()`, `memset()`, `_fill()`, `memcpy()`, `memccpy()`, `memcmp()`, and `strstr()`. The optimization has also been implemented for eServers p630 and POWER4 processors. The millicode functions are new routines that exist in the AIX kernel. All programs compiled and run on AIX 5L Version 5.2 use the new millicode routines. The AIX 5L Version 5.2 of the libc library routines for these functions simply branch to the millicode routines in the kernel. Regardless of what machine a program is compiled and bound on, it will always use the correct millicode for the machine it is running on, since the millicode is contained in the kernel and the machine copies in the appropriate version of the routines at boot time. All of these libc routines are bound statically to avoid the time code cost of calling a shared library routine.

12.8 Ultimedia and PCMCIA device restrictions

AIX 5.1 no longer supports the following devices:

- ▶ AIX Ultimedia Services Audio and Video devices

In the past, the support of audio in AIX was accomplished by the Ultimedia Services (UMS) toolbox and API found on the AIX 4.3.3 Bonus Pack. The overall audio strategy has changed from UMS to JavaSound. The JavaSound API can be found on base AIX 5.x.

- ▶ PCMCIA device support

12.9 Diagnostics enhancements

The following enhancements have been made to the AIX 5L diagnostics utility.

12.9.1 Turboways PCI ATM adapter diagnostic enhancements (5.1.0)

The Turboways PCI ATM adapter provides full-duplex network connections at a rate of 155 Mbps. There are two versions available: Multi-Mode Fiber (MMF) connector and Unshielded Twisted Pair (UTP).

For example, to invoke diagnostic on the ATM adapter `atm0`, use the command:

```
# diag -d atm0
```

The Diagnostic Application performs hardware problem determination on configured hardware. In AIX 5L Version 5.1, for the ATM adapter, the diagnostic screens have been enhanced to show a running progress of the test being executed on the adapter. The Diagnostic Application will also analyze the error

log for specific errors logged against the adapter; appropriate action is taken if a error is found (this could be from nothing to posting a Service Request Number (SRN)).

Software prerequisites

In order for the diagnostic application to execute properly, the following software must be installed:

- ▶ devices.pci.14107c00.diag (required for both MMF and UTP adapters)
- ▶ devices.pci.14104e00.diag (required for MMF adapter only)
- ▶ bos.diag

Figure 12-1 and Figure 12-2 on page 773 show an example of the Advanced diagnostic routine when the Diagnostic Application is running. The bottom section of the screen changes as different tests are being run on the adapter. Figure 12-3 on page 773 shows the diagnostic panel when the test has been completed.

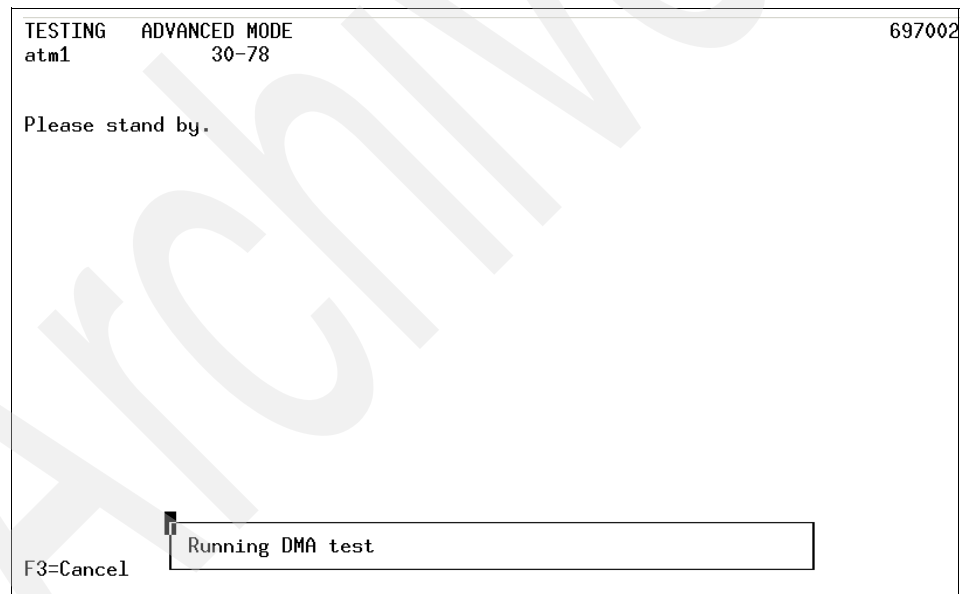


Figure 12-1 Diagnostic panel for running DMA test

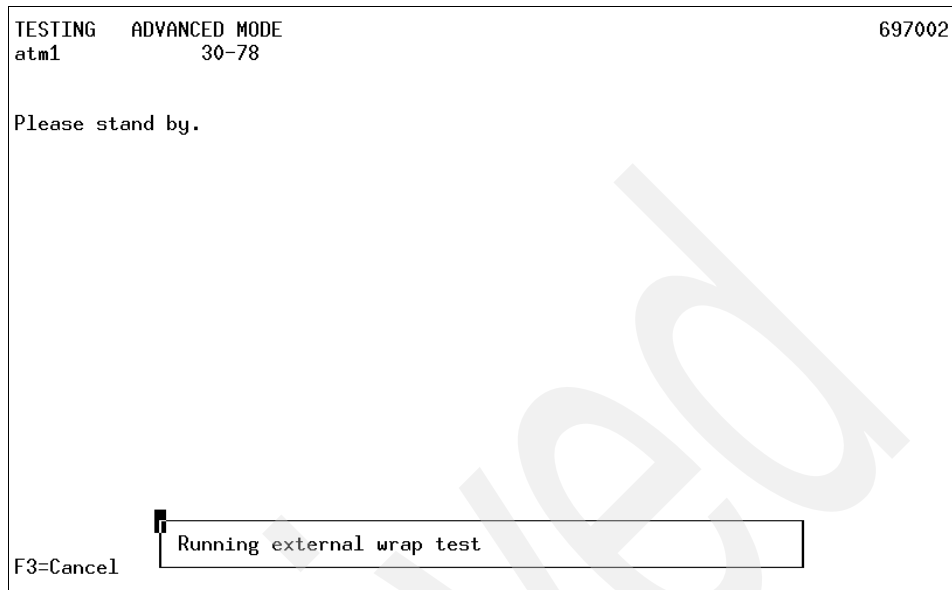


Figure 12-2 Diagnostic panel for running external wrap test

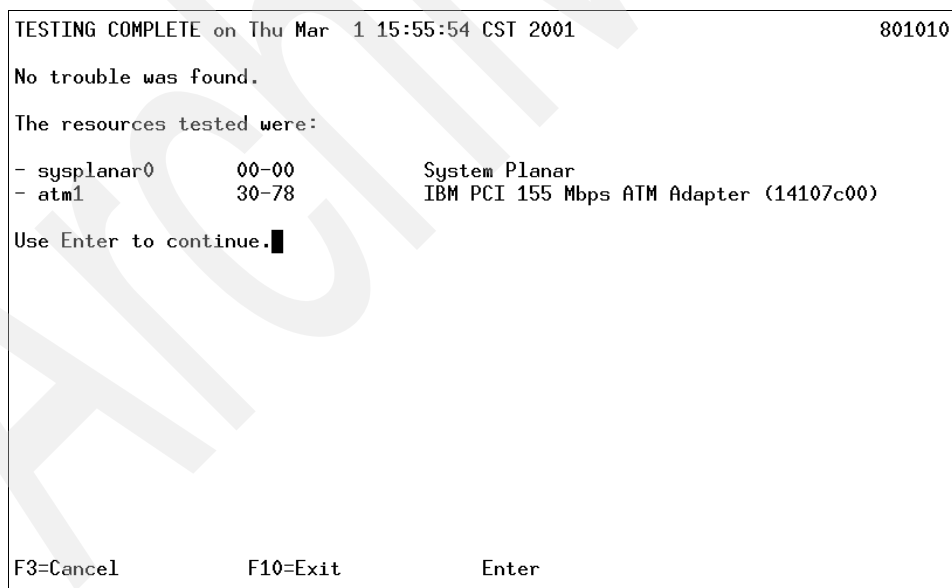


Figure 12-3 Diagnostic panel for test complete

12.9.2 LS-120 floppy drive diagnostic support (5.1.0)

The LS120 is a floppy disk drive that uses laser-formatted diskettes that have a capacity of 120 MB. The 3.5 inch floppy diskette drive diagnostic application has been modified to support the LS-120 diskette drive. To enter the diagnostic menus, log into the server as the root user and type `diag`. The diagnostic routines are the same as those for the 1.44 MB floppy drive.

12.9.3 Physical location codes (5.2.0)

With AIX 5L Version 5.2 the diagnostics panel now shows the physical location codes of devices instead of the AIX logical location as it did in the past. It is useful to determine directly where the devices are located without an exhausting cross-reference.

Figure 12-4 shows the physical device location in the diagnostic selection panel.

```
ADVANCED DIAGNOSTIC SELECTION                                     801006

From the list below, select any number of resources by moving
the cursor to the resource and pressing 'Enter'.
To cancel the selection, press 'Enter' again.
To list the supported tasks for the resource highlighted, press 'List'.

Once all selections have been made, press 'Commit'.
To exit without selecting a resource, press the 'Exit' key.

[MORE...5]
scsi2          P1-I8/Z1          Wide/Ultra-2 SCSI I/O Controller
hdisk2         P1-I8/Z1-A8       16 Bit LVD SCSI Disk Drive (9100 MB)
hdisk3         P1-I8/Z1-A9       16 Bit LVD SCSI Disk Drive (9100 MB)
ses0           P1-I8/Z1-Af       SCSI Enclosure Services Device
scsi3          P1-I8/Z2          Wide/Ultra-2 SCSI I/O Controller
hdisk4         P1-I8/Z2-A8       16 Bit LVD SCSI Disk Drive (9100 MB)
hdisk5         P1-I8/Z2-A9       16 Bit LVD SCSI Disk Drive (9100 MB)
ses1          P1-I8/Z2-Af       SCSI Enclosure Services Device
[MORE...23]

F1=Help          F4=List          F7=Commit          F10=Exit
F3=Previous Menu
```

Figure 12-4 Diagnostics panel

The error log entry and the `lsconf` command have also been modified to show the physical location device instead of the AIX logical location.

12.10 Common Character Mode support for AIX (5.1.0)

AIX 5L Version 5.1 allows support of Common Character Mode (CCM). CCM is an interface defined for graphic display adapters, which allows the graphics display to be used as an install console even though the adapter-specific device driver is not on the AIX boot media. With CCM, adapters supporting the interface will be recognized, configured, and made operational by AIX without the installation of the adapter-specific software.

Note: This function will be available only on Common Hardware Reference Platforms (CHRP) systems.

12.10.1 PCI Common Character Mode

Common Character Mode is a software and firmware mechanism defined for PCI graphics display adapters to provide a text-based interface for AIX installation on CHRP machines.

CCM makes use of the existing LFT interface to display drivers through a set of function pointers that each display adapter has currently provided. For CCM, these functions form the device-independent module, and this module resides in the boot image of the AIX installation CD. Device-dependent (specific) code will be part of the firmware residing in each adapter ROM. The common character mode device-independent code (CCM) communicates with the common character mode device dependent code (CDD) to get the device initialized and to perform any rendering operation as needed.

12.10.2 Device driver configuration

When AIX system configuration determines a display adapter is CCM capable and there is no device software package available for this device, it configures this graphics display adapter in CCM mode. From the ODM information, the system configuration knows about the PCI CCM configuration method and calls it.

12.11 AIX configuration commands (5.2.0)

Version 5.2 introduces enhancements to commands previously packaged with AIX.

12.11.1 The prtconf command

The **prtconf** command without any flags displays the system model, machine serial, processor type, number of processors, processor clock speed, CPU type, total memory size, network information, file system information, paging space information, and devices information. Version 5.2 introduces flags for this command. The command syntax is as follows, and the most commonly used flags are provided in Table 12-1.

Table 12-1 New flags for prtconf

Flags	Description
-c	Displays CPU type, for example, 32-bit or 64-bit
-k	Displays the kernel in use, for example, 32-bit or 64-bit
-L	Displays LPAR partition number and partition name if this is an LPAR partition; otherwise returns -1 NULL
-m	Displays system memory
-s	Displays processor clock speed in MHz
-v	Displays the VPD found in the Customized VPD object class for devices

Examples of this command are shown as follows:

```
# prtconf -k
Kernel Type: 32-bit
# prtconf -m
Memory Size: 512 MB
# prtconf -s
Processor Clock Speed: 332 MHz
```

12.11.2 The lsconf command

The **lsconf** command is provided for Linux affinity and has the same flags as the **prtconf** command.

12.12 Hardware support (5.2.0)

AIX 5L Version 5.2 exclusively supports PCI architecture machines. Support for Microchannel Bus Architecture (MCA), Personal Computer Memory Card International Association (PCMCIA), and Instrumentation Systems and Automation Society (ISA) devices has been withdrawn.

There is also a minimum hardware requirement for Version 5.2 of 128 MB of RAM and 2.2 GB of disk space. This section outlines the devices and machines that are no longer supported under Version 5.2

Version 5.2 withdraws support for the following architectures:

- ▶ MCA (built-in and plug-in)
- ▶ PCMCIA (built-in and plug-in)
- ▶ ISA (PReP built-in and plug-in, although CHRP built-in support remains)
- ▶ ISA (CHRP plug-in)

Version 5.2 withdraws support for the following processors:

- ▶ Power 1
- ▶ Power 2
- ▶ Power Single Chip (RSC)
- ▶ Power 2 Single Chip (P2SC)
- ▶ 601
- ▶ 603

Version 5.2 withdraws support for PReP-specific functions for the following packages:

- ▶ PReP PAL
- ▶ PReP desktop power management (hibernate)
- ▶ All IDE support
- ▶ All plug-in ISA adapter support
- ▶ All PReP built-in ISA adapter support (although support for CHRP built-in ISA support remains)
- ▶ PReP boot image from AIX Install CD-ROM
- ▶ PReP boot image from AIX Standalone Diagnostics CD-ROM

Version 5.2 withdraws support for selected PCI adapters that are only supported on PReP platforms, as provided in Table 12-2.

Table 12-2 Version 5.2 withdrawn PCI adapter support

Feature	Description
2408	10-95 F/W SCSI SE, PCI/SHORT/32BIT/5V
2409	10-95 F/W SCSI DIFF, EXT ONLY, PCI/SHORT/32BIT/5V

Feature	Description
2638	04-97 VIDEO CAPTURE(NTSC/PAL/SECAM), PCI/LONG/32BIT/5V
2648	06-95 (GXT150P) PCI/SHORT/32BIT/5V, GRAPHICS ADAPTER
2657	10-95 S15 GRAPHICS ADPTR, PCI/SHORT/32BIT/5V, WEITEK P9100
2837	04-97 MVP MULTI-MONITOR ADPTR, PCI/LONG/32BIT/3.3 OR 5V
2839	GXT100P Graphics Adapter
2854	10-96 (GXT500P), PCI/LONG/32BIT/3.3 OR 5V, GRAPHICS ADAPTER
2855	10-96 (GXT550P), PCI/LONG/32BIT/3.3 OR 5V, GRAPHICS ADAPTER
2856	06-95 PCI/SHORT/32BIT/3.3 OR 5V, 7250 ATTACH ADAPTER
7252	GXT1000, 7250-002 Internal Graphics Accelerator
7253	GXT1000, 7250-002 with graphics feature
7254	Video Output Option
8242	06-95 10/100BASET ETHERNET PCI/SHORT/32BIT/5V, (3COM)

Version 5.2 withdraws support for PReP-specific ISA adapters (plug-ins), as provided in Table 12-3.

Table 12-3 Version 5.2 withdrawn PReP-specific ISA adapter support

Feature	Description
2647	06-95 VIDEO CAPTURE ENHANCEMENT, ISA/SHORT
2701	10-95 4 PORT SDLC, ISA/LONG, EIA 232/V.35/X.21, (GALE)
2931	10-95 8-PORT, ISA/LONG, EIA232 ADPTR/FAN-OUT BOX
2932	04-96 8-PORT, ISA/LONG, EIA232/422 ADAPTER
2933	10-95 128-PORT, ISA/LONG, EIA232 ASYNCH CONTROLLER
2961	10-95 1 PORT X.25, SDLC, PPP, ISA/LONG, ADAPTER (C1X)
2971	06-95 TOKEN RING ADAPTER, ISA
2981	06-95 ETHERNET ADAPTER, ISA, RJ45/BNC
8240	06-95 A/M 3COM ETHERNET ISA/SHORT TP ONLY

Feature	Description
8241	06-95 A/M 3COM ETHERNET ISA/SHORT BNC/AUI

Note: Often a CHRP package would pre-req or co-req a PreP package to pull in required files to in order for the package to work. These selected files have now been moved to the CHRP packages and so no longer have a dependency on the PReP package, which has been removed.

Version 5.2 withdraws support for the following ISA adapters (plug-ins), even though they may run on a pSeries machine that is supported by Version 5.2. These include, but are not limited to, the adapters identified in Table 12-4.

Table 12-4 Version 5.2 withdrawn ISA adapter support

Feature	Description
2931	8-PORT, ISA/LONG, EIA232 ADPTR/FAN-OUT BOX
2933	128-PORT, ISA/LONG, EIA232 ASYNC CONTROLLER
2932	8-PORT, ISA/LONG, EIA232/422 ADPTR/FAN-OUT BOX
2961	1 PORT X.25, SDLC, PPP, ISA/LONG, ADAPTER
2701	4 PORT SDLC, ISA/LONG, EIA 232/V.35/X.21

AIX Version 4.3 removed support for all AIX notebooks. All remaining PReP notebook support has been withdrawn from Version 5.2.

CHRP power management support is withdrawn in Version 5.2.

All MCA support is withdrawn in Version 5.2. The primary packages and support include:

- ▶ MCA PAL
- ▶ All plug-in and built-in MCA support
- ▶ MCA boot image from AIX Install CD-ROM
- ▶ MCA boot image from AIX Standalone Diagnostics CD-ROM
- ▶ Pegasus and other MCA-specific commands

In some cases a CHRP plug-in and built-in I/O package will prerequisite or corequisite an MCA package to pull in required files. In all cases the CHRP packages have been rebuilt to include the files that they require, thus removing any dependency on the MCA package. The MCA package has also been removed.

Version 5.2 withdraws support for all PCI RS/6000 systems based on the PReP architecture and corresponding features including, but not limited to, the following, noting that all notebook support was withdrawn with Version 4.3, as provided in Table 12-5.

Table 12-5 Version 5.2 PCI RS/6000 withdrawn support listing

Systems	Family	Systems	Family
7020-0U0	40P	6015-066	
7020-SPE	40P	7248-100	43P
7020-B1B	40P	7248-120	43P
7020-B1C	40P	7248-132	43P
7020-D1D	40P	7043-140	
7020-D2D	40P	7043-240	
7020-D4E	40P	7024-E20	
6042-850	Notebook	7024-E30	
7247-821	Notebook	7025-F30	
7247-822	Notebook	7025-F40	
7247-823	Notebook	7317-F3L	
7247-860	Notebook	7026-H10	
6050	All models	6070	All models

Version 5.2 withdraws support for all MCA RS/6000 models and corresponding features including, but not limited to, the machines listed in Table 12-6.

Table 12-6 Version 5.2 MCA RS/6000 withdrawn support listing

Systems			
7006-41T	7006-41W	7006-42T	7006-42W
7007-N40	7008-M20	7008-M2A	7009-C10
7009-C20	7010-120	7010-130	7010-140
7010-150	7010-160	7011-220	7011-22G
7011-22S	7011-22W	7011-230	7011-23E 230E
7011-23S	7011-23T	7011-23W	7011-250

Systems			
7011-25E 250E	7011-25F 250FTURBO	7011-25S	7011-25T
7011-25W	7012-320	7012-32E 320E	7012-32H
7012-340	7012-34H	7012-350	7012-355
7012-360	7012-365	7012-36T 36T	7012-370
7012-375	7012-37T 37T	7012-380	7012-390
7012-397	7012-39H	7012-G02	7012-G30
7012-G40	7013-520	7013-52H	7013-530
7013-53E 530E	7013-53H	7013-540	7013-550
7013-55E 550E	7013-55L	7013-55S 550S	7013-560 560
7013-56F 560F	7013-570	7013-57F 570F	7013-580
7013-58F 580F	7013-58H	7013-590	7013-591
7013-595	7013-59H	7013-J01	7013-J30
7013-J40	7013-J50	7015-930	7015-950
7015-95E 950E	7015-970	7015-97B	7015-97E 970E
7015-97F 970F	7015-980	7015-98B	7015-98E 980E
7015-98F 980F	7015-990	7015-99E 990E	7015-99F 990F
7015-99J 990J	7015-99K 990K	7015-R10	7015-R20
7015-R21	7015-R24	7015-R30	7015-R3U R30U
7015-R40	7015-R4U R40U	7015-R50	7015-R5U R50U
7030-397	7030UPGRD	7030-3AT	7030-3BT
7030-3CT	7202-900		

Version 5.2 withdraws support for MCA-based SP nodes to the machines listed in Table 12-7 on page 782.

Table 12-7 Version 5.2 MCA-based SP nodes withdrawn support

Feature	Description
2001	62 MHz Thin Nodes
2002	66 MHz Thin Nodes
2003	66 MHz Wide Node
2004	66 MHz Thin Nodes
RPQ	66 MHz Wide (59H)
2005	77 MHz Wide Node
2006	112 MHz High Node
2007	135 MHz Wide Node
2008	120 MHz Thin Nodes
2009	200 MHz High Node
2022	160 MHz Thin Nodes

Version 5.2 withdraws support for the devices listed in Table 12-8.

Table 12-8 Version 5.2 device support withdrawn

Feature		
7027-HSC	PDOG	SE
7027-HSD	PDOG	DIFF
7236-001	ADEC	DRWR
7317-D10	DSK	DRWR
7318-P10		
7318-S20		
7319-100		
7319-110		

National language support

The national language support (NLS) environment is defined by a combination of language and geographic or cultural requirements. These conventions consist of four basic components:

- ▶ Translated language of the screens, panels, and messages
- ▶ Language convention of the geographical area and culture
- ▶ Language of the keyboard
- ▶ Language of the documentation

In an effort to support more languages, several enhancements have been made.

13.1 Input methods for Chinese locales (5.1.0)

In AIX 5L Version 5.1, the simplified Chinese locale (GBK, Zh_CN) has been enhanced with some new or upgraded input methods (IME). The following topics are discussed in the subsequent sections:

- ▶ Intelligent ABC
- ▶ BiaoXing Ma
- ▶ Zheng Ma
- ▶ PinYin
- ▶ Internal code

The updates of the input methods under the GBK locale has affected the bos.loc.iso.Zh_CN fileset.

13.1.1 Input methods window

By default, all supported input methods (including ABC, PinYin, Zheng Ma, BiaoXing Ma, and internal code) are in the enabled status. You can change its status by pressing the Ctrl+F12 keys and then selecting input method to enable or disable it (see Figure 13-1).



Figure 13-1 Window of Chinese input method

Key

The key is:

1. Window title.
2. Name of Input Methods: Including ABC, PinYin, Zheng Ma, Biao Xing Ma, and Internal Code IME.
3. Status of Input Method: ON/OFF. When the switch is ON, this input method is enabled. When the switch is OFF, it is disabled.

13.1.2 Intelligent ABC Input Method

Intelligent ABC Input Method (Figure 13-2) is a Chinese input method that is based on the phonetic representation of Chinese characters. It is very easy to study and master for Chinese people. With the aid of BiXing code, which is based on the basic stroke that constructs the glyph of Chinese character, ABC Input Method can input the GBK Chinese character (including GB code) easily.

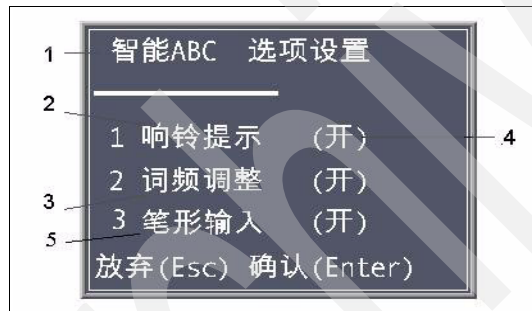


Figure 13-2 ABC Input Method setting window

Key

The key is:

1. Window of ABC Input Method setting.
2. Ring Indication option: If the switch is ON, the system will beep when an error code is generated.
3. Word Frequency Adjustment option: If the switch is ON, the ABC work frequency adjustment function will work as designed.
4. Switch option (ON/OFF): If the switch is OFF, the corresponding function in ABC IME will be disabled. The default is ON.
5. BiXing Code Input option: If the switch is ON, you can press the keypad to input some GBK Chinese characters; otherwise, BiXing input will be ignored.

13.1.3 BiaoXing Ma Input Method

BiaoXing Ma Input Method (Figure 13-3) is a kind of Chinese input method in which a Chinese character is divided into several components known as radicals according to its writing orders.

BiaoXingMa IME has three options: Ring indication, External code indication, and Displaying as striking.



Figure 13-3 BiaoXing Ma Input Method setting window

Key

The key is:

1. Name of BiaoXing Ma IME setting window.
2. Ring Indication option: If the switch is ON, the system will beep when an error code is generated.
3. External Code Indication option: If the switch is ON, the system will prompt what kind of external code will be generated next for corresponding candidate Chinese character.
4. Switch option (ON/OFF): If the switch is OFF, the corresponding function will be disabled. The default is ON.
5. Displaying as Striking Function option.

13.1.4 Zheng Ma Input Method

Zheng Ma Input Method (Figure 13-4 on page 787) is a Chinese input method that is based on the grapheme representation of a Chinese word. According to the modality information of the Chinese character, every word or phrase is

assigned a code, which is called graphemic code. ZhengMa is a kind of graphemic code input method.



Figure 13-4 Zheng Ma Input Method setting window

Key

The key is:

1. Name of Zheng Ma IME setting window.
2. Ring Indication option: If the switch is ON, the system will beep when an error code is generated.
3. External Code Indication option: If the switch is ON, the system will prompt what kind of external code will be generated next for the corresponding candidate Chinese character.
4. Switch option (ON/OFF): If the switch is OFF, the corresponding function will be disabled. Default is ON.
5. Displaying as Striking Function option.

13.1.5 PinYin Input Method

PinYin Input Method (Figure 13-5 on page 788) is a Chinese input method that is based on the phonetic representation of Chinese characters. According to the phonetic word building theory, a Chinese character can be divided into one or several phonemes according to its pronunciation.

PinYin Input Method is very similar with the QuanPin mode of Intelligent ABC Input Method, and its input manipulation is completely compliant with the standards of the Chinese Phonetic Scheme. This input method can input all the Chinese characters that are included in the Chinese extended Internal Code Specification.



Figure 13-5 PinYin Input Method setting window

Key

The key is:

1. Name of PinYin IME setting window.
2. Ring Indication option: If the switch is ON, the system will beep when an error code is generated.
3. Displaying as Striking Function option.
4. Switch option (ON/OFF): If the switch is OFF, the corresponding function will be disabled. The default is ON.

13.1.6 Internal Code Input Method

Internal Code Input Method (Figure 13-6) is an input method that complies with the code table defined in GBK (Chinese Internal Code Specification) and Unicode System Version 2 (UCS2). You can select one of them by pressing the Ctrl+F11 keys. (GBK is the default).



Figure 13-6 Internal Code Input Method setting window

Key

The key is:

1. Name of Internal Code IME setting window.
2. Ring Indication option: If the switch is ON, the system will beep when an error code is generated.
3. GBK Internal Code option: If the switch is ON, GBK Internal Code will be used. If the switch is OFF, UNICODE will be used instead. The default is the GBK Internal Code.
4. Switch option (ON/OFF).

13.2 Euro support for non-European countries (5.1.0)

AIX already provides full Euro enablement for all supported languages and territories through the UTF-8/Unicode locale environments. However, in AIX 5L Version 5.1, many of the existing country-specific codesets have been modified to incorporate the Euro symbol. These modifications are summarized in Table 13-1.

Table 13-1 Modified locales for using Euro

Existing codeset name	Euro symbol value	Locales using this codeset
ISO8859-7	0xA4	el_GR (Greece)
IBM-922	0xA4	Et_EE (Estonia)
IBM-921	0xA4	Lv_LV (Latvia) Lt_LT (Lithuania)
IBM-1046	0xFF	Ar_AA (Arabic)
IBM-1129	0xA4	Vi_VN (Vietnam)
big5	0xA3E1	Zh_TW (Trad. Chinese)

To enable the use of the Euro symbol, you have to install all the needed fonts for the specific language environment. The fonts are listed in Table 13-2.

Table 13-2 Locale settings versus font fileset

Locale	Font fileset
el_GR (Greece)	X11.fnt.iso7
Et_EE (Estonia)	X11.fnt.ucs.com

Locale	Font fileset
Lv_LV (Latvia), Lt_LT (Lithuania)	X11.fnt.ucs.com
Ar_AA (Arabic)	X11.fnt.ibm1046
Vi_VN (Vietnam)	X11.fnt.ucs.com
Zh_TW (Trad. Chinese)	X11.fnt.ucs.com

13.2.1 Testing the Euro glyph

To test the Euro glyph, invoke the `/usr/dt/bin/dtterm` or `/usr/bin/X11/aixterm` terminal. (The `/usr/bin/X11/xterm` terminal does not support international locales.) Use the **echo** command for checking the existence of the Euro glyph:

```
# echo "\0244"
```

You can also check the keyboard mappings with the following command:

```
# xmodmap -pke | grep EuroSign
keycode 27 = e E EuroSign
```

You can compile and run the following program to test the output of all printable one-byte characters:

```
#include <stdio.h>
main()
{
    int i;
    printf("    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f  \n");
    printf("-----\n");
    for(i=0x20; i<256; i++) {
        if(i == 0x80) i+= 0x20;
        if (i%16 == 0)
            printf("%x : ",i);
        if (i==0xa0)
            putchar(' ');
        else
            putchar(i);
        putchar(' ');
        putchar(' ');
        if (i%16 == 15)
            printf("\n");
        }
        printf("\n");
    }
}
```

13.3 National language support Euro (5.2.0)

On January 1, 2002, the European Monetary Union (EMU) which consisted of the following countries, finalized the conversion of their national currency to the euro (common European currency):

Austria, Belgium, Finland, France, Germany, Greece, Ireland, Italy, Luxembourg, Netherlands, Portugal, and Spain

In most participating countries, a dual circulation period will last between four weeks and two months. After that, national bank notes and coins will cease to be legal tender, and the euro bank notes and coins will become the sole currency throughout the Euro area.

Once the dual circulation period is over, you will still be able to exchange your national bank notes and coins for euro bank notes and coins at your national central bank either indefinitely or for a very long period of time (at least ten years in the case of bank notes). Concerning national coins, in most cases this period is limited to a few years.

The use of the Euro currency symbol and the currency formatting rules concerning it have become the default currency handling methods in AIX locales for those countries that are EMU members. A complete list is provided in Table 13-3.

Table 13-3 List of euro-enabled locales

Language/territory	UTF-8 locale name	ISO locale name	IBM-1252 locale name
Catalan/Spain	CA_ES.UTF-8	ca_ES.8859-15	ca_ES.IBM-1252
Dutch/Belgium	NL_BE.UTF-8	nl_BE.8859-15	nl_BE.IBM-1252
Dutch/Netherlands	NL_NL.UTF-8	nl_NL.8859-15	nl_NL.IBM-1252
English/Belgium	EN_BE.UTF-8	en_BE.8859-15	N/A
English/Ireland	EN_IE.UTF-8	en_IE.8859-15	N/A
Finnish/Finland	FI_FI.UTF-8	fi_FI.8859-15	fi_FI.IBM-1252
French/Belgium	FR_BE.UTF-8	fr_BE.8859-15	fr_BE.IBM-1252
French/France	FR_FR.UTF-8	fr_FR.8859-15	fr_FR.IBM-1252
French/Luxembourg	FR_LU.UTF-8	fr_LU.8859-15	N/A
German/Austria	DE_AT.UTF-8	de_AT.8859-15	N/A

Language/territory	UTF-8 locale name	ISO locale name	IBM-1252 locale name
German/Germany	DE_DE.UTF-8	de_DE.8859-15	de_DE.IBM-1252
German/Luxembourg	DE_LU.UTF-8	de_LU.8859-15	N/A
Greek/Greece	EL_GR.UTF-8	el_GR.ISO8859-7	N/A
Italian/Italy	IT_IT.UTF-8	it_IT.8859-15	it_IT.IBM-1252
Portuguese/Portugal	PT_PT.UTF-8	pt_PT.8859-15	pt_PT.IBM-1252
Spanish/Spain	ES_ES.UTF-8	es_ES.8859-15	es_ES.IBM-1252

Note that legacy codesets that do not contain the Euro symbol at all (for example, ISO8859-1) are not changed. These locales will continue to format currency values using each country's traditional currency formatting rules.

If traditional national currency formatting is desired, the LC_MONETARY category can be set by the application with the `setlocale()` subroutine or by the user with the LC_MONETARY environment variable to `XX_XX@preeuro`, where `XX_XX` is the language territory designation for the current locale. For example, to change the currency symbol EUR of the DE_DE locale, to the traditional symbol DM, issue the following command:

```
export LC_MONETARY=DE_DE@preeuro
```

The following command may be used to review the currency formatting for the current locale:

```
locale -k LC_MONETARY
```

The output of the command is similar to the following:

```
int_curr_symbol="DEM "  
currency_symbol="DM"  
mon_decimal_point=","  
mon_grouping="3"  
mon_thousands_sep="."  
positive_sign=""  
negative_sign="-"  
int_frac_digits=2  
frac_digits=2  
p_cs_precedes=0  
p_sep_by_space=1  
n_cs_precedes=0  
n_sep_by_space=1  
p_sign_posn=1  
n_sign_posn=1  
debit_sign=""
```

```
credit_sign=""
left_parenthesis=""
right_parenthesis=""
```

13.4 Korean keyboard enablement (5.1.0)

AIX 5L Version 5.1 now provides support for the alternate 103 Korean keyboard. This includes the Korean/English switch key, which is called Hangul. This key is located between the space bar and the right Alt key. There is a Chinese key, called Hanja, that is located between the left Alt key and the space bar.

Keyboard definitions will be added to support this 103-key keyboard in all possible AIX environments. Xmodmap and imkeymap support for X will be provided. LFT support is not possible because the LFT environment does not have the capacity for multi-byte encoding.

The keyboard definitions for the Korean locale will be based on IBM keyboard number 450. Figure 13-7 illustrates the keyboard layout.



Figure 13-7 Korean keyboard

13.5 NLS: Unicode Extension B Enhancement (5.2.0)

Version 5.2 lays the framework to support the GB18030-2000 codeset standard. This is a new Chinese standard that specifies an extended codepage and mapping table to Unicode Extension B.

The following section overviews the changes that have been made in Version 5.2 to allow the integration of the GB18030 codeset and Unicode Extension B.

13.5.1 Enhancements to Version 5.2

A Chinese mandate has been issued stating that any software application released for the Chinese market will have to incorporate support for GB18030. There is no deadline for this as yet, although Version 5.2 makes preparation for this change.

This support is for an additional 48,000 characters beyond the 20,902 that is supported by AIX in previous releases, in terms of font sets, input methods, and printer enablement.

Version 5.2 has the following enhancements to lay the framework for full support in the future:

- ▶ Implementation of the Unicode X output method (XOM).
- ▶ 64-bit enablement of all AIX base libraries (Unicode Extension B is only supported in the 64-bit environment).
- ▶ UTF-8 encoding becomes a maximum of 4 bytes per character (instead of 3 as in previous releases).
- ▶ Universal UCS Converter has been expanded from UCS-2 (2-byte) to UTF-32 (4-byte) encoding and incorporated into Version 5.2. UCS is used to convert source codeset to Unicode and then into the target codeset.
- ▶ the **iconv** command now allows conversion for UTF8 (expanded to handle 4-byte characters), UTF-32, UTF-16, UTF-16BE and UTF-16LE, and UTF-32 encoding.
- ▶ Version 5.2 provides the ability to convert from GB18030 to and from other commonly used codesets, including UTF-32.

13.6 Unicode XOM enhancement (5.2.0)

Version 5.2 enhances performance of the use of UCS-2 fonts when running under X-Windows and Motif applications.

UCS-2 fonts contain over 36,000 characters. In previous releases of AIX, the complete font set would be loaded even though the majority of the font set will not be needed by an application. The size of the set of fonts typically grows with each revision of AIX as greater functionality is provided.

Version 5.2 takes advantage of the X11R6 font feature, which allows an application to load only a subset of a font. The X Output Method (XOM) now uses on demand loading of only the font set that is needed, as opposed to loading all font sets at once whether required or not.

There are sixteen font subsets with 4096 characters in each subset. The font sets currently supported include:

AR_AA, AR_AE, AR_BH, AR_EG, AR_JO, AR_KW, AR_LB, AR_OM, AR_QA, AR_SA, AR_SY, AR_TN, BE_BY, BG_BG, CA_ES, CS_CZ, DA_DK, DE_AT, DE_CH, DE_DE, DE_LU, EL_GR, EN_AU, EN_BE, EN_CA, EN_GB, EN_IE, EN_IN, EN_NZ, EN_US, EN_ZA, ES_AR, ES_CL, ES_CO, ES_ES, ES_MX, ES_PE, ES_PR, ES_UY, ES_VE, ET_EE, FI_FI, FR_BE, FR_CA, FR_CH, FR_FR, FR_LU, HE_IL, HI_IN, HR_HR, HU_HU, IS_IS, IT_CH, IT_IT, JA_JP, KO_KR, LT_LT, LV_LV, MK_MK, NL_BE, NL_NL, NO_NO, PL_PL, PT_BR, PT_PT, RO_RO, RU_RU, SH_SP, SH_YU, SK_SK, SL_SI, SQ_AL, SR_SP, SR_YU, SV_SE, TH_TH, TR_TR, UK_UA, UNIVERSAL, VI_VN, ZH_CN, ZH_TW.

13.7 Additional locale support (5.2.0)

The support for locales provided in Table 13-4 has been added in AIX 5L Version 5.2.

Table 13-4 Additional locales

Language/territory	Abbreviation	Codeset	Keyboard definition
Arabic/Algeria	ar_DZ AR_DZ	ISO8859-6 ¹ UTF-8 ¹	ar_AA
Arabic/Morocco	ar_MA AR_MA	ISO8859-6 ¹ , UTF-8 ¹	ar_AA
Arabic/Yemen	ar_YE AR_YE	ISO8859-6 ¹ , UTF-8 ¹	ar_AA
Chinese (simplified)/Singapore	ZH_SG	UTF-8 ¹	zh_CN
Chinese/Hong Kong (simplified)	ZH_HK	UTF-8 ¹	zh_CN
English/Hong Kong	en_HK EN_HK	ISO8859-15, UTF-8 ¹	en_US
¹ Denotes that the bidirectional and UTF-8 locales will not have LFT keymap support.			

Language/territory	Abbreviation	Codeset	Keyboard definition
English/Philippines	en_PH EN_PH	ISO8859-15, UTF-8 ¹	en_US
English/Singapore	en_SG EN_SG	ISO8859-15, UTF-8 ¹	en_US
Indonesian/Indonesia	id_ID ID_ID	ISO8859-15, UTF-8 ¹	en_US
Malay/Malaysia	ms_MY MS_MY	ISO8859-15, UTF-8 ¹	en_US
Spanish/Bolivia	es_BO ES_BO	ISO8859-15, UTF-8 ¹	es_ES
Spanish/Costa Rica	es_CR ES_CR	ISO8859-15, UTF-8 ¹	es_ES
Spanish/Dominican Republic	es_DO ES_DO	ISO8859-15, UTF-8 ¹	es_ES
Spanish/Ecuador	es_EC ES_EC	ISO8859-15, UTF-8 ¹	es_ES
Spanish/El Salvador	es_SV ES_SV	ISO8859-15, UTF-8 ¹	es_ES
Spanish/Guatemala	es_GT ES_GT	ISO8859-15, UTF-8 ¹	es_ES
Spanish/Honduras	es_HN ES_HN	ISO8859-15, UTF-8 ¹	es_ES
Spanish/Nicaragua	es_NI ES_NI	ISO8859-15, UTF-8 ¹	es_ES
Spanish/Panama	es_PA ES_PA	ISO8859-15, UTF-8 ¹	es_ES
Spanish/Paraguay	es_PY ES_PY	ISO8859-15, UTF-8 ¹	es_ES
Spanish/United States	es_US ES_US	ISO8859-15, UTF-8 ¹	es_ES
¹ Denotes that the bidirectional and UTF-8 locales will not have LFT keymap support.			

13.8 Removal of obsolete locales (5.2.0)

Table 13-5 provides a list of locales based on the IBM-850 codeset that were removed from AIX 5L Version 5.2.

Table 13-5 Obsolete locales

Locale	Language	Territory
Ca_ES	Catalan	Spain
Da_DK	Danish	Denmark
De_CH	German	Switzerland
De_DE	German	Germany
En_GB	English	Great Britain
En_US	English	United States
Es_ES	Spanish	Spain
Fi_FI	Finnish	Finland
Fr_BE	French	Belgium
Fr_CA	French	Canada
Fr_CH	French	Switzerland
Fr_FR	French	France
Is_IS	Icelandic	Iceland
It_IT	Italian	Italy
NL_BE	Dutch	Belgium
NL_NL	Dutch	Netherlands
No_NO	Norwegian	Norway
Pt_PT	Portuguese	Portugal
Sv_SE	Swedish	Sweden

13.9 Unicode 3.1 support (5.2.0)

The Unicode standard is the most widely accepted standard in the computer industry for the encoding of the various languages of the world. On May 16, 2001, the 3.1 version of Unicode was published. This latest version of the

standard has increased the character set and has updated sections on character properties, the bidirectional rendering algorithm, and other text properties such as line breaking and collation rules for internationalized text.

Prior to AIX 5L Version 5.2, the locale support on AIX is based on Unicode Version 2.0.14. With AIX 5L Version 5.2, support for the current 3.1 version of Unicode has been added.

Prior to AIX 5L Version 5.2, Unicode data is represented on disk as UTF-8 encoded values. Unicode values are encoded as either 1, 2, or 3 byte quantities. With the addition of the Extension B characters, the UTF-8 encoding becomes a maximum of 4 bytes per character instead of 3. Table 13-6 summarizes the algorithm used to encode Unicode characters as a UTF-8 string.

Table 13-6 Unicode encoding as UTF-8

Unicode value	Unicode binary value	UTF-16 (binary)	UTF-8 1st byte	UTF-8 2nd byte	UTF-8 3rd byte	UTF-8 4th byte
U+0000 - U+007F	00000000 0xxxxxxx	00000000 0xxxxxxx	0xxxxxxx			
U+0080 - U+07FF	00000yyy yyxxxxxx	00000yyy yyxxxxxx	110yyyyy	10xxxxxx		
U+0800 - U+FFFF	zzzzyyyy yyxxxxxx	zzzzyyyy yyxxxxxx	1110zzzz	10yyyyyy	10xxxxxx	
U+10000 - U+10FFFF	uuuuuzzz zyyyyyyx xxxxx	110110ww wwzzzzyy + 110111yy yyxxxxxx	11110uuu	10uuzzzz	10yyyyyy	10xxxxxx

Where uuuuu = wwww + 1, to account for Plane 16 characters. For example, the CJK Extension B character with Unicode Value U+25A73 would be encoded in UTF-8 as binary 11100000 10100101 10101001 10110011 or a hexadecimal value of F0 A5 A9 B3.

All of the new Unicode 3.1 characters are added to the existing UTF-8 locale definitions and their character properties are consistent with the properties as defined in the Unicode character database Version 3.1 at:

<http://www.unicode.org/Public/3.1-Update/UnicodeData-3.1.0.txt>

The Numeric Input Method was added in AIX 5L Version 5.2. The Numeric Input Method allows users to input Unicode characters directly, regardless of what language they are using.

Note: Unicode Extension B characters, which are characters beyond 0xffff, cannot be displayed because of font limitations.

13.10 NLS JISX0213 compliance (5.2.0)

JISX0213 is a Japanese codeset standard that is an extension of JISX0208. This new standard adds additional 4344 Japanese characters for character displaying and input. The additional characters consist of 1908 JIS Level 3 Kanji characters and 2436 JIS Level 4 Kanji characters. JISX0213 enablement is implemented on the Unicode Extension B enhancement of Japanese UTF-8 locale (JA_JP) for 64-bit applications with the following functional enhancements:

- ▶ Maximum of 4 bytes per character in UTF-8 encoding
- ▶ Expansion from UCS-2(2-byte) to UTF-32(4-byte) of Universal UCS Converter
- ▶ Implementation of the Unicode X Output Method (XOM)

For the first release of AIX Version 5.2, JIS X0213 is provided as a technology preview and support is limited to JA_JP 64-bit applications with the following restrictions:

- ▶ Range of code point for input: UCS-2
- ▶ Character set to be displayed: JISX0208 and JISX0212

AIXIM allows users to chose Kuten Input Mode for JISX0208 only, up to JIS Level 3 characters, or up to Level 4 (full JISX0213) characters. Level 3 and Level 4 characters can be registered into the new JISX0213 user dictionary.

A new dictionary utility is provided to maintain JISX0213 characters. Support for JIS X0212 requires installation of following filesets.

bos.iconv.ucs.com	Unicode Base Converters for AIX Code Sets/Fonts
bos.loc.com.JP	Common Locale Support - Japanese
bos.loc.com.utf	Common Locale Support - UTF-8
bos.loc.utf.JA_JP	Base System Locale UTF Code Set - Japanese

You can get more information in the /usr/lpp/jls/doc/README.jisx0213.utf or /usr/lpp/jls/doc/README.jisx0213.pc file.

Abbreviations and acronyms

ABI	Application Binary Interface	BFF	Backup File Format
AC	Alternating Current	BI	Business Intelligence
ACL	Access Control List	BIND	Berkeley Internet Name Domain
ADSM	ADSTAR Distributed Storage Manager	BIST	Built-In Self-Test
ADSTAR	Advanced Storage and Retrieval	BLAS	Basic Linear Algebra Subprograms
AFPA	Adaptive Fast Path Architecture	BLOB	Binary Large Object
AFS	Andrew File System	BLV	Boot Logical Volume
AH	Authentication Header	BOOTP	Boot Protocol
AIO	Asynchronous I/O	BOS	Base Operating System
AIX	Advanced Interactive Executive	BPF	Berkeley Packet Filter
ANSI	American National Standards Institute	BSC	Binary Synchronous Communications
APAR	Authorized Program Analysis Report	BSD	Berkeley Software Distribution
API	Application Programming Interface	CA	Certificate Authority
AppA	Application Audio	CAD	Computer-Aided Design
AppV	Application Video	CAE	Computer-Aided Engineering
ARP	Address Resolution Protocol	CAM	Computer-Aided Manufacturing
ASCI	Accelerated Strategic Computing Initiative	CATE	Certified Advanced Technical Expert
ASCII	American National Standards Code for Information Interchange	CATIA	Computer-Graphics Aided Three-Dimensional Interactive Application
ASR	Address Space Register	CCM	Common Character Mode
ATM	Asynchronous Transfer Mode	CD	Compact Disk
AuditRM	Audit Log resource manager	CDE	Common Desktop Environment
AUI	Attached Unit Interface	CDLI	Common Data Link Interface
AWT	Abstract Window Toolkit	CD-R	CD Recordable
BCT	Branch on Count	CD-ROM	Compact Disk-Read Only Memory

CE	Customer Engineer	DASD	Direct Access Storage Device
CEC	Central Electronics Complex	DAT	Digital Audio Tape
CFD	Computational Fluid Dynamics	DBCS	Double Byte Character Set
CFM	Configuration File Manager	DBE	Double Buffer Extension
CGE	Common Graphics Environment	DC	Direct Current
CHRP	Common Hardware Reference Platform	DCE	Distributed Computing Environment
CIM	Common Information Model	DCUoD	Dynamic Capacity Upgrade on Demand
CISPR	International Special Committee on Radio Interference	DDC	Display Data Channel
CLI	Command Line Interface	DDS	Digital Data Storage
CLIO/S	Client Input/Output Sockets	DE	Dual-Ended
CLVM	Concurrent LVM	DES	Data Encryption Standard
CMOS	Complimentary Metal-Oxide Semiconductor	DFL	Divide Float
CMP	Certificate Management Protocol	DFP	Dynamic Feedback Protocol
COFF	Common Object File Format	DFS	Distributed File System
COLD	Computer Output to Laser Disk	DGD	Dead gateway detection
CPU	Central Processing Unit	DH	Diffie-Hellman
CRC	Cyclic Redundancy Check	DHCP	Dynamic Host Configuration Protocol
CRL	Certificate Revocation List	DIMM	Dual Inline Memory Module
CSID	Character Set ID	DIP	Direct Insertion Probe
CSM	Cluster Systems Management	DIT	Directory Information Tree
CSR	Customer Service Representative	DIVA	Digital Inquiry Voice Answer
CSS	Communication Subsystems Support	DLPAR	Dynamic LPAR
CSU	Customer Set-Up	DLT	Digital Linear Tape
CUoD	Capacity Upgrade on Demand	DMA	Direct Memory Access
CWS	Control Workstation	DMT	Directory Management Tool
DAD	Duplicate Address Detection	DMTF	Distributed Management Task Force
DAS	Dual Attach Station	DN	Distinguished Name
		DNLC	Dynamic Name Lookup Cache
		DNS	Domain Naming System
		DOE	Department of Energy
		DOI	Domain of Interpretation
		DOM	Document Object Model

DOS	Disk Operating System	ERRM	Event Response resource manager
DPCL	Dynamic Probe Class Library	ESID	Effective Segment ID
DRAM	Dynamic Random Access Memory	ESP	Encapsulating Security Payload
DRM	Dynamic Reconfiguration Manager	ESSL	Engineering and Scientific Subroutine Library
DS	Differentiated Service	ETML	Extract, Transformation, Movement, and Loading
DSA	Dynamic Segment Allocation	F/C	Feature Code
DSE	Diagnostic System Exerciser	F/W	Fast and Wide
DSMIT	Distributed SMIT	FC	Fibre Channel
DSU	Data Service Unit	FCAL	Fibre Channel Arbitrated Loop
DTD	Document Type Definition	FCC	Federal Communication Commission
DTE	Data Terminating Equipment	FCP	Fibre Channel Protocol
DW	Data Warehouse	FDDI	Fiber Distributed Data Interface
DWA	Direct Window Access	FDPR	Feedback Directed Program Restructuring
EA	Effective Address	FDX	Full Duplex
EC	Engineering Change	FIFO	First In/First Out
ECC	Error Checking and Correcting	FLASH EPROM	Flash Erasable Programmable Read-Only Memory
ECN	Explicit Congestion Notification	FLIH	First Level Interrupt Handler
EEPROM	Electrically Erasable Programmable Read Only Memory	FMA	Floating point Multiply Add operation
EFI	Extensible Firmware Interface	FPR	Floating Point Register
EHD	Extended Hardware Drivers	FPU	Floating Point Unit
EIA	Electronic Industries Association	FRCA	Fast Response Cache Architecture
EIM	Enterprise Identity Mapping	FRU	Field Replaceable Unit
EISA	Extended Industry Standard Architecture	FSRM	File System resource manager
ELA	Error Log Analysis	FTP	File Transfer Protocol
ELF	Executable and Linking Format	FTP	File Transfer Protocol
EMU	European Monetary Union	GAI	Graphic Adapter Interface
EOF	End of File		
EPOW	Environmental and Power Warning		

GAMESS	General Atomic and Molecular Electronic Structure System	IAR	Instruction Address Register
GID	Group ID	IBM	International Business Machines
GPFS	General Parallel File System	ICCCM	Inter-Client Communications Conventions Manual
GPR	General-Purpose Register	ICE	Inter-Client Exchange
GUI	Graphical User Interface	ICELib	Inter-Client Exchange library
GUID	Globally Unique Identifier	ICMP	Internet Control Message Protocol
HACMP	High Availability Cluster Multi Processing	ID	Identification
HACWS	High Availability Control Workstation	IDE	Integrated Device Electronics
HBA	Host Bus Adapters	IDL	Interface Definition Language
HCON	IBM AIX Host Connection Program/6000	IDS	Intelligent Decision Server
HDX	Half Duplex	IEEE	Institute of Electrical and Electronics Engineers
HFT	High Function Terminal	IETF	Internet Engineering Task Force
HIPPI	High Performance Parallel Interface	IHS	IBM HTTP Server
HiPS	High Performance Switch	IHV	Independent Hardware Vendor
HiPS LC-8	Low-Cost Eight-Port High Performance Switch	IIOIP	Internet Inter-ORB Protocol
HMC	Hardware Management Console	IJG	Independent JPEG Group
HMT	Hardware Multithreading	IKE	Internet Key Exchange
HostRM	Host resource manager	ILMI	Integrated Local Management Interface
HP	Hewlett-Packard	ILS	International Language Support
HPF	High Performance FORTRAN	IM	Input Method
HPSSDL	High Performance Supercomputer Systems Development Laboratory	INRIA	Institut National de Recherche en Informatique et en Automatique
HP-UX	Hewlett-Packard UNIX	IP	Internetwork Protocol (OSI)
HTML	Hyper-text Markup Language	IPL	Initial Program Load
HTTP	Hypertext Transfer Protocol	IPSec	IP Security
Hz	Hertz	IrDA	Infrared Data Association (which sets standards for infrared support including protocols for data interchange)
I/O	Input/Output		
I²C	Inter Integrated-Circuit Communications		

IRQ	Interrupt Request	LAPI	Low-Level Application Programming Interface
IS	Integrated Service	LDAP	Lightweight Directory Access Protocol
ISA	Industry Standard Architecture, Instruction Set Architecture	LDIF	LDAP Directory Interchange Format
ISAKMP	Internet Security Association Management Protocol	LED	Light Emitting Diode
ISB	Intermediate Switch Board	LFD	Load Float Double
ISDN	Integrated-Services Digital Network	LFT	Low Function Terminal
ISMP	InstallShield Multi-Platform	LID	Load ID
ISNO	Interface Specific Network Options	LLNL	Lawrence Livermore National Laboratory
ISO	International Organization for Standardization	LMB	Logical Memory Block
ISV	Independent Software Vendor	LP	Logical Partition
ITSO	International Technical Support Organization	LPAR	Logical Partitioning
IXFR	Incremental Zone Transfer	LP64	Long-Pointer 64
JBOD	Just a Bunch of Disks	LPI	Lines Per Inch
JCE	Java Cryptography Extension	LPP	Licensed Program Product
JDBC	Java Database Connectivity	LPR/LPD	Line Printer/Line Printer Daemon
JFC	Java Foundation Classes	LRU	Least Recently Used
JFS	Journaled File System	LTG	Logical Track Group
JSSE	Java Secure Sockets Extension	LV	Logical Volume
JTAG	Joint Test Action Group	LVCB	Logical Volume Control Block
JVMPI	Java Machine Profiling Interface	LVD	Low Voltage Differential
KDC	Key Distribution Center	LVM	Logical Volume Manager
L1	Level 1	MAP	Maintenance Analysis Procedure
L2	Level 2	MASS	Mathematical Acceleration Subsystem
L3	Level 3	MAU	Multiple Access Unit
LAM	Loadable Authentication Module	MBCS	Multi-Byte Character Support
LAN	Local Area Network	Mbps	Megabits Per Second
LANE	Local Area Network Emulation	MBps	Megabytes Per Second
		MCA	Micro Channel Architecture
		MCAD	Mechanical Computer-Aided Design
		MCM	Multichip Module

MDF	Managed Object Format	NIM	Network Installation Management
MDI	Media Dependent Interface	NIS	Network Information Service
MES	Miscellaneous Equipment Specification	NL	National Language
MFLOPS	Million of Floating point Operations Per Second	NLS	National Language Support
MII	Media Independent Interface	NT-1	Network Terminator-1
MIB	Management Information Base	NTF	No Trouble Found
MIP	Mixed-Integer Programming	NTP	Network Time Protocol
MLR1	Multi-Channel Linear Recording 1	NUMA	Non-Uniform Memory Access
MMF	Multi-Mode Fibre	NUS	Numerical Aerodynamic Simulation
MODS	Memory Overlay Detection Subsystem	NVRAM	Non-Volatile Random Access Memory
MP	Multiprocessor	NWP	Numerical Weather Prediction
MPC-3	Multimedia PC-3	OACK	Option Acknowledgment
MPI	Message Passing Interface	OCS	Online Customer Support
MPIO	Multipath I/O	ODBC	Open DataBase Connectivity
MPOA	Multiprotocol over ATM	ODM	Object Data Manager
MPP	Massively Parallel Processing	OEM	Original Equipment Manufacturer
MPS	Mathematical Programming System	OLAP	Online Analytical Processing
MSS	Maximum Segment Size	OLTP	Online Transaction Processing
MST	Machine State	ONC+	Open Network Computing
MTU	Maximum Transmission Unit	OOUI	Object-Oriented User Interface
MWCC	Mirror Write Consistency Check	OSF	Open Software Foundation, Inc.
MX	Mezzanine Bus	OSL	Optimization Subroutine Library
NBC	Network Buffer Cache	OSLp	Parallel Optimization Subroutine Library
NCP	Network Control Point	P2SC	POWER2 Single/Super Chip
ND	Neighbor Discovery	PAG	Process Authentication Group
NDP	Neighbor Discovery Protocol	PAM	Pluggable Authentication Mechanism
NDS	Novell Directory Services	PAP	Privileged Access Password
NFB	No Frame Buffer		
NFS	Network File System		
NHRP	Next Hop Resolution Protocol		

PBLAS	Parallel Basic Linear Algebra Subprograms	PRNG	Pseudo-Random Number Generator
PCB	Protocol Control Block	PSE	Portable Streams Environment
PCI	Peripheral Component Interconnect	PSSP	Parallel System Support Program
PDT	Paging Device Table	PTF	Program Temporary Fix
PDU	Power Distribution Unit	PTPE	Performance Toolbox Parallel Extensions
PE	Parallel Environment	PTX	Performance Toolbox
PEDB	Parallel Environment Debugging	PV	Physical Volume
PEX	PHIGS Extension to X	PVC	Permanent Virtual Circuit
PFS	Perfect Forward Security	PVID	Physical Volume Identifier
PGID	Process Group ID	QMF	Query Management Facility
PHB	Processor Host Bridges	QoS	Quality of Service
PHY	Physical Layer	QP	Quadratic Programming
PID	Process ID	RAID	Redundant Array of Independent Disks
PID	Process ID	RAM	Random Access Memory
PIOFS	Parallel Input Output File System	RAN	Remote Asynchronous Node
PKCS	Public-Key Cryptography Standards	RAS	Reliability, Availability, and Serviceability
PKI	Public Key Infrastructure	RDB	Relational DataBase
PKR	Protection Key Registers	RDBMS	Relational Database Management System
PMTU	Path MTU	RDF	Resource Description Framework
POE	Parallel Operating Environment	RDISC	ICMP Router Discovery
POP	Power-On Password	RDN	Relative Distinguished Name
POSIX	Portable Operating Interface for Computing Environments	RDP	Router Discovery Protocol
POST	Power-On Self-test	RFC	Request for Comments
POWER	Performance Optimization with Enhanced Risc (Architecture)	RIO	Remote I/O
PPC	PowerPC	RIP	Routing Information Protocol
PPM	Piecewise Parabolic Method	RIPL	Remote Initial Program Load
PPP	Point-to-Point Protocol	RISC	Reduced Instruction-Set Computer
PREP	PowerPC Reference Platform	RMC	Resource Monitoring and Control

ROLTP	Relative Online Transaction Processing	SEPBU	Scalable Electrical Power Base Unit
RPA	RS/6000 Platform Architecture	SGI	Silicon Graphics Incorporated
RPC	Remote Procedure Call	SGID	Set Group ID
RPL	Remote Program Loader	SHLAP	Shared Library Assistant Process
RPM	Redhat Package Manager	SID	Segment ID
RSC	RISC Single Chip	SIT	Simple Internet Transition
R SCT	Reliable Scalable Cluster Technology	SKIP	Simple Key Management for IP
RSE	Register Stack Engine	SLB	Segment Lookaside Buffer
RSVP	Resource Reservation Protocol	SLIH	Second Level Interrupt Handler
RTC	Real-Time Clock	SLIP	Serial Line Internet Protocol
RVSD	Recoverable Virtual Shared Disk	SLR1	Single-Channel Linear Recording 1
SA	Secure Association	SM	Session Management
SACK	Selective Acknowledgments	SMB	Server Message Block
SAN	Storage Area Network	SMIT	System Management Interface Tool
SAR	Solutions Assurance Review	SMP	Symmetric Multiprocessor
SAS	Single Attach Station	SMS	System Management Services
SASL	Simple Authentication and Security Layer	SNG	Secured Network Gateway
SBCS	Single-Byte Character Support	SNIA	Storage Networking Industry Association
ScaLAPACK	Scalable Linear Algebra Package	SNMP	Simple Network Management Protocol
SCB	Segment Control Block	SOI	Silicon-on-Insulator
SCSI	Small Computer System Interface	SP	IBM RS/6000 Scalable POWER parallel Systems
SCSI-SE	SCSI-Single Ended	SP	Service Processor
SDK	Software Development Kit	SPCN	System Power Control Network
SDLC	Synchronous Data Link Control	SPEC	System Performance Evaluation Cooperative
SDR	System Data Repository	SPI	Security Parameter Index
SDRAM	Synchronous Dynamic Random Access Memory	SPM	System Performance Measurement
SE	Single Ended		

SPOT	Shared Product Object Tree	UDF	Universal Disk Format
SPS	SP Switch	UDI	Uniform Device Interface
SPS-8	Eight-Port SP Switch	UIL	User Interface Language
SRC	System Resource Controller	ULS	Universal Language Support
SRN	Service Request Number	UNI	Universal Network Interface
SSA	Serial Storage Architecture	UP	Uniprocessor
SSC	System Support Controller	USB	Universal Serial Bus
SSL	Secure Socket Layer	USLA	User-Space Loader Assistant
STFDU	Store Float Double with Update	UTF	UCS Transformation Format
STP	Shielded Twisted Pair	UTM	Uniform Transfer Model
SUID	Set User ID	UTP	Unshielded Twisted Pair
SUP	Software Update Protocol	UUCP	UNIX-to-UNIX Communication Protocol
SVC	Switch Virtual Circuit	VACM	View-based Access Control Model
SVC	Supervisor or System Call	VESA	Video Electronics Standards Association
SWVPD	Software Vital Product Data	VFB	Virtual Frame Buffer
SYNC	Synchronization	VG	Volume Group
TCB	Trusted Computing Base	VGDA	Volume Group Descriptor Area
TCE	Translate Control Entry	VGSA	Volume Group Status Area
Tcl	Tool Command Language	VHDCI	Very High Density Cable Interconnect
TCP/IP	Transmission Control Protocol/Internet Protocol	VIPA	Virtual IP Address
TCQ	Tagged Command Queuing	VLAN	Virtual Local Area Network
TGT	Ticket Granting Ticket	VMM	Virtual Memory Manager
TLB	Translation Lookaside Buffer	VP	Virtual Processor
TLS	Transport Layer Security	VPD	Vital Product Data
TOS	Type Of Service	VPN	Virtual Private Network
TPC	Transaction Processing Council	VSD	Virtual Shared Disk
TPP	Toward Peak Performance	VSM	Visual System Manager
TSE	Text Search Engine	VSS	Versatile Storage Server
TSE	Text Search Engine	VT	Visualization Tool
TTL	Time To Live	WAN	Wide Area Network
UCS	Universal Coded Character Set	WBEM	Web-based Enterprise Management
UDB EEE	Universal Database and Enterprise Extended Edition		

WLM	Workload Manager
WTE	Web Traffic Express
XCOFF	Extended Common Object File Format
XIE	X Image Extension
XIM	X Input Method
XKB	X Keyboard Extension
XL F	XL Fortran
XML	Extended Markup Language
XOM	X Output Method
XPM	X Pixmap
XSSO	Open Single Sign-on Service
XTF	Extended Distance Feature
XVFB	X Virtual Frame Buffer

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 814.

- ▶ *AIX 5L Workload Manager (WLM)*, SG24-5977
- ▶ *AIX Reference for Sun Solaris Administrators*, SG24-6584
- ▶ *AIX Version 4.3 Differences Guide*, SG24-2014
- ▶ *Introducing VERITAS Foundation Suite for AIX*, SG24-6619
- ▶ *Running Linux Applications on AIX*, SG24-6033

Other resources

These publications are also relevant as further information sources:

- ▶ *Performance Toolbox Version 2 and 3 Guide and Reference*, SC23-2625
- ▶ *Reliable Scalable Cluster Technology Version 2 Release 1 Resource Monitoring and Control Guide and Reference*, SC23-4345
- ▶ W. Richard Stevens, *UNIX Network Programming, Volume 1: Networking APIs: Sockets and XTI*, Second Edition, Prentice Hall, 1997, Product Number 013490012X

Referenced Web sites

These Web sites are also relevant as further information sources:

- ▶ Agfa-Gevaert Group
<http://www.agfa.com>
- ▶ AIX
<http://www.ibm.com/servers/aix/products/aixos/linux/index.html>

- ▶ AIX Toolbox for Linux Applications home page
<http://www.ibm.com/servers/aix/products/aixos/linux/>
- ▶ AT&T Center for Internet Research
<http://www.aciri.org>
- ▶ ATM specifications
<http://www.atmforum.com/standards/approved.html>
- ▶ Cisco Systems, Inc.
<http://www.cisco.com>
- ▶ Counterpane Labs home page
<http://www.counterpane.com/yarrow.html>
- ▶ Distributed Management Task Force, Inc.
<http://www.dmtf.org>
- ▶ Dynamic Probe Class Library
<http://www.cs.wisc.edu/~paradyn/DPCL>
- ▶ fvwm2 window managersources download
<http://fvwm.org> or <http://xwinman.org>,
- ▶ GNOME project home page
<http://www.gnome.org>
- ▶ GNU coding standards
http://www.gnu.org/prep/standards_toc.html
- ▶ GNU project home page
<http://www.gnu.org>
- ▶ IBM AIX Web browsers home page
<http://www.ibm.com/servers/aix/browsers/index.html>
- ▶ IBM developerWorks Web site OpenSSH package download
<http://oss.software.ibm.com/developerworks/projects/opensshi>
- ▶ IBM SecureWay Directory information
<http://www-4.ibm.com/software/network/directory>
- ▶ Inline JFS2 log sizing information
http://publib16.boulder.ibm.com/pseries/en_US/infocenter/base
- ▶ Internet Engineering Task Force
<http://www.ietf.org>

- ▶ Internic root server download
<ftp://ftp.rs.internic.net/domain/named.root>
- ▶ JAVA information
<http://www.ibm.com/developerworks/java/jdk/aix/>
- ▶ JAVA Cryptography Extension
<http://java.sun.com/products/jce>
- ▶ JAVA Secure Socket Extension
<http://java.sun.com/products/jsse>
- ▶ KDE project home page
<http://www.kde.com>
- ▶ Korn Shell home page
<http://www.kornshell.com>
- ▶ Linux FreeS/WAN project home page
<http://www.freeswan.org>
- ▶ Log size information
http://publib16.boulder.ibm.com/pseries/en_US/infocenter/base
- ▶ **lsof** command download
<ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/lsof/lsof-4.61-2.aix5.1.ppc.rpm>
- ▶ Public Key Cryptography
<http://www.rsasecurity.com/rsalabs/pkcs/index.html>
- ▶ RedHat
<http://www.redhat.com>
- ▶ RFC information sources
<http://www.ietf.org/rfc.html>
- ▶ RPM packages - a useful link
<http://www-1.ibm.com/servers/aix/products/aixos/linux/download.html>
- ▶ SecureWay Directory
<http://www-4.ibm.com/software/network/directory>
- ▶ Sendmail standards
<http://www.sendmail.org>
- ▶ OpenSSH home page
<http://www.openssh.or>

- ▶ Selected publications of Pravin Bhagwat
<http://www.cs.umd.edu/~pravin/publications/publist.htm>
- ▶ Storage Network Industry Association
<http://www.snia.org>
- ▶ The Open Group
<http://www.opennc.com>
- ▶ Unicode character database version 3.1
<http://www.unicode.org/Public/3.1-Update/UnicodeData-3.1.0.txt>
- ▶ Uniform Device Driver (UDI) home page
<http://www.projectudi.org>
- ▶ University of Maryland
<http://www.cs.umd.edu>
- ▶ Virtual Desktop for X Windows
<http://fvwm.org>
- ▶ X/Open Single Signon
<http://www.opennc.com/pubs/catalog/u039.htm>
- ▶ X Windows Manager
<http://xwinman.org>

How to get IBM Redbooks

You can order hardcopy Redbooks, as well as view, download, or search for Redbooks at the following Web site:

ibm.com/redbooks

You can also download additional materials (code samples or diskette/CD-ROM images) from that site.

IBM Redbooks collections

Redbooks are also available on CD-ROMs. Click the CD-ROMs button on the Redbooks Web site for information about all the CD-ROMs offered, as well as updates and formats.

Index

Symbols

- .indirect
 - indirect blocks
 - JFS performance 219
- .indirect block 219
- .times file 89
- /dev/nsmb0 device 531
- /dev/random 652
- /dev/urandom 652
- /etc/cdromd.conf file 257
- /etc/dfpd.conf 546
- /etc/dns/named.conf 441
- /etc/ftpaccess.ctl 509
- /etc/hosts 587
- /etc/ipsec.conf 539
- /etc/ipsec.secrets 539
- /etc/irs.conf 587
- /etc/isakmpd.conf file 544
- /etc/mail/alias 387
- /etc/mail/aliases.db 387
- /etc/mail/aliases.pag 387
- /etc/mail/sendmail.cf 387
- /etc/mkcifs_fs script 531
- /etc/netsvc.conf 587
- /etc/policyd.conf 433, 438
- /etc/rc.net 474
- /etc/resolv.lldap 588
- /etc/rndc.conf 443
- /etc/security/audit/config 591
- /etc/security/audit/events 590
- /etc/tunables 425
- /mkcd/cd_fs 374
- /mkcd/cd_image 374
- /mkcd/mksysb_image 374–375
- /proc 688
 - see also proc pseudo file system 684
- /proc/pid#/cwd 688
- /proc/pid#/fd 688
- /sbin/helpers/mount_cifs mount helper 531
- /tmp/hosts.ldif 585
- /usr/include/net/frca.h 521
- /usr/include/sys/limits. 463
- /usr/include/sys/limits.h 221

- /usr/lib/boot 297
- /usr/lib/drivers/nsmbdd device driver 531
- /usr/lib/drivers/qos 432
- /usr/lib/methods/cfgnsmb configuration method 531
- /usr/samples/tcpip/anon.users.ftp 512
- /usr/samples/tcpip/libpcap 523
- /usr/sbin/db_file.dhcpd 296
- /usr/sbin/policyd 432
- /var/adm/ras/trcfile trace report file 273
- _AIO_AIX_SOURCE macro 20

Numerics

- 32-bit
 - binary compatibility 12
 - kernel extension 767
- 32-bit DWA 30
- 32bit, WLM process type 49
- 64-bit
 - binary compatibility 12
 - FRCA API 521
 - kernel extension 767
- 64bit
 - WLM process type 49
- 64-bit applications 30
- 64-bit DWA 30
- 64-bit indirect mode 30
- 64-bit kernel
 - CAPP/EAL4+ 647
- 64-bit kernel, JFS2 236
- 7 554

A

- ABI
 - Sun user thread API 696
- accelerator
 - accessibility for Web-based system Manager 353
- ACCEP_LICENSES field , bosinst.data file 361
- accept() routine 489
- accounting system 68
- acctcom command 68
- active MWCC 210

- active_dgd parameter 478
- adding routes 464
- address resolution protocol, enhancement 487
- addresses, virtual IP 495
- address-to-nodename translation 486
- administration
 - workload manager 38
- adump
 - adump.report 315
 - snap flag 315
- adump command 315
- Advanced Menu 80
- affinity
 - System V 663
- AGFA rasterizer 767
- AI_ADDRCONFIG 490
- AI_ALL 490
- AI_NUMERICSERV 490
- AI_V4MAPPED 490
- AI_V4MAPPED flag 490
- aio.h header file 21
- AIX
 - Version 5.2 Migration 323
- AIX 5L Version 5.2
 - Platform support 776
- aix database message 356
- AIX enhanced file system 224
- AIX Fast Connect 525
- AIX inter operability
 - CSM 165
- AIX LPP packages 736
- AIX source affinity for Linux applications 762
- alias 387
- alias command, KDB 298
- aliases database 387
- aliases file, sendmail 386
- aliases, networking 497
- alignment interrupts 397
- alog command 151
- alstat command 397
- alt_disk_install command 372
- alternate configurations
 - workload manager 57
- Alternate disk install 318
 - BOS installation time 318
 - Filesets 319
- anon.users.ftp FTP configuration file 512
- ANSI, terminal emulation 391
- anti-spam, sendmail 386
- aopt command 409
- API 387
 - dual-semantic 762
 - EIM 649
 - Java security 570
 - Linux 762
 - performance monitor 397
 - perfstat_alloc 422
 - perfstat_cpu 422
 - perfstat_cpu_total 422
 - perfstat_disk 422
 - perfstat_disk_total 422
 - perfstat_memory_total 422
 - perfstat_netinterface 422
 - perfstat_netinterface_total 422
 - perfstat_pagingspace1 422
 - perfstat_protocol 422
 - Sun user thread 696
- applet mode 339, 347
- application path names (WLM) 48
- application tags (WLM) 49
- Ar_AA locale, Euro support 789
- ar_DZ locale 795
- ar_MA locale 795
- ar_YE locale 795
- architecture
 - Web-based System Manager 329
- argsbuf 748
- ARP 468
- as pseudo file 685
- as_att() function call 12
- assignment rules 62
- assignment rules (WLM) 45
- Asynchronous I/O 20
- asynchronous transfer mode, 5.2 enhancements 553
- atfork handler, pthread library 16
- ATM 553
 - bridge 547
 - Checksum offload 555
 - Control timer 554
 - diagnostic 771
 - Dynamic MTU design 555
 - entstat command 552
 - Forward disconnect timer 554
 - IBM 2216 547
 - IEEE 802.3 Ethernet emulation 547
 - IEEE 802.5 token ring emulation 547
 - IP fragmentation 548

- LAN Emulation device driver 547
- LANE2 554
- LE client 547
- mpcstat command 552
- MPOA 548
- MPOA client 547
- MTU size 548
- Standard Ethernet emulation 547
- tokstat command 552
- trace, MPOA 548
- ATM LANE
 - Ethernet debug tracing 552
 - frame size 549
 - token ring debug tracing 552
- atrm command 666
- Attribute value grouping
 - Workload manager 83
- Attribute value grouping configuration
 - Web-based system manager 84
- attributes
 - basic user 571
 - classes 42
 - extended user 571
 - Kerberos user 578
 - registry 572
- attributes, localshm 44
- audio
 - drivers 770
 - filesets 770
 - hardware 770
- audit events, LDAP 590
- Audit log dialogs 153
- audit log resource manager 152
- audit plug-in, LDAP 589
- audit service, LDAP 592
- AuditRM
 - see also audit log resource manager 152
- auth
 - authentication module 574
- auth, VPN function mapping 540
- authby, VPN function mapping 540
- authentication method
 - DCE 571
 - Kerberos 575
 - standard AIX 571
- AUTO SYNC
 - lsvg output field 197
- auto, VPN function mapping 539
- autoconf command, Linux 753
- autoconf6 command 491
- AutoFS
 - multi-threaded 253
- automake command, Linux 753
- Automated offline
 - tprof enhancement 412
- automatic assignment (WLM) 45
- automount facility 257
 - /etc/cdromd.conf 257
- automountd
 - multi-threaded 253
- B**
 - B+-tree (JFS2) 225
 - backsnap command 249
 - bos.rte.filesystem 250
 - backups
 - snapshot support 213
 - Bar Display 74
 - baseDN 585
 - bash2 command, Linux 753
 - basic tunnel connection, VPN 542
 - BeginCriticalSection() system call 21
 - Berkeley database 386
 - Berkeley DB 386
 - Berkeley Packet Filter 522
 - bffcreate command 365
 - BiaoXing Ma input method 786
 - big volume group 215
 - big5, Euro support 789
 - bin.bin 218
 - binary compatibility 12, 355
 - binary format address 487
 - BIND 9 439
 - BIND service 587
 - bindintcpu command 133
 - bindprocessor command 133
 - bison command, Linux 753
 - BiXing 785
 - block size 225
 - blocks, mkiv and extendlv command 216
 - Bonus Pack 251
 - Boot LED display
 - LEDs
 - second line display 285
 - bootlist command 185
 - BOS install 747
 - BOS install, software license agreement 362

- BOS installation
 - enable 64-bit kernel and JFS2 237
- BOS installation, desktop selection 750
- bos.adt.include 697
- bos.adt.lib 697
- bos.alt_disk_install.boot_images
 - Alternate disk install 319
- bos.alt_disk_install.rte
 - Alternate Disk Install 319
- bos.cdmount fileset 257
- bos.msg
 - System V Print 727
- bos.perf.tools fileset 396
- bos.rte.control
 - WLMRM 86
- bos.rte.filesystem
 - backsnap command 250
 - snapshot command 250
- bos.rte.libc 486
- bos.svprint
 - System V Print 727
- bos.sysmgt.trcgui_samp fileset 273
- bos.terminfo.svprint.data
 - System V Print 727
- bosboot command 398
 - KDB 297
- bosdebug command, enable HMT 769
- bosinst.data DUMPDEVICE 278
- bosinst.data file
 - ACCEP_LICENSES field 361
 - CONSOLE field 752
 - DESKTOP field 752
 - desktop selection 752
 - INSTALL_64BIT_KERNEL field 238
 - software license agreement 359
- bosinst.data large_dumplv 278
- bosinst.data SIZE_GB 278
- bosinst.data, dump device 278
- buffers per CPU 494
- burn_cd command 375
- bzip2 command, Linux 753

C

- C++ Compiler, System V affinity 664
- C++, weak symbol support 664
- C7_cumwait
 - ATM 554
- C7_retry

- ATM 554
- C7_wait
 - ATM 554
- Ca_ES locale 797
- ca_ES.8859-15 ISO locale 791
- CA_ES.UTF-8 UTF-8 Locale 791
- cache file system 254
- CacheFS 254
- cachefs 254
- cachefslog command 254
- cachefswssize command 255
- callouts 769
 - events 770
 - prochr structure 770
 - PROCHR_TERMINATE 770
 - register 770
 - unregister 770
- cancellation cleanup handler, pthread library 16
- CAP_BYPASS_RAC_VMM, chuser 144
- CAPP/EAL4+
 - Installation steps 643
 - TCB, 64-bit kernel, JFS2 647
- Cascading Style Sheets 566
- case translation CD-ROM 260
- CD Studio 373
- cdcheck command 258
- CDE desktop, BOS installation 750
- cdeject command 259
- cdmount command 259
- cdrecord 373
- cdrfs 258
- cdrfs file system 258
- CD-ROM
 - automount facility 257
- CD-ROM mount 260
- cdromd daemon 257
- cdumount command 259
- cdutil command 259
- CDWrite Version 1.3 373
- CE 484
- Centralized logging 164
 - CSM 164
- Certificate Management Protocol, Java 570
- cfgmgr command 171, 284, 747
- cfgnsmb configuration method 531
- chargefee command 68
- chauthent command 661
- chcod command 132
- chcondition command 160

- check_core command 311
- Checksum offload
 - ATM 555
- checksum packet 490
- chfilt command 659
- chfn command
 - loadable module support 573
- chfs command 222
- chgaio fast path 21
- chgposixaio SMIT fastpath 21
- chgroup command
 - loadable module support 573
- chgrpmem command
 - loadable module support 573
- child 171
- child zones 451
- chinese characters 784
- chlicense command 364
- chlv command
 - passive MWCC 211
- chps command 378
- chpv command
 - hot spare disk support 187
- chresponse command 161
- chroot command 512
- chrsrc command 160
- chsh command
 - loadable module support 573
- chuser command
 - loadable module support 573
- chvg command
 - hot spare disk support 187
 - supporting different LTG sizes 196
- chvg command enhancements 212
- CIFS 526
- CIM
 - Interface Definition Language 28
 - Management Object Format 28
 - Object Manager 27
 - Schema 28
- CIMOM 27
- class
 - inheritance 43
- class assignment rules 47
- class definition, LDAP 591
- class name (WLM) 48
- classes 37
- classification process 45
- cloning
 - route 466
- cluster device reservation 172
- Cluster System Management (CSM) 162, 164
 - Configuration file manager (CFM) 163
 - Consumability 164
 - CSM Database 164
 - CSM HMC enhancements 164
 - Distributed shell 163
 - Domain Management 162
 - dsh command 163
 - EERM 163
 - Functionality 162
 - Hardware control 163
 - Hardware Control and Integration 164
 - Inter operability AIX and Linux 165
 - Probe manager 163
 - Remote console 163
 - rpower command 163
 - SNMP 165
 - software maintenance 164
- cmd
 - shd action 283
- cmdstat tools 399
- CMP, Java 570
- cn=hosts 588
- codeset 795
 - big5 789
 - Euro support 789
 - IBM-1046 789
 - IBM-1129 789
 - IBM-921 789
 - IBM-922 789
 - ISO8859-7 789
- collecting core information 310
- command
 - geninstall 734
- command tool 342
- commands
 - acctcom 68
 - adump 315
 - alog 151
 - alstat 397
 - alt_disk_install 372
 - aopt 409
 - autoconf, Linux 753
 - autoconf6 491
 - automake, Linux 753
 - bash2, Linux 753
 - bffcreate 365

bindintcpu 133
 bindprocessor 133
 bison, Linux 753
 bootlist 185
 bosboot 398
 bosdebug, HMT 769
 burn_cd 375
 bzip2, Linux 753
 cachefslog 254
 cachefswsize 255
 cdcheck 258
 cdeject 259
 cdmount 259
 cdutil 259
 cfgmgr 171, 284, 747
 chargefree 68
 chauthnt 661
 chcod 132
 chcondition 160
 check_core 311
 chfilt 659
 chfn 573
 chfs 222
 chgprmem 573
 chgroup 573
 chlicense 364
 chlv 211
 chps 378
 chpv 187
 chresponse 161
 chroot 512
 chsrc 160
 chsh 573
 chuser 573
 chvg 187, 196, 212
 compare_report 368
 configassist 347
 cpio 313
 crfs 222, 233
 curt 416
 cvs, Linux 753
 dbx 294
 dd 383
 defragfs 220
 devinstall 747
 df 254
 diag 265, 771
 diffutils, Linux 753
 dig 449, 456
 dnssec-keygen 447
 dnssec-signed 451
 dnssec-signzone 451
 du 221
 dumpcheck 276
 elm, Linux 753
 emacs, Linux 753
 emstat 397
 errdemon 264
 errpt 264
 expfilt 660
 export 588
 extendlv 216
 extendlv, performance improvement 214
 extendvg, performance improvement 214
 FDPR 409
 filemon 239, 397, 402
 fileplace 239, 396
 fileutils, Linux 753
 findutils, Linux 753
 flex, Linux 753
 fractrl 521
 fsck 234
 g++, Linux 753
 gawk, Linux 753
 gcc, Linux 753
 gdb, Linux 753
 gencopy 367, 738
 genfilt 659
 geninstall 740
 genkex 396
 genkld 396
 genld 397
 gennames 402, 416
 gensyms 416
 gettext, Linux 753
 ghostscript, Linux 753
 git, Linux 753
 glade, Linux 761
 grep, Linux 753
 guile, Linux 753
 gv, Linux 753
 gzip, Linux 753
 hostldif 585
 ifconfig down 477
 ike 541, 544
 ikedb 540, 653
 importvg 221
 indent, Linux 753

infocenter 355
install_wizard 741
installp 736, 740, 747
installp, software license agreement 359
instdev 747
inulag 358
iostat 185, 399–400
ipfilter 396
iptrace 493
joinvg 214
kdb 297
ksh93 25
ksysv, Linux 755
kuser, Linux 754
ledit command 145
less, Linux 753
libtool, Linux 753
lockstat 398
locktrace 398
logevent 151
logform 235
lppmgr 367
lquerypv 197
lsactdef 160
lsattr 387, 497, 551
lsaudrec 152, 155
lsauthent 662
lscondition 161
lscondresp 161
lsdev 496, 550
lsfilt 659
lsgroup 573
lslicense 364
lslv 210, 235
lsnf 257
lsnf, Linux 753
lspath 177
lsps 379
lsresponse 161
lsrset 52, 124, 142
lsrsrc 160
lsrsrcde 160
lsuser 572–573, 578
lsvg 197
lsvpd 132
lvostat 198, 207
m4, Linux 753
migratelp 198, 207
mkcd 218

mkcfsmnt 254
mkclass 44
mkcondition 160
mkcondresp 161
mkdev 185, 496
mkfs 222, 233
mkgroup 573
mkitab 257
mkkrb5clnt 576
mkkrb5srv 576
mklv, performance improvement 214
mkpath 174
mkramdisk 222
mkresponse 161
mkrr_fs 375
mkrsrc 160
mkseckrb5 576
mkuser 572–573, 578
mkvg 196, 208
mkvg, performance improvement 214
mobip6ctrl 504
mount 236, 256, 260, 531
mpage, Linux 753
mpcstat 552
ncftp, Linux 753
net share 526
netpmon 396
netstat 461, 490
netstat -C 478
nfso 424
nim 321
nimadm 319
no 424, 466, 469, 484, 488, 491, 563
notifyevent 151
nsupdate 452
ntpdate 442
od 687
passwd 573, 650
pax 277, 313
pprof 396
ps 253
python, Linux 753
qosadd 435, 437
qoslist 436
qosmod 436, 438
qosremove 436, 438
recreatevg 208
redefinevg 187
refresh 449

- refsrc 160
- rep-gtk, Linux 753
- restore 312
- restvg 372
- rmaudrec 155
- rmctrl 146
- rmcondition 160
- rmcondresp 161
- rmdev 185
- rmgroup 573
- rmpath 175
- rmresponse 161
- rmrsrc 160
- rmss 396
- rmuser 573
- rndc 443
- route 464, 473
- rpm 736
- rpm, Linux 756
- rsync, Linux 753
- sar 399
- schedtune 424
- sed, Linux 753
- shconf 281
- shrinkps 382
- shutdown 384
- sh-utils, Linux 753
- snap 277, 313
- snapcore 310
- splat 419
- splitvg 213
- stopcondresp 161
- stripnm 397
- su 650
- svmon 396, 402
- swapoff 378
- tar 277, 313, 315
- tar, Linux 753
- tcl/tk, Linux 753
- tcdevgrp 270
- tcsh, Linux 753
- textutils, Linux 753
- tg 273
- topas 397
- tprof 396, 409
- tprof command 416
- trace 15, 267, 275, 315, 418, 494
- transfig, Linux 753
- trcrpt 269, 272, 401

- truss 693
- udisetaup 747
- unmount 256
- unzip, Linux 753
- uudecode 384
- uuencode 383
- varyonvg 187
- vim, Linux 753
- vmstat 399–400
- vmtune 424
- vpdadd 23
- vpddel 23
- wallevent 151
- wget, Linux 753
- wlmmmon 71
- wlmpref 71
- wlmsat 71
- xfig, Linux 753
- xpdf, Linux 753
- zip, Linux 753
- zoo, Linux 753
- zsh, Linux 753
- commands (JFS2) 229
- commands impfilt 660
- commands lspp 369
- commands pstat 416
- Common Character Mode 775
- Common Information Model (CIM) 28
 - Installation 30
 - Logical information flow 29
 - Pegasus 28
- compare_report command 368
- comparison chart (JFS2) 224
- compatibility
 - NFS 228
 - workload manager 50
- compatibility (JFS2) 227, 229
- complex inode lock 220
- complex lock 769
- compound loadable module 574
- compression 225
- concurrent group 579
- condition
 - RMC concept 146
- condition variables, pthread 16
- Conditions plug-in 153
- conditions, system monitoring 156
- configassist command 347
- Configuration Assistant Task 347

- configuration change
 - QoS 433
 - RMC property 158
- Configuration file manager
 - CSM 163
- confset 88
- confsetcntrl command 89
- congestion 484–485
- Congestion Experienced 484
- Congestion Window Reduced 484
- CONSOLE field, desktop selection 752
- container 332
- contents area 330
- context switch
 - avoidance 21
 - BeginCriticalSection() 21
 - critical section 21
 - dispatcher 21
 - EnableCriticalSections() 21
 - multithreaded applications 22
- Control timeout multiplier
 - ATM 554
- Control timeout value
 - ATM 554
- Control timer
 - ATM 554
- Controlled access protection profile
 - Installation 642
- conv command, KDB 298
- core dump
 - collecting information 310
 - core file 308
 - CORE_NAMING variable 308
 - file naming 308
- CORE_NAMING, core file naming 308
- coredump() system call 277
- correspondent node, IPv6 502
- cost, regarding network routes 469
- cpio command 313, 666
- cpsd daemon 544
- CPU (topas) 408
- CPU Guard 134
 - UE-Gard 136
- CPU resource
 - workload manager 60
- CPU trace, snap 315
- CPU usage, tprof command 409
- CPUList 494
- create logical volume, mklv command 216

- create volume group, mkvg command 208
- cred pseudo file 685
- crfs command 222, 233
- cron daemon 385
- cron logfile 385
- cron logging 385
- crontab file 386
- Cryptographic Library 660
- cryptographic random number generation 651
- Cryptography Extension, Java 570
- cryptosecure hash generation 660
- CSM Database 164
- CSM HMC enhancements 164
- ctl pseudo file 685
- CUoD 131
- CuPath Class 173
- CuPathAt 174
- curt command 416
- custom tools 342
- cvs command, Linux 753
- CWR 484
- cyclic multiplexing 476

D

- Da_DK locale 797
- daemons
 - automountd 253
 - cdromd 257
 - cpsd 544
 - cron 385
 - dfpd 545
 - IBM.AuditRMd 160
 - IBM.ERrmd 160
 - IBM.FSrmmd 160
 - IBM.HostRMd 160
 - isakmpd 541, 544
 - kadmind 577
 - krb5kdc 577
 - named9 440
 - nsupdate9 440
 - qdaemon, enable debugging 726
 - rmcd 160
 - routed 467
 - rpc.statd 253
 - shdaemon 281
 - syslogd 492, 544
 - tmd 544
 - xmtrend 71

- yppasswordd 611
- data availability 251
- data heap, DSA allocation 12
- data management tunnel 654
- data segment 493
- date command 667
- db
 - identification module 574
- db library, Linux 753
- db.root file 442
- db_file.dhcpd library 296
- DBM, Berkeley 387
- DBX 292
 - print 292
 - tracebacks 307
- dbx command 294
- dcal command, KDB 298
- dd command
 - span option 383
- DDNS 454
- de_AT.8859-15 ISO locale 791
- DE_AT.UTF-8 UTF-8 Locale 791
- De_CH locale 797
- De_DE locale 797
- de_DE.8859-15 ISO locale 792
- DE_DE.UTF-8 UTF-8 Locale 792
- de_LU.8859-15 ISO locale 792
- DE_LU.UTF-8 UTF-8 Locale 792
- dead gateway detection 466
- debug_trace option, MPOA 548
- debugger
 - DBX 292
 - KDB 297
- debugging 544
- debugging for JetDirect backend 726
- debugging for qdaemon 726
- decryption 660
- dedicated dump device 277
- dedicated dump, ratio to memory 279
- default browser 349
- Default.Default class 69
- defragfs command 220
- defragmentation 225, 251
- defragmentation (JFS2) 226
- Defunct process harvesting 22
- deleting routes 465
- demo license 251
- DESKTOP field, bosinst.data file 752
- desktops

- BOS installation 750
- CDE, GNOME, KDE 750
- Linux 759–760
- NIM installation 752
- detecting an MPIO capable device 172
- development
 - FDPR 409
- device dependent fileset 31
- device discover 171
- device driver 168–169
- device independent fileset 31
- device reservation algorithm 171
- device reservation policy 171
 - NO_RESERVE 171
 - PR_EXCLUSIVE 171
 - PR_SHARED 172
 - SINGLE_PATH 171
- Devices
 - Version 5.2 support withdrawal 782
- devinstall command 747
- df command 254, 667
 - commands df 221
- dfmounts command 669
- DFP 545
- DFP Agent 545
- dfpd daemon 545
- dfshares command 668
- DGD 466
 - passive options 475
- dgd_packets_lost 473–474
- dgd_ping_time 473–474
- dgd_retry_time 474
- diag command 265, 771
- diag, diskette drive diagnostics 774
- diagnostics
 - link to error log 265
 - physicals location 774
- diagnostics, LS120 diskette drive 774
- Diffie-Hellman group 5 652
- diffutils command, Linux 753
- dig command 449, 456
- Digital certificate support, VPN 542
- dircmp command 669
- direct I/O, for NBC 516
- Direct Window Access 30
- Directory name lookup cache 218
- directory organization 225
- diskette diag 774
- diskette diagnostics 774

- diskIO resource 61
- dispgid command 670
- dispuid command 670
- Distributed shell
 - CSM 163
- DLPAR
 - CPU sparing 133
- dmp_del service 279
- DNLC
 - long filenames 218
 - Slab allocators 219
- DNS 439
 - starting 442
- DNS protocol, regarding BIND 439
- DNS security, regarding BIND 439
- DNSSEC 450
- dnssec-keygen command 447
- dnssec-signed command 451
- dnssec-signzone command 451
- Document Object Model 566
- Document Type Definition 653
- documentation library
 - performance monitor 397
- DOI 543
- domain
 - methods.cfg attribute 574
- Domain Management
 - CSM 162
- domain name service 439
- Domain of Interpretation 543
- DPCL, Dynamic Probe Class Library 20
- dsh command 163
- du command 221
- dual-semantic APIs 762
- dump
 - core file 308
 - KDB subcommand 299
- dump command, KDB 299
- dump device 277
- dump device bosinst.data 278
- dump facility 279
- dump, VPN function mapping 539
- dumpcheck command 276
- DUMPDEVICE bosinst.data 278
- dumpdir, VPN function mapping 539
- dumpfs command
 - JFS2 snapshot image 249
- DVD 372
- DVD-RAM

- automount facility 257
- DVD-ROM support 769
- DWA 30
- dynamic CPU deallocation 136
- dynamic CPU sparing 133
- dynamic data buffer cache 513
- dynamic DNS 440, 452
- Dynamic Feedback Protocol 545
- dynamic host configuration protocol 504
- Dynamic MTU design
 - ATM 555
- Dynamic Probe Class Library tool 20
- dynamic processor deallocation 134
- dynamic segment allocation 12

E

- EACCES error code 258
- EADS chip 287
- EAI_NONAME error 490
- ECN 484
- ECN Capable Transport 485
- ECN-capable 484
- ECN-capable router 485
- ECT 485
- EEH Enhanced I/O error handling 289
- EERM
 - CSM 163
- EIM 649
- el_GR locale, Euro support 789
- el_GR.ISO8859-7 ISO locale 792
- EL_GR.UTF-8 UTF-8 Locale 792
- electronic software license agreements
 - see also software license agreement 357
- elm command, Linux 753
- emacs command, Linux 753
- emstat command 397
- emulation 397
- en_BE.8859-15 ISO locale 791
- EN_BE.UTF-8 UTF-8 Locale 791
- En_GB locale 797
- en_HK locale 795
- en_IE.8859-15 ISO locale 791
- EN_IE.UTF-8 UTF-8 Locale 791
- en_PH locale 796
- en_SG locale 796
- En_US locale 797
- Enable system backups
 - Version 5.2 migration 329

- Enable trusted computing base
 - Version 5.2 migration 329
- EnableCriticalSections() system call 21
- enabling KDB, bosboot 297
- encryption 660
- EndCriticalSection() system call 21
- enhanced file system
 - JFS2 230
- Enhanced I/O Error Handling (EEH) 287
- Enhanced I/O error handling, EEH 289
- enhanced login privacy 649
- enhanced notification support, BIND 9 455
- enlightenment window manager, Linux 753
- Enterprise Identity Mapping 649
- entstat command, LANE trace 552
- environment variable
 - \$HOME 345
- environment variable LDR_CNTRL 13
- errdaemon 267
- errdemon command
 - duplicate error elimination 264
- errlog
 - shd action 283
- ERRM
 - see also event response resource manager 148
- error log
 - link to diagnostics 265
 - PCI FRU isolation 291
- Error log entry 25
- errpt command
 - intermediate output format 264
 - KDB subcommand 299
- errresume 266
- errsave 267
- es_BO locale 796
- es_CR locale 796
- es_DO locale 796
- es_EC locale 796
- Es_ES locale 797
- es_ES.8859-15 ISO locale 792
- ES_ES.UTF-8 UTF-8 Locale 792
- es_GT locale 796
- es_HN locale 796
- es_NI locale 796
- es_PA locale 796
- es_PY locale 796
- es_SV locale 796
- es_US locale 796
- ESP, VPN 540
- Et_EE locale, Euro support 789
- EtherChannel 556
 - network interface backup 556
- EtherChannel backup
 - Configuration 562
 - mkdev 563
- EtherChannel backup adapter 561
- Ethernet
 - interrupt saturation 767
- Ethernet adapters
 - vlan 563
- Ethernet interface backup, EtherChannel 556
- Euro support 789, 791
 - Ar_AA, locale 790
 - big5 789
 - el_GR locale 789
 - Et_EE locale 789
 - IBM-1046 789
 - IBM-1129 789
 - IBM-921 789
 - IBM-922 789
 - ISO8859-7 789
 - Lt_LT, locale 790
 - Lv_LV, locale 790
 - Vi_VN, locale 790
 - Zh_TW, locale 790
- European Monetary Union 791
- Evaluation assurance level 4+
 - Installation 642
- evaluation version 251
- event expression
 - RMC concept 147
- Event notification
 - Workload manager 86
- Event Response Resource Manager
 - command line interface 160
- event response resource manager 148
- events, system monitoring 155
- exclusive lock 526
- exec() system call 46, 49
- expfilt command 660
- Explicit Congestion Notification 484, 493
- export command 588
- extend logical volume , extendlv command 217
- Extended Hostname support 389
- extendlv command 216
- extendlv command performance improvement 214
- extendvg command performance improvement 214

F

- fail_over algorithm 180
- failover mode 171
- fast connect 525
 - lock 526
 - memory mapped 529
 - mode 526
 - search caching 528
 - sh_oplockfiles 526
 - sh_searchcache 526
 - sh_sendfile 526
 - user level security 527
 - user name mapping 526
 - Windows Terminal Server 528
- fast device configuration
 - cfgmgr command 284
- Fast Recovery algorithm 484
- Fast Retransmit algorithm 484
- fastpaths
 - chgaio 21
 - chgposixaio 21
- FAST storage subsystem 212
- FDPR command 409
- Fi_FI locale 797
- fi_FI.8859-15 ISO locale 791
- FI_FI.UTF-8 UTF-8 Locale 791
- fibre channel 261
- file descriptor limit, 5.2 enhancement 221
- file pages (vmstat) 400
- file size 225
- file system 258
 - block size 225
 - compatibility 227
 - defragmentation 226
 - directory organization 225–226
 - extent based addressing 225
 - file size 225
 - fragments 225
 - inline log 236
 - inode allocation 226
 - log device 235
 - NFS 228
 - quotas 225
 - RMC resource manager 158
 - size limit 225
 - udfs 258
 - variable block size 225
 - VFS 234
- filemon command 239, 397, 402
- fileplace command 239, 396
- filesets bos.sysmgt.trcgui_samp 273
- filesets installed, compare 369
- filesets modcrypt.base.includes 661
- filesets, remove 367
- fileutils command, Linux 753
- findutils command, Linux 753
- firewall hooks 523
- First Period 77
- fixed 49
- fixed license number 364
- flex command, Linux 753
- fnlib library, Linux 753
- focus (WLM) 59
- Font rasterizer 390
- fork 46
- format
 - log device 235
- Forward disconnect timer
 - ATM 554
- forwardcontrol, VPN function mapping 539
- Fr_BE locale 797
- fr_BE.8859-15 ISO locale 791
- FR_BE.UTF-8 UTF-8 Locale 791
- Fr_CA locale 797
- Fr_CH locale 797
- Fr_FR locale 797
- fr_FR.8859-15 ISO locale 791
- FR_FR.UTF-8 UTF-8 Locale 791
- fr_LU.8859-15 ISO locale 791
- FR_LU.UTF-8 UTF-8 Locale 791
- fragmentation (JFS2) 226
- fragmentation, TCP/IP 483
- fragments 225
- framework
 - web-based system manager 345
- FRCA 513
- fractrl command 521
- freehostent 487
- fsck command 234
 - JFS2 snapshot image 250
- fsdb command 249
- FSRM
 - see also file system resource manager 158
- FTP server enhancements, 5.2 509
- ftpaccess.ctl 509
- ftphearld.txt 509
- Functionality
 - CSM 162

Functionality of EEH
PCI FRU isolation 288

functions
as_att() 12
mmap() 12
nbc_vfs_flag 517
nbc_vno_flag() 517
nbc_vptosid 518
rmmmap_create() 12
setlocale() 792
functions of ECN 484
functions shmact() 12

G

g++ command, Linux 753
gated daemon 467
gateways, regarding networking 468
gathering core files 310
gawk command, Linux 753
GB blocks, file system 221
GB, mklv and extendlv command 216
GB18030
Unicode Extension B 794
UTF-32 794
GB18030, text editor 221
GBK, simplified chinese locale 784
gcc command, Linux 753
gcc compiler 294
gdb command, Linux 753
gecko layout engine 566
gencopy command 367, 738
genfilt command 659
geninstall command 734, 740
geninstall.summary file 749
genkex command 396, 411
genkld command 396, 411
genld command 397
gennames command 402, 416
gensyms command 416
getconf command 392, 670
getdev command 671
getgrp command 672
gethostbyaddr subroutine 486
getipnodebyaddr subroutine 486
getipnodebyname subroutine 486
getrusage() system call 20
gettext command, Linux 753
ghostscript command, Linux 753

GIF 343
git command, Linux 753
glade command, Linux 761
global lock, TCBHEAD_LOCK 484
GLX extension 30
GNOME desktop, BOS installation 750
GNOME desktop, Linux 753, 759
GNU Compiler, System V affinity 664
GNU software, compile and install 764, 812
graphic display adapter
common character mode 775
graphical framework, Linux 758
graphics adapters 31
graphics performance 30
grep command, Linux 753
group (WLM) 48
groups command 673
groups, IKE 536
GSSAPI 661
gtk+ library, Linux 753
GTX4000P 31
GTX6000P 31
guile command, Linux 753
gv command, Linux 753
gzip command, Linux 753

H

HACMP concurrent mode cluster 171
Hangul, Korean keyboard 793
Hanja, Korean keyboard 793
hardware
DVD-ROM support 769
Hardware control
CSM 163
Hardware Multithreading (HMT) 768
Hardware requirement
Version 5.2 migration 323
hardware topology 142
hash generation 660
hash tables 488
hcal command, KDB 298
heterogeneous environment 251
HKWD_LDR trace hook 275
HKWD_LDR_CHKPT trace hook 276
HKWD_LDR_ERR trace hook 276
HKWD_LDR_KMOD trace hook 276
HKWD_LDR_PROC trace hook 276
HMT 768

- home agent node, IPv6 502
- hookid 274
- host bus adapter 261
- Host overview plug-in 153
 - Delete a Process 154
 - File system utilization 153
 - Increase Paging Space 154
 - IP address 153
 - Paging space 153
 - Reconnect to RMC System 154
- host resource class 158
- host resource manager 158
- hostent structure 487
- HostRM
 - see also host resource manager 158
- hostslidif command 585
- HOT SPARE
 - lsvg output field 197
- HTTP 1.1 520
- HTTP GET kernel extension 518
- HTTP, split-connection proxy system 481

I

- IBM 2216 547
- IBM HTTP Server 347, 513
- IBM SecureWay Directory 584
- IBM Web 356
- IBM.AuditRMd
 - RMC daemon 160
- IBM.ERrmd
 - RMC daemon 160
- IBM.FSrmd
 - RMC daemon 160
- IBM.HostRMd
 - RMC daemon 160
- IBM.WLM
 - Workload manager 86
- IBM-1046, Euro support 789
- IBM-1129, Euro support 789
- IBM-850 codeset 797
- IBM-921, Euro support 789
- IBM-922, Euro support 789
- ICMP6_FILTER 489
- iconv command 794
- id_ID locale 796
- IDL, CIM 28
- IEEE 802.1Q 563
- IEEE 802.3 format, MPOA 550

- ifconfig down command 477
- IHS 347
- IKE 536, 544
 - groups 536
 - import/export, Linux 538
 - ipsec.conf, Linux 539
 - logging 544
 - negotiation 544
 - policies 536
 - wild cards 536
- ike command 541, 544
- ikedb command 540, 653
- ikelifetime, VPN function mapping 540
- IKETransform 653
- image file, rename 365
- IMCPV6 messages 490
- IME, input methods 784
- IMNSearch
 - text search engine 354
- impfilt command 660
- import Linux IPSEC 538
- Import user volume groups
 - Version 5.2 migration 329
- import volume groups (JFS2) 227
- importvg command 221
- in-addr.arpa 445
- inbound_fw 524
- inbound_fw_free_args, firewall hooks option 525
- inbound_fw_save_args, firewall hooks option 525
- incremental zone transfer 440
- incremental zone transfers 454
- indent command, Linux 753
- indirect block, JFS 219
- indirect mode 30
- infocenter command 355
- Information Center 355
- informational exchange 543
- Initial control timeout
 - ATM 554
- inline log 236
- inode (JFS2) 226
- inode command, KDB 299
- inode lock
 - INODE_UNLOCK 220
 - IREAD_LOCK 220
 - ISIMPLE_LOCK 220
 - ISIMPLE_UNLOCK 220
 - IWRITE_LOCK 220
- inode lock, complex 220

- INODE_UNLOCK() macro 220
- inodes 219, 225
- inpcb hash tables 488
- input methods 784
 - BiaoXing Ma 786
 - Intelligent ABC 785
 - Internal code 788
 - PinYin 787
 - Zheng Ma 786
- inst_status 749
- install RPM package 758
- Install Shield Multi Platform 747
- INSTALL_64BIT_KERNEL field, bosinst.data file 238
- install_lwcf_handler() system call 307
- install_wizard command 741
- Installation
 - Common Information Model (CIM) 30
- installation image 742
- Installation settings 745
- Installation wizard 740
- installation, desktop selection 750
- installp command 736, 740, 747
- installp command, software license agreement 359
- installp.summary 749
- InstallShield Multi-Platform 349, 734, 740
- instdev command 747
- Instrumentation Systems and Automation Society (ISA) 776
- INT_MAX 463
- integrated local management interface 553
- Intelligent ABC Input Method 785
- Inter operability
 - CSM 165
- interface backup, EtherChannel 556
- Interface Definition Language, CIM 28
- interface specific routes 464
- interfaces
 - load balancing 461
 - VPN function mapping 539
- Internal code Input method 788
- international locales, Euro support 790
- Internet Engineering Task Force 505
- Internet Key Exchange 544
 - command line interface 537
- Internet key exchange 652
- interrupt saturation 767
- inulag command 358
- invoking the install_wizard 741
- io
 - shd detection scheme 284
- ioctl system call
 - IOCINFO 196
- ioo command 241
- iostat command 185, 399–400
- IP fragmentation, MPOA 548
- IP packet filtering, VPN 542
- IP security, start/stop 542
- IP_FINDPMTU socket option 466
- ip_filtr_in_hook 524
- ip_len field 549
- ip_output, firewall hooks option 524
- ip_output_post_fw, firewall hooks option 525
- ipfilter command 396
- ipforwarding, no option 475
- ipintr_noqueue, firewall hooks option 524
- ipintr_noqueue_post_fw, firewall hooks option 524
- ipintr_noqueue2, firewall hooks option 524
- IPSEC, import/export 538
- iptrace command 493
- iptrace log file 493
- IPv4 Protocol 486
- IPv4-compatible address 486
- IPv4-mapped IPv6 address 486
- IPv6
 - 5.2 enhancements 456
 - BIND support 439
 - IPv6 Protocol 486
 - IPv6, regarding dead gateways 469
 - IPV6_CHECKSUM 489
 - IREAD_LOCK() macro 220
 - ls_IS locale 797
 - ISA adapter
 - Version 5.2 support withdrawal 779
 - ISAKMP 536, 543
 - isakmp_events 544
 - isakmpd daemon 541, 544
 - ISIMPLE_LOCK() macro 220
 - ISIMPLE_UNLOCK() macro 220
 - ISMP 349, 734, 740, 747
 - ISO8859-7, Euro support 789
 - It_IT locale 797
 - it_IT.8859-15 ISO locale 792
 - IT_IT.UTF-8 UTF-8 Locale 792
 - IWRITE_LOCK() macro 220
 - IXFR 454

J

- Java 330
 - Cryptography Extension 570
 - installed version 27
 - Secure Sockets Extension 570
 - security 570
- Java 1.3 339
- Java Profiler Agent
 - tprof enhancement 412
- Java profiling, tprof command 410
- Java Virtual Machine Profiling Interface 410
- JCE 570
- JetDirect backend, enable debugging 726
- JFS 251
 - .indirect 219
 - DNLC 218
 - mount option 219
- jfs_vnc_enter 239
- jfs_vnc_init 239
- jfs_vnc_lookup 239
- jfs_vnc_purge 239
- jfs_vnc_remove 239
- JFS2 224
 - 64-bit kernel 237
 - BOS installation 237
 - CAPP/EAL4+ 647
 - DNLC 218
 - filemon command 239
 - fileplace command 239
 - inline log 236
 - log device 235
 - migration installation 237
 - overwrite installation 237
 - reserved heuristic 240
 - rootvg support 236
 - vnode cache 239
- JFS2 large file system enablement 240
- JFS2 Performance 240
- JFS2 performance
 - ioo command 241
 - j2_maxRandomWrite 241
 - minfree 241
 - snapshot map 243
 - syncd 241
 - vmo command 241
 - vmstat command 241
 - vmtune command 241
- JFS2 snapshot backup 250
- logredo command 250

- JFS2 snapshot image 242
 - backsnap command 249
 - Creation
 - Web-based System Manager 244
 - defragfs command 250
 - dumpfs command 249
 - fsck command 250
 - fsdb command 249
 - Functionality 242
 - mount command 249
 - snapshot command 249
- JFS2LOG 235
- JISX0213 799
- joinvg command 214
- journaled file system
 - JFS2 224
- journaling 251
- JSSE 570
- JVMPI 410

K

- kadmind
 - Kerberos admin daemon 577
- KB, mklv and extendlv command 216
- KBD
 - enable debugger 297
- KDB
 - alias command 298
 - bosboot command 297
 - clear variables 300
 - conv command 298
 - dcal command 298
 - dump 299
 - dump command 299
 - errpt command 299
 - hcal command 298
 - inode command 299
 - link command 298
 - list variables 300
 - lke command 299
 - mbuf command 299
 - netm command 299
 - p command 299
 - proc command 300
 - set \$repeat command 298
 - set scroll command 298
 - sock command 300
 - sr64 command 300

- status command 300
- th command 300
- thread command 300
- varlist command 300
- varrm command 300
- kdb 301
 - ndd subcommand 302
 - netstat subcommand 302
 - output redirection facility 301
 - print subcommand 302
 - route sub-command 303
 - rtentry sub-command 305
 - rxnode sub-command 305
 - set edit subcommand 301
 - set logfile subcommand 301
 - set loglevel subcommand 301
 - symptom subcommand 302
 - trcstart and trcstop subcommands 307
 - which subcommand 302
- kdb command 297
- kdb di subcommand 301
- KDE desktop, BOS installation 750
- KDE desktop, Linux 753, 760
- keep alive enhancements 488
- Kerberos 661
- Kerberos attributes 578
- Kerberos Version 5 575
- kernel debugger
 - KDB 297
- kernel extension 168
- kernel locks 398
- kernel service
 - proch_reg() 770
 - proch_unreg() 770
 - prochadd() 769
 - prochdel() 769
- kernel service dmp_ctl 280
- kernel tuning 387
- kernel_lock 769
- key management tunnel 653
- keyboard
 - Euro support 790
 - Hangul, Korean 793
 - Hanja, Korean 793
 - Korean support 793
- keyexchange, VPN function mapping 540
- keyingtries, VPN function mapping 540
- keylife, VPN function mapping 540
- klipsdebug, VPN function mapping 539

- KRB5files
 - Kerberos registry value 578
- krb5kdc
 - Kerberos KDC daemon 577
- ksh93 command 25
- ksysv command, Linux 755
- kuser command, Linux 754

L

- LAN Emulation Clients 553
- LANE2
 - ATM 554
- language environment, Euro support 789
- large data type support 12
- large file enabled
 - cachefs 254
- Large page 143
- large page support
 - svmon command 406
- large_dump bosinst.data 278
- large_dumplv, bosinst.data 278
- last command 673
- Last Period 77
- launch plug-ins 334
- LDAP 583
 - audit events 590
 - audit plug-in 589
 - audit service 592
 - class definition 591
 - security plug-in 589
- LDAP, sendmail 386
- ldd command 674
- ldedit command 145
- LDIF 585
- LDR_CNTRL environment variable 13–14
- LDR_CNTRL=LARGE_PAGE_DATA variable 145
- LE client 547
- left, VPN function mapping 539
- leftfirewall 540
- leftfirewall, VPN function mapping 540
- leftid, VPN function mapping 540
- leftnexthop, VPN function mapping 540
- leftsasigkey, VPN function mapping 540
- leftsubnet, VPN function mapping 540
- leftupdown, VPN function mapping 540
- less command, Linux 753
- LFS
 - DNLC 218

- lg_dumplv, dump device 278
- libcrypto.a library 445
- libdns_secure.a library 445
- libjpeg library, Linux 753
- LIBPATH environment variable, Perl 26
- libpng library, Linux 753
- library routines 486
- libtiff library, Linux 753
- libtool command, Linux 753
- license agreements
 - see also software license agreement 357
- lightweight core file support 307
- limit log file 493
- limitations
 - multipath routing 465
- Limited Transmit algorithm 493
- limited_transmit 493
- Limits on class resources
 - Workload manager 94
- limits.h 221
- linkcommand, KDB 298
- Linux 735
 - APIs 762
 - autoconf command 753
 - automake command 753
 - bash2 command 753
 - bison command 753
 - bzip2 command 753
 - cvs command 753
 - db library 753
 - desktops 759–760
 - diffutils command 753
 - elm command 753
 - emacs command 753
 - enlightenment window manager 753
 - fileutils command 753
 - findutils command 753
 - flex command 753
 - fnlib library 753
 - g++ command 753
 - gawk command 753
 - gcc command 753
 - gdb command 753
 - gettext command 753
 - ghostscript command 753
 - git command 753
 - glade command 761
 - GNOME desktop 753, 759
 - graphical framework 758
 - grep command 753
 - gtk+ library 753
 - guile command 753
 - gv command 753
 - gzip command 753
 - import/export VPN 538
 - indent command 753
 - install RPM package 758
 - IPSEC import/export 539
 - KDE desktop 753, 760
 - ksysv command 755
 - kuser command 754
 - less command 753
 - libjpeg library 753
 - libpng library 753
 - libtiff library 753
 - libtool command 753
 - ls of command 753
 - m4 command 753
 - mpage command 753
 - ncftp command 753
 - ncurses library 753
 - python command 753
 - qt library 753
 - readline library 753
 - rep-gtk command 753
 - rpm command 756
 - RPM database 756
 - rsync command 753
 - sawfish window manager 753
 - sed command 753
 - sh-utils command 753
 - slang library 753
 - source affinity 762
 - system management 754
 - tar command 753
 - tcl/tk command 753
 - tcsh command 753
 - textutils command 753
 - transfig command 753
 - unzip command 753
 - vim command 753
 - Web-based System Manager client 352
 - wget command 753
 - xfig command 753
 - xpdf command 753
 - zip command 753
 - zoo command 753
 - zsh command 753

- Linux affinity
 - lsconf command 776
- Linux inter operability
 - CSM 165
- listdgrp command 674
- lke command, KDB 299
- ln command 675
- load balance mode 172
- load balancing
 - gateways 461
- loadable module
 - auth option 574
 - authentication 573
 - compound 574
 - db option 574
 - files 571
 - identification 573
- loader trace hooks 416
- locale support 795
- locales
 - obsolete 797
- location codes 774
- lock 526
 - exclusive 526
 - multiple read/write 20
 - opportunistic 526
 - pthread_cond_t type 19
 - pthread_mutex_t type 19
 - pthread_rwlock_t type 19
 - pthreads 16
 - rec_mutex type 19
 - spinlock_t type 19
- lock tracing 398
- lockstat command 398
- locktrace command 398
- log
 - Web-based System Manager 340
- log device 235
- log file of syslog 545
- LOG_KERN facility 492
- logevent command
 - RMC command 151
- logform command 235
- logging level 544
- logging protocol 544
- logical CPU numbers 124
- Logical Volume Manager 251–252
- logical volume serialization 215
- logical volume, extendlv command 217
- logical volume, mklv command 216
- login
 - shd action 283
- login.cfg, login security 650
- logins command 675
- logredo command 250
- long filenames
 - DNLC 218
- lost packets, networking 471
- lppmgr command 367
- lppsourc 321
- lquerypv
 - LTG size 197
- LS120 diskette drive 774
- lsactdef command 160
- lsattr command 497
- lsattr command, MPOA 551
- lsattr command, NCARGS 387
- lsaudrec command 155
 - RMC command 152
- lsauthent command 662
- lscondition command 161
- lscondresp command 161
- lsconf command
 - Linux affinity 776
- lsdev command 496
- lsdev command, MPOA 550
- lsfilt command 659
- lsgrupp command
 - loadable module support 573
- lslicense command 364
- lspp command 369
- lsiv command
 - JFS2 235
 - passive MWCC 210
- lsiof command 257
- lsiof command, Linux 753
- lspath command 177
- lsps command 379
- lsresponse command 161
- lsrset command 52, 124, 142
- lsrsrc command 160
- lsrsrcde command 160
- lsuser command
 - DCE user 572
 - Kerberos user 578
 - loadable module support 573
- lsvg command
 - new options 197

- lsvpd command 132
- lswlmconf command 89
- Lt_LT locale, Euro support 789
- LTG
 - support for different sizes 196
- LTG size
 - lsvg output field 197
- lun 168
- LUN, increasing 212
- LV, extendlv command 218
- Lv_LV locale, Euro support 789
- LVM
 - hot-spot management 198
 - RAID support 211
 - thread-safe liblvm.a 211
- LVM (JFS2) 228
- lvmstat command 198, 207
- lwcf 307
 - install_lwcf_handler() 307
 - mt_trce() 307
- lwp pseudo file directory 686
- lwpcpl pseudo file 687
- lwpsinfo pseudo file 687
- lwpstatus pseudo file 687

M

- m4 command, Linux 753
- mach command 676
- macros
 - INODE_UNLOCK() 220
 - IREAD_LOCK() 220
 - ISIMPLE_LOCK 220
 - ISIMPLE_UNLOCK() 220
 - IWRITE_UNLOCK 220
- macros_AIO_AIX_SOURCE 20
- mail, sendmail 386
- maintenance level 36
- MakeDisc Version 1.3-Beta2 373
- malloc 15
- MALLOCBUCKETS 15
- MALLOCDEBUG variable 15
- MALLOCMULTIHEAP 14
- manage AIX from PC 351
- Manage certificates, VPN 542
- managed machine 347
- Management Object Format, CIM 28
- management tool 233
- manual assignment (WLM) 45

- Manual offline
 - tprof enhancement 413
- map pseudo file 685
- Matsushita LF-D291 373
- MAX_RT_COST 463
- MAX_RT_COST setting 469
- maxdata, very large program support 13
- MB blocks, file system 221
- MB, mkiv and extendlv command 216
- mbuf command, KDB 299
- mbufs 299
- MCA packages
 - Version 5.2 support withdrawal 779
- MCA platform SP systems
 - Version 5.2 support withdrawal 781
- MCA platform systems
 - Version 5.2 support withdrawal 780
- MCM 142
- memory address space 30
- Memory affinity 428
- memory resource
 - workload manager 61
- memory, dump device size 277
- MEMORY_AFFINITY environment variable 142
- message of the day, FTP 509
- message submission agent 386
- methods.cfg
 - domain attribute 574
 - options attribute 574
 - program attribute 574
 - program_64 attribute 574
- Micro profiling
 - tprof enhancement 410
- Microchannel Bus Architecture (MCA) 776
- migratelp command 198, 207
- migration (JFS2) 228
- migration install, software license agreement 362
- migration installation, JFS2 237
- Migration steps
 - Version 5.2 324
- millicode functions 771
- minfree
 - JFS2 performance 241
- mkcd command 218
- mkcdimg Version 2.0 373
- mkcfsmnt command 254
- mkcifs_fs boot time script 531
- mkclass command 44
- mkcondition command 160

- mkcondresp command 161
- mkdev
 - EtherChannel backup 563
- mkdev command 185, 496
- mkfs command 222
 - JFS2 233
- mkgroup command
 - loadable module support 573
- mkisofs 373
- mkkitab command 257
- mkkrb5clnt command 576
- mkkrb5srv command 576
- mklv command performance improvement 214
- mkpath command 174
- mkramdisk command 222
- mkresponse command 161
- mkroute, SMIT panel 476
- mkrr_fs command 375
- mkrrsrc command 160
- mkseckrb5 command 576
- mksysb
 - DVD 372
- mkuser command
 - DCE user 572
 - Kerberos user 578
 - loadable module support 573
- mkvg command 208
 - supporting different LTG sizes 196
- mkvg command performance 214
- mmap() function call 12
- mnemonics
 - accessibility for Web-based System Manager 353
- Mobile IPv6 502
- mobile node, IPv6 502
- mobip6ctrl command 504
- modcrypt.base.includes fileset, crypto 661
- mode 526
- modular exponentiation group primes 653
- MOF, CIM 28
- monitor
 - RMC concept 146
- monitoring 340
- monitoring, system monitoring 153
- mount 219
- mount command 256, 531
 - JFS2 snapshot image 249
- mount command, CD-ROM
 - nocase option 260
 - upcase option 260
- mount option
 - mind 219
 - nomind 219
- mount_cifs mount helper 531
- Mozilla 1.0.1 566
- mpage command, Linux 753
- mpcstat command, LANE trace 552
- MPIO 168
 - adapter 170
 - algorithm 171
 - bootlist command 185
 - child 171
 - cluster 172
 - concepts 169
 - detecting an MPIO capable device 172
 - device 170
 - device discover 171
 - device driver 168
 - device reservation policy 171
 - NO_RESERVE 171
 - PR_EXCLUSIVE 171
 - PR_SHARED 172
 - SINGLE_PATH 171
 - fail_over algorithm 180
 - failover mode 171
 - HACMP concurrent mode cluster 171
 - kernel extension 168
 - load balance mode 172
 - lspath command 177
 - lun 168
 - mkdev command 185
 - mkpath 174
 - ODM 170
 - parent 170, 184
 - path 168, 171
 - path disable 178
 - path enable 178
 - path management 169
 - path missing 178
 - PCM ODM 172
 - PCMKE 168, 172
 - PCMRTL 168
 - PR_SHARED 172
 - rmdev command 185
 - rmpath command 175
 - round_robin algorithm 180
 - run-time loadable configuration module 168
 - SCSI scsd 169

- SDD 169
- SINGLE_PATH 171
- Unique device identifier 171
- volume group 170
- MPIO path defined 178
- MPIO path detected 179
- MPIO path failed 178
- MPOA 548
 - client 547
 - commands
 - lsattr 551
 - lsdev 550
 - debug tracing 552
 - debug_trace option 548
 - ICMP 550
 - IEEE 802.3 Ethernet format 550
 - IP fragmentation 548
 - IPv4 support 548
 - layer 549
 - network 549
 - Resolution Reply 549
 - shortcut 549
 - Standard Ethernet format 550
 - Token Ring 552
 - token ring format 550
 - trace 548
- ms_MY locale 796
- mt_trce() system call 307
- MTU size 465
- Multi function adapter
 - PCI FRU isolation 288
- multiheap malloc 14
- multipath I/O 461
- multipath routing 460
- multiple segments
 - JFS 219
- multiple volume dd 383
- Multiprotocol over ATM 548
- multithreading 253
- mutex, pthreads 16
- MWCC 209
 - active 210
 - passive 210

N

- name resolution
 - LDAP 583
- named.conf network configuration file 441

- named9 daemon 440
- Nano profiling
 - tprof enhancement 412
- NAS client 662
- National Language Support 783
- NBC, network buffer cache 513
- nbc_vfs_flag function 517
- nbc_vno_flag() function 517
- nbc_vnode_in_dev() function 517
- nbc_vptosid() function 518
- NCARGS parameter 387
- ncftp command, Linux 753
- ncurses library, Linux 753
- net share command 526
- netgroup, file 255
- netgroup, NFS 255
- netm command, KDB 299
- netmasks 461
 - DGD 475
- netpmon 401
- netpmon command 396
- NetQuestion
 - text search engine 354
- Netscape Web browser 566
- netstat -C command 478
- netstat command 461, 490
- network
 - vlan 563
- network adapter resource class 158
- network addresses
 - virtual IP 495
- Network Buffer Cache 516
- Network Installation Client Tasks 741
- network interface backup
 - configuring 556
 - interface 559
 - supported adapters 556
- network option 492
 - limited_transmit 493
 - RFC2414 492
 - sodebug 492
 - tcp_ecn 493
 - tcp_init_window 492
 - tcp_newreno 492
- network option tcp_ecn 484
- networking enhancements 431
- NFS 228, 256
 - statd multithread 253
- NFS cache 255

- NFS, CacheFS 254
- NFS, netgroup 255
- nfso command 424
- nim command 321
- NIM installation
 - bosinst.data file, desktop selection 752
 - desktop selection 752
 - INSTALL_64BIT_KERNEL field 238
 - JFS2 support 238
- NIM Machines 741
- NIM master 741
- NIM, install_wizard 740
- nimadm command 319
- NIS maps 611
- NIST Advanced Encryption Standard 660
- NL_BE locale 797
- nl_BE.8859-15 ISO locale 791
- NL_BE.UTF-8 UTF-8 Locale 791
- NL_NL locale 797
- nl_NL.8859-15 ISO locale 791
- NL_NL.UTF-8 UTF-8 Locale 791
- NLS 783
 - GB18030 794
 - iconv command 794
 - Unicode extension B 793
 - Universal UCS Converter 794
- no command 424, 466, 469, 484, 488, 491
- no command option
 - vlan 563
- No_NO locale 797
- NO_RESERVE 171
- nocase mapping 260
- nocase translation 260
- nodename-to-address translation 486
- Notification payload 543
- Notify payload 543
- Notify, regarding BIND 440
- notifyevent command
 - RMC command 151
- Novell Directory Services 505
- NS records 455
- nsmbdd device driver 531
- NSORDER environment variable 588
- nsupdate command 452
- nsupdate9 daemon 440
- ntpdate command 442
- Numeric Input Method 798
- NVRAM 267
- NXDOMAIN error 458

O

- object classes
 - LDAP 584
- Object Manager, CIM 27
- object pseudo file directory 686
- obsolete locales 797
- od command 687
- ODM 170
- ODM CuPath Class 173
- ODM CuPathAt Class 174
- ODM PdAt class 174
- ODM PdPathAt Class 173
- odm_run_method 748
- Online management 251
- online support 355
- Online volume management 251
- OpenGL 30
- OpenSSH library 445, 619
- OpenSSL RPM 445, 619
- OpenType font 390
- openx(), automount facility 258
- operational state
 - RMC property 158
- opportunistic locks 526
- options
 - methods.cfg attribute 574
- outbound_fw 524
- outbound_fw_free_args, firewall hooks option 525
- outbound_fw_save_args, firewall hooks option 525
- overlap, block size 384
- overview plug-ins 333

P

- p command, KDB 299
- P2SC
 - Version 5.2 support withdrawal 777
- package format 735
- package list 748
- packages DWA 31
- packet capture library 522
- paging device resource class 158
- paging space
 - deactivation 378
 - decreasing size 380
- paging space, dump device 277
- PAM 614
- Panasonic Cw-7502-B 373
- parallel jobs 307

- parallel operating environment 423
- parameters of the getipbynodeaddr 486
- parameters of the getipnodebyname 486
- parent 170–171
- parent device 184
- passive dead gateway detection 468
- passive mode (WLM) 36
- passive MWCC 210
- passive_dgd 474
- passive_dgd field 471
- passwd command 650
 - loadable module support 573
- password prompt, login security 650
- path 168, 171
- path defined 178
- path detected 179
- path disable 178
- path enable 178
- path failed 178
- Path management 174
- path management 169
- path missing 178
- pax command 277, 313
- PC Client 346
- PCI bridge adapters
 - PCI FRU isolation 288
- PCI FRU isolation 287
 - adapter types 288
 - error log entry 291
 - Functionality of EEH 288
- PCM ODM 172
- PCMCIA 771
- PCMKE 168, 172, 174
- PCMRTL 168
- PdAt class 174
- PdPathAt Class 173
- Pegasus
 - Common Information Model (CIM) 28
- per share 526
- PercentTotUsed
 - RMC monitored property example 147
- perfagent 397
- perfagent.tools fileset 396
- performance
 - alignment 397
 - emulation 397
 - fast connect 528
 - FDPR 409
 - iostat 185, 400
 - svmon 402
 - vmstat 400
- performance monitor API 397
- performance monitor topas 406
- performance toolbox (WLM) 36
- performance tools 396
 - curt 416
 - splat 419
- perfstat APIs 421
- perfstat_alloc API 422
- perfstat_cpu API 422
- perfstat_cpu_total API 422
- perfstat_disk API 422
- perfstat_disk_total API 422
- perfstat_memory_total API 422
- perfstat_netinterface API 422
- perfstat_netinterface_total API 422
- perfstat_pagingspace API 422
- perfstat_protocol API 422
- Perl 5.6 26
- permissions, root file system 218
- Personal Computer Memory Card International Association (PCMCIA) 776
- petabytes 225
- pfs, VPN function mapping 540
- physical volume resource class 158
- pid_t ue_proc_register() function call 135
- pid_t ue_proc_unregister() function call 135
- ping, regarding DGD 472
- PinYin input method 787
- piohpnf, enable debugging 727
- pkgfile 747
- pkgname 748
- PKI 570
- plock 49
- plock() system call 49
- Pluggable Authentication Mechanism 614
- plug-in 331
- pluto, VPN function mapping 539
- plutobackgroundload, VPN function mapping 539
- plutodebug, VPN function mapping 539
- plutoload, VPN function mapping 539
- plutostart, VPN function mapping 539
- plutowait, VPN function mapping 539
- PMTU 465
- policies, IKE 536
- policies, networking 432
- policy agent daemon 436
- policyd.conf 438

- POSIX compliant AIO 20
- Post processing
 - tprof enhancement 413
- postpluto, VPN function mapping 539
- power failure,error logging 267
- Power PC 224
- Power1
 - Version 5.2 support withdrawal 777
- Power2
 - Version 5.2 support withdrawal 777
- PP size, mkvg command 208
- PP, extendlv command 218
- pp_login
 - shd parameter 283
- pprof command 396
- PR_EXCLUSIVE 171
- PR_SHARED 172
- preferences 345
 - web-based system manager 339
- PReP functions
 - Version 5.2 support withdrawal 777
- PReP PCI adapter
 - Version 5.2 support withdrawal 777
- PReP platform systems
 - Version 5.2 support withdrawal 780
- PReP specific ISA adapters
 - Version 5.2 support withdrawal 778
- PReP Support
 - Version 5.2 support withdrawal 777
- prepluto, VPN function mapping 539
- print
 - DBX subcommand 292
- prio
 - shd detection scheme 281–282
- PRNG 651
- Probe manager
 - CSM 163
- problem determination
 - parallel jobs 307
- proc command, KDB 300
- proc pseudo file system
 - as pseudo file 685
 - cred pseudo file 685
 - ctl pseudo file 685
 - lwp pseudo file directory 686
 - lwpctl pseudo file 687
 - lwpsinfo pseudo file 687
 - lwpstatus pseudo file 687
 - map pseudo file 685
 - object pseudo file directory 686
 - psinfo pseudo file 685
 - sigact pseudo file 685
 - status pseudo file 685
 - sysent pseudo file 685
 - vfs entry 684
- procrcd command 689
- process accounting 68
- process type (WLM) 49
- processor resource class 158
- procfiles command 689
- procflags command 689
- Proch callouts 769
- proch_reg() kernel service 770
- proch_unreg() kernel service 770
- prochadd() kernel service 769
- prochdel() kernel service 769
- prochr structure 770
- PROCHR_TERMINATE, event type 770
- procldd command 690
- procmap command 690
- procrun command 692
- procsig command 690
- procstack command 691
- procstop command 691
- proctools 688
- proctree command 692
- procwait command 692
- procwdx command 689
- Profiling
 - tprof enhancement 410
- profiling for Java applications 410
- program
 - methods.cfg attribute 574
- program resource class 158
- program_64
 - methods.cfg attribute 574
- programming
 - DBX 292
- protocol
 - CIFS 526
 - NFS 228, 253
 - smb 525
 - SNMP 345
- proxy daemon 544
- proxy systems, split-connection 481
- prtconf command 776
- ps command 677
- pseudo file directories 686

- pseudo files 685, 687
- pseudo-random number generator 651
- psinfo pseudo file 685
- pstat command 416
- Pt_PT locale 797
- pt_PT.8859-15 ISO locale 792
- PT_PT.UTF-8 UTF-8 Locale 792
- pthdb_atfork() function 16
- pthdb_atfork_arg() function 16
- pthdb_atfork_child() function 16
- pthdb_atfork_parent() function 16
- pthdb_atfork_prepare() function 16
- pthdb_atfork_type() function 16
- pthdb_cleanup() function 16
- pthdb_cleanup_arg() function 16
- pthdb_cleanup_func() function 16
- pthread
 - condition variables 16
 - debug library 15
 - library call 20
 - lock 19
 - lock attributes 16
 - multiple read/write locks 20
 - mutex attributes 16
 - pthdb_atfork 16
 - pthdb_atfork_arg 16
 - pthdb_atfork_child 16
 - pthdb_atfork_parent 16
 - pthdb_atfork_prepare 16
 - pthdb_atfork_type 16
 - pthdb_cleanup 16
 - pthdb_cleanup_arg 16
 - pthdb_cleanup_func 16
 - pthdb_pthread_owner_resource() 19
 - pthdb_pthread_waiter_resource() 19
 - pthdb_resource_handle() 19
 - pthdb_resource_type() 19
 - pthread_rwlock_t data type 20
 - unregister atfork handler 16
- pthread_atfork() system call 16
- pthread_atfork_np() system call 16
- pthread_atfork_unregister_np() system call 16
- pthread_getusage_bp library call 20
- PTX 397
- public key certificates, PRNG 651
- Public Key Infrastructure 570
- publications 355
- Public-Key Cryptography Standards, PKCS 571
- pwck command 677

- python command, Linux 753

Q

- qdaemon, enable debugging 726
- QoS 432
- QoS manager 436
- qosadd command 435, 437
- qoslist command 436
- qosmod command 436, 438
- qosremove command 436, 438
- qt library, Linux 753
- quality of service 432
- QuanPin input method 787
- query RPM database, Linux 758
- quot command 678
- quotas 225

R

- RAID 251
- RAID 0 251
- RAID 0+1 251
- RAID 1 251
- RAID 1+0 251
- RAID 5 251
- RAID support 211
- ramdisk 222
- random device 652
- random number generator 651
- RAS
 - check_core command 311
 - core file naming 308
 - inline log 236
 - isakmpd daemon 541
 - JetDirect, enable debugging 726
 - qdaemon, enable debugging 726
 - snapcore command 308
- Rasterizer, font 390
- rcmds 661
- readline library, Linux 753
- Real time
 - tprof enhancement 412
- realm
 - Kerberos Version 5 configuration 577
- rearm expression
 - RMC concept 147
- reboot
 - shd action 284
- Reconnect 154

- recording preferences 71
- recreatevg command 208
- Redbooks Web site 814
 - Contact us xxxix
- redefinevg command 187
- RedHat Package Manager 734, 740, 756
- refresh command 449
- refsrc command 160
- register callouts 770
- registry
 - user attribute 572
- rekeyfuzz, VPN function mapping 540
- remote access servers 506
- Remote console
 - CSM 163
- remote hostname 389
- remote system administration 346
- Reno algorithm 492
- rep-gtk command, Linux 753
- Report Browser 72
- report displays 73
- Report Properties Panel 76
- request-ixfr option 455
- requests for comments 505
- reserve_policy 174
- Reserved field of the TCP header 484
- reserved heuristic
 - JFS2 240
- resolv.lidap file 588
- resource class
 - host 158
 - network adapter 158
 - paging device 158
 - physical volume 158
 - processor 158
 - program 158
 - RMC concept 146
- Resource Description Framework 566
- resource limits 68
- resource manager
 - audit log 152
 - event response 148
 - file system 158
 - host 158
- Resource Monitoring and Control
 - command line interface 160
 - see also RMC 145
- resource sets (WLM) 51
- resource-usage 68

- response
 - RMC concept 146
- restore command 376
 - Functionality 377
 - handling of sparse files 312
- restricted users, FTP 509
- restvg command 372
- RFC
 - 1122 467
 - 2407 544
 - 2408 544
 - 2409 544
 - 2414 492
 - 2553 486
 - 2582 492
 - 816 467
 - RFC1886 457–458
 - RFC1995 454
 - RFC1996 455
 - RFC2241 505
 - RFC2553 489
 - RFC2610 505
 - RFC2874 456, 458
 - RFC2937 505–506
 - RFC3011 506
- Ricoh MP6201SE 6XR-2X 373
- right, VPN function mapping 540
- rightfirewall, VPN function mapping 540
- rightid, VPN function mapping 540
- rightnextthop, VPN function mapping 540
- rightrsasigkey, VPN function mapping 540
- rightsubnet, VPN function mapping 540
- rightupdown, VPN function mapping 540
- Rijndael 660
- rmaudrec command 155
- RMC
 - audit log resource manager 152
 - condition 146
 - configuration change property 158
 - event expression 147
 - event response resource manager 148
 - file system resource manager 158
 - host resource class 158
 - host resource manager 158
 - IBM.AuditRMd 160
 - IBM.ERrmd 160
 - IBM.FSrmmd 160
 - IBM.HostRMd 160
 - logevent command 151

- lsaudrec command 152
- monitor 146
- network adapter resource class 158
- notifievent command 151
- operational state property 158
- packaging 146
- paging device resource class 158
- PercentTotUsed 147
- physical volume resource class 158
- processor resource class 158
- program resource class 158
- rearm expression 147
- resource class 146
- response 146
- rmcd 160
- wallevent command 151
- rmcctrl command
 - RMC control program 146
- rmcd
 - RMC control daemon 160
- rmcondition command 160
- rmcondresp command 161
- rmdev command 185
- rmfs command
 - commands
 - rmfs 221
- rmgroup command
 - loadable module support 573
- rmmmap_create() function call 12
- rmpath command 175
- rmresponse command 161
- rmrsrc command 160
- rmss command 396
- rmuser command
 - loadable module support 573
- rndc command 443
- rndc.conf 443
- rndc-confgen command
 - commands
 - rndc-confgen 443
- root
 - file system permissions 218
- root zone file 450
- root.system 218
- rootvg, JFS2 236
- round_robin algorithm 180
- route command 464, 473
- routed daemon 467
- routers 485
- routes
 - cost of 463, 469
 - interface specific 464
- routine nbc_locate() function call 516
- routing
 - multipath 460
- rpc.statd daemon 253
- RPM 734, 740
 - command options 756
 - database 756
 - install 758
 - query database 758
- rpm 747
- rpm command 736
- rpm command, Linux 756
- rpm installer 735
- RPM packages 735
- rpower command 163
- RSC
 - Version 5.2 support withdrawal 777
- rset 52
- rset registry 52
- rsync command, Linux 753
- RTAS log, UE-Gard 136
- run-time loadable configuration module 168

S

- SA 543
- SA_SIGINFO handler 135
- SACK mechanism 493
- SAP, CD-ROM mapping 260
- SAP, CD-ROM translation 260
- sar command 399
- sawfish window manager, Linux 753
- SC_DIAGNOSTIC flag, automount facility 258
- schedtune command 424
- scheduler (WLM) 34
- script language, Perl 5.6 26
- SCSI scsd 169
- SDD 169
- secret, VPN 540
- secure rcmds 661
- Secure Sockets Extension, Java 570
- Secure Sockets Layer, Java 570
- SecureWay Directory 584
 - audit events 590
 - audit plug-in 589
 - audit service 592

- class definition 591
- security plug-in 589
- security
 - enhanced login privacy 649
 - FTP 509
 - password prompt 650
 - r-commands 661
 - TSIG 447
- Security Association 543
- Security Association Payload 543
- security plug-in, LDAP 589
- Security, Internet Key Exchange 536
- security, Java 570
- security, Perl installation 26
- security, regarding BIND 439
- sed command, Linux 753
- segment allocation, very large program support 12
- Sendmail 387
- sendmail 386
 - sendmail, aliases file 386
 - sendmail, anti-spam 386
 - sendmail, virtual hosting 386
 - sendmail.cf 386
- serialization, logical volume 215
- server consolidation 34
- server message block 525
- Service Location Protocol 505
- session log 340
- set \$pretty (DBX) 293
- set \$repeat command, KDB 298
- set scroll command, KDB 298
- setclock command
 - commands
 - setclock 442
- setgid() system call 49
- setlocale() function 792
- setpri() system calls 49
- settime command 679
- setting of the log level 544
- setuid() system call 49
- setuname command 679
- sh_oplockfiles 526
- sh_searchcache 526
- sh_sendfile 526
- shconf command
 - shd configuration 281
- shd
 - cmd action 283
 - configuration 281
 - errlog action 283
 - lio detection scheme 284
 - login action 283
 - pp_login parameter 283
 - prio detection scheme 281–282
 - reboot action 284
 - shdaemon 281
 - ss_pp parameter 283
 - warning action 283
- shdaemon 281
- shell attribute
 - value changed 25
- shmat()
 - fast connect 529
- shmat() function call 12
- shortcut, MPOA 549
- shrinkps command 382
- shutdown command
 - logging 384
- sh-utils command, Linux 753
- sigact pseudo file 685
- SIGBUS signal 135
- SIGBUS signal, UE-Gard 136
- simplified chinese locale, GBK 784
- single function adapter
 - PCI FRU Isolation 288
- single segment
 - JFS 219
- single system monitoring 153
- single transmission window 493
- SINGLE_PATH 171
- SIZE_GB bosinst.data 278
- Slab allocators
 - DNLC 219
- slang library, Linux 753
- SLB, Dynamic Feedback Protocol 545
- Slow-start algorithm 484
- smb
 - fast connect 525
- SMB file system support 530
- SMBFS
 - installation 531
- SMIT
 - chgaio fastpath 21
 - chgposixaio 21
 - System V print 727
- SMP 34
- snap
 - adump 315

- TCP/IP information 313
- Workload Manager information 315
- snap command 277, 313
- snapcore command 310
- Snapshot image (JFS2) 242
- Snapshot map
 - JFS2 Performance 243
- snapshot backups 213
- snapshot command 249
 - bos.rte.filesystem 250
- Snapshot Display 74
- snapshots 80
- SNMP 345
 - Cluster System Management (CSM) 165
- SO_KEEPALIVE 488
- sock command, KDB 300
- sodebug 492
- software license agreement
 - agreement database 358
 - agreement file 357
 - BOS install 362
 - bosinst.data file 359
 - handling 357
 - installp command 359
 - inulag command 358
 - migration install 362
- Software maintenance
 - CSM 164
- Software Overview 741
- Solaris affinity 687
 - /proc 688
 - /proc/pid#/cwd 688
 - /proc/pid#/fd 688
- Application binary interface (ABI) 696
- atrm command 666
- cpio command 666
- date command 667
- df command 667
- dfmounts command 669
- dfshares command 668
- dircmp command 669
- dispgid command 670
- dispuuid command 670
- getconf command 670
- getdev command 671
- getgrp command 672
- groups command 673
- last command 673
- ldd command 674
- listgrp command 674
- ln command 675
- logins command 675
- mach command 676
- proccred command 689
- procfiles command 689
- proclags command 689
- procldd command 690
- procmmap command 690
- procrun command 692
- procsig command 690
- procstack command 691
- proctop command 691
- proctree command 692
- procwait command 692
- procwdx command 689
- ps command 677
- pTools 688
- pwck command 677
- settime command 679
- setuname command 679
- Sun user thread API 696
- Sun user thread API filesets 697
- swap command 679
- truss command 695
- umountall command 680
- wall command 680
- weak symbol support 664
- whodo command 681
- zdump command 681
- zic command 682
- Solaris commands, Solaris affinity 665
- Solaris tools 687
- source affinity for Linux applications 762
- span multi volumes 383
- sparse files
 - handling by restore 312
- splat command 419
 - enablement 420
- split-connection proxy system 481
- splivg command 213
- sr64, KDB 300
- ss_pp
 - shd parameter 283
- SSL, Java 570
- standard congestion control algorithms 484
- Standard Ethernet format, MPOA 550
- startsrc command, enable qdaemon debugging 726

- startsrc command, regarding DNS 442
- statd (NFS) 253
- status command, KDB 300
- status pseudo file 685
- stopcondresp command 161
- Storage Networking Industry Association 261
- striprm command 397, 411
- structure of devinstall 747
- su command 650
- subclass 38
- subsystems
 - RMC 146
- superclass 38, 70
- Sv_SE locale 797
- svmon command 396, 402, 406
- swap command 679
- swapoff command 378
- switch.prt command 728
- symlinks, for BIND 440
- syncd
 - JFS2 performance 241
- sysent pseudo file 685
- syslog 544
- syslog daemon 492, 544
- syslog enhancement 492
- syslog, VPN function mapping 539
- system calls
 - BeginCriticalSection() 21
 - coredump() 277
 - EnableCriticalSections() 21
 - EndCriticalSection() 21
 - exec() 46, 49
 - getusage() 20
 - install_lwcf_handler() 307
 - ioctl(IOCINFO) 196
 - mt_trce() 307
 - plock() 49
 - pthread_atfork() 16
 - pthread_atfork_np() 16
 - pthread_atfork_unregister_np() 16
 - pthread_getrusage_bp 20
 - setgid() 49
 - setpri() 49
 - setuid() 49
 - tracing using truss 693
- system dump facility 279
- system hang detection
 - see also shd 281
- System information command
 - getconf 392
 - getconf command 392
- System management
 - Mksysb on CD-R or DVD 372
 - generic backup CD 372
 - non-bootable volume group backup 372
 - personal system backup CD 372
 - tested software and hardware 372
- system management, Linux 754
- system monitoring 153
 - CPU cycles 153
 - Number of processors 153
 - Operating system level 153
 - Serial number 153
- system resources 74
- System V
 - affinity 663
- System V affinity 678
- System V init editor, Linux 755
- System V Print
 - bos.msg 727
 - bos.svprint 727
 - bos.terminfo.svprint.data 727
- System V print
 - switch.prt command 728
- System V print SMIT enablement 727
- System.Default class 69

T

- tables
 - routing 463
- Tabulation Display 75
- tar 315
- tar command 277, 313
- tar command, Linux 753
- TCB
 - CAPP/EAL4+ 647
 - Tivoli risk manager
 - tcbck command 648
 - Tivoli risk manager integration 648
- tcbck command 648
- TCBHEAD_LOCK, global lock 484
- tcl/tk command, Linux 753
- TCP keep alive enhancements 488
- TCP/IP 484
 - ACK 486
 - Fast Recovery algorithm 492
 - fragmentation 483

- interface backup, EtherChannel 556
- layer 484
- receiver 484
- segment size max. 465
- sender 484
- splicing 481
- TCP/IP collection, snap 313
- tcp_ecn 484, 493
- tcp_init_window 492
- tcp_inpcb_hashtab_siz 488
- TCP_KEEPCNT 489
- TCP_KEEPIIDLE 488
- TCP_KEEPINTVL 488
- tcp_newreno 492
- TCP-endpoint 485
- tcsh command, Linux 753
- telnet, ANSI terminal emulation 391
- terabytes 225
- TERM environment variable, ANSI 391
- terminal emulation, ANSI 391
- text search engine 354
 - binary compatibility 355
- textutils command, Linux 753
- tgx command 273
- th command, KDB 300
- thread 253
- thread command, KDB 300
- threads (vmstat) 400
- tier 79
- Tier/Class Menu 79
- Time based configuration
 - Web-based System Manager configuration 90
 - Workload manager 88
- time_t 12
- timeouts
 - gateway 470
 - network 470
- tips area 344
- Tivoli 648
 - CSM 164
- tivoli management agent 648
- tivoli netview 346
- Tivoli risk manager integration
 - TCB 648
- TLS, Java 570
- tmd daemon 544
- Token Ring
 - multi protocol over atm 552
- token ring format, MPOA 550
- tokstat command, LANE trace 552
- Toolbox for Linux Applications 733
- tools, cmdstat 399
- topas command 397
- total Logins
 - Workload manager
 - Class resource limits 95
- totalConnectTime
 - Workload manager
 - Class resource limits 94
- totalCPU
 - Workload manager
 - Class resource limits 94
- totalDiskIO
 - Workload manager
 - Class resource limits 94
- totalLogins
 - Workload manager
 - Class resource limits 95
- totalProcesses
 - Workload manager
 - Class resource limits 95
- totalThreads
 - Workload manager
 - Class resource limits 95
- tprof command 396, 409, 416
- tprof enhancement
 - Automated offline 412
 - clarifying post processing and manual offline 414
 - Further features 416
 - Java Profiler Agent 412
 - Manual offline 413
 - Micro profiling 410
 - Nano profiling 412
 - Post processing 413
 - Profiling 410
 - Real time 412
- tprof enhancements 411
- trace
 - buffer size 268
 - event groups 268
 - single mode 267
 - single-buffer 267
- trace command 15, 267, 275, 315, 418, 494
- trace file, sample 418
- trace group
 - hook IDs 269
- trace hook 495

- trace, MPOA 548
- tracebacks 307
- transaction signature 439
- transfig command, Linux 753
- transmit informational data 543
- Transport Layer Security, Java 570
- trcevgrp command 270
- trcfile 494
- trcpt command 401
- trcpt
 - events groups 269
- trcpt command 269, 272
- Trend Box
 - Workload Manager
 - Trend Box 77
- TrueType fonts 390
- TrueType rasterizer 390
- truss command 695
 - system call tracing 693
- trusted root zone file 450
- TSIG security 447
- Tunnel Manager 544
- Turboways PCI ATM adapter 771
- type, VPN function mapping 539

U

- UCS2 788
- udfs file system 258
- UDI 734, 740
- UDI formatted device drivers 747
- UDID 171
- udisetaup command 747
- udp_inpcb_hashtab_siz 488
- udp_pmtu_discover option 465
- UE-Gard 136
 - system calls 135
- UFST code 390
- Ultimedia 771
- umountall command 680
- unicode 3.1 support 797
- Unicode Extension B 793
- Unicode extension B
 - iconv command 794
- Unicode System Version 2 788
- Unicode XOM enhancement 794
- Uniform Device Interface 734, 740
- Unique device identifier 171
- unique_id 174

- Universal disk Format, mkcd command 375
- Universal UCS Converter
 - Unicode extension B 794
- unix_mp 297
- unix_up 297
- unmount command 256
- unregister callouts 770
- unzip command, Linux 753
- urandom file 652
- user (WLM) 48
- user administration, Linux 754
- user attributes
 - basic 571
 - extended 571
 - Kerberos 578
 - registry 572
- user interface
 - web-based system manager 330
- user level security 527
- user names 387
- user-network interface 553
- UTF-32
 - GB18030 794
- utmp command 392
- utmpd
 - utmp command 392
- uudecode command 384
- uuencode command 383

V

- varlist command, KDB 300
- varrm command, KDB 300
- varyonvg command
 - read-only mode 187
- VERITAS Enterprise Administrator (VEA) 251
- VERITAS File System 251
- VERITAS Foundation Suite for AIX 250
- VERITAS NetBackup 250
- VERITAS Volume Manager 251
- Version 5.2
 - EtherChannel backup adapter 561
 - JFS2 large file system enablement 240
 - Migration steps 324
 - restore command 376
 - Zombie harvesting 22
- Version 5.2 AIX migration 323
- Version 5.2 migration
 - Enable system backups 329

- Enable trusted computing Base 329
- Features 324
- Hardware requirement 323
- Import user volume groups 329
- preparation 323
- very large program support 12
- VFS 234
- Vi_VN locale, Euro support 789
- video
 - quality of service 434
- views support, regarding BIND 459
- vim command, Linux 753
- VIPA system management tasks 496
- virtual adapters
 - vlan 563
- virtual hosting, sendmail 386
- virtual IP address 499
- virtual IP addresses 495
- virtual LAN adapters 564
- Virtual Local Area Networks 563
- virtual memory manager 34
- Virtual Private Network, security 536
- vmo command 241
- vmstat command 241, 399–400
- vmtune command 241, 424
- vnc_enter 239
- vnc_init 239
- vnc_lookup 239
- vnc_purge 239
- vnc_remove 239
- vnode cache, JFS2 239
- vnode pointer 517
- volume group, mkvg command 209
- vpdadd command 23
- vpddel command 23

W

- wall command 680
- wallevent command
 - RMC command 151
- warning
 - shd action 283
- WBEM, CIM 27
- weak and global links, Solaris affinity 665
- Weak symbol support
 - Solaris affinity 664
- weak symbol support, Solaris affinity 664
- WEB browser 348

- web tool 342
- Web-Based Enterprise Management, CIM 27
- Web-based System Manager 329
 - accessibility 353
 - architecture 329
 - container 332
 - contents area 330
 - custom tools 342
 - ISAKMP 536
 - JFS2 233
 - launch plug-ins 334
 - Linux client 352
 - modes of operation 338
 - monitoring 153
 - monitoring, conditions 156
 - monitoring, events 155
 - overview-plug-ins 333
 - PC Client 346
 - plug-in 331
 - preferences 345
 - result window 343
 - security enhancements 658
 - session log 340
 - single system monitoring 153
 - SNMP integration 345
 - tips area 344
 - tivoli netview integration 346
 - Tunnel Manager 544
 - VPN, Task and Overview 542
 - Workload manager
 - Class resource limits 95
- Web-based system manager
 - Attribute value grouping configuration 84
- Web-based System Manager
 - JFS2 snapshot image 244
- wget command, Linux 753
- whodo command 681
- Width of Interval 77
- wild cards, IKE 536
- window manager, Linux 753
- WLM 68
 - 32bit 49
 - 64bit 49
 - fixed 49
 - plock 49
- WLM data 76
- wlm, localshm 44
- wlm, shared memory segment 44
- wlm, class accounting 68

- WLM_Console 72
- wlmcntrl command
 - Workload manager 95
- wlmon command 71
- wlmpcrf command 71
- WLMRM
 - bos.rte.control 86
- wlmstat command 71
 - Workload manager 95
- Workload Manager
 - accounting 68
 - Advanced Menu 80
 - assignment rules 62
 - Bar Display 74
 - class 68
 - class attributes 42
 - classes 37
 - classname 68
 - disk allocations 71
 - diskIO resource 61
 - First Period 77
 - inheritance 43
 - Last Period 77
 - overview 34
 - Report Browser 72
 - report displays 73
 - Report Properties 76
 - resources 68
 - Snapshot Display 74
 - subclass 38
 - superclass 38
 - Tabulation Display 75
 - Tier/Class Menu 79
 - Tiers 79
 - Width of Interval 77
 - WLM_Console 72
 - wlmon 71
 - wlmpcrf 71
- Workload manager
 - .times file 89
 - Attribute value grouping 83
 - Class resource limits
 - Web-based System Manager configuration 95
 - confset 88
 - confsetcntrl command 89
 - Event notification 86
 - IBM.WLM 86
 - Limits on class resources 94
 - lswlmonf command 89
 - new resource types 83
 - Time based configuration
 - Web-based System Manager configuration 90
 - Time based configurations 88
 - total limit on user defined classes 99
 - use of attribute value grouping 83
 - wlmcntrl command 95
 - WLMRM 86
 - wlmstat command 95
- workload manager
 - topas 406
- workload manager (svmon) 402
- Workload Manager information, snap 315
- Workload Manager Resource Manager (WLMRM) 86
- Write-behind (j2_maxRandomWrite)
 - JFS2 performance 241

X

- X Window server 30
- X/Open Single Sign-on Service 614
- X11R6
 - Unicode XOM enhancement 795
- XCOFF 409
- xfig command, Linux 753
- XML-based User Interface Language 566
- xmtrend daemon 71
- xmwlm daemon
 - daemons
 - xmwlm 71
- xpdf command, Linux 753
- XPG 5 20
- Xprofiler 422
- XSSO 614
- X-Windows performance profiler 422

Y

- Yamaha CRW4416SX 373
- Yamaha CRW8424S 373
- Yarrow engine 652
- yppasswordd daemon 611

Z

- zdump command 681
- Zh_CN locale 784

ZH_HK locale 795
ZH_SG locale 795
Zh_TW locale, Euro support 789
Zheng Ma Input Method 786
zic command 682
zip command, Linux 753
Zombie harvesting
 Prior to Version 5.2 22
 Version 5.2 22
zone files 445
zone transfers 454
zoo command, Linux 753
zsh command, Linux 753



Redbooks

ALX5L Differences Guide

Version 5.2 Edition



Redbooks

AIX 5L Differences Guide Version 5.2 Edition

**AIX - The industrial
strength UNIX
operating system**

**An expert's guide to
the new release**

**Version 5.0 through
5.2 enhancements
explained**

This IBM Redbook focuses on the differences introduced in AIX 5L through Version 5.2 when compared to AIX Version 4.3.3. It is intended to help system administrators, developers, and users understand these enhancements and evaluate potential benefits in their own environments.

AIX 5L introduces many new features, including Linux and System V affinity, dynamic LPAR, multipath I/O, 32- and 64-bit kernel and application support, virtual IP, Quality of Service enhancements, enhanced error logging, dynamic paging space reduction, hot-spare disk management, advanced Workload Manager, JFS2 snapshot image, and others. The availability of Web-based System Manager for Linux continues AIX's move towards a standard, unified interface for system tools. There are many other enhancements available with AIX 5L, and you can explore them in this redbook.

For customers who are familiar with AIX 5L Version 5.1, features that are new in AIX 5L Version 5.2 are indicated by a version number (5.2.0) in the title of the section.

**INTERNATIONAL
TECHNICAL
SUPPORT
ORGANIZATION**

**BUILDING TECHNICAL
INFORMATION BASED ON
PRACTICAL EXPERIENCE**

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks