IBM
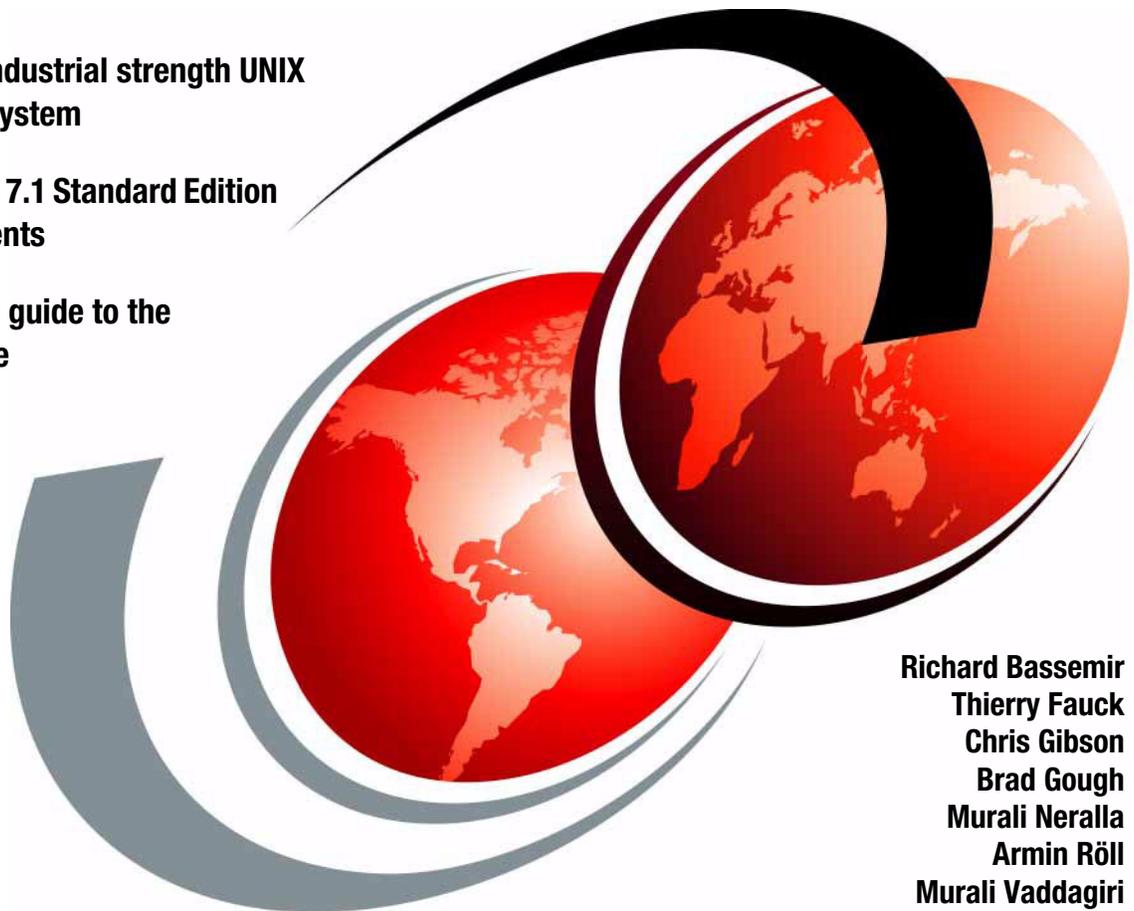
IBM AIX Version 7.1 Differences Guide

AIX - The industrial strength UNIX operating system

AIX Version 7.1 Standard Edition enhancements

An expert's guide to the new release

Richard Bassemir
Thierry Fauck
Chris Gibson
Brad Gough
Murali Neralla
Armin Röll
Murali Vaddagiri

Redbooks

ibm.com/redbooks

IBM

International Technical Support Organization

IBM AIX Version 7.1 Differences Guide

December 2010

Note: Before using this information and the product it supports, read the information in "Notices" on page xiii.

First Edition (December 2010)

This edition applies to AIX Version 7.1 Standard Edition, program number 5765-G98.

Contents

Figures

Tables

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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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 illustrate 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.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

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

Active Memory™	GPFS™	PowerPC®
AIX 5L™	HACMP™	PowerVM™
AIX®	IBM Systems Director Active	POWER®
BladeCenter®	Energy Manager™	pSeries®
Blue Gene®	IBM®	Redbooks®
DB2®	LoadLeveler®	Redbooks (logo) ®
developerWorks®	Parallel Sysplex®	Solid®
Electronic Service Agent™	Power Systems™	System p5®
Enterprise Storage Server®	POWER3™	System p®
eServer™	POWER4™	System Storage®
Everyplace®	POWER5™	Systems Director VMControl™
GDPS®	POWER6®	Tivoli®
Geographically Dispersed	POWER7™	WebSphere®
Parallel Sysplex™	PowerHA™	Workload Partitions Manager™

The following terms are trademarks of other companies:

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

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

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

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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

Preface

This IBM® Redbooks® publication focuses on the enhancements to IBM AIX® Version 7.1 Standard Edition. It is intended to help system administrators, developers, and users understand these enhancements and evaluate potential benefits in their own environments.

AIX Version 7.1 introduces many new features, including:

► Domain Role Based Access Control

► Workload Partition enhancements

► Topas performance tool enhancements

► Terabyte segment support

► Cluster Aware AIX functionality

AIX Version 7.1 offers many other new enhancements, and you can explore them all in this publication.

For clients who are not familiar with the enhancements of AIX through Version 5.3, a companion publication, *AIX Version 6.1 Differences Guide*, SG24-7559, is available.

The team who wrote this book

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

Richard Bassemir is an IBM Certified Consulting IT Specialist in the ISV Business Strategy and Enablement organization in the Systems and Technology Group located in Austin, Texas. He has seven years of experience in IBM System p® technology. He has worked at IBM for 33 years. He started in mainframe design, design verification, and test, and moved to Austin to work in the Software Group on various integration and system test assignments before returning to the Systems and Technology Group to work with ISVs to enable and test their applications on System p hardware.

Thierry Fauck is a Certified IT Specialist working in Toulouse, France. He has 25 years of experience in Technical Support with major HPC providers. As system administrator of the French development lab, his areas of expertise include AIX,

VIOS, SAN, and PowerVM™. He is currently leading an FVT development team for WPAR and WPAR mobility features. He authored a white paper on WPARs and actively contributed to the WPAR IBM Redbooks publication. This is his second AIX Differences Guide publication.

Chris Gibson is an AIX and PowerVM specialist. He works for Southern Cross Computer Systems, an IBM Business Partner located in Melbourne, Australia. He has 11 years of experience with AIX and is an IBM Certified Advanced Technical Expert - AIX. He is an active member of the AIX community and has written numerous technical articles on AIX and PowerVM for IBM developerWorks®. He also writes his own AIX blog on the IBM developerWorks website. Chris is also available online on Twitter (@cgibbo). This is his second Redbooks publication having previously co-authored the NIM from A to Z in AIX 5L™ book.

Brad Gough is a technical specialist working for IBM Global Services in Sydney, Australia. Brad has been with IBM since 1997. His areas of expertise include AIX, PowerHA™, and PowerVM. He is an IBM Certified Systems Expert - IBM System p5® Virtualization Technical Support and IBM eServer™ p5 and pSeries® Enterprise Technical Support AIX 5L V5.3. This is his third IBM Redbooks publication.

Murali Neralla is a Senior Software Engineer in the ISV Business Strategy and Enablement organization. He is also a Certified Consulting IT Specialist. He has over 15 years of experience working at IBM. Murali currently works with the Financial Services Sector solution providers to enable their applications on IBM Power Systems™ running AIX.

Armin Röll works as a Power Systems IT specialist in Germany. He has 15 years of experience in Power Systems and AIX pre-sales technical support and, as a team leader, he fosters the AIX skills community. He holds a degree in experimental physics from the University of Hamburg, Germany. He co-authored the AIX Version 4.3.3, the AIX 5L Version 5.0, the AIX 5L Version 5.3 and the AIX 6.1 Differences Guide IBM Redbooks.

Murali Vaddagiri is a Senior Staff Software Engineer working for IBM Systems and Technology Group in India. He has over 7 years of experience in AIX operating system and PowerHA development. He holds a Master of Science degree from BITS, Pilani, India. His areas of expertise include security, clustering, and virtualization. He has filed nine US patents and authored several disclosure publications in these areas.

Scott Vetter, PMP, managed the project that produced this publication. Scott has also authored a number of IBM Redbooks publications.

Special thanks to the following people for their contributions to this project:

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

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 books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

► Use the online **Contact us** review Redbooks form found at:

  ibm.com/redbooks

► Send your comments in an email to:

  redbooks@us.ibm.com

► Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

► Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

► Follow us on Twitter:

  http://twitter.com/ibmredbooks

► Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

► Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

► Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

1

Application development and debugging

This chapter describes the major AIX Version 7.1 enhancements that are part of the application development and system debug category, including:

1.1 AIX binary compatibility

IBM guarantees that applications, whether written in-house or supplied by an application provider, will run on AIX 7.1 if they currently run on AIX 6.1 or AIX 5L—without recompilations or modification. Even well-behaved 32-bit applications from AIX V4.1, V4.2, and V4.3 will run without recompilation.

Refer to the following for further information regarding binary compatibility:

http://www.ibm.com/systems/power/software/aix/compatibility/

1.2 Improved performance using 1 TB segments

In AIX V7.1, 1 TB segments are an autonomic operating system feature designed to improve performance of 64-bit large memory applications. This enhancement optimizes performance when using shared memory regions (shmat/mmap). New restricted **vmo** options are available to change the operating system policy. A new VMM_CNTRL environment variable is available to alter per process behavior.

> **Important:** Restricted tunables should not be changed without direction from IBM service.

1 TB segment aliasing improves performance by using 1 TB segment translations on Shared Memory Regions with 256 MB segment size. This support is provided on all 64-bit applications that use Shared Memory Regions. Both directed and undirected shared memory attachments are eligible for 1 TB segment aliasing.

If an application qualifies to have its Shared Memory Regions use 1 TB aliases, the AIX operating system uses 1 TB segment translations without changing the application. This requires using the shm_1tb_shared **vmo** tunable, shm_1tb_unshared **vmo** tunable, and esid_allocator **vmo** tunable.

The shm_1tb_shared **vmo** tunable can be set on a per-process basis using the SHM_1TB_SHARED= VMM_CNTRL environment variable. The default value is set dynamically at boot time based on the capabilities of the processor. If a single Shared Memory Region has the required number of ESIDs, it is automatically changed to a shared alias. The acceptable values are in the range of 0 to 4 KB (require approximately 256 MB ESIDs in a 1 TB range).

Example 1-1 on page 3 shows valid values for shm_1tb_shared tunable parameter.

Example 1-1   The shm_1tb_shared tunable

```
#vmo -F -L shm_1tb_shared
NAME                        CUR    DEF    BOOT   MIN    MAX    UNIT            TYPE
     DEPENDENCIES
--------------------------------------------------------------------------------
shm_1tb_shared              0      12     12     0      4K     256MB segments  D
--------------------------------------------------------------------------------
#
```

The shm_1tb_unshared **vmo** tunable can be set on a per-process basis using the SHM_1TB_UNSHARED= VMM_CNTRL environment variable. The default value is set to 256. The acceptable values are in the range of 0 to 4 KB. The default value is set cautiously (requiring the population of an up to 64 GB address space) before moving to an unshared 1 TB alias.

The threshold number is set to 256 MB segments at which a shared memory region is promoted to use a 1 TB alias. Lower values must cautiously use the shared memory regions to use a 1 TB alias. This can lower the segment look-aside buffer (SLB) misses but can also increase the page table entry (PTE) misses, if many shared memory regions that are not used across processes are aliased.

Example 1-2 shows valid values for the shm_1tb_unshared tunable parameter.

Example 1-2   The shm_1tb_unshared tunable

```
#vmo -F -L shm_1tb_unshared
NAME                        CUR    DEF    BOOT   MIN    MAX    UNIT            TYPE
     DEPENDENCIES
--------------------------------------------------------------------------------
shm_1tb_unshared            256    256    256    0      4K     256MB segments  D
--------------------------------------------------------------------------------
#
```

The esid_allocator **vmo** tunable can be set on a per-process basis using the ESID_ALLOCATOR= VMM_CNTRL environment variable. The default value is set to 0 for AIX Version 6.1 and 1 for AIX Version 7.1. Values can be either 0 or 1. When set to 0, the old allocator for undirected attachments is enabled. Otherwise, a new address space allocation policy is used for undirected attachments.

This new address space allocator attaches any undirected allocation (such as SHM and MMAP) to a new address range of 0x0A00000000000000 - 0x0AFFFFFFFFFFFFFF in the address space of the application.

The allocator optimizes the allocations in order to provide the best possible chances of 1 TB alias promotion. Such optimization can result in address space holes, which are considered normal when using undirected attachments.

Directed attachments are done for the 0x0700000000000000 - 0x07FFFFFFFFFFFFFF range, thus preserving compatibility with earlier versions. In certain cases where this new allocation policy creates a binary compatibility issue, the legacy allocator behavior can be restored by setting the tunable to 0.

Example 1-3 shows valid values for the esid_allocation tunable parameter.

Example 1-3   The esid_allocator tunable

```
# vmo -F -L esid_allocator
NAME                         CUR   DEF   BOOT   MIN   MAX   UNIT        TYPE
    DEPENDENCIES
--------------------------------------------------------------------------------
esid_allocator               1     1     1      0     1     boolean        D
--------------------------------------------------------------------------------
#
```

Shared memory regions that were not qualified for shared alias promotion are grouped into 1 TB regions. In a group of shared memory regions in a 1 TB region of the application's address space, if the application exceeds the threshold value of 256 MB segments it is promoted to use an unshared 1 TB alias.

In applications where numerous shared memory is attached and detached, lower values of this threshold can result in increased PTE misses. Applications that only detach shared memory regions at exit can benefit from lower values of this threshold.

To avoid causing the environments name space conflicts, all environment tunables are used under the master tunable VMM_CNTRL. The master tunable is specified with the @ symbol separating the commands.

An example for using VMM_CNTRL is:

```
VMM_CNTRL=SHM_1TB_UNSHARED=32@SHM_1TB_SHARED=5
```

Take Note: 32-bit applications are not affected by either `vmo` or environment variable tunable changes.

All **vmo** tunables and environment variables have analogous **vm_pattr** commands. The exception is the esid_allocator tunable. This tunable is not present in the vm_pattr options to avoid situations where portions of the shared memory address space are allocated before running the command.

If using AIX Runtime Expert, the shm_1tb_shared, shm_1tb_unshared and esid_allocator tunables are all in the vmoProfile.xml profile template.

1.3 Kernel sockets application programming interface

To honor the increasing client and ISV demand to code environment- and solution-specific kernel extensions with socket level functionality, AIX V7.1 and AIX V6.1 with TL 6100-06 provide a documented kernel sockets application programming interface (API). The kernel service sockets API is packaged with other previously existing networking APIs in the base operating system 64-bit multiprocessor runtime fileset bos.mp64.

The header file `/usr/include/sys/kern_socket.h`, which defines the key data structures and function prototypes, is delivered along with other existing header files in the bos.adt.include fileset. As provided in Table 1-1, the implementation of the new programming interface is comprised of 12 new kernel services for TCP protocol socket operations. The API supports the address families of both IPv4 (AF_INET) and IPv6 (AF_INET6).

Table 1-1   Kernel service socket API

TCP protocol socket operation	Kernel service name	Function
Socket creation	kern_socreate	Creates a socket based on the address family, type, and protocol.
Socket binding	kern_sobind	Associates the local network address to the socket.
Socket connection	kern_soconnect	Establishes connection with a foreign address.
Socket listen	kern_solisten	Prepares to accept incoming connections on the socket.
Socket accept	kern_soaccept	Accepts the first queued connection by assigning it to the new socket.

TCP protocol socket operation	Kernel service name	Function
Socket get option	kern_sogetopt	Obtains the option associated with the socket, either at the socket level or at the protocol level.
Socket set option	kern_sosetopt	Sets the option associated with the socket, either at the socket level or at the protocol level.
Socket reserve operation to set send and receive buffer space	kern_soreserve	Enforces the limit for the send and receive buffer space for a socket.
Socket shutdown	kern_soshutdown	Closes the read-half, write-half, or both read and write of a connection.
Socket close	kern_soclose	Aborts any connections and releases the data in the socket.
Socket receive	kern_soreceive	The routine processes one record per call and tries to return the number of bytes requested.
Socket send	kern_sosend	Passes data and control information to the protocol associated send routines.

For a detailed description of each kernel service, refer to *Technical Reference: Kernel and Subsystems, Volume 1*, SC23-6612 of the AIX product documentation at:

http://publib.boulder.ibm.com/infocenter/aix/v6r1/topic/com.ibm.aix.
kerneltechref/doc/ktechrf1/ktechrf1.pdf

1.4 UNIX08 standard conformance

The POSIX UNIX® standard is periodically updated. Recently, a draft standard for Issue 7 has been released. It is important from both an open standards and a client perspective to implement these new changes to the standards.

AIX V7.1 has implemented IEEE POSIX.1-200x The Open Group Base Specifications, Issue 7 standards in conformance with these standards.

The Base Specifications volume contains general terms, concepts, and interfaces of this standard, including utility conventions and C-language header definitions. It also contains the definitions for system service APIs and subroutines, language-specific system services for the C programming language, and API issues, including portability, error handling, and error recovery.

The Open Group Base Specifications, Issue 7 can be found at:

http://www.unix.org/2008edition

In adherence to IEEE POSIX.1-200x The Open Group Base Specifications, Issue 7 standards, several enhancements were made in AIX V7.1.

New system calls were added so that users can open a directory and then pass the returned file descriptor to a system call, together with a relative path from the directory. The names of the new system calls in general were taken from the existing system calls with an *at* added at the end. For example, an accessxat() system call has been added, similar to accessx(), and openat() for an open().

There are several advantages when using these enhancements . For example, you can implement a per-thread current working directory with the newly added system calls. Another example: you can avoid race conditions where part of the path is being changed while the path name parsing is ongoing.

Table 1-2 shows a subset of new library functions and system calls that are added.

Table 1-2   Short list of new library functions and system calls

System calls	
acessxat	mknodat
chownxat	openat
faccessat	openxat
fchmodat	readlinkat
fchownat	renameat
fexecve	stat64at
fstatat	statx64at
futimens	statxat
kopenat	symlinkat
linkat	ulinkat

System calls	
mkdirat	utimensat
mkfifoat	

Example 1-4 shows how applications can make use of these calls. The overall effect is the same as if you had done an open call to the path `dir_path/filename`.

Example 1-4   A sample application call sequence

```
..........
dirfd = open(dir_path, …);
..........
accessxat(dirfd, filename, ….);
.........
fd = openat(dirfd, filename, …);
..........
```

Table 1-3 provides a subset of added routines that are the same as isalpha, isupper, islower, isdigit, isxdigit, isalnum, isspace, ispunct, isprint, isgraph, and iscntrl subroutines respectively, except that they test character C in the locale represented by Locale, instead of the current locale.

Table 1-3   New library functions to test characters in a locale

Name	
isupper_l	ispunct_l
islower_l	isprint_l
isdigit_l	isgraph_l
isxdigit_l	iscntrl_l
isspace_l	isalnum_l

1.4.1 stat structure changes

The stat, stat64, and stat64x structures are changed. A new st_atim field, of type struct timespec, replaces the old st_atime and st_atime_n fields:

```
struct timespec {
        time_t tv_sec; /* seconds */
        long tv_nsec; /* and nanoseconds */
};
The old fields are now macros defined in <sys/stat.h> file:
```

```
#define st_atime       st_atim.tv_sec
#define st_mtime       st_mtim.tv_sec
#define st_ctime       st_ctim.tv_sec
#define st_atime_n     st_atim.tv_nsec
#define st_mtime_n     st_mtim.tv_nsec
#define st_ctime_n     st_ctim.tv_nsec
```

1.4.2 open system call changes

Two new open flags are added to the open() system call:

#include <fcntl.h>

int open(const char *path, int oflag, ...);

▶ O_DIRECTORY

If the path field does not name a directory, open() fails and sets errno to ENOTDIR.

▶ O_SEARCH

Open a directory for search; open() returns an error EPERM if there is no search permission.

> **Of interest:** The O_SEARCH flag value is the same as the O_EXEC flag. Therefore, the result is unspecified if this flag is applied to a non-directory file.

1.4.3 utimes system call changes

The utimes() system call is changed as follows:

#include <sys/stat.h>

utimes(const char *fname, const struct timeval times[2]);

▶ If either of the times parameter timeval structure tv_usec fields have the value UTIME_OMIT, then this time value is ignored.

▶ If either of the times parameter timespec structure tv_usec fields have the value UTIME_NOW, then this time value is set to the current time.

This provides a way in which the access and modify times of a file can be better adjusted.

1.4.4 futimens and utimensat system calls

Two new system calls, futimens() and utimensat(), are added. Both provide nanosecond time accuracy, and include the UTIME_OMIT and UTIME_NOW functionality. The utimensat() call is for path names, and futimens() is for open file descriptors.

int utimensat(int dirfd, const char *fname, const struct timespec times[2], int flag);

int futimens(int fd, const struct timespec times[2]);

1.4.5 fexecve system call

The new fexecve system call is added as follows:

#include <unistd.h>

int fexecve(int fd, const char *argp[], const char *envp[]);

The fexecve call works same as the execve() system call, except that it takes a file descriptor of an open file instead of a pathname of a file. The fexecve call may not be used with RBAC commands (the file must have DAC execution permission).

For a complete list of changes, refer to AIX V7.1 documentation at:

http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=/com.ibm.aix.ntl/releasenotes_kickoff.htm

1.5 AIX assembler enhancements

This section discusses the enhancements made to the assembler in AIX V7.1.

1.5.1 Thread Local Storage (TLS) support

Thread Local Storage (TLS) support has been present in the IBM XL C/C++ compiler for some time. The compiler's -qtls option enables recognition of the __thread storage class specifier, which designates variables that are allocated from threadlocal storage.

When this option is in effect, any variables marked with the __thread storage class specifier are treated as local to each thread in a multithreaded application.

At runtime, an instance of each variable is created for each thread that accesses it, and destroyed when the thread terminates. Like other high-level constructs that you can use to parallelize your applications, thread-local storage prevents race conditions to global data, without the need for low-level synchronization of threads.

The TLS feature is extended to the assembler in AIX V7.1 to allow the assembler to generate object files with TLS functionality from an associated assembler source file.

1.5.2 TOCREL support

Recent versions of the IBM XL C/C++ compilers support compiler options (for example -qfuncsect, -qxflag=tocrel) that can reduce the likelihood of TOC overflow. These compiler options enable the use of new storage-mapping classes and relocation types, allowing certain TOC symbols to be referenced without any possibility of TOC overflow.

The TOCREL functionality is extended to the assembler in AIX V7.1. This allows the assembler to generate object files with TOCREL functionality from an associated assembler source file.

1.6 Malloc debug fill

Malloc debug fill is a debugging option with which you can fill up the allocated memory with a certain pattern.

The advantage of using this feature for debugging purposes is that it allows memory to be *painted* with some user-decided initialized value. This way, it can then be examined to determine if the requested memory has subsequently been used as expected by the application. Alternatively, an application could fill in the memory itself in the application code after returning from malloc, but this requires recompilation and does not allow the feature to be toggled on or off at runtime.

For example, you might fill the spaces with a known string, and then look (during debug) to see what memory has been written to or not, based on what memory allocations are still filled with the original fill pattern. When debugging is complete, you can simply unset the environment variable and rerun the application.

Syntax for enablingthe Malloc debug fill option is as follows:

```
#export MALLOCDEBUG=fill:pattern
```

where `pattern` can be octal or hexadecimal numbers specified in the form of a string.

The following example shows that a user has enabled the Malloc debug fill option and set the fill `pattern` to string `abc.`

```
#export MALLOCDEBUG=fill:"abc"
```

Table 1-4 shows the fill pattern for a user allocating eight bytes of memory with a fill pattern of `abc.`

Table 1-4   Malloc abc fill pattern

1	2	3	4	5	6	7	8
a	b	c	a	b	c	a	b

Important: `pattern` can be octal or hexadecimal numbers specified in the form of a string. The pattern `\101` is treated as the octal notation for character `A`. The pattern `\x41` is treated as the hexadecimal notation for character `A`.

The fill pattern is parsed byte by byte, so the maximum that can be set for fill pattern is "\xFF" or "\101". If you set the fill pattern as "\xFFA", then it will be taken as hex FF and char A. If you want A also to be taken as hex, the valid way of specifying is "\xFF\xA". The same holds true for octal—if you set the fill pattern as "\101102", then it will be taken as octal 101 and string "102".

If an invalid octal number is specified, for example `\777` that cannot be contained within 1 byte, it will be stored as `\377`, the maximum octal value that can be stored within 1 byte.

1.7 proc_getattr and proc_setattr enhancements

AIX 6.1 TL6 and 7.1 provide Application Programming Interfaces (API) proc_getattr and proc_setattr to allow a process to dynamically change its core dump settings.

The procattr_t structure that is passed to the API is as follows:

```
typedef struct {
    uchar core_naming;  /* Unique core file name */
    uchar core_mmap;    /* Dump mmap'ed regions in core file */
    uchar core_shm;     /* Dump shared memory regions in core file */
    uchar aixthread_hrt;/* Enable high res timers */
} procattr_t;
```

The following sections discuss new attributes for the proc_getattr and proc_setattr system calls.

1.7.1 Core dump enhancements

The API supports enabling, disabling, and querying the settings for the following core dump settings:

CORE_NAMING   Controls whether unique core files should be created with unique names.

CORE_MMAP   Controls whether the contents of mmap() regions are written into the core file.

CORE_NOSHM   Controls whether the contents of system V shared memory regions are written into the core file.

Applications can use these interfaces to ensure that adequate debug information is captured in cases where they dump core.

Example 1-5 provides syntax of these two APIs.

Example 1-5 proc_getattr(), proc_setattr() APIs

```
#include <sys/proc.h>

int proc_getattr (pid, attr, size)
pid_t pid;
procattr_t *attr;
uint32_t size;

The proc_getattr subroutines allows a user to retrieve the current
state of certain process attributes. The information is returned in the
structure procattr_t defined in sys/proc.h


int proc_setattr (pid, attr, size)
pid_t pid;
procattr_t *attr;
uint32_t size;

The proc_setattr subroutines allows a user to set selected attributes
of a process. The list of selected attributes is defined in structure
procattr_t defined in sys/proc.h
```

1.7.2 High resolution timers

The API supports setting the high resolution timers. SHIGHRES enables high-resolution timers for the current process.

1.8 Disabled read write locks

The existing complex locks used for serialization among threads work only in a process context. Because of this, complex locks are not suitable for the interrupt environment.

When simple locks are used to serialize heavily used disabled critical sections which could be serialized with a shared read/write exclusive model, performance bottlenecks may result.

AIX 7.1 provides kernel services for shared read/write exclusive locks for use in interrupt environments. These services can be used in kernel or kernel extension components to get improved performance for locks where heavy shared read access is expected. Table 1-5 lists these services.

Table 1-5   Kernel and kernel extension services

Index	Kernel service
1	**drw_lock_init** **Purpose** Initialize a disabled read/write lock. **Syntax** #include<sys/lock_def.h> void drw_lock_init(lock_addr) drw_lock_t lock_addr ; **Parameters** lock_addr - Specifies the address of the lock word to initialize.

Index	Kernel service
2	**drw_lock_read** **Purpose** Lock a disabled read/write lock in read-shared mode. **Syntax** #include<sys/lock_def.h> void drw_lock_read(lock_addr) drw_lock_t lock_addr ; **Parameters** lock_addr - Specifies the address of the lock word to lock.
3	**drw_lock_write** **Purpose** Lock a disabled read/write lock in write-exclusive mode. **Syntax** #include<sys/lock_def.h> void drw_lock_write(lock_addr) drw_lock_t lock_addr ; **Parameters** lock_addr - Specifies the address of the lock word to lock.
4	**drw_lock_done** **Purpose** Unlock a disabled read/write lock. **Syntax** #include<sys/lock_def.h> void drw_lock_done(lock_addr) drw_lock_t lock_addr ; **Parameters** lock_addr - Specifies the address of the lock word to unlock.

Index	Kernel service
5	**drw_lock_write_to_read** **Purpose** Downgrades a disabled read/write lock from write exclusive mode to read-shared mode. **Syntax** #include<sys/lock_def.h> void drw_lock write_to_read(lock_addr) drw_lock_t lock_addr ; **Parameter** lock_addr - Specifies the address of the lock word to lock.
6	**drw_lock_read_to_write** **drw_lock_try_read_to_write** **Purpose** Upgrades a disabled read/write from read-shared to write exclusive mode. **Syntax** #include<sys/lock_def.h> boolean_t drw_lock read_to_write(lock_addr) boolean_t drw_lock try_read_to_write(lock_addr) drw_lock_t lock_addr ; **Parameters** lock_addr - Specifies the address of the lock word to lock.
7	**drw_lock_islocked** **Purpose** Determine whether a drw_lock is held in either read or write mode. **Syntax** #include<sys/lock_def.h> boolean_t drw_lock_islocked(lock_addr) drw_lock_t lock_addr ; **Parameters** lock_addr - Specifies the address of the lock word.

Index	Kernel service
8	**drw_lock_try_write** **Purpose** Immediately acquire a disabled read/write lock in write-exclusive mode if available. **Syntax** #include<sys/lock_def.h> boolean_t drw_lock try_write(lock_addr); drw_lock_t lock_addr ; **Parameters** lock_addr - Specifies the address of the lock word to lock.

1.9 DBX enhancements

The following sections discuss the dbx enhancements that were first made available in AIX V7.1 and AIX V6.1 TL06.

1.9.1 Dump memory areas in pointer format

A new option (`p` to print a pointer or address in hexadecimal format) is added to the dbx `display` subcommand to print memory areas in pointer format. Example 1-6 displays five pointers (32-bit) starting from address location 0x20000a90.

Example 1-6   Display 32-bit pointers

```
(dbx) 0x20000a90 /5p
0x20000a90:  0x20000bf8 0x20000bb8 0x00000000 0x20000b1c
0x20000aa0:  0x00000000
```

Example 1-7 displays five pointers (64-bit) starting from address location 0x0ffffffffffffa88.

Example 1-7   Display 64-bit pointers

```
(dbx) 0x0ffffffffffffa88/5p
0x0ffffffffffffa88:  0x0000000110000644 0x0000000110000664
0x0ffffffffffffa98:  0x000000011000064c 0x0000000110000654
0x0ffffffffffffaa8:  0x000000011000065c
```

1.9.2 dbx environment variable print_mangled

A new dbx environment variable called print_mangled is added. It is used to determine whether to print the C++ functions in mangled form or demangled form. The default value of print_mangled is unset. If set, dbx prints mangled function names. This feature allows you to use both mangled and demangled C++ function names with dbx subcommands. This applies for binaries compiled in debug mode (-g compiled option) and for binaries compiled in non-debug mode.

Example 1-8 demonstrates exploiting the print_mangled environment variable while setting a break point in the function1() overloaded function.

Example 1-8   The print_mangeled dbx environment variable

```
(dbx) st in function1
1. example1.function1(char**)
2. example1.function1(int)
3. example1.function1(int,int)
Select one or more of [1 - 3]: ^C
(dbx) set $print_mangled
(dbx) st in function1
1. example1.function1__FPPc
2. example1.function1__Fi
3. example1.function1__FiT1
Select one or more of [1 - 3]: ^C
```

Example 1-9 demonstrates how to reset the print_mangled environment variable with the **unset** command.

Example 1-9   The unset print_mangled dbx environment variable

```
(dbx) unset $print_mangled
(dbx) st in function1
1. example1.function1(char**)
2. example1.function1(int)
3. example1.function1(int,int)
Select one or more of [1 - 3]:
```

1.9.3 DBX malloc subcommand enhancements

The following dbx `malloc` subcommand enhancements are made in AIX 7.1:

► The `malloc allocation` subcommand of dbx was allowed only when the AIX environment variable MALLOCDEBUG=log was set. This restriction is removed in AIX 7.1.

► The output of `malloc freespace` subcommand of dbx is enhanced to display the memory allocation algorithms. Example 1-10 displays the output of the `malloc freespace` subcommand.

Example 1-10   The malloc freespace dbx subcommand output

```
(dbx) malloc freespace
Freespace Held by the Malloc Subsystem:

    ADDRESS         SIZE HEAP      ALLOCATOR
0x20002d60      57120    0      YORKTOWN
(dbx)q
# export MALLOCTYPE=3.1

(dbx) malloc freespace
Freespace Held by the Malloc Subsystem:

    ADDRESS         SIZE HEAP      ALLOCATOR
0x20006028         16    0            3.1
0x20006048         16    0            3.1
.............
.............
(dbx)
```

► A new argument (the address of a memory location) is added to the `malloc` subcommand. This dbx subcommand will fetch and display the details of the node to which this address belongs.

Example 1-11 displays the address argument of the `malloc` subcommand.

Example 1-11   The address argument of the malloc subcommand

```
(dbx) malloc 0x20001c00
Address 0x20001c00 node details :

Status : ALLOCATED

    ADDRESS         SIZE HEAP      ALLOCATOR
0x20000c98       4104    0       YORKTOWN
(dbx)
```

```
(dbx) malloc 0x20002d60
Address 0x20002d60 node details :

Status : FREE

    ADDRESS         SIZE HEAP     ALLOCATOR
0x20002d60         57120   0       YORKTOWN
(dbx)
```

1.10 ProbeVue enhancements

In November 2007, AIX V6.1 introduced the ProbeVue dynamic tracing facility for
both performance analysis and problem debugging. ProbeVue uses the Vue
scripting and programming language to dynamically specify trace points and
provide the actions to run at the specified trace points. ProbeVue supports
location and event probe points, which are categorized by common
characteristics into probe types. Previous AIX releases support the following
probe types:

► User function entry probes for C programs (or `uft` probes)

► User function entry probes for C++ programs (or `uftxlc++` probes)

► User function entry probes for Java™ programs (or `uftjava` probes)

► System call entry or exit probes (or `syscall` probes)

► Extended system call entry and exit probes (or `syscallx` probes)

► System trace hook probes (or `systrace` probes)

► Probes that fire at specific time intervals (or `interval` probes)

ProbeVue associates a probe manager with each probe type. As such the probe
manager denotes the software code that defines and provides a set of probe
points of the same probe type to the ProbeVue dynamic tracing framework. AIX
supports the following probe managers:

► User function probe manager (`uft`, `uftxlc++`, `uftjava` probes)

► System call probe manager (`syscall` probes)

► Extended System Call Probe Manager (`syscallx` probes)

► System trace probe manager (`systrace` probes)

► Interval probe manager (`interval` probes)

The following features were added in AIX V7.1 and AIX V6.1 TL 6100-06 to further enhance the usability and functionality of the ProbeVue dynamic tracing facility:

► `uft` probe manager support for Fortran programs

► Introduction of user function exit probes

► Module name support in user function probes

► Dynamic tracing of C++ code without direct C++ compiler assistance

► New *associative array* data type for the Vue programming language

► Access to current process, thread, and user area related information

► Process specific scope of interval probes for profiling programs

1.10.1 User function probe manager for Fortran

The dynamic tracing capabilities of AIX have been extended by allowing ProbeVue to probe Fortran executables through the `uft` probe type. The probe specification, argument access and ProbeVue function usage in probe actions for Fortran function probes are similar to other `uft` probes with the following differences:

► ProbeVue supports all required basic data types but you have to map the Fortran data types to ProbeVue data types and use the same in the Vue script. The mapping of Fortran data types to ProbeVue data types is listed in Table 1-6.

Table 1-6   Fortran to ProbeVue data type mapping

Fortran data type	ProbeVue data type
INTEGER * 2	short
INTEGER * 4	int / long
INTEGER * 8	long long
REAL	float
DOUBLE PRECISION	double

Fortran data type	ProbeVue data type
| COMPLEX | No equivalent basic data type. This data type needs to be mapped to a structure as shown below:<br><br>```
typedef struct complex {
        float a;
        float b;
} COMPLEX;
``` |
LOGICAL	int (The Fortran standard requires logical variables to be the same size as INTEGER/REAL variables.)
CHARACTER	char
BYTE	signed char

► Fortran passes IN scalar arguments of internal procedures by value, and other arguments by reference. Arguments passed by reference should be accessed with copy_userdata().

► Routine names in a Fortran program are case insensitive. But, while specifying them in a ProbeVue script, they should be in lowercase.

► Fortran stores arrays in column-major form, whereas ProbeVue stores them in row-major form.

► Intrinsic or built-in functions cannot be probed with ProbeVue. All Fortran routines listed in the XCOFF symbol table of the executable or linked libraries can be probed. ProbeVue uses the XCOFF symbol table to identify the location of these routines. However, the prototype for the routine has to be provided by you and ProbeVue tries to access the arguments according to the prototype provided. For routines where the compiler mangles the names, the mangled name should be provided.

► While Fortran can have header files, most applications do not use this capability. ProbeVue does not support direct inclusion of Fortran header files. However, a mapping of Fortran data types to ProbeVue data types can be provided in a ProbeVue header file and specified with the **-I** option of the **probevue** command.

1.10.2 User function exit probes

Since the initial implementation of ProbeVue, user function entry probes are supported. AIX V7.1 and the related TL 6100-06 of AIX V6.1 also allow to probe

user function exits. The new keyword `exit` must be used in the location field of the uft probe point to enable the dynamic tracing of user function exits. The function return value can be accessed with the __rv built-in class variable. Example 1-12 shows a Vue script segment that enables the dynamic tracing of errors returned by the fictitious user function foo().

Example 1-12   Vue script segment for tracing foo() user function exits

```
/*To track the errors returned by foo() user function, you can write a
script like this*/

@@uft:$__CPID:*:foo:exit
        when (__rv < 0)
{
        printf("\nThe foo function failed with error code %d",__rv);
}
```

1.10.3 Module name support in user probes

The user function trace uft probe manager has been enhanced to allow the module name of a function to be specified for the uft and uftxlc++ probe types. (The uft and uftxlc++ probe types are associated with the same uft probe manager.) The third field of the uft and uftxlc++ 5-tuple probe specification no longer needs to be set to * (asterisk wildcard) as in the past but can now be used to limit the dynamic tracing for a given user function to the instances defined in a particular library or object name. Only archive and object names are allowed in a module name specification.

Example 1-13 shows several options to define library module names for the fictitious user function foo(). The foo() function may be included in the libc.a archive or the shr.o object module. (In any of the uft probe specifications the dynamic tracing is limited to the foo() function calls made by the process with the process ID 4094.)

Example 1-13   Module name specification syntax

```
@@uft:4094:*:foo:entry              #Function foo in any module
@@uft:4094:libc.a:foo:entry         #Function foo in any module in any
                                     archive named libc.a
@@uft:4094:libc.a(shr.o):foo:entry  #Function foo in the shr.o module
                                     in any archive named libc.a
```

1.10.4 ProbeVue support for pre-compiled C++ header files

In previous AIX releases Probevue required the installation of the IBM XL C/C++ compiler on every system where dynamic tracing of C++ applications was intended to be done. The C++ compiler support was needed to process the C++ header files included in the ProbeVue script.

Beginning with AIX V7.1 and AIX V6.1 TL 6100-06, the C++ header files can be preprocessed on a dedicated system where the C++ compiler is available by using the `-P` option of the **probevue** command. By default **probevue** will generate an output file with the same name as the input C++ header files but extended with a `.Vue` suffix. The preprocessed header files can then be transferred to any other system to be used there as include files with the `-I` option of the **probevue** command to trace C++ applications.

1.10.5 Associative array data type

The Vue language accepts four special data types in addition to the traditional C-89 data types:

► String data type

► List data type

► Timestamp data type

► Associative array data type

While the first three data types are supported since ProbeVue was initially implemented in AIX V6.1, the associative array data type is new to AIX V7.1 and AIX V6.1 TL 6100-06. An associative array is a map or look-up table consisting of a collection of keys and their associated values. There is a 1 to 1 mapping between keys and values. Associative arrays are supported by Perl, ksh93, and other programming languages.

The following operations are available for the associative array data type:

► Adding a key-value pair, updating value

► Searching a key

► Deleting a key

► Checking for a key

► Increment or decrement operation on the associative array values

► Printing the associative array contents

► Clearing the associative array contents

- ▶ Quantize on associative array
- ▶ Lquantize on associative array

1.10.6 Built-in variables for process- and thread-related information

In addition to the special built-in variables, __arg1 through __arg32, and __rv, the Vue programming language also defines a set of general-purpose built-in variables. Built-in class variables are essentially functions, but are treated as variables by ProbeVue. The list of supported general-purpose built-in variables has been extended by four additional variables to get access to process- and thread-related information:

__curthread   Built-in variable to access data related to the current thread.

__curproc   Built-in variable to access data related to the current process.

__ublock   Built-in variable providing access to the user area (process ublock) related information.

__mst   Built-in variable to access the hardware register content of the current thread's Machine State Save Area (MST).

These built-in variables cannot be used in systrace, BEGIN, and END probe points. Also they can be used in interval probes only if a process ID (PID) is specified. A set of members are defined for each built-in function which retrieve the data from the context of the thread or process running the probe.

Table 1-7 provides information that can be accessed using the -> operator on the __curthread built-in variable.

Table 1-7   Members of the __curthread built-in variable

Member name	Description
tid	Thread ID
pid	Process ID
policy	Scheduling policy
pri	Priority
cpuusage	CPU usage
cpuid	Processor to which the current thread is bound to
sigmask	Signal blocked on the thread

Member name	Description
lockcount	Number of kernel lock taken by the thread

Table 1-8 provides information that can be accessed using the -> operator on the __curproc built-in variable.

Table 1-8   Members of the __curproc built-in variable

Member name	Description
pid	Process ID
ppid	Parent process ID
pgid	Process group ID
uid	Real user ID
suid	Saved user ID
pri	Priority
nice	Nice value
cpu	Processor usage
adspace	Process address space
majflt	I/O page fault
minflt	Non I/O page fault
size	Size of image in pages
sigpend	Signals pending on the process
sigignore	Signals ignored by the process
sigcatch	Signals being caught by the process
forktime	Creation time of the process
threadcount	Number of threads in the process

Table 1-9 provides information that can be accessed using the -> operator on the __ublock built-in variable.

Table 1-9   Members of the __ublock built-in variable

Member name	Description
text	Start of text

Member name	Description
tsize	Text size (bytes)
data	Start of data
sdata	Current data size (bytes)
mdata	Maximum data size (bytes)
stack	Start of stack
stkmax	Stack max (bytes)
euid	Effective user ID
uid	Real user ID
egid	Effective group ID
gid	Real group ID
utime_sec	Process user resource usage time in seconds
stime_sec	Process system resource usage time in seconds
maxfd	Max fd value in user

Table 1-10 provides information that can be accessed using the -> operator on the __mst built-in variable.

Table 1-10   Members of the __mst built-in variable

Member name	Description
r1 — r10	General purpose register r1 to r10
r14 — r31	General purpose register r14 to r31
iar	Instruction address register
lr	Link register

1.10.7 Interval probes for profiling programs

The interval probe manager provides probe points that fire at a user-defined time interval. The probe points are not located in kernel or application code, but instead are based on wall clock time interval-based probe events.

The interval probe manager is useful for summarizing statistics collected over an interval of time. It accepts a 4-tuple probe specification in the following format:

```
@@interval:<pid>:clock:<time_interval>
```

In previous AIX releases the second field only accepted an asterisk (*) wild card and the interval probe was fired for all processes. A ProbeVue user had the option to reference the process ID of a particular thread through the use of the __pid built-in variable in an interval probe predicate to ensure that the probe is hit in the context of the given process.

But this configuration does not guarantee that the probe would be fired for the process at the specified intervals. This restriction has been lifted and a ProbeVue user can now also specify the process ID of a particular program in the second field of the interval probe 4-tuple. In this way an application can be profiled by interval-based dynamic tracing. Because of this new capability interval probes with specified process IDs are referred to as *profiling interval probes*. Note that only one profiling interval probe can be active for any given process.

Also, the stktrace() user-space access function and the __pname() built-in variable are now allowed in interval probes when a process ID is provided in the probe specification. The stktrace trace capture function formats and prints the stack trace and the general purpose __pname built-in function provides access to the process name of a traced thread.

In addition to the improved process scope control the granularity of the timer interval has been enhanced as well.

The initial implementation required to specify the timer interval in integral multiples of 100 ms. This requirement is still valid for interval probes without process ID. Thus, probe events that are apart by 100 ms, 200 ms, 300 ms, and so on, are the only ones allowed in non-profiling interval probes.

But for interval probes with process ID specified, non-privileged users are now entitled to specify intervals in integral multiples of 10 ms. Thus, probe events that are apart by 10 ms, 20 ms, 30 ms, and so on, are allowed for normal users in profiling interval probes. The global root user has an even higher flexibility to configure probe intervals. The time intervals only need to be greater or equal to the configurable minimum interval allowed for the global root user. The minimum timer interval can be set as low as 1 ms with the `probevctrl` command using the `-c` flag in conjunction with the `min_interval` attribute. The `min_interval` attribute value is always specified in milliseconds. The command `/usr/sbin/bosboot -a` must be run for a change to take effect in the next boot.

2

File systems and storage

This chapter describes the major AIX Version 7.1 enhancements that are part of the file system and connected storage, including:

► 2.1, "LVM enhancements" on page 30

► 2.2, "Hot files detection in JFS2" on page 35

29

2.1 LVM enhancements

This section discusses LVM enhancements in detail.

2.1.1 LVM enhanced support for solid-state disks

Solid®-state disks (SSDs) are a very popular option for enterprise storage requirements. SSDs are unique in that they do not have any moving parts and thus perform at electronic speeds without mechanical delays (moving heads or spinning platters) associated with traditional spinning Hard Disk Drives (HDDs). Compared to traditional HDDs, the characteristics of SSDs enable a higher level of I/O performance in terms of greater throughput and lower response times for random I/O. These devices are ideal for applications that require high IOPS/GB and/or low response times.

AIX V7.1 includes enhanced support in the AIX Logical Volume Manager (LVM) for SSD. This includes the capability for LVM to restrict a volume group (VG) to only contain SSDs and the ability to report that a VG only contains SSDs. This feature is also available in AIX V6.1 with the 6100-06 Technology Level.

Traditionally, a volume group can consist of physical volumes (PVs) from a variety of storage devices, such as HDDs. There was no method to restrict the creation of a volume group to a specific type of storage device. The LVM has been enhanced to allow for the creation of a volume group to a specific storage type, in this case SSDs. The ability to restrict a volume group to a particular type of disk can assist in enforcing performance goals for the volume group.

For example, a DB2® database may be housed on a set of SSDs for best performance. Reads and writes in that VG will only perform as fast as the slowest disk. For this reason it is best to restrict this VG to SSDs only. To maximize performance, the mixing of SSD and HDD hdisks in the same volume group must be restricted.

The creation, extension, and maintenance of an SSD VG must ensure that the restrictions are enforced. The following LVM commands have been modified to support this enhancement and enforce the restriction:

► `lsvg`

► `mkvg`

► `chvg`

► `extendvg`

► `replacepv`

The LVM device driver has been updated to support this enhancement. The changes to the LVM device driver and commands rely upon the successful identification of an SSD device. To determine whether a disk is an SSD, the IOCINFO operation is used on the disk's ioctl() function. Using the specified bits, the disk can be examined to determine if it is an SSD device. The structures, devinfo and scdk64 are both defined in `/usr/include/sys/devinfo.h`. If DF_IVAL (0x20) is set in the flags field of the devinfo structure, then the flags field in the scdk64 structure is valid. The flags can then be examined to see if DF_SSD (0x1) is set.

For information about configuring SSD disks on an AIX system, refer to the following websites:

http://www.ibm.com/developerworks/wikis/display/WikiPtype/Solid+State+Drives

http://www.ibm.com/developerworks/wikis/display/wikiptype/movies

To confirm the existence of the configured SSD disk on our lab system, we used the `lsdev` command, as shown in Example 2-1.

Example 2-1   Output from the lsdev command showing SSD disks

```
# lsdev -Cc disk
hdisk0  Available 01-08-00 Other SAS Disk Drive
hdisk1  Available 01-08-00 Other SAS Disk Drive
hdisk2  Available 01-08-00 Other SAS Disk Drive
hdisk3  Available 01-08-00 Other SAS Disk Drive
hdisk4  Available 01-08-00 SAS Disk Drive
hdisk5  Available 01-08-00 Other SAS Disk Drive
hdisk6  Available 01-08-00 SAS Disk Drive
hdisk7  Available 01-08-00 SAS Disk Drive
hdisk8  Available 01-08-00 Other SAS Disk Drive
hdisk9  Available 01-08-00 SAS RAID 0 SSD Array
hdisk10 Available 01-08-00 SAS RAID 0 SSD Array
hdisk11 Available 01-08-00 SAS RAID 0 SSD Array
```

The `mkvg` command accepts an additional flag, `-X`, to indicate that a new VG must reside on a specific type of disk. This effectively restricts the VG to this type of disk while the restriction exists. The following list describes the options to the `-X` flag.

-X none   This is the default setting. This does not enforce any restriction. Volume group creation can use any disk type.

-X SSD   At the time of creation, the volume group is restricted to SSD devices only.

In Example 2-2, we create an SSD restricted volume, named dbvg, using an SSD disk.

Example 2-2   Creating an SSD restricted VG

```
# lsdev -Cc disk | grep hdisk9
hdisk9  Available 01-08-00 SAS RAID 0 SSD Array
# mkvg -X SSD -y dbvg hdisk9
dbvg
```

Important: Once a PV restriction is turned on, the VG can no longer be imported on a version of AIX that does not support PV type restrictions.

Even if a volume group PV restriction is enabled and then disabled, it will no longer be possible to import it on a version of AIX that does not recognize the PV type restriction.

The use of the -I flag on a PV restricted VG is not allowed.

Two examples of when this limitation should be considered are:

► When updating the AIX level of nodes in a cluster. There will be a period of time when not all nodes are running the same level of AIX.

► When reassigning a volume group (exportvg/importvg) from one instance of AIX to another instance of AIX that is running a previous level of the operating system.

The **lsvg** command will display an additional field, PV RESTRICTION, indicating whether a PV restriction is set for a VG. If the VG has no restriction, the field will display none. The **lsvg** command output shown in Example 2-3 is for a volume group with a PV restriction set to SSD.

Example 2-3   The volume group PV RESTRICTION is set to SSD

```
# lsvg dbvg
VOLUME GROUP:     dbvg            VG IDENTIFIER:  00c3e5bc00004c000000012b0d2be925
VG STATE:         active          PP SIZE:        128 megabyte(s)
VG PERMISSION:    read/write      TOTAL PPs:      519 (66432 megabytes)
MAX LVs:          256             FREE PPs:       519 (66432 megabytes)
LVs:              0               USED PPs:       0 (0 megabytes)
OPEN LVs:         0               QUORUM:         2 (Enabled)
TOTAL PVs:        1               VG DESCRIPTORS: 2
STALE PVs:        0               STALE PPs:      0
ACTIVE PVs:       1               AUTO ON:        yes
MAX PPs per VG:   32512
MAX PPs per PV:   1016            MAX PVs:        32
LTG size (Dynamic): 256 kilobyte(s)  AUTO SYNC:    no
```

```
HOT SPARE:          no                        BB POLICY:      relocatable
MIRROR POOL STRICT: off
PV RESTRICTION:     SSD
```

The **chvg** command accepts an additional flag, **-X**, to set or change the device type restriction on a VG. The following list describes the options available.

-X none   Removes any PV type restriction on a VG.

-X SSD   Places a PV type restriction on the VG if all the underlying disks are of type SSD. An error message is displayed if one or more of the existing PVs in the VG do not meet the restriction.

In Example 2-4 we first remove the PV type restriction from the volume group and then set the PV type restriction to SSD.

Example 2-4   Changing the PV type restriction on a volume group

```
# chvg -X none dbvg
# lsvg dbvg
VOLUME GROUP:       dbvg                 VG IDENTIFIER:  00c3e5bc00004c000000012b0d2be925
VG STATE:           active               PP SIZE:        128 megabyte(s)
VG PERMISSION:      read/write           TOTAL PPs:      519 (66432 megabytes)
MAX LVs:            256                  FREE PPs:       519 (66432 megabytes)
LVs:                0                    USED PPs:       0 (0 megabytes)
OPEN LVs:           0                    QUORUM:         2 (Enabled)
TOTAL PVs:          1                    VG DESCRIPTORS: 2
STALE PVs:          0                    STALE PPs:      0
ACTIVE PVs:         1                    AUTO ON:        yes
MAX PPs per VG:     32512
MAX PPs per PV:     1016                 MAX PVs:        32
LTG size (Dynamic): 256 kilobyte(s)      AUTO SYNC:      no
HOT SPARE:          no                   BB POLICY:      relocatable
MIRROR POOL STRICT: off
PV RESTRICTION:     none

# chvg -X SSD  dbvg
# lsvg dbvg
VOLUME GROUP:       dbvg                 VG IDENTIFIER:  00c3e5bc00004c000000012b0d2be925
VG STATE:           active               PP SIZE:        128 megabyte(s)
VG PERMISSION:      read/write           TOTAL PPs:      519 (66432 megabytes)
MAX LVs:            256                  FREE PPs:       519 (66432 megabytes)
LVs:                0                    USED PPs:       0 (0 megabytes)
OPEN LVs:           0                    QUORUM:         2 (Enabled)
TOTAL PVs:          1                    VG DESCRIPTORS: 2
STALE PVs:          0                    STALE PPs:      0
ACTIVE PVs:         1                    AUTO ON:        yes
MAX PPs per VG:     32512
MAX PPs per PV:     1016                 MAX PVs:        32
```

```
LTG size (Dynamic): 256 kilobyte(s)        AUTO SYNC:      no
HOT SPARE:           no                     BB POLICY:      relocatable
MIRROR POOL STRICT: off
PV RESTRICTION:      SSD
```

If we attempt to create a volume group, using a non-SSD disk with an SSD PV type restriction, the command will fail, as shown in Example 2-5.

Example 2-5   Attempting to create an SSD restricted VG with a non-SSD disk

```
# lsdev -Cc disk | grep hdisk1
hdisk1  Available 01-08-00 Other SAS Disk Drive
# mkvg -X SSD -y dbvg hdisk1
0516-1930 mkvg: PV type not valid for VG restriction.
        Unable to comply with requested PV type restriction.
0516-1397 mkvg: The physical volume hdisk1, will not be added to
the volume group.
0516-862 mkvg: Unable to create volume group.
```

Access to and control of this functionality is available via LVM commands only. At this time there are no SMIT panels for **mkvg** or **chvg** to set or change the restriction.

The **extendvg** and **replacepv** commands have been modified to honor any PV type restrictions on a volume group. For example, when adding a disk to an existing volume group with a PV restriction of SSD, the **extendvg** command ensures that only SSD devices are allowed to be assigned, as shown in Example 2-6.

If you attempt to add a mix of non-SSD and SSD disks to an SSD restricted volume group, the command will fail. If any of the disks fail to meet the restriction, all of the specified disks are not added to the volume group, even if one of the disks is of the correct type. The disks in Example 2-6 are of type SAS (hdisk7) and SSD (hdisk10). So even though hdisk10 is SSD, the volume group extension operation does not add it to the volume group because hdisk7 prevents it from completing successfully.

Example 2-6   Attempting to add a non-SSD disk to an SSD restricted volume group

```
# lsdev -Cc disk | grep hdisk7
hdisk7  Available 01-08-00 SAS Disk Drive
# extendvg -f dbvg hdisk7
0516-1254 extendvg: Changing the PVID in the ODM.
0516-1930 extendvg: PV type not valid for VG restriction.
        Unable to comply with requested PV type restriction.
0516-1397 extendvg: The physical volume hdisk7, will not be added to
```

```
the volume group.
0516-792 extendvg: Unable to extend volume group.

# lsdev -Cc disk | grep hdisk7
hdisk7  Available 01-08-00 SAS Disk Drive
# lsdev -Cc disk | grep hdisk10
hdisk10 Available 01-08-00 SAS RAID 0 SSD Array
# extendvg -f dbvg hdisk7 hdisk10
0516-1930 extendvg: PV type not valid for VG restriction.
        Unable to comply with requested PV type restriction.
0516-1397 extendvg: The physical volume hdisk7, will not be added to
the volume group.
0516-1254 extendvg: Changing the PVID in the ODM.
0516-792 extendvg: Unable to extend volume group.
```

When using the `replacepv` command to replace a disk, in an SSD restricted VG, the command will allow disks of that type only. If the destination PV is not the correct device type, the command will fail.

Currently, only the SSD PV type restriction is recognized. In the future, additional strings may be added to the PV type definition, if required, to represent newly supported technologies.

Mixing both non-SSD and SSD disks in a volume group that does not have a PV type restriction is still possible, as shown in Example 2-7. In this example we created a volume group with a non-SSD disk (`hdisk7`) and an SSD disk (hdisk9). This will work because we did not specify a PV restriction with the `-X SSD` option with the `mkvg` command.

Example 2-7   Creating a volume with both non-SSD and SSD disks

```
# lsdev -Cc disk | grep hdisk7
hdisk7  Available 01-08-00 SAS Disk Drive
# lsdev -Cc disk | grep hdisk9
hdisk9  Available 01-08-00 SAS RAID 0 SSD Array
# mkvg -y dbvg hdisk7 hdisk9
dbvg
```

2.2 Hot files detection in JFS2

Solid-state disks (SSDs) offer a number of advantages over traditional hard disk drives (HDDs). With no seek time or rotational delays, SSDs can deliver substantially better I/O performance than HDDs. The following white paper,

Positioning Solid State Disk (SSD) in an AIX environment, discusses these advantages in detail:

http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101560

In order to maximize the benefit of SSDs it is important to only place data on them that requires high throughput and low response times. This data is referred to as *hot* data or *hot* files. Typically a *hot* file can be described as a file that is read from or written to frequently. It could also be a file that is read from or written to in large chunks of data.

Before making a decision to move suspected hot files to faster storage (for example SSDs), users of a file system need to determine which files are actually hot. The files must be monitored for a period of time in order to identify the best candidates.

AIX V7.1 includes enhanced support in the JFS2 file system for solid-state disks (SSDs). JFS2 has been enhanced with the capability to capture and report per-file statistics related to the detection of hot files that can be used to determine whether a file should be placed on an SSD. These capabilities enable applications to monitor and determine optimal file placement. This feature is also available in AIX V6.1 with the 6100-06 Technology Level.

JFS2 Hot File Detection (HFD) enables the collection of statistics relating to file usage on a file system. The user interface to HFD is through programming functions only. HFD is implemented as a set of ioctl function calls. The enhancement is designed specifically so that application vendors can integrate this function into their product code.

There is no AIX command line interface to the JFS2 HFD function or the statistics captured by HFD ioctl function calls. However, the `filemon` command can be used to provide global hot file detection for all file systems, logical volumes and physical disks on a system.

These calls are implemented in the j2_ioctl function, where any of the HFD_* ioctl calls cause the j2_fileStats function to be called. This function handles the ioctl call and returns zero for success, or an error code on failure. When HFD is active in a file system, all reads and writes of a file in that file system cause HFD counters for that file to be incremented. When HFD is inactive, the counters are not incremented.

The HFD mechanism is implemented as several ioctl calls. The calls expect an open file descriptor to be passed to them. It does not matter which file in the file system is opened for this, because the system simply uses the file descriptor to identify the file system location and lists or modifies the HFD properties for the JFS2 file system.

The ioctl calls are defined in the `/usr/include/sys/hfd.h` header file. The contents of the header file are shown in Example 2-8.

Example 2-8   The /usr/include/sys/hfd.h header file

```
/* IBM_PROLOG_BEGIN_TAG                                               */
/* This is an automatically generated prolog.                        */
/*                                                                    */
/* $Source: aix710 bos/kernel/sys/hfd.h 1$ */
/*                                                                    */
/* COPYRIGHT International Business Machines Corp. 2009,2009           */
/*                                                                    */
/* Pvalue: p3 */
/* Licensed Materials - Property of IBM                               */
/*                                                                    */
/* US Government Users Restricted Rights - Use, duplication or        */
/* disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  */
/*                                                                    */
/* Origin: 27 */
/*                                                                    */
/* $Header: @(#) 1 bos/kernel/sys/hfd.h, sysj2, aix710, 0950A_710 2009-11-30T13:35:35-06:00$ */
/*                                                                    */
/* IBM_PROLOG_END_TAG                                                 */

/* %Z%%M%       %I%  %W% %G% %U%  */


/*
 * COMPONENT_NAME: (SYSJ2) JFS2 Physical File System
 *
 * FUNCTIONS: Hot Files Detection (HFD) subsystem header
 *
 * ORIGINS: 27
 *
 * (C) COPYRIGHT International Business Machines Corp. 2009
 * All Rights Reserved
 * Licensed Materials - Property of IBM
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
 */

#ifndef _H_HFD
#define _H_HFD

#include <sys/types.h>
#include <sys/ioctl.h>

#define HFD_GET        _IOR('f', 118, int)          /* get HFD flag */
#define HFD_SET        _IOW('f', 117, int)          /* set HFD flag */
```

```
#define HFD_END           _IOW('f', 116, int)          /* terminate HFD */
#define HFD_QRY           _IOR('f', 115, hfdstats_t)    /* qet HFD stats */

/* Hot File Detection (HFD) ioctl specific structs and flags { */

typedef struct per_file_counters {
        ino64_t         c_inode;
        uint64_t        c_rbytes;
        uint64_t        c_wbytes;
        uint64_t        c_rops;
        uint64_t        c_wops;
        uint64_t        c_rtime;
        uint64_t        c_wtime;
        uint32_t        c_unique;
} fstats_t;

typedef struct hfd_stats_request {
        uint64_t        req_count;
        uint32_t        req_flags;
        uint32_t        req_resrvd;
        uint64_t        req_cookie;
        fstats_t        req_stats[1];
} hfdstats_t;

/* } Hot File Detection (HFD) ioctl specific structs and flags */

#endif /* _H_HFD */
```

The HFD ioctl calls are summarized as follows:

HFD_GET   A file descriptor argument is passed to this call, which contains an open file descriptor for a file in the desired file system. This ioctl call takes a pointer to an integer as its argument and returns the status of the HFD subsystem for the file system. If the returned integer is zero, then HFD is not active. Otherwise, HFD is active. All users can submit this ioctl call.

HFD_SET   A file descriptor argument is passed to this call, which contains an open file descriptor for a file in the desired file system. This ioctl call takes a pointer to an integer as its argument. The integer needs to be initialized to zero before the call to disable HFD and to a non-zero to activate it. If the call would result in no change to the HFD state, no action is performed, and the call returns with success. If the user is not authorized, the call will return an EPERM error condition.

If HFD has not been active for the file system since it was mounted, it is initialized and memory is allocated for the HFD

counters. Additional memory is allocated as required as files in the file system are read from, or written to. The HFD file counters are initialized to zeroes when they are allocated or reused (for example, when a file is deleted). When the file system is unmounted, the HFD subsystem is terminated in the file system. The allocated memory is freed at this time. If HFD is deactivated, the counters are not incremented, but they are not reset either.

HFD_END   This call causes the HFD subsystem to be terminated and memory allocated to it to be freed. Calling it while HFD is active in the file system causes an EBUSY error condition. If the user is not authorized, the call will return an EPERM error condition.

If the file system is activated again, the statistics counters will restart from zeroes. A file descriptor argument is passed to this call, which contains an open file descriptor for a file in the desired file system. This ioctl call takes only a NULL pointer as an argument. Passing any other value causes an EINVAL error condition.

HFD_QRY   A file descriptor argument is passed to this call, which contains an open file descriptor for a file in the desired file system. This ioctl call takes a pointer to an hfdstats_t structure as an argument. The structure must be initialized before the call, and it returns the current HFD statistics for active files in the file system.

If the argument is not a valid pointer, the call returns an EFAULT error condition. If the pointer is NULL, the call returns an EINVAL error condition. If HFD is not active, the call returns an ENOENT error condition. Depending on the passed-in values for the fields in the structure, the call returns different data in the same structure. If the user is not authorized, the call returns an EPERM error condition.

The statistics counters for an active file are not reset. To find hot files, the HFD_QRY ioctl call must be performed many times, over a set time interval. The statistics for each interval are calculated by subtracting the statistics values for each counter at the end and at the beginning of the interval.

The hfdstats_t structure contains a one-element long array of fstats_t structures. Each structure contains the following fields: c_inode, c_unique, c_rops, c_wops, c_rbytes, c_wbytes, c_rtime, and c_wtime. These fields contain statistics of the file in question. The c_rops and c_wops fields contain the count of the read and write operations for the file. The c_rbytes and c_wbytes fields contain the number of bytes read from or written to the file. The c_rtime and c_wtime fields contain, respectively, the total amount of time spent in the read and write operations for

the file. The c_inode and c_unique fields contain the inode and generation numbers of the file.

In addition, the mount and unmount functionality has been enhanced to allocate and free data structures required by the HFD subsystem. The j2_rdwr function has also been modified to increment HFD statistics counters. The file statistics collected for a file system are not saved when the file system is unmounted.

It is possible to activate, deactivate and terminate HFD for a file system. Per-file statistics are collected and can be retrieved via the programming interface. If HFD is activated for a file system, there is minimal impact to the file system's performance and resource usage. After HFD is activated for a file system, its inodes will be write locked for the first read or write operation. A performance overhead associated with HFD would not be more than 2 % on a system with adequate memory, as measured by a standard file system test benchmark for read/write activity.

HFD uses memory to store the per-file statistics counters. This may cause a large increase in memory use while HFD is active. The extra memory is kept even when HFD is no longer active, until the file system is unmounted or HFD is terminated.

The memory requirement is about 64 bytes per active file. A file is considered active if it has had at least one read or write while HFD has been active for the file system. However, the extra memory will not grow larger than the memory required by the number of files equal to the maximum number of inodes in the JFS2 inode cache (as specified by the j2_inodeCacheSize ioo tuning parameter).

Since HFD is used only for statistics, its memory is not saved during a system dump, or live dump. The kdb and KDB utilities have been enhanced to print the values of the mount inode i_j2fstats and the inode i_fstats fields. There are no additional trace hooks associated with HFD. The HFD memory heap can be inspected using kdb `heap`, `pile`, and `slab` commands.

Only authorized users may change the state of or retrieve statistics for an HFD-enabled file system. HFD uses the PV_KER_EXTCONF privilege. To enable a program to modify the HFD state or to query active files, the program must have the appropriate privilege first. For example, the following set of commands would allow all users to run a program named /tmp/test to enable HFD on the /testfs file system:

```
# setsecattr -c secflags=FSF_EPS accessauths=ALLOW_ALL
innateprivs=PV_KER_EXTCONF /tmp/test
# setkst
# su - guest
$ /tmp/test /testfs ON
```

```
HFD is now active
```

The following sample code demonstrates how the HFD_QRY ioctl call can be used to find hot files in a file system, as shown in Example 2-9 on page 41.

The print_stats function would need to run qsort (or another sort function) to find hot files in the file system. The comparison function for the sort would need to have the selection criteria for a hot file built in, for example whether to use the number of bytes read or number of bytes written field. It also needs to check the c_inode and c_unique numbers and subtract the statistics counters of the two arrays to determine the count for the interval.

The req_count field allows you to determine how large an array should be set in order to allocate data. The req_stats array contains entries for the statistics for each active file at the time of the HFD_QRY call. Each entry has the inode number of the file in the c_inode field. If a file is deleted, its entry becomes available for reuse by another file. For that reason, each entry also contains a c_unique field, which is updated each time the c_inode field changes.

The ioctl (fd, HFD_QRY, &Query) call returns per-file I/O statistics in the Query structure. There are three methods for using the HFD_QRY ioctl call.

► To query a single file, the passed-in value for req_count is zero. The c_inode field is also zero. This call returns file statistics for the file being referenced by the passed-in file descriptor. This method is useful for monitoring a single file.

► To query all active files, the passed-in field for req_count is zero. This call returns with the req_count field set to the number of elements needed in the req_stats array. The size of the array is set so that all of the data available at that point (that is the number of all active files) is stored.

► To query some active files in a file system, the passed-in field for req_count is set to a positive value. This call returns up to this many entries (req_count) in the req_stats array. If the passed-in value of the req_stats array is large enough to contain the number of active files, the req_cookie field is set to zero on return. HFD_QRY is called repeatedly until all entries are returned.

Important: The example code is for demonstration purposes only. It does not cater for any error handling, and does not take into account potential changes in the number of active files.

Example 2-9   Example HFD_QRY code

```
int          fd, SetFlag, Count;
hfdstats_t   Query;
hfdstats_t   *QueryPtr1, *QueryPtr2;
```

```
fd = open(./filesystem., O_RDONLY);      /* get a fd */
SetFlag = 1;
ioctl(fd, HFD_SET, &SetFlag);            /* turn on HFD */
Query.req_count = 0;
ioctl(fd, HFD_QRY, &Query);              /* find no of entries */
Count = Query.req_count + 1000; /* add some extras */
Size = sizeof(Query) + (Count . 1) * sizeof(fstats_t);
QueryPtr1 = malloc(Size);
QueryPtr2 = malloc(Size);
QueryPtr2->req_count = Count;
QueryPtr2->req_cookie = 0;
ioctl(fd, HFD_QRY, QueryPtr2);           /* get the data in 2 */
while (Monitor) {
             sleep(TimeInterval);
             QueryPtr1->req_count = Count;
             QueryPtr1->req_cookie = 0;
             ioctl(fd, HFD_QRY, QueryPtr1);  /* get the data in 1 */
             print_stats(QueryPtr1, QueryPtr2); /* print stats 1 - 2 */
sleep(TimeInterval);
             QueryPtr2->req_count = Count;
             QueryPtr2->req_cookie = 0;
             ioctl(fd, HFD_QRY, QueryPtr2);  /* get the data in 2 */
             print_stats(QueryPtr2, QueryPtr1); /* print stats 2 - 1 */
}
SetFlag = 0;
ioctl(fd, HFD_SET, &SetFlag);            /* turn off HFD */
ioctl(fd, HFD_END, NULL);                /* terminate HFD */
```

3

Workload Partitions and resource management

This chapter discusses Workload Partitions (WPARs). WPARs are virtualized software-based partitions running within an instance of AIX. They are available in AIX V7.1 and AIX V6.1. This chapter contains the following sections:

► 3.1, "Trusted kernel extension loading and configuration" on page 44
► 3.2, "WPAR list of features" on page 50
► 3.3, "Versioned Workload Partitions (VWPAR)" on page 50
► 3.4, "Device support in WPAR" on page 68
► 3.5, "WPAR RAS enhancements" on page 95
► 3.6, "WPAR migration to AIX Version 7.1" on page 98

3.1 Trusted kernel extension loading and configuration

Trusted kernel extension loading and configuration allows the global administrator to select a set of kernel extensions that can then be loaded from within a system WPAR.

By default, dynamic loading of a kernel extension in a WPAR returns a message:

```
sysconfig(SYS_KLOAD): Permission denied
```

In the following examples, Global> will refer to the prompt for a command issued in the Global instance of AIX. # will be the prompt inside the WPAR.

3.1.1 Syntax overview

As user, a new flag `-X` for the `mkwpar` and `chwpar` commands is available. Multiple `-X` flags can be specified to load multiple kernel extensions.

The syntax described in man pages for the commands is as follows:

```
-X [exportfile=/path/to/file |[kext=[/path/to/extension|ALL]]
        [local=yes | no] [major=yes | no]
```

where the specification can be direct (using `kext=`) or through a stanza (`exportfile=`). It will work when private to a WPAR or shared with Global.

To remove an explicit entry for an exported kernel extension, use the following command:

```
chwpar -K -X [kext=/path/to/extension|ALL] wparname
```

Consideration: If the kernel extension is loaded inside a workload partition, it will not be unloaded from the Global until the WPAR is stopped or rebooted. A restart of the workload partition will be required to completely unexport the kernel extension from the workload partition.

The kext path specification must match a value inside the workload partition's configuration file. This must either be a fully qualified path or ALL if kext=ALL had previously been used.

3.1.2 Simple example monitoring

The following reference to kernel extension loading, on the IBM DeveloperWorks website, provides a good examples. Refer to *Writing AIX kernel extensions* at the following location:

http://www.ibm.com/developerworks/aix/library/au-kernelext.html

Using the example from that site with a default WPAR creation would result in output similar to what is shown in Example 3-1.

Example 3-1   Creation of a simple WPAR

```
Global> mkwpar -n testwpar
mkwpar: Creating file systems...
 /
 /home
 /opt
 /proc
 /tmp
 /usr
 /var
.........
Mounting all workload partition file systems.
x ./usr
syncroot: Processing root part installation status.
syncroot: Installp root packages are currently synchronized.
syncroot: RPM root packages are currently synchronized.
syncroot: Root part is currently synchronized.
syncroot: Returns Status = SUCCESS
Workload partition testwpar created successfully.
mkwpar: 0960-390 To start the workload partition, execute the following
as root: startwpar [-v] testwpar

Global> startwpar testwpar
Starting workload partition testwpar.
Mounting all workload partition file systems.
Loading workload partition.
Exporting workload partition devices.
Exporting workload partition kernel extensions.
Starting workload partition subsystem cor_test.
0513-059 The cor_test Subsystem has been started. Subsystem PID is
7340192.
Verifying workload partition startup.
```

When the WPAR is created, Example 3-2 shows how to access it and see if we can load a kernel extension.

Example 3-2   Trying to load a kernel extension in a simple WPAR

```
Global> clogin testwpar
**********************************************************************
*
*   Welcome to AIX Version 7.1!
* *
*   Please see the README file in /usr/lpp/bos for information pertinent
to    *
*   this release of the AIX Operating System.
*
* *
**********************************************************************
# ls
Makefile          hello_world.kex  loadkernext.o     sample.log
README            hello_world.o    main
hello_world.c     loadkernext      main.c
hello_world.exp   loadkernext.c    main.o
# ./loadkernext -q hello_world.kex
Kernel extensionKernel extension is not present on system.
# ./loadkernext -l hello_world.kex
sysconfig(SYS_KLOAD): Permission denied
```

As expected, we are unable to load the kernel extension (`Permission denied`).

The aim is to create a new system WPAR with the kernel extension parameter as shown in Example 3-3 using the `-X` parameter of the `mkwpar` command. We verify the existence of the kernel extension in the Global instance.

Example 3-3   Successful loading of kernel extension

```
Global> mkwpar -X kext=/usr/src/kernext/hello_world.kex local=yes -n testwpar2
mkwpar: Creating file systems...
 /
 /home
 /opt
 /proc
....
syncroot: Processing root part installation status.
syncroot: Installp root packages are currently synchronized.
syncroot: RPM root packages are currently synchronized.
syncroot: Root part is currently synchronized.
syncroot: Returns Status = SUCCESS
```

```
Workload partition testwpar2 created successfully.
mkwpar: 0960-390 To start the workload partition, execute the following as root:
startwpar [-v] testwpar2

Global> cd /usr/src/kernext
Global> ./loadkernext -q hello_world.kex
Kernel extensionKernel extension is not present on system.
Global> ./loadkernext -l hello_world.kex
Kernel extension kmid is 0x50aa2000.
Global> genkex | grep hello
f1000000c0376000    2000 hello_world.kex
Global> ls
Makefile        hello_world.kex  loadkernext.o    sample.log
README          hello_world.o    main
hello_world.c   loadkernext      main.c
hello_world.exp loadkernext.c    main.o
Global> cat sample.log
Hello AIX World!
```

The **loadkernext -q** command queries the state of the module. The **-l** option is used for loading the module. If the command is successful, it returns the kmid value. The **genkex** command also confirms that the kernel extension is loaded on the system. The loaded module will write output to sample.log file in the current working directory.

3.1.3 Enhancement of the lspwar command

The **lswpar** command has been enhanced with the flag **X** to list detailed kernel extension information for each requested workload partition in turn, as shown in Example 3-4.

Example 3-4   Parameter -X of the lswpar command

```
Global> lswpar -X
lswpar: 0960-679 testwpar2 has no kernel extension configuration.
Name   EXTENSION NAME                    Local  Major  Status
-----------------------------------------------------------------
test2  /usr/src/kernext/hello_world.kex  yes    no     ALLOCATED
```

3.1.4 mkwpar -X local=yes|no parameter impact

Since we specified the parameter local=yes in the previous example (Example 3-3 on page 46), the GLOBAL instance does not see that kernel

extension—it is private to the WPAR called testwpar2. The query command in Example 3-5 shows it is not running on the system.

Example 3-5   Loading kernel extension

```
Global> uname -a
AIX Global 1 7 00F61AA64C00
Global> cd /usr/src/kernext
Global> ./loadkernext -q hello_world.kex
Kernel extension is not present on system.
```

A change of that parameter to local=no will share the extension with the Global as demonstrated in the output shown in Example 3-6.

Example 3-6   Changing type of kernel extension and impact to Global

```
Global> chwpar -X local=no kext=/usr/src/kernext/hello_world.kex
testwpar2
Global> lswpar -X
lswpar: 0960-679 testwpar2 has no kernel extension configuration.
Name    EXTENSION NAME                    Local   Major   Status
-----------------------------------------------------------------
test2  /usr/src/kernext/hello_world.kex  no      no      ALLOCATED

Global> startwpar testwpar2
Starting workload partition testwpar2.
Mounting all workload partition file systems.
Loading workload partition.
Exporting workload partition devices.
Exporting workload partition kernel extensions.
Starting workload partition subsystem cor_test2.
0513-059 The cor_test2 Subsystem has been started. Subsystem PID is
10879048.
Verifying workload partition startup.
Global> pwd
/usr/src/kernext
Global> ./loadkernext -q hello_world.kex
Kernel extension is not available on system.
```

The last command (`./loadkernext -q hello_world.kex`) is verifying that it is allocated but not yet used.

But when we make use of it within the WPAR, it is available both in the WPAR and in the Global. Note that the kmid is coherent in both environments (Example 3-7 on page 49).

Example 3-7   WPAR and Global test of extension

```
Global> clogin testwpar2
**********************************************************************


*  Welcome to AIX Version 7.1!

*  Please see the README file in /usr/lpp/bos for information pertinent
to    *
*  this release of the AIX Operating System.
*
* *
**********************************************************************
Last login: Wed Aug 25 18:38:28 EDT 2010 on /dev/Global from 75011p01

# cd /usr/src/kernext
# ./loadkernext -q hello_world.kex
Kernel extension is not present on system.
# ./loadkernext -l hello_world.kex
Kernel extension kmid is 0x50aa3000.

# exit
Global> uname -a
AIX 75011p01 1 7 00F61AA64C00
Global> ./loadkernext -q hello_world.kex
Kernel extension is there with kmid 1353330688 (0x50aa3000).
Global> genkex | grep hello
f1000000c0378000      2000 hello_world.kex
```

> **Note:** The `mkwpar -X` command has updated the configuration file named
> `/etc/wpars/test2.cf` with a new entry related to that kernel extension:
>
> ```
> extension:
>         checksum =
> "4705b22f16437c92d9cd70babe8f6961e38a64dc222aaba33b8f5c9f4975981a:12
> 82772343"
>         kext = "/usr/src/kernext/hello_world.kex"
>         local = "no"
>         major = "no"
> ```

An unload of the kernel extension on one side would appear to be unloaded from
both sides.

3.2 WPAR list of features

With AIX 6.1 TL4 the capability to create a WPAR with its root file systems on a storage device dedicated to that WPAR was introduced. This is called a rootvg WPAR. With AIX 6.1 TL6, the capability to have VIOS-based VSCSI disks in a WPAR has been introduced. With AIX 7.1, the support of kernel extension load and VIOS disks and their management within a WPAR was added, allowing a rootvg WPAR that supports VIOS disks.

3.3 Versioned Workload Partitions (VWPAR)

A new product named AIX 5.2 Workload Partitions for AIX 7 supports the creation of an AIX 5.2 environment in a versioned workload partition (VWPAR). Applications running in a Versioned WPAR will interact with the legacy AIX environment in the user space.

All the features mentioned in 3.2, "WPAR list of features" on page 50 are supported in a Versioned WPAR.

This topic describes the support of that Versioned WPAR support with a runtime environment of level AIX 5.2 in an AIX 7.1 WPAR. Runtime environment means commands, libraries, and kernel interface semantics.

The examples refer to a Global> prompt when issued from the Global AIX instance. The # prompt is provided from within a Versioned WPAR.

3.3.1 Benefits

The capability to run an AIX 5.2 environment inside an AIX 7.1 WPAR has the following advantages:

▶ Ability to run AIX 5.2 on new hardware (POWER7 processor-based systems)

▶ Ability to extend service life for that old version of AIX

▶ Ability to run AIX 5.2 binary applications on new hardware without changing the user-space environment

3.3.2 Requirements and considerations

The AIX 5.2 Workload Partitions product has several considerations in order to transparently run AIX 5.2 in a WPAR on AIX 7.1.

> **Important:** AIX 5.2 Workload Partition for AIX 7 is an optional separate product (LPP) that runs on top of AIX 7.1

The requirements are as follows:

► For an AIX 5.2 system to be integrated in the Versioned WPAR, it must have the final service pack (TL10 SP8 or 5200-10-08) installed.

> **Take Note:** The AIX 5.2 environment is not provided with the LPP.

► The product will only be supported on POWER7 technology-based hardware.
► NFS server is not supported in a Versioned WPAR.
► Device support in the Versioned WPAR is limited to devices directly supported in an AIX 7.1 WPAR.
► No PowerHA support is available in a Versioned WPAR.
► Versioned WPAR needs to be private, meaning that /usr and /opt cannot be shared with Global.
► Some commands and libraries from the AIX 5.2 environment that have extensive dependencies on data from the kernel extensions are replaced with the corresponding 7.1 command or library.
► Some additional software may need to be installed into the Versioned WPAR.

Some additional considerations for the user:

► When a kernel extension is loaded in a WPAR 7.1, it is flagged as a private module (3.1, "Trusted kernel extension loading and configuration" on page 44). On the Global side, you may see multiple instances of the same module even if it is not used.
► Kernel extensions cannot be used to share data between WPARs.
► Versioned WPARs get support for /dev/[k]mem but it is limited to around 25 symbols (the symbols being used in AIX 5.2). There is no access to other symbols.

3.3.3 Creation of a basic Versioned WPAR AIX 5.2

Creation of a Versioned WPAR requires the following steps, discussed in detail in the following sections:

► Creating an AIX 5.2 `mksysb` image
► Installing the support images for Versioned WPAR

- ► Creating the Versioned WPAR
- ► Starting the WPAR and its management

mksysb image

From a running AIX 5.2 system, you must create an `mksysb` image using the `mksysb` command. This can be available as a file, a disk, a CD or DVD, or on tape.

As most of the AIX 5.2 systems used to have one root JFS file system, migration to the current layout will be handled at the time of WPAR creation. JFS file systems will also be restored as JFS2 file systems because a rootvg WPAR does not support JFS.

In our example, we have used an AIX 5.2 TL10 SP8 `mksysb` image file.

Install the required LPP for Versioned WPAR support

In order to install the appropriate LPPs in a Versioned WPAR during the WPAR creation, you need to have the following packages available in `/usr/sys/inst.images`:

- ► bos.wpars
- ► wio.common
- ► vwpar.52

On the installation media DVD, the LPP packages to install with `installp` command are called vwpar.images.52 and vwpar.images.base. When these two packages are installed, they will place the three required packages listed above into `/usr/sys/inst.images`.

If you do not have the required packages installed, you will receive a message stating that some software is missing, as shown in Example 3-8.

Example 3-8   Missing vwpar packages installation message

```
Global> mkwpar -C -B mksysb52_TL10_SP8 -n vers_wpar1
mkwpar: 0960-669 Directory /usr/sys/inst.images does not contain the
software required to create a versioned workload partition.
```

> **Note:** If you did a manual copy of the packages you need to execute the `inutoc` command to update the catalog file .toc to include the packages you just added.

Creating a basic Versioned WPAR

The command to create a system WPAR is `mkwpar`. It has been enhanced to support the creation of a Versioned WPAR. The command flags relating to the creation of a Versioned WPAR are:

`/usr/sbin/mkwpar ... [-C] [-E directory] [-B wparbackupdevice] [-D ... xfactor=n]`

-C   Specify Versioned WPAR creation. This option is valid only when additional versioned workload partition software has been installed.

-B   Specifies the 5.2 `mksysb` image to be used to populate the WPAR.

-D   xfactor=*n*. The new attribute *xfactor* of the -D option allows the administrator to control the expansion of a compressed JFS file system. The default value is 1 and the maximum value is 8.

-E *directory*   The directory that contains the filesets required to install the Versioned WPAR. The directory specification is optional because it is allowing an alternative location in place of `/usr/sys/inst.images` to be specified.

Running the command will populate the file systems from the `mksysb` image.

Since all JFS file systems will be restored as JFS2 file systems when creating a versioned workload partition with its own root volume group, and JFS does not support compression, the file system size may no longer be sufficient to hold the data. The new attribute *xfactor* of the -D option allows the administrator to control the expansion of the file system. The default value is 1 and the maximum value is 8.

Other results from the mkwpar command

For a Versioned WPAR, the `mkwpar` command will create namefs mounts for the /usr and /opt file systems from the Global in the mount list for the WPAR at /nre/usr and /nre/opt, respectively.

Simple Versioned WPAR creation output using an mksysb image file

The initial command using an `mksysb` image file called `mksysb52_TL10_SP8` would be:

`mkwpar -C -B mksysb52_TL10_SP8 -n vers_wpar1`

The output is similar to that shown in Example 3-9 on page 54.

Example 3-9   Simple Versioned WPAR creation

```
Global>  /usr/sbin/mkwpar -C -B mksysb52_TL10_SP8 -n vers_wpar1
Extracting file system information from backup...
mkwpar: Creating file systems...
 /
Creating file system '/' specified in image.data
 /home
Creating file system '/home' specified in image.data
 /opt
Creating file system '/opt' specified in image.data
 /proc
 /tmp
Creating file system '/tmp' specified in image.data
 /usr
Creating file system '/usr' specified in image.data
 /var
Creating file system '/var' specified in image.data
Mounting all workload partition file systems.
New volume on /var/tmp/mksysb52_TL10_SP8:
Cluster size is 51200 bytes (100 blocks).
The volume number is 1.
The backup date is: Tue Jun  8 12:57:43 EDT 2010
Files are backed up by name.
The user is root.
x         5473 ./bosinst.data
x         8189 ./image.data
x       133973 ./tmp/vgdata/rootvg/backup.data
x            0 ./home
x            0 ./home/lost+found
x            0 ./opt
x            0 ./opt/IBMinvscout
x            0 ./opt/IBMinvscout/bin
x         2428 ./opt/IBMinvscout/bin/invscoutClient_PartitionID
x     11781523 ./opt/IBMinvscout/bin/invscoutClient_VPD_Survey
x            0 ./opt/LicenseUseManagement
..................
The total size is 1168906634 bytes.
The number of restored files is 28807.
+-----------------------------------------------------------------------+
                     Pre-installation Verification...
+-----------------------------------------------------------------------+
Verifying selections...done
Verifying requisites...done
Results...
```

```
SUCCESSES
---------
  Filesets listed in this section passed pre-installation verification
  and will be installed.

  Selected Filesets
  -----------------
  bos.wpars 7.1.0.1                         # AIX Workload Partitions
  vwpar.52.rte 1.1.0.0                      # AIX 5.2 Versioned WPAR Runti...
  wio.common 6.1.3.0                        # Common I/O Support for Workl...

  << End of Success Section >>

FILESET STATISTICS
------------------
    3  Selected to be installed, of which:
        3  Passed pre-installation verification
  ----
    3  Total to be installed


+-----------------------------------------------------------------------+
                    Installing Software...
+-----------------------------------------------------------------------+

installp: APPLYING software for:
        bos.wpars 7.1.0.1
........
+-----------------------------------------------------------------------+
                    Summaries:
+-----------------------------------------------------------------------+

Installation Summary
--------------------
Name                    Level           Part        Event       Result
-------------------------------------------------------------------------
bos.wpars               7.1.0.1         USR         APPLY       SUCCESS
bos.wpars               7.1.0.1         ROOT        APPLY       SUCCESS
wio.common              6.1.3.0         USR         APPLY       SUCCESS
wio.common              6.1.3.0         ROOT        APPLY       SUCCESS
vwpar.52.rte            1.1.0.0         USR         APPLY       SUCCESS
vwpar.52.rte            1.1.0.0         ROOT        APPLY       SUCCESS
Workload partition vers_wpar1 created successfully.
```

```
mkwpar: 0960-390 To start the workload partition, execute the following as root:
startwpar [-v] vers_wpar1
```

Listing information about Versioned WPAR in the system

A new parameter `L` has been added to the `lswpar -t` command to list Versioned WPARs.

Example 3-10 shows the difference between the simple `lswpar` and the `lswpar -t L` commands.

Example 3-10   lswpar queries

```
Global> lswpar
Name          State  Type  Hostname   Directory             RootVG WPAR
-------------------------------------------------------------------------
vers_wpar1  D      S     vers_wpar1  /wpars/vers_wpar1   no
wpar1         D      S     wpar1        /wpars/wpar1         no

Global> lswpar -t L
Name          State  Type  Hostname   Directory             RootVG WPAR
-------------------------------------------------------------------------
vers_wpar1  D      S     vers_wpar1  /wpars/vers_wpar1   no
```

Example 3-11 shows the results when using several other options with the `lswpar` command. Information on kernel extensions can be viewed with the `-X` option. Device information for each WPAR can be viewed with the `-D` option. Mount information can be viewed with the `-M` option. The last query with **lswpar -M** shows that the WPAR file systems have been allocated in the Global system rootvg disk.

Example 3-11   Multiple lswpar queries over Versioned WPAR

```
Global> lswpar -X vers_wpar1
lswpar: 0960-679 vers_wpar1 has no kernel extension configuration.

Global> lswpar -D vers_wpar1
Name          Device Name     Type     Virtual Device  RootVG  Status
-------------------------------------------------------------------------
vers_wpar1  /dev/null       pseudo                              ALLOCATED
vers_wpar1  /dev/tty        pseudo                              ALLOCATED
vers_wpar1  /dev/console    pseudo                              ALLOCATED
vers_wpar1  /dev/zero       pseudo                              ALLOCATED
vers_wpar1  /dev/clone      pseudo                              ALLOCATED
vers_wpar1  /dev/sad        clone                               ALLOCATED
vers_wpar1  /dev/xti/tcp    clone                               ALLOCATED
```

```
vers_wpar1  /dev/xti/tcp6   clone                          ALLOCATED
vers_wpar1  /dev/xti/udp    clone                          ALLOCATED
vers_wpar1  /dev/xti/udp6   clone                          ALLOCATED
vers_wpar1  /dev/xti/unixdg clone                          ALLOCATED
vers_wpar1  /dev/xti/unixst clone                          ALLOCATED
vers_wpar1  /dev/error      pseudo                         ALLOCATED
vers_wpar1  /dev/errorctl   pseudo                         ALLOCATED
vers_wpar1  /dev/audit      pseudo                         ALLOCATED
vers_wpar1  /dev/nvram      pseudo                         ALLOCATED
vers_wpar1  /dev/kmem       pseudo                         ALLOCATED

Global> lswpar -M vers_wpar1
Name        MountPoint                  Device      Vfs     Nodename  Options
---------------------------------------------------------------------------
vers_wpar1  /wpars/vers_wpar1           /dev/fslv00 jfs2
vers_wpar1  /wpars/vers_wpar1/home      /dev/lv01   jfs
vers_wpar1  /wpars/vers_wpar1/nre/opt   /opt        namefs            ro
vers_wpar1  /wpars/vers_wpar1/nre/sbin  /sbin       namefs            ro
vers_wpar1  /wpars/vers_wpar1/nre/usr   /usr        namefs            ro
vers_wpar1  /wpars/vers_wpar1/opt       /dev/fslv01 jfs2
vers_wpar1  /wpars/vers_wpar1/proc      /proc       namefs            rw
vers_wpar1  /wpars/vers_wpar1/tmp       /dev/fslv02 jfs2
vers_wpar1  /wpars/vers_wpar1/usr       /dev/fslv03 jfs2
vers_wpar1  /wpars/vers_wpar1/var       /dev/fslv05 jfs2

Global> lsvg -l rootvg | grep vers
fslv00          jfs2    1    1    1   closed/syncd  /wpars/vers_wpar1
lv01            jfs     1    1    1   closed/syncd
/wpars/vers_wpar1/home
fslv01          jfs2    1    1    1   closed/syncd
/wpars/vers_wpar1/opt
fslv02          jfs2    1    1    1   closed/syncd
/wpars/vers_wpar1/tmp
fslv03          jfs2    18   18   1   closed/syncd
/wpars/vers_wpar1/usr
fslv05          jfs2    1    1    1   closed/syncd
/wpars/vers_wpar1/var
```

startwpar

The `startwpar` command gives a standard output, except that a message is
displayed stating that the WPAR is not configured as checkpointable. This is
because the file systems are on the Global root disk; see Example 3-12 on
page 58.

Example 3-12   startwpar of a Versioned WPAR

```
Global> startwpar vers_wpar1
Starting workload partition vers_wpar1.
Mounting all workload partition file systems.
Loading workload partition.
Exporting workload partition devices.
Exporting workload partition kernel extensions.
Starting workload partition subsystem cor_vers_wpar1.
0513-059 The cor_vers_wpar1 Subsystem has been started. Subsystem PID is 10289366.
startwpar: 0960-239 The workload partition vers_wpar1 is not configured to be
checkpointable.
Verifying workload partition startup.
```

Accessing a Versioned WPAR

To access a WPAR, you need to define the WPAR with an address and connect to it using **telnet** or **ssh** commands.

However, for some administrative commands you can use the **clogin** command to log on to the WPAR.

> **Note:** The **clogin** process is not part of the WPAR and prevents WPAR mobility.

Within the WPAR, you can list the file systems mounted as well as list the drivers loaded in a Versioned WPAR, as shown in Example 3-13.

Example 3-13   Commands in a Versioned WPAR

```
Global> clogin vers_wpar1
*******************************************************************************
*                                                                             *
*                                                                             *
*  Welcome to AIX Version 5.2!                                                *
*                                                                             *
*                                                                             *
*  Please see the README file in /usr/lpp/bos for information pertinent to    *
*  this release of the AIX Operating System.                                  *
*                                                                             *
*                                                                             *
*******************************************************************************
Last unsuccessful login: Tue Apr 13 12:35:04 2010 on /dev/pts/1 from
p-eye.austin.ibm.com
Last login: Tue Jun  8 11:53:53 2010 on /dev/pts/0 from varnae.austin.ibm.com
```

```
# uname -a
AIX vers_wpar1 2 5 00F61AA64C00
# df
Filesystem     512-blocks      Free %Used    Iused %Iused Mounted on
Global            131072    106664   19%     1754    13% /
Global            131072    126872    4%       17     1% /home
Global            786432    402872   49%     7044    14% /nre/opt
Global           1572864   1158728   27%    10137     8% /nre/sbin
Global           4849664   1184728   76%    41770    24% /nre/usr
Global            131072     35136   74%      778    16% /opt
Global                 -         -    -         -     - /proc
Global            131072    126520    4%       22     1% /tmp
Global           2359296    133624   95%    25300    59% /usr
Global            131072    111368   16%      350     3% /var
# lsdev
aio0       Available  Asynchronous I/O (Legacy)
inet0      Defined    Internet Network Extension
posix_aio0 Available  Posix Asynchronous I/O
pty0       Available  Asynchronous Pseudo-Terminal
sys0       Available  System Object
wio0       Available  WPAR I/O Subsystem
```

The command reports it is running an AIX 5.2 system. Its host name has been modified to be the WPAR name. AIX 7.1 binaries are found under the /nre/opt, /nre/sbin, and /nre/usr file systems.

The **lsdev** command reports the available devices in the Versioned WPAR. They are the ones expected to be in AIX 7.1 WPAR (3.4, "Device support in WPAR" on page 68).

Use of /nre commands in a Versioned WPAR

Some commands are available in the directory /nre/usr/bin. These are the AIX 7.1 binaries. Example 3-14 displays the result of using them in a Versioned WPAR. In our example, the AIX 5.2 commands are located in /usr. These files are not intended to be used directly in the Versioned WPAR. They are only intended to be used in situations where the native environment has to be used for proper behavior in the Versioned WPAR.

Example 3-14   Execution of a AIX 7.1 binary command in a Versioned WPAR

```
# /nre/usr/bin/who
Could not load program /nre/usr/bin/who:
Symbol resolution failed for who because:
        Symbol __strcmp (number 3) is not exported from dependent
```

```
        module /usr/lib/libc.a(shr.o).
      Symbol ___strcpy (number 5) is not exported from dependent
        module /usr/lib/libc.a(shr.o).
Examine .loader section symbols with the 'dump -Tv' command.


# /usr/bin/who
root        Global      Sep  2 15:48      (Global)
```

> **Note:** You should not attempt to execute the AIX 7.1 commands under /nre
> directly.

3.3.4 Creation of an AIX Version 5.2 rootvg WPAR

Because rootvg WPARs reside on a rootvg disk exported to the WPAR, which is
distinct from the Global system rootvg, it must be specified in the `mkwpar`
command by using the `-D` option.

The simplest `mkwpar` command to create a rootvg Versioned WPAR is:

```
mkwpar -D devname=hdisk? rootvg=yes [xfactor=[1-8]] [-O] -C -B
<mksysb_device> -n VersionedWPARname
```

The command has the following considerations:

► Multiple `-D` options can be specified if multiple disks have to be exported.

► The rootvg=yes specification means that these disks will be part of the WPAR
  rootvg. Otherwise, the disk would be exported to the WPAR as a data disk,
  separate from the rootvg.

► The `-O` flag overwrites the existing volume group data on the disk, or creates a
  new one.

► The xfactor parameter has been described in "Creating a basic Versioned
  WPAR" on page 53.

> **Note:** The storage devices exportable to a Version WPAR are devices that
> can be exported to an AIX 7.1 WPAR, and that includes devices not
> supported by standalone AIX 5.2.

Example 3-15 on page 61 shows the use of the `mkwpar` command to create a
Versioned WPAR using hdisk4 and the mksysb image called
mksysb52_TL10_SP8. The device hdisk4 is a disk without any volume group.
Therefore, there is no need to specify the `-O` (override) option on the `mkwpar`
command.

Example 3-15   rootvg Versioned WPAR creation

```
Global> mkwpar -C -B mksysb52_TL10_SP8 -n vers_wpar2 -D devname=hdisk4 rootvg=yes
<
Extracting file system information from backup...
Creating workload partition's rootvg. Please wait...
mkwpar: Creating file systems...
 /
Creating file system '/' specified in image.data
 /admin
 /home
Converting JFS to JFS2
Creating file system '/home' specified in image.data
 /opt
Creating file system '/opt' specified in image.data
 /proc
 /tmp
Creating file system '/tmp' specified in image.data
 /usr
Creating file system '/usr' specified in image.data
 /var
Creating file system '/var' specified in image.data
Mounting all workload partition file systems.
New volume on /var/tmp/mksysb52_TL10_SP8:
Cluster size is 51200 bytes (100 blocks).
The volume number is 1.
The backup date is: Tue Jun  8 12:57:43 EDT 2010
Files are backed up by name.
The user is root.
x         5473 ./bosinst.data
x         8189 ./image.data
x       133973 ./tmp/vgdata/rootvg/backup.data
x            0 ./home
x            0 ./home/lost+found
x            0 ./opt
x            0 ./opt/IBMinvscout
...........
+-------------------------------------------------------------------------------+
                          Summaries:
+-------------------------------------------------------------------------------+

Installation Summary
--------------------
Name                          Level          Part       Event       Result
-------------------------------------------------------------------------------
```

```
bos.net.nis.client           7.1.0.0      ROOT      APPLY      SUCCESS
bos.perf.libperfstat         7.1.0.0      ROOT      APPLY      SUCCESS
bos.perf.perfstat            7.1.0.0      ROOT      APPLY      SUCCESS
bos.perf.tools               7.1.0.0      ROOT      APPLY      SUCCESS
bos.sysmgt.trace             7.1.0.0      ROOT      APPLY      SUCCESS
clic.rte.kernext             4.7.0.0      ROOT      APPLY      SUCCESS
devices.chrp.base.rte        7.1.0.0      ROOT      APPLY      SUCCESS
devices.chrp.pci.rte         7.1.0.0      ROOT      APPLY      SUCCESS
devices.chrp.vdevice.rte     7.1.0.0      ROOT      APPLY      SUCCESS
devices.common.IBM.ethernet 7.1.0.0       ROOT      APPLY      SUCCESS
devices.common.IBM.fc.rte    7.1.0.0      ROOT      APPLY      SUCCESS
devices.common.IBM.mpio.rte 7.1.0.0       ROOT      APPLY      SUCCESS
devices.common.IBM.scsi.rte 7.1.0.0       ROOT      APPLY      SUCCESS
devices.fcp.disk.array.rte   7.1.0.0      ROOT      APPLY      SUCCESS
devices.fcp.disk.rte         7.1.0.0      ROOT      APPLY      SUCCESS
devices.fcp.tape.rte         7.1.0.0      ROOT      APPLY      SUCCESS
devices.scsi.disk.rte        7.1.0.0      ROOT      APPLY      SUCCESS
devices.tty.rte              7.1.0.0      ROOT      APPLY      SUCCESS
bos.mp64                     7.1.0.0      ROOT      APPLY      SUCCESS
bos.net.tcp.client           7.1.0.0      ROOT      APPLY      SUCCESS
bos.perf.tune                7.1.0.0      ROOT      APPLY      SUCCESS
perfagent.tools              7.1.0.0      ROOT      APPLY      SUCCESS
bos.net.nfs.client           7.1.0.0      ROOT      APPLY      SUCCESS
bos.wpars                    7.1.0.0      ROOT      APPLY      SUCCESS
bos.net.ncs                  7.1.0.0      ROOT      APPLY      SUCCESS
wio.common                   7.1.0.0      ROOT      APPLY      SUCCESS
Finished populating scratch file systems.
```
Workload partition vers_wpar2 created successfully.
```
mkwpar: 0960-390 To start the workload partition, execute the following as root:
startwpar [-v] vers_wpar2
```

When the Versioned WPAR is created, hdisk4 is allocated to the WPAR and
contains the rootvg for that WPAR. Example 3-16 shows that file system layout of
a rootvg Versioned WPAR is different from the layout of a non-rootvg Versioned
WPAR as shown in Example 3-11 on page 56.

Example 3-16   Rootvg Versioned WPAR file system layout

```
Global> lswpar -D | grep disk
vers_wpar2  hdisk4          disk                yes    ALLOCATED
Global> lswpar -M vers_wpar2
Name       MountPoint                        Device     Vfs    Nodename  Options
-------------------------------------------------------------------------------
vers_wpar2  /wpars/vers_wpar2                /dev/fslv10 jfs2
vers_wpar2  /wpars/vers_wpar2/etc/objrepos/wboot  /dev/fslv11  jfs2
vers_wpar2  /wpars/vers_wpar2/opt            /opt        namefs            ro
```

| vers_wpar2 | /wpars/vers_wpar2/usr | /usr | namefs | ro |

For our rootvg Versioned WPAR, two file systems, `/dev/fslv010` and `/dev/fslv11`, which will be used to bootstrap the WPAR, have been created. They are located on the Global rootvg disk.

Startwpar of a rootvg Versioned WPAR

For a rootvg Versioned WPAR, a minimal file system set is created in the Global's rootvg and is used to bootstrap the WPAR and synchronize device information between the WPAR and the Global. They are mounted as / and `/etc/objrepos/wboot` during startup of the WPAR. Then they are overmounted with the WPAR file systems.

Example 3-17   Startwpar of a rootvg Versioned WPAR

```
Global> startwpar vers_wpar2
Starting workload partition vers_wpar2.
Mounting all workload partition file systems.
Loading workload partition.
Exporting workload partition devices.
hdisk4 Defined
Exporting workload partition kernel extensions.
Starting workload partition subsystem cor_vers_wpar2.
0513-059 The cor_vers_wpar2 Subsystem has been started. Subsystem PID is 4456646.
startwpar: 0960-239 The workload partition vers_wpar2 is not configured to be
checkpointable.
Verifying workload partition startup.
```

Device information queries from a rootvg Versioned WPAR

The rootvg Versioned WPAR has all the standard file systems mounted from its own rootvg, plus read-only namefs mounts from the Global. These namefs mounts are the native runtime environment file systems called `/nre/usr`, `/nre/opt`, and `/nre/sbin`. There is also a root file system mounted from the Global to bootstrap the WPAR (see Example 3-16 on page 62) and a /etc/objrepos/wboot mount that is used to synchronize device information between the WPAR and the Global. The layout is displayed using the **df** command in Example 3-18.

Example 3-18   Devices and file systems in a rootvg Versioned WPAR

```
Global> clogin vers_wpar2
****************************************************************************
*                                                                        *
*                                                                        *
```

```
*  Welcome to AIX Version 5.2!                                              *
*                                                                          *
*                                                                          *
*  Please see the README file in /usr/lpp/bos for information pertinent to *
*  this release of the AIX Operating System.                               *
*                                                                          *
*                                                                          *
****************************************************************************
Last unsuccessful login: Tue Apr 13 12:35:04 2010 on /dev/pts/1 from
p-eye.austin.ibm.com
Last login: Tue Jun  8 11:53:53 2010 on /dev/pts/0 from varnae.austin.ibm.com

# uname -a
AIX vers_wpar2 2 5 00F61AA64C00
# df
Filesystem       512-blocks      Free %Used     Iused %Iused Mounted on
Global             131072     104472   21%      1795    14% /
/dev/hd4           131072     104472   21%      1795    14% /
Global            4849664    1184728   76%     41770    24% /nre/usr
Global             786432     402872   49%      7044    14% /nre/opt
Global            1572864    1158704   27%     10163     8% /nre/sbin
/dev/hd2          2359296     117536   96%     25300    62% /usr
/dev/hd10opt       131072      33088   75%       778    17% /opt
/dev/hd11admin     131072     128344    3%         4     1% /admin
/dev/hd1           131072     128344    3%         4     1% /home
/dev/hd3           131072     124472    6%        22     1% /tmp
/dev/hd9var        131072     109336   17%       350     3% /var
Global             131072     128336    3%         5     1% /etc/objrepos/wboot
Global                  -          -    -         -      - /proc
# lsdev
fscsi0     Available 00-00-02 WPAR I/O Virtual Parent Device
hd1        Available          Logical volume
hd2        Available          Logical volume
hd3        Available          Logical volume
hd4        Available          Logical volume
hd10opt    Available          Logical volume
hd11admin  Available          Logical volume
hd9var     Available          Logical volume
hdisk0     Available 00-00-02 MPIO Other DS4K Array Disk
inet0      Defined            Internet Network Extension
pty0       Available          Asynchronous Pseudo-Terminal
rootvg     Available          Volume group
sys0       Available          System Object
wio0       Available          WPAR I/O Subsystem
```

3.3.5 Content of the vwpar.52 package

The vwpar.52 package would install the following files in your WPAR. These are the files required to overlay 5.2 commands and libraries that have kernel data dependencies with an AIX 7.1 version of the file.

Example 3-19   The vwpar.52 lpp content

```
Cluster size is 51200 bytes (100 blocks).
The volume number is 1.
The backup date is: Wed Aug 11 20:03:52 EDT 2010
Files are backed up by name.
The user is BUILD.
          0 ./
       1063 ./lpp_name
          0 ./usr
          0 ./usr/lpp
          0 ./usr/lpp/vwpar.52
     189016 ./usr/lpp/vwpar.52/liblpp.a
          0 ./usr/lpp/vwpar.52/inst_root
       1438 ./usr/lpp/vwpar.52/inst_root/liblpp.a
          0 ./usr/aixnre
          0 ./usr/aixnre/5.2
          0 ./usr/aixnre/5.2/bin
       8718 ./usr/aixnre/5.2/bin/timex
       4446 ./usr/aixnre/5.2/bin/nrexec_wrapper
          0 ./usr/aixnre/5.2/ccs
          0 ./usr/aixnre/5.2/ccs/lib
          0 ./usr/aixnre/5.2/ccs/lib/perf
      40848 ./usr/aixnre/5.2/ccs/lib/librtl.a
     320949 ./usr/aixnre/5.2/ccs/lib/libwpardr.a
          0 ./usr/aixnre/5.2/lib
          0 ./usr/aixnre/5.2/lib/instl
     186091 ./usr/aixnre/5.2/lib/instl/elib
      60279 ./usr/aixnre/5.2/lib/instl/instal
    2008268 ./usr/aixnre/5.2/lib/liblvm.a
     291727 ./usr/aixnre/5.2/lib/libperfstat.a
       1012 ./usr/aixnre/5.2/lib/perf/libperfstat_updt_dictionary
          0 ./usr/aixnre/bin
       3524 ./usr/aixnre/bin/nre_exec
       4430 ./usr/aixnre/bin/nrexec_wrapper
          0 ./usr/aixnre/diagnostics
          0 ./usr/aixnre/diagnostics/bin
        939 ./usr/aixnre/diagnostics/bin/uspchrp
          0 ./usr/aixnre/lib
```

```
   0 ./usr/aixnre/lib/boot
   0 ./usr/aixnre/lib/boot/bin
1283 ./usr/aixnre/lib/boot/bin/bootinfo_chrp
1259 ./usr/aixnre/lib/boot/bin/lscfg_chrp
   0 ./usr/aixnre/lib/corrals
4446 ./usr/aixnre/lib/corrals/nrexec_wrapper
   0 ./usr/aixnre/lib/instl
4438 ./usr/aixnre/lib/instl/nrexec_wrapper
   0 ./usr/aixnre/lib/methods
4446 ./usr/aixnre/lib/methods/nrexec_wrapper
   0 ./usr/aixnre/lib/methods/wio
   0 ./usr/aixnre/lib/methods/wio/common
4470 ./usr/aixnre/lib/methods/wio/common/nrexec_wrapper
4430 ./usr/aixnre/lib/nrexec_wrapper
   0 ./usr/aixnre/lib/ras
4438 ./usr/aixnre/lib/ras/nrexec_wrapper
   0 ./usr/aixnre/lib/sa
4438 ./usr/aixnre/lib/sa/nrexec_wrapper
   0 ./usr/aixnre/objclass
3713 ./usr/aixnre/objclass/PCM.friend.vscsi.odmadd
 353 ./usr/aixnre/objclass/PCM.friend.vscsi.odmdel
2084 ./usr/aixnre/objclass/adapter.vdevice.IBM.v-scsi.odmadd
 234 ./usr/aixnre/objclass/adapter.vdevice.IBM.v-scsi.odmdel
6575 ./usr/aixnre/objclass/disk.vscsi.vdisk.odmadd
 207 ./usr/aixnre/objclass/disk.vscsi.vdisk.odmdel
   0 ./usr/aixnre/pmapi
   0 ./usr/aixnre/pmapi/tools
4446 ./usr/aixnre/pmapi/tools/nrexec_wrapper
   0 ./usr/aixnre/sbin
4430 ./usr/aixnre/sbin/nrexec_wrapper
4508 ./usr/aixnre/sbin/nrexec_trace
4374 ./usr/aixnre/sbin/nrexec_no64
   0 ./usr/aixnre/sbin/helpers
4438 ./usr/aixnre/sbin/helpers/nrexec_wrapper
   0 ./usr/aixnre/sbin/helpers/jfs2
4446 ./usr/aixnre/sbin/helpers/jfs2/nrexec_wrapper
4544 ./usr/aixnre/sbin/helpers/jfs2/nrexec_mount
   0 ./usr/aixnre/sbin/perf
   0 ./usr/aixnre/sbin/perf/diag_tool
4462 ./usr/aixnre/sbin/perf/diag_tool/nrexec_wrapper
2526 ./usr/aixnre/sbin/stubout
6641 ./usr/ccs/lib/libcre.a
   0 ./usr/lib/corrals
37789 ./usr/lib/corrals/manage_overlays
4096 ./usr/lib/objrepos/overlay
```

```
       4096 ./usr/lib/objrepos/overlay.vc
The total size is 3260358 bytes.
The number of archived files is 78.
```

3.3.6 Creation of a relocatable Versioned WPAR

Creation of a relocatable Versioned WPAR using the command line interface
(CLI) or a script would require the WPAR file systems to be located on an NFS
server.

> **Note:** The Versioned WPAR must have a private /usr and /opt. The `mkwpar`
> command should include the **-l** option and the /opt and /usr specifications.
>
> If you do not use the **-l** option, the system would issue a message such as:
>
> ```
> mkwpar: 0960-578 Workload partition directory /wpars/mywpar/opt is
> empty.  Quitting.
> ```

The creation should be done using the WPAR Manager, but in our example a
script requiring a name for the WPAR is provided in Example 3-20.

Example 3-20   Creation of MYWPAR

```
#!/usr/bin/ksh93
MYWPAR=$1
ADDRESS=A.B.C.D
NFSSERVER=mynfsserver

mkwpar -n $MYWPAR -h $MYWPAR \
-N interface=en0 netmask=255.255.255.0 address=$ADDRESS \
-r -l \
-C -B mksysb_5200-10-08-0930 \
-M directory=/      vfs=nfs host=$NFSSERVER dev=/nfs/$MYWPAR/root \
-M directory=/opt   vfs=nfs host=$NFSSERVER dev=/nfs/$MYWPAR/opt \
-M directory=/usr   vfs=nfs host=$NFSSERVER dev=/nfs/$MYWPAR/usr \
-M directory=/home  vfs=nfs host=$NFSSERVER dev=/nfs/$MYWPAR/home \
-M directory=/tmp   vfs=nfs host=$NFSSERVER dev=/nfs/$MYWPAR/tmp \
-M directory=/var   vfs=nfs host=$NFSSERVER dev=/nfs/$MYWPAR/var \
-c
```

We have included the **-r** option to get a copy of the network resolution
configuration from the global definitions. The checkpointable option **-c** has also
been specified.

3.3.7 SMIT interface

There is a new SMIT fastpath menu called vwpar for creating Versioned WPARs from `mksysb` images and from SPOTs. It is similar to the advance WPAR creation menu with new flags for the image to be loaded. It requires the vwpar.sysmgt package being installed.

3.4 Device support in WPAR

AIX 6.1 TL4 introduced the capability of creating a system WPAR with the root file systems on storage devices dedicated to the WPAR. Such a workload partition is referred to as a rootVG WPAR.

AIX 6.1 TL 6 introduced the support for VIOS-based VSCSI disks in a WPAR.

SAN support for rootvg system WPAR released with AIX 6.1 TL 6 provided the support of individual devices (disk or tapes) in a WPAR.

The result is that without the action of a Global AIX instance system administrator, the WPAR administrator can manage the adapter as well as the storage devices attached to it. There is no difference in syntax managing the device from the Global AIX instance or from the WPAR.

The controller example used will be the support of the Fibre Channel adapter introduced with AIX 7.1.

The following flow details user commands, behavior and outputs related to all these features. In the following sections, commands issued from the AIX Global instance are prefixed with Global>. Commands issued from the WPAR are prefixed with the WPAR name (for example wpar2>). WPAR examples are named wpar1, wpar2, and so on.

> **Note:** The Fibre Channel (FC) adapter can be either a physical or a virtual fibre channel adapter.

3.4.1 Global device listing used as example

Initially the test environment is running in an LPAR that is attached to an FC adapter with no disk.

From the Global, the `lscfg` command provides a familiar listing (Example 3-21 on page 69).

Example 3-21   Physical adapter available from Global

```
Global> lscfg | grep fc
+fcs0              U5802.001.0086848-P1-C2-T1              8Gb PCI
Express Dual Port FC Adapter (df1000f114108a03)
* fcnet0           U5802.001.0086848-P1-C2-T1              Fibre
Channel Network Protocol Device
+ fscsi0           U5802.001.0086848-P1-C2-T1              FC
SCSI I/O Controller Protocol Device
+ fcs1             U5802.001.0086848-P1-C2-T2              8Gb
PCI Express Dual Port FC Adapter (df1000f114108a03)
```

3.4.2 Device command listing in an AIX 7.1 WPAR

For our example, we created a single system WPAR using the `mkwpar -n wpar1`
command which creates a WPAR with JFS2 file systems included in the current
Global rootvg volume. Example 3-22 shows the output of the creation, the output
of the `lswpar` command queries for the file systems, as well as a display of the
Global rootvg disk content.

Example 3-22   Simple WPAR file system layout

```
Global> mkwpar -n wpar1
.....
syncroot: Processing root part installation status.
syncroot: Installp root packages are currently synchronized.
syncroot: RPM root packages are currently synchronized.
syncroot: Root part is currently synchronized.
syncroot: Returns Status = SUCCESS
Workload partition wpar1 created successfully.
mkwpar: 0960-390 To start the workload partition, execute the following as root:
startwpar [-v] wpar1

Global> lswpar
Name    State  Type  Hostname  Directory      RootVG WPAR
---------------------------------------------------------
wpar1   D      S     wpar1     /wpars/wpar1   no

Global> lswpar -M
Name    MountPoint         Device       Vfs      Nodename  Options
-----------------------------------------------------------------
wpar1   /wpars/wpar1       /dev/fslv00  jfs2
wpar1   /wpars/wpar1/home  /dev/fslv01  jfs2
wpar1   /wpars/wpar1/opt   /opt         namefs             ro
```

```
wpar1  /wpars/wpar1/proc  /proc         namefs          rw
wpar1  /wpars/wpar1/tmp   /dev/fslv02  jfs2
wpar1  /wpars/wpar1/usr   /usr          namefs          ro
wpar1  /wpars/wpar1/var   /dev/fslv03  jfs2

Global> 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    8      8      1    open/syncd    N/A
hd8                     jfs2log   1      1      1    open/syncd    N/A
hd4                     jfs2      4      4      1    open/syncd    /
hd2                     jfs2      37     37     1    open/syncd    /usr
hd9var                  jfs2      12     12     1    open/syncd    /var
hd3                     jfs2      2      2      1    open/syncd    /tmp
hd1                     jfs2      1      1      1    open/syncd    /home
hd10opt                 jfs2      6      6      1    open/syncd    /opt
hd11admin               jfs2      2      2      1    open/syncd    /admin
lg_dumplv               sysdump   16     16     1    open/syncd    N/A
livedump                jfs2      4      4      1    open/syncd
/var/adm/ras/livedump
fslv00                  jfs2      2      2      1    closed/syncd  /wpars/wpar1
fslv01                  jfs2      1      1      1    closed/syncd  /wpars/wpar1/home
fslv02                  jfs2      2      2      1    closed/syncd  /wpars/wpar1/tmp
fslv03                  jfs2      2      2      1    closed/syncd  /wpars/wpar1/var
```

When we start the WPAR (see Example 3-23) there is a mention of devices and
kernel extensions loading.

Example 3-23   Start of the WPAR

```
Global> startwpar wpar1
Starting workload partition wpar1.
Mounting all workload partition file systems.
Loading workload partition.
Exporting workload partition devices.
Exporting workload partition kernel extensions.
Starting workload partition subsystem cor_wpar1.
0513-059 The cor_wpar1 Subsystem has been started. Subsystem PID is
10158202.
Verifying workload partition startup.
```

In an AIX 7.1 system WPAR we can find a new entry in the **lscfg** command
output called WPAR I/O. This is the heart of the storage virtualization in a WPAR.

This feature allows use of the usual AIX commands related to devices such as lsdev, lscfg, cfgmgr, mkdev, rmdev, chdev, and lsvpd.

In Example 3-24, we log in to the system WPAR and issue the **lscfg** command that mentions the WPAR I/O subsystem entry.

Example 3-24   The lscfg display in a simple system WPAR

```
Global> clogin wpar1
*******************************************************************************
*                                                                             *
*                                                                             *
*  Welcome to AIX Version 7.1!                                                 *
*                                                                             *
*                                                                             *
*  Please see the README file in /usr/lpp/bos for information pertinent to    *
*  this release of the AIX Operating System.                                  *
*                                                                             *
*                                                                             *
*******************************************************************************
Last login: Tue Aug 31 15:27:43 EDT 2010 on /dev/Global from 7501lp01

wpar1:/> lscfg
INSTALLED RESOURCE LIST

The following resources are installed on the machine.
+/- = Added or deleted from Resource List.
*   = Diagnostic support not available.

  Model Architecture: chrp
  Model Implementation: Multiple Processor, PCI bus

+ sys0              System Object
* wio0              WPAR I/O Subsystem
```

The software packages being installed in the WPAR are as shown in Example 3-25.

Example 3-25   Packages related to wio installed in WPAR

```
wpar1:/> lslpp -L | grep wio
  wio.common              7.1.0.0    C     F    Common I/O Support for
  wio.fcp                 7.1.0.0    C     F    FC I/O Support for Workload
  wio.vscsi               7.1.0.0    C     F    VSCSI I/O Support for Workload
```

And when the specific package is installed, the subclass support is installed in `/usr/lib/methods/wio`. Support for Fibre Channel is called fcp and virtual SCSO disk support is called vscsi, as shown in Example 3-26.

Example 3-26   Virtual device support abstraction layer

```
wpar1:/> cd /usr/lib/methods/wio
wpar1:/> ls -R
common  fcp     vscsi
./common:
cfg_wpar_vparent  cfgwio          defwio

./fcp:
configure     unconfigure

./vscsi:
configure     unconfigure
# file /usr/lib/methods/wio/common/defwio
/usr/lib/methods/wio/common/defwio: executable (RISC System/6000) or object module
# /usr/lib/methods/wio/common/defwio
wio0
# lsdev | grep wio
wio0  Available  WPAR I/O Subsystem
```

3.4.3 Dynamically adding a Fibre Channel adapter to a system WPAR

Following our environment example, dynamically adding an FC channel adapter to the WPAR will be done with the `chwpar -D` command, as shown in Example 3-27. This `chwpar` command is referred to as an export process, but it does not do the `cfgmgr` command to update the device listing.

The Fibre Channel adapter mentioned is the one found in Global, as seen in Example 3-22 on page 69.

In the output shown in Example 3-27, we log in to the WPAR and verify Fibre Channel information coherency comparing to Global.

Example 3-27   Dynamically adding an FC adapter to a running WPAR

```
Global> chwpar -D devname=fcs0 wpar1
fcs0 Available
fscsi0 Available
fscsi0 Defined
line = 0
Global> lswpar -D wpar1
```

```
Name    Device Name      Type      Virtual Device RootVG Status
----------------------------------------------------------------
wpar1   fcs0             adapter                         EXPORTED
wpar1   /dev/null        pseudo                          EXPORTED
....

Global> clogin wpar1
**********************************************************************
*
*  Welcome to AIX Version 7.1!
*
*  Please see the README file in /usr/lpp/bos for information pertinent to    *
*  this release of the AIX Operating System.                                  *
**********************************************************************
Last login: Thu Aug 26 14:33:49 EDT 2010 on /dev/Global from 7501lp01

wpar1:/> lsdev
inet0 Defined    Internet Network Extension
pty0  Available  Asynchronous Pseudo-Terminal
sys0  Available  System Object
wio0  Available  WPAR I/O Subsystem

wpar1:/> fcstat fcs0
Error accessing ODM
Device not found
wpar1:/> lspath
wpar1:/> cfgmgr
wpar1:/> lspath
wpar1:/> fcstat fcs0
Error accessing ODM
VPD information not found

wpar1:/> lsdev
fcnet0 Defined   00-00-01 Fibre Channel Network Protocol Device
fcs0   Available 00-00    8Gb PCI Express Dual Port FC Adapter
fscsi0 Available 00-00-02 FC SCSI I/O Controller Protocol Device
inet0  Defined            Internet Network Extension
pty0   Available          Asynchronous Pseudo-Terminal
sys0   Available          System Object
wio0   Available          WPAR I/O Subsystem
```

> **Note:** Dynamic allocation of the adapter to the WPAR requires a `cfgmgr`
> command update to update the ODM and make the new adapter and device
> available.
>
> That dynamic allocation is referred to during the export process to the WPAR.

Change in the config file related to that device addition

At that point the WPAR configuration file located in `/etc/wpars/wpar1.cf` has
been updated with a new device entry, listed in Example 3-28.

Example 3-28   /etc/wpars/wpar1.cf entry update for device fcs0

```
device:
        devname = "fcs0"
        devtype = "6"
```

lsdev output from Global

A new flag, `-x`, to the `lsdev` command allows printing of exported devices;
Example 3-29.

Example 3-29   lsdev -x output

```
Global> lsdev -x | grep -i export
fscsi0     Exported   00-00-02     FC SCSI I/O Controller Protocol Device
```

3.4.4 Removing of the Fibre Channel adapter from Global

When the Fibre Channel adapter is allocated to a running WPAR, it is busy on the
Global side and cannot be removed; Example 3-30.

Example 3-30   rmdev failure for a busy device

```
Global> rmdev -dl fcs0 -R
fcnet0 deleted
rmdev: 0514-552 Cannot perform the requested function because the
        fscsi0 device is currently exported.
```

3.4.5 Reboot of LPAR keeps Fibre Channel allocation

From the previous state, reboot of the LPAR keeps the Fibre Channel allocation
to the WPAR, as shown in Example 3-31 on page 75.

Example 3-31   Fibre Channel adapter queries from Global after reboot

```
Global> lscfg | grep fc
+ fcs0            U5802.001.0086848-P1-C2-T1              8Gb PCI Express Dual
Port FC Adapter (df1000f114108a03)
* fcnet0          U5802.001.0086848-P1-C2-T1               Fibre Channel
Network Protocol Device
+ fscsi0          U5802.001.0086848-P1-C2-T1              FC SCSI I/O
Controller Protocol Device
+ fcs1            U5802.001.0086848-P1-C2-T2              8Gb PCI Express Dual
Port FC Adapter (df1000f114108a03)
* fcnet1          U5802.001.0086848-P1-C2-T2               Fibre Channel
Network Protocol Device
+ fscsi1          U5802.001.0086848-P1-C2-T2              FC SCSI I/O
Controller Protocol Device

Global> lswpar -Dq wpar1
wpar1  fcs0           adapter                     ALLOCATED
wpar1  /dev/null      pseudo                      ALLOCATED
....
Global> lswpar
Name    State  Type  Hostname  Directory      RootVG WPAR
---------------------------------------------------------
wpar1  D      S      wpar1     /wpars/wpar1  no
```

Since the WPAR wpar1 is not started, we can now remove the Fibre Channel adapter from the Global. The result is seen in Example 3-32 and confirm that a WPAR cannot start if it is missing some adapters.

Example 3-32   Removal of the Fibre Channel adapter from the Global

```
Global> rmdev -dl fcs0 -R
fcnet0 deleted
sfwcomm0 deleted
fscsi0 deleted
fcs0 deleted

Global> lswpar -D wpar1
Name    Device Name     Type     Virtual Device  RootVG  Status
-----------------------------------------------------------------
wpar1                   adapter                          MISSING
Global> startwpar wpar1
**********************************************************************
ERROR
ckwpar: 0960-586 Attributes of fcs0 do not match those in ODM.
```

```
ERROR
ckwpar: 0960-587 fcs0 has un-supported subclass type.


**********************************************************************
```

Removal of the adapter using the **chwpar** command corrects the situation. The
lswpar command shows the device is not missing or allocated any more. The
WPAR is now able to start, as shown in Example 3-33.

Example 3-33   Removal of missing device allows WPAR start

```
Global> chwpar -K -D devname=fcs0 wpar1
Global> lswpar -D wpar1
Name    Device Name     Type    Virtual Device  RootVG  Status
------------------------------------------------------------------
wpar1  /dev/null        pseudo                          ALLOCATED
wpar1  /dev/tty         pseudo                          ALLOCATED
wpar1  /dev/console     pseudo                          ALLOCATED
wpar1  /dev/zero        pseudo                          ALLOCATED
wpar1  /dev/clone       pseudo                          ALLOCATED
wpar1  /dev/sad         clone                           ALLOCATED
wpar1  /dev/xti/tcp     clone                           ALLOCATED
wpar1  /dev/xti/tcp6    clone                           ALLOCATED
wpar1  /dev/xti/udp     clone                           ALLOCATED
wpar1  /dev/xti/udp6    clone                           ALLOCATED
wpar1  /dev/xti/unixdg  clone                           ALLOCATED
wpar1  /dev/xti/unixst  clone                           ALLOCATED
wpar1  /dev/error       pseudo                          ALLOCATED
wpar1  /dev/errorctl    pseudo                          ALLOCATED
wpar1  /dev/audit       pseudo                          ALLOCATED
wpar1  /dev/nvram       pseudo                          ALLOCATED
wpar1  /dev/kmem        pseudo                          ALLOCATED

Global> startwpar wpar1
Starting workload partition wpar1.
Mounting all workload partition file systems.
Replaying log for /dev/fslv04.
Replaying log for /dev/fslv05.
Replaying log for /dev/fslv06.
Replaying log for /dev/fslv07.
Loading workload partition.
Exporting workload partition devices.
Exporting workload partition kernel extensions.
Starting workload partition subsystem cor_wpar2.
```

```
0513-059 The cor_wpar2 Subsystem has been started. Subsystem PID is
7012438.
Verifying workload partition startup.
```

3.4.6 Disk attached to Fibre Channel adapter

If you have disks attached to your Fibre Channel adapter, the previous **lsdev**
command display will be updated accordingly.

Disks are called *end-point devices,* meaning they do not have or cannot have
children devices.

In the test environment, we used four Fibre Channel disks attached to the
system. On one of them (hdisk1) a volume group named lpar1data from the
Global was created.

From the Global point of view, the devices commands output can be seen in
Example 3-34.

Example 3-34   Devices commands issued on the Global

```
Global> lsdev -c adapter
ent0    Available       Virtual I/O Ethernet Adapter (l-lan)
ent1    Available       Virtual I/O Ethernet Adapter (l-lan)
fcs0    Available 00-00 8Gb PCI Express Dual Port FC Adapter (df1000f114108a03)
fcs1    Available 00-01 8Gb PCI Express Dual Port FC Adapter (df1000f114108a03)
vsa0    Available       LPAR Virtual Serial Adapter
vscsi0 Available        Virtual SCSI Client Adapter
Global> lsdev -c disk
hdisk0 Available         Virtual SCSI Disk Drive
hdisk1 Available 00-00-02 MPIO Other DS4K Array Disk
hdisk2 Available 00-00-02 MPIO Other DS4K Array Disk
hdisk3 Available 00-00-02 MPIO Other DS4K Array Disk
hdisk4 Available 00-00-02 MPIO Other DS4K Array Disk
Global> lspath -t
Enabled hdisk0 vscsi0 0
Enabled hdisk1 fscsi0 0
Enabled hdisk2 fscsi0 0
Enabled hdisk3 fscsi0 0
Enabled hdisk4 fscsi0 0
Global> fcstat -d -e fcs0 | head -24

FIBRE CHANNEL STATISTICS REPORT: fcs0
```

```
Device Type: 8Gb PCI Express Dual Port FC Adapter (df1000f114108a03)
(adapter/pciex/df1000f114108a0)
Serial Number: 1B02104269
Option ROM Version: 02781174
ZA: U2D1.11X4
World Wide Node Name: 0x20000000C99FC71E
World Wide Port Name: 0x10000000C99FC71E


FC-4 TYPES:
  Supported: 0x0000012000000000000000000000000000000000000000000000000000000000
  Active:    0x0000010000000000000000000000000000000000000000000000000000000000


FC-4 TYPES (ULP mappings):
  Supported ULPs:
        Internet Protocol (IP) over Fibre Channel (IETF RFC2625)
        Small Computer System Interface (SCSI) Fibre Channel Protocol (FCP)
  Active ULPs:
        Small Computer System Interface (SCSI) Fibre Channel Protocol (FCP)
Class of Service: 3
Port Speed (supported): 8 GBIT
Port Speed (running):   4 GBIT
Port FC ID: 0x010200


Global> lspv
hdisk0          00f61aa68cf70a14                       rootvg          active
hdisk1          00f61aa6b48ad819                       lpar1data       active
hdisk2          00f61aa6b48b0139                       None
hdisk3          00f61aa6b48ab27f                       None
hdisk4          00f61aa6b48b3363                       None
Global> lspv hdisk1
PHYSICAL VOLUME:   hdisk1               VOLUME GROUP:    lpar1data
PV IDENTIFIER:     00f61aa6b48ad819 VG IDENTIFIER
00f61aa600004c000000012aba12d483
PV STATE:          active
STALE PARTITIONS:  0                    ALLOCATABLE:     yes
PP SIZE:           64 megabyte(s)       LOGICAL VOLUMES: 0
TOTAL PPs:         799 (51136 megabytes) VG DESCRIPTORS:  2
FREE PPs:          799 (51136 megabytes) HOT SPARE:       no
USED PPs:          0 (0 megabytes)       MAX REQUEST:     256 kilobytes
FREE DISTRIBUTION: 160..160..159..160..160
USED DISTRIBUTION: 00..00..00..00..00
MIRROR POOL:       None
```

3.4.7 Startwpar error if adapter is busy on Global

As the volume group is active from the Global environment, it prevents the WPAR to load the Fibre Channel device. To demonstrate this, we try to start again the WPAR that is supposed to have Fibre Channel adapter fcs0 allocated to it. The WPAR will start, but the adapter is not EXPORTED to (not available for use by) the WPAR

Example 3-35   WPAR1 start error message if disk is busy

```
Global> startwpar wpar1
Starting workload partition wpar1.
Mounting all workload partition file systems.
Loading workload partition.
Exporting workload partition devices.
Method error (/usr/lib/methods/ucfgdevice):
        0514-062 Cannot perform the requested function because the
                 specified device is busy.

mkFCAdapExport:0: Error 0
Exporting workload partition kernel extensions.
Starting workload partition subsystem cor_wpar2.
0513-059 The cor_wpar2 Subsystem has been started. Subsystem PID is 9240666.
Verifying workload partition startup.

Global> clogin wpar1 lsdev
inet0     Defined     Internet Network Extension
pty0      Available   Asynchronous Pseudo-Terminal
sys0      Available   System Object
vg00      Available   Volume group
wio0      Available   WPAR I/O Subsystem
Global> lswpar -D
Name    Device Name       Type      Virtual Device  RootVG  Status
----------------------------------------------------------------
wpar1   fcs0              adapter                            ALLOCATED
```

> **Note:** Controller devices or end-point devices in AVAILABLE state are not exported to WPARs. They must be in DEFINED state.

3.4.8 Startwpar with a Fibre Channel adapter defined

To start the WPAR and have the Fibre Channel loaded you need to quiesce that adapter making the volume group not allocated on the Global side. A `varyoffvg` command as shown in Example 3-36 on page 80 starts the WPAR.

Example 3-36   Startwpar with Fibre Channel device available for WPAR use

```
Global> varyoffvg lpar1data
Global> lspv hdisk1
0516-010 : Volume group must be varied on; use varyonvg command.
PHYSICAL VOLUME:    hdisk1                   VOLUME GROUP:    lpar1data
PV IDENTIFIER:      00f61aa6b48ad819 VG IDENTIFIER
00f61aa600004c000000012aba12d483
PV STATE:           ???????
STALE PARTITIONS:   ???????                  ALLOCATABLE:     ???????
PP SIZE:            ???????                  LOGICAL VOLUMES: ???????
TOTAL PPs:          ???????                  VG DESCRIPTORS:  ???????
FREE PPs:           ???????                  HOT SPARE:       ???????
USED PPs:           ???????                  MAX REQUEST:     256 kilobytes
FREE DISTRIBUTION:  ???????
USED DISTRIBUTION:  ???????
MIRROR POOL:        ???????
Global> lspv
hdisk0        00f61aa68cf70a14                    rootvg          active
hdisk1        00f61aa6b48ad819                    lpar1data
hdisk2        00f61aa6b48b0139                    None
hdisk3        00f61aa6b48ab27f                    None
hdisk4        00f61aa6b48b3363                    None

Global> startwpar wpar1
Starting workload partition wpar1.
Mounting all workload partition file systems.
Loading workload partition.
Exporting workload partition devices.
hdisk1 Defined
hdisk2 Defined
hdisk3 Defined
hdisk4 Defined
sfwcomm0 Defined
fscsi0 Defined
line = 0
Exporting workload partition kernel extensions.
Starting workload partition subsystem cor_wpar2.
0513-059 The cor_wpar2 Subsystem has been started. Subsystem PID is 6029534.
Verifying workload partition startup.
```

So when WPAR is running, we can display the Fibre Channel and its associated devices from the WPAR side, as shown in Example 3-37 on page 81.

Example 3-37   Devices in the WPAR

```
Global> clogin wpar1
***********************************************************************
* *
*  Welcome to AIX Version 7.1!                                      * *
*  Please see the README file in /usr/lpp/bos for information pertinent
* to this release of the AIX Operating System.                       * *
***********************************************************************
Last login: Sat Aug 28 15:33:14 EDT 2010 on /dev/Global from 7501lp01

wpar1:/> lsdev
fcnet0 Defined   00-00-01 Fibre Channel Network Protocol Device
fcs0   Available 00-00    8Gb PCI Express Dual Port FC Adapter (df1000f114108a03)
fscsi0 Available 00-00-02 FC SCSI I/O Controller Protocol Device
hdisk0 Available 00-00-02 MPIO Other DS4K Array Disk
hdisk1 Available 00-00-02 MPIO Other DS4K Array Disk
hdisk2 Available 00-00-02 MPIO Other DS4K Array Disk
hdisk3 Available 00-00-02 MPIO Other DS4K Array Disk
inet0  Defined            Internet Network Extension
pty0   Available          Asynchronous Pseudo-Terminal
sys0   Available          System Object
wio0   Available          WPAR I/O Subsystem
wpar1:/> lspath
Enabled hdisk0 fscsi0
Enabled hdisk1 fscsi0
Enabled hdisk2 fscsi0
Enabled hdisk3 fscsi0

wpar1:/> lscfg
INSTALLED RESOURCE LIST

The following resources are installed on the machine.
+/- = Added or deleted from Resource List.
*   = Diagnostic support not available.

  Model Architecture: chrp
  Model Implementation: Multiple Processor, PCI bus

+ sys0            System Object
* wio0            WPAR I/O Subsystem
+ fcs0            8Gb PCI Express Dual Port FC Adapter (df1000f114108a03)
* fcnet0          Fibre Channel Network Protocol Device
+ fscsi0          FC SCSI I/O Controller Protocol Device
* hdisk0          MPIO Other DS4K Array Disk
```

```
* hdisk1           MPIO Other DS4K Array Disk
* hdisk2           MPIO Other DS4K Array Disk
* hdisk3           MPIO Other DS4K Array Disk

wpar1:/> lspv
hdisk0         00f61aa6b48ad819                   None
hdisk1         00f61aa6b48b0139                   None
hdisk2         00f61aa6b48ab27f                   None
hdisk3         00f61aa6b48b3363                   None
```

Since the Fibre Channel adapter is in use by the WPAR, this also means that all its child devices are allocated to the WPAR. The disks are not visible; see Example 3-38.

Example 3-38   Disk no longer visible from Global

```
Global> lspv
hdisk0         00f61aa68cf70a14                    rootvg          active
Global> lsvg
rootvg
lpar1data
Global> lsvg lpar1data
0516-010 : Volume group must be varied on; use varyonvg command.
Global> varyonvg lpar1data
0516-013 varyonvg: The volume group cannot be varied on because
        there are no good copies of the descriptor area.
```

> **Note:** The `lsdev -x` command gives you the list of exported devices to WPAR.
>
> When a device is exported, the `mkdev`, `rmdev`, `mkpath`, and `chgpath` commands fail. The `cfgmgr` command takes no action.

3.4.9 Disk commands in the WPAR

In a WPAR, disk commands are available as usual, as shown in Example 3-39.

Example 3-39   Creation of volume in a WPAR

```
wpar1:/> mkvg -y wpar1data hdisk1
wpar1data
wpar1:/> lspv
hdisk0         00f61aa6b48ad819                   None
hdisk1         00f61aa6b48b0139                   wpar1data       active
hdisk2         00f61aa6b48ab27f                   None
```

```
hdisk3          00f61aa6b48b3363                        None
wpar1:/> importvg hdisk0
synclvodm: No logical volumes in volume group vg00.
vg00
wpar1:/> lspv
hdisk0          00f61aa6b48ad819               vg00          active
hdisk1          00f61aa6b48b0139               wpar1data     active
hdisk2          00f61aa6b48ab27f               None
hdisk3          00f61aa6b48b3363               None
wpar1:/> mklv vg00 10
lv00
wpar1:/> lsvg vg00
VOLUME GROUP:     vg00                 VG IDENTIFIER:
00f61aa600004c000000012aba12d483
VG STATE:         active               PP SIZE:       64 megabyte(s)
VG PERMISSION:    read/write           TOTAL PPs:     799 (51136 megabytes)
MAX LVs:          256                  FREE PPs:      789 (50496 megabytes)
LVs:              1                    USED PPs:      10 (640 megabytes)
OPEN LVs:         0                    QUORUM:        2 (Enabled)
TOTAL PVs:        1                    VG DESCRIPTORS: 2
STALE PVs:        0                    STALE PPs:     0
ACTIVE PVs:       1                    AUTO ON:       yes
MAX PPs per VG:   32512
MAX PPs per PV:   1016                 MAX PVs:       32
LTG size (Dynamic): 256 kilobyte(s)    AUTO SYNC:     no
HOT SPARE:        no                   BB POLICY:     relocatable
PV RESTRICTION:   none
wpar1:/> lsvg -l vg00
vg00:
LV NAME           TYPE       LPs     PPs     PVs LV STATE      MOUNT POINT
lv00              jfs        10      10      1   closed/syncd  N/A
```

3.4.10 Access to the Fibre Channel attached disks from the Global

As seen previously in Example 3-38 on page 82, when the Fibre Channel device is exported to the WPAR, the disks are not visible from the Global.

To gain access to the disks from the Global, one simple solution is to stop the WPAR, as demonstrated in Example 3-40.

Example 3-40   Stopping WPAR releases Fibre Channel allocation

```
Global> stopwpar wpar1
Stopping workload partition wpar1.
```

```
Stopping workload partition subsystem cor_wpar2.
0513-044 The cor_wpar2 Subsystem was requested to stop.
stopwpar: 0960-261 Waiting up to 600 seconds for workload partition to halt.
Shutting down all workload partition processes.
fcnet0 deleted
hdisk0 deleted
hdisk1 deleted
hdisk2 deleted
hdisk3 deleted
fscsi0 deleted
0518-307 odmdelete: 1 objects deleted.
wio0 Defined
Unmounting all workload partition file systems.
Global> lspv
hdisk0          00f61aa68cf70a14                    rootvg          active
Global> cfgmgr
lspv
Method error (/usr/lib/methods/cfgefscsi -l fscsi1 ):
        0514-061 Cannot find a child device.
Global> lspv
hdisk0          00f61aa68cf70a14                    rootvg          active
hdisk1          00f61aa6b48ad819                    lpar1data
hdisk2          00f61aa6b48b0139                    None
hdisk3          00f61aa6b48ab27f                    None
hdisk4          00f61aa6b48b3363                    None
Global>
Global> lsvg -l lpar1data
0516-010 : Volume group must be varied on; use varyonvg command.
Global> varyonvg lpar1data
Global> lsvg -l lpar1data
lpar1data:
LV NAME               TYPE      LPs      PPs      PVs  LV STATE      MOUNT POINT
lv00                  ???       10       10       1    closed/syncd  N/A
```

> **Note:** When the WPAR is removed or stopped, all device instances are
> removed from the WPAR allocation, so they should be available from the
> Global.

3.4.11 Support of Fibre Channel devices in the mkwpar command

The adapter specification is handled with the **-D** parameter on the **mkwpar**
command.

```
mkpwar -n wpar2 -D devname=fcs0
```

The **mkwpar -D** option in the man page is shown in Example 3-41.

Example 3-41   mkwpar -D option syntax

```
-D [devname=device name | devid=device identifier] [rootvg=yes | no]
        [devtype=[clone | pseudo | disk | adapter | cdrom | tape]] [xfactor=n]
            Configures exporting or virtualization of a Global device into the
            workload partition every time the system starts. You can specify
            more than one -D flag to allocate multiple devices. Separate the
            attribute=value by blank spaces. You can specify the following
            attributes for the -D flag:
```

The devname specification can be a controller name (see previous examples) or a end-point device name like a disk name. If not specified, the devtype will be queried from the Global ODM databases.

When you specify a devname or a devid, the **mkwpar** command will modify the WPAR definition to include the specified adapter or device.

Creation of a rootvg system WPAR

If the rootvg flag is set to yes, the root file system of the WPAR will exist on the specified disk device (see Example 3-42).

Example 3-42   Creation of a rootvg system WPAR

```
Global> mkwpar -n wpar2 -D devname=hdisk3 rootvg=yes -O
Creating workload partition's rootvg. Please wait...
mkwpar: Creating file systems...
 /
 /admin
...
wio.common                      7.1.0.0        ROOT      APPLY      SUCCESS
Finished populating scratch file systems.
Workload partition wpar2 created successfully.
mkwpar: 0960-390 To start the workload partition, execute the following as root:
startwpar [-v] wpar2

Global> lswpar -M wpar2
Name    MountPoint                       Device      Vfs      Nodename  Options
--------------------------------------------------------------------------------
wpar2   /wpars/wpar2                     /dev/fslv05  jfs2
wpar2   /wpars/wpar2/etc/objrepos/wboot  /dev/fslv06  jfs2
wpar2   /wpars/wpar2/opt                 /opt         namefs            ro
wpar2   /wpars/wpar2/usr                 /usr         namefs            ro
Global> lswpar -D wpar2
Name    Device Name     Type     Virtual Device  RootVG  Status
```

```
-----------------------------------------------------------------
wpar2  /dev/null        pseudo                      ALLOCATED
....
wpar2  hdisk3           disk                   yes  ALLOCATED
```

> **Note:** In the preceding examples, `/dev/fslv05` and `/dev/fslv06` are the file systems used to start the rootvg WPAR and contain the bare minimum elements to configure the WPAR storage devices.

Rootvg system WPAR creation failure when device busy

Attempting to create a rootvg WPAR using a device that has already been exported to another WPAR will fail. For example, if a Fibre Channel adapter has been exported to an Active WPAR (wpar1), this adapter is owned by wpar1 until it is freed. The adapter may be released by either stopping the WPAR or removing the device from within the WPAR with the **rmdev** command. If a WPAR administrator attempts to create a WPAR using the same FC adapter, an error message is displayed stating that the device is busy. The WPAR creation fails (Example 3-43).

Example 3-43   Mkwpar failure if end-point device is busy

```
Global> mkwpar -n wpar2 -D devname=hdisk3 rootvg=yes
Creating workload partition's rootvg. Please wait...
mkwpar: 0960-621 Failed to create a workload partition's rootvg. Please
use -O flag to overwrite hdisk3.
        If restoring a workload partition, target disks should be in
available state.
Global> mkwpar -n wpar2 -D devname=hdisk3 rootvg=yes -O
mkwpar: 0960-619 Failed to make specified disk, hdisk3, available.
```

> **Note:** The `mkwpar -O` command may be used to force the overwrite of an existing volume group on the given set of devices specified with the `-D` rootvg=yes flag directive.

Rootvg system WPAR overview

When the system WPAR has been created (see Example 3-42 on page 85), two devices have been created in the Global rootvg disk for management and startup purpose: One for the root mount point and the other for the ODM customizing to be made during the export phase (Example 3-44).

Example 3-44   Listing of the rootvg system WPAR file systems from the Global

```
Global> lswpar -M wpar2
Name   MountPoint                         Device    Vfs     Nodename  Options
```

```
--------------------------------------------------------------------------------
wpar2  /wpars/wpar2                          /dev/fslv05  jfs2
wpar2  /wpars/wpar2/etc/objrepos/wboot  /dev/fslv06  jfs2
wpar2  /wpars/wpar2/opt                      /opt         namefs          ro
wpar2  /wpars/wpar2/usr                      /usr         namefs          ro

Global> lspv -l hdisk0 |  grep wpar2
fslv05                   2        2       00..02..00..00..00      /wpars/wpar2
fslv06                   1        1       00..01..00..00..00
/wpars/wpar2/etc/objrepos/wboot
```

Devices that have been allocated and exported to a WPAR are placed into a
Defined state in the Global instance. If the WPAR administrator was to run the
lsdev command from the global instance, prior to exporting the device to a
WPAR, it will be seen that the device is in an Available state. Once the device is
exported to a WPAR, the **lsdev** command will report the device as Defined from
the Global instance (Example 3-45).

Example 3-45   Allocated devices to a WPAR not available to Global

```
Global> lswpar -D wpar2| grep disk
wpar2  hdisk3            disk                        yes     ALLOCATED
Global>
Global> lsdev -x
L2cache0    Available           L2 Cache
...
fcnet0     Defined    00-00-01   Fibre Channel Network Protocol Device
fcnet1     Defined    00-01-01   Fibre Channel Network Protocol Device
fcs0       Available 00-00       8Gb PCI Express Dual Port FC Adapter
(df1000f114108a03)
fcs1       Available 00-01       8Gb PCI Express Dual Port FC Adapter
(df1000f114108a03)
fscsi0     Available 00-00-02    FC SCSI I/O Controller Protocol Device
fscsi1     Available 00-01-02    FC SCSI I/O Controller Protocol Device
fslv00     Available            Logical volume
fslv01     Available            Logical volume
fslv02     Available            Logical volume
fslv03     Available            Logical volume
fslv04     Available            Logical volume
fslv05     Available            Logical volume
fslv06     Available            Logical volume
hd1        Defined              Logical volume
hd2        Defined              Logical volume
hd3        Defined              Logical volume
hd4        Defined              Logical volume
```

```
hd5        Defined              Logical volume
hd6        Defined              Logical volume
hd8        Defined              Logical volume
hd10opt    Defined              Logical volume
hd11admin  Defined              Logical volume
hd9var     Defined              Logical volume
hdisk0     Available            Virtual SCSI Disk Drive
hdisk1     Defined   00-00-02   MPIO Other DS4K Array Disk
hdisk2     Defined   00-00-02   MPIO Other DS4K Array Disk
hdisk3     Available 00-00-02   MPIO Other DS4K Array Disk
hdisk4     Defined   00-00-02   MPIO Other DS4K Array Disk
...
Global> lspv
hdisk0         00f61aa68cf70a14                    rootvg        active
hdisk3         00f61aa6b48ab27f                    None
Global> lspv -l hdisk3
0516-320 : Physical volume 00f61aa6b48ab27f0000000000000000 is not assigned to a
volume group.
```

Startwpar of the rootvg system WPAR

The **startwpar** command effectively processes the export phase and associates
the devices to the WPAR. In case of the rootvg specification, the disk name
appears in the listing. It also mentions that the kernel extension dynamic loading
is being used to load the Fibre Channel and the wio driver (see Example 3-46).

Example 3-46   Startwpar of a rootvg WPAR on a Fibre Channel disk

```
Global> startwpar wpar2
Starting workload partition wpar2.
Mounting all workload partition file systems.
Loading workload partition.
Exporting workload partition devices.
hdisk3 Defined
Exporting workload partition kernel extensions.
Starting workload partition subsystem cor_wpar3.
0513-059 The cor_wpar3 Subsystem has been started. Subsystem PID is
8650994.
Verifying workload partition startup.
```

Note: An FC controller would not be exported explicitly but would be implicitly
exported when the **cfgmgr** command is being launched by the /etc/rc.bootc
script.

Within the rootvg WPAR the file system structure is referencing internal devices (/dev/...) from the rootvg disk as well as file systems mounted from Global since we did not create private file systems. We can also see that the root mount point mounted from the Global is over-mounted with the local device (Example 3-47).

Example 3-47   File systems of the rootvg WPAR seen from inside the WPAR

```
Global> clogin wpar2 df
Filesystem     512-blocks      Free %Used    Iused %Iused Mounted on
Global             262144    200840   24%     2005     9% /
/dev/hd4           262144    200840   24%     2005     9% /
Global            4063232    448200   89%    41657    44% /usr
Global             786432    427656   46%     7008    13% /opt
/dev/hd11admin     131072    128312    3%        5     1% /admin
/dev/hd1           131072    128312    3%        5     1% /home
/dev/hd3           262144    256864    3%        9     1% /tmp
/dev/hd9var        262144    220368   16%      349     2% /var
Global             131072    128336    3%        5     1% /etc/objrepos/wboot
Global                  -         -    -         -     - /proc
Global> clogin wpar2 lspv
hdisk0         00f61aa6b48ab27f                        rootvg          active
```

And the device listing is also as expected with disks and drivers wio and fscsi0, as shown in Example 3-48.

Example 3-48   lsdev in a rootvg system WPAR

```
Global> clogin wpar2 lsdev
fscsi0     Available 00-00-02 WPAR I/O Virtual Parent Device
hd1        Available          Logical volume
hd3        Available          Logical volume
hd4        Available          Logical volume
hd11admin  Available          Logical volume
hd9var     Available          Logical volume
hdisk0     Available 00-00-02 MPIO Other DS4K Array Disk
inet0      Defined            Internet Network Extension
pty0       Available          Asynchronous Pseudo-Terminal
rootvg     Available          Volume group
sys0       Available          System Object
wio0       Available          WPAR I/O Subsystem
```

Fibre Channel controller cannot be shared

Because we started wpar2 rootvg system WPAR, the Fibre Channel controller can be exported to wpar1 system WPAR since one of its children is busy. As

such, wpar1 WPAR start will not load the fcs0 controller and some warning messages appear on the console (Example 3-49).

Example 3-49   Exclusive device allocation message

```
Global> startwpar wpar1
Starting workload partition wpar1.
Mounting all workload partition file systems.
Loading workload partition.
Exporting workload partition devices.
rmdev: 0514-552 Cannot perform the requested function because the
        hdisk3 device is currently exported.
mkFCAdapExport:0: Error 0
Exporting workload partition kernel extensions.
Starting workload partition subsystem cor_wpar2.
0513-059 The cor_wpar2 Subsystem has been started. Subsystem PID is 8585362.
Verifying workload partition startup.

Global> lswpar
Name    State Type  Hostname  Directory     RootVG WPAR
---------------------------------------------------------
wpar1  A     S     wpar1    /wpars/wpar1  no
wpar2  A     S     wpar2    /wpars/wpar2  yes
Global> lswpar -D
Name    Device Name     Type     Virtual Device  RootVG  Status
-------------------------------------------------------------------
wpar1  fcs0            adapter                          ALLOCATED
.....
wpar2  hdisk3          disk     hdisk0          yes     EXPORTED
```

End-point devices are separated

However, the other disks (end-point devices) can be allocated to another WPAR if the Fibre Channel controller has not been explicitly exported.

We can now create a new rootvg system WPAR on disk hdisk4. A summary of the console messages issued from the `mkwpar` command is listed in Example 3-50. The `startwpar` command console messages are also included.

Example 3-50   New rootvg system WPAR creation

```
Global> mkwpar -O -D devname=hdisk4 rootvg=yes -n wpar3
.....
syncroot: Processing root part installation status.
syncroot: Installp root packages are currently synchronized.
syncroot: RPM root packages are currently synchronized.
```

```
syncroot: Root part is currently synchronized.
syncroot: Returns Status = SUCCESS
......
Exporting a workload partition's rootvg. Please wait...
Cleaning up the trace of a workload partition's rootvg population...
mkwpar: Removing file system /wpars/wpar3/usr.
mkwpar: Removing file system /wpars/wpar3/proc.
mkwpar: Removing file system /wpars/wpar3/opt.
Creating scratch file system...
Populating scratch file systems for rootvg workload partition...
Mounting all workload partition file systems.
x ./usr
x ./lib
....
Installation Summary
--------------------
Name                         Level      Part     Event     Result
-------------------------------------------------------------------------------
bos.net.nis.client           7.1.0.0    ROOT     APPLY     SUCCESS
bos.perf.libperfstat         7.1.0.0    ROOT     APPLY     SUCCESS
bos.perf.perfstat            7.1.0.0    ROOT     APPLY     SUCCESS
bos.perf.tools               7.1.0.0    ROOT     APPLY     SUCCESS
bos.sysmgt.trace             7.1.0.0    ROOT     APPLY     SUCCESS
clic.rte.kernext             4.7.0.0    ROOT     APPLY     SUCCESS
devices.chrp.base.rte        7.1.0.0    ROOT     APPLY     SUCCESS
devices.chrp.pci.rte         7.1.0.0    ROOT     APPLY     SUCCESS
devices.chrp.vdevice.rte     7.1.0.0    ROOT     APPLY     SUCCESS
devices.common.IBM.ethernet 7.1.0.0     ROOT     APPLY     SUCCESS
devices.common.IBM.fc.rte    7.1.0.0    ROOT     APPLY     SUCCESS
devices.common.IBM.mpio.rte 7.1.0.0     ROOT     APPLY     SUCCESS
devices.common.IBM.scsi.rte 7.1.0.0     ROOT     APPLY     SUCCESS
devices.fcp.disk.array.rte   7.1.0.0    ROOT     APPLY     SUCCESS
devices.fcp.disk.rte         7.1.0.0    ROOT     APPLY     SUCCESS
devices.fcp.tape.rte         7.1.0.0    ROOT     APPLY     SUCCESS
devices.scsi.disk.rte        7.1.0.0    ROOT     APPLY     SUCCESS
devices.tty.rte              7.1.0.0    ROOT     APPLY     SUCCESS
bos.mp64                     7.1.0.0    ROOT     APPLY     SUCCESS
bos.net.tcp.client           7.1.0.0    ROOT     APPLY     SUCCESS
bos.perf.tune                7.1.0.0    ROOT     APPLY     SUCCESS
perfagent.tools              7.1.0.0    ROOT     APPLY     SUCCESS
bos.net.nfs.client           7.1.0.0    ROOT     APPLY     SUCCESS
bos.wpars                    7.1.0.0    ROOT     APPLY     SUCCESS
bos.net.ncs                  7.1.0.0    ROOT     APPLY     SUCCESS
wio.common                   7.1.0.0    ROOT     APPLY     SUCCESS
Finished populating scratch file systems.
```

```
Workload partition wpar3 created successfully.
mkwpar: 0960-390 To start the workload partition, execute the following as root:
startwpar [-v] wpar3
Global>

Global> startwpar wpar3
Starting workload partition wpar3.
Mounting all workload partition file systems.
Loading workload partition.
Exporting workload partition devices.
hdisk4 Defined
Exporting workload partition kernel extensions.
Starting workload partition subsystem cor_wpar4.
0513-059 The cor_wpar4 Subsystem has been started. Subsystem PID is 7405614.
Verifying workload partition startup.
```

And from the global instance we can see that both disks are exported
(Example 3-51).

Example 3-51   Global view of exported disks to rootvg WPARs

```
Global> lswpar -D
Name    Device Name       Type      Virtual Device  RootVG  Status
------------------------------------------------------------------
wpar1  fcs0              adapter                            ALLOCATED
...
wpar2  hdisk3            disk      hdisk0          yes     EXPORTED
...
wpar3  hdisk4            disk      hdisk0          yes     EXPORTED
Global> lsdev -x | grep -i export
hdisk3    Exported  00-00-02   MPIO Other DS4K Array Disk
hdisk4    Exported  00-00-02   MPIO Other DS4K Array Disk
```

3.4.12 Config file created for the rootvg system WPAR

When a system WPAR is being created, a config file is also created in
/etc/wpars and includes the rootvg device specification as well as the rootvg
WPAR type, as shown in Example 3-52.

Example 3-52   /etc/wpars/wpar3.cf listing

```
Global> cat /etc/wpars/wpar3.cf
general:
        name = "wpar3"
        checkpointable = "no"
```

```
              hostname = "wpar3"
              privateusr = "no"
              directory = "/wpars/wpar3"
              ostype = "0"
              auto = "no"
              rootvgwpar = "yes"
              routing = "no"

resources:
              active = "yes"
.....
device:
              devid = "3E213600A0B8000291B080000E299059A3F460F1815
FAStT03IBMfcp"
              devtype = "2"
              rootvg = "yes"
```

3.4.13 Removing an FC-attached disk in a running system WPAR

It is not possible to remove the rootvg disk of the system WPAR when it is active
since it is busy, as shown in Example 3-53.

Example 3-53   Rootvg disk of a rootvg WPAR cannot be removed if WPAR is active

```
Global> chwpar -K -D devname=hdisk4 wpar3
chwpar: 0960-604 the device with devname, hdisk4, is still being used
in the WPAR.
chwpar: 0960-018 1 errors refreshing devices.
```

3.4.14 Mobility considerations

The use of rootvg devices and Fibre Channel in a system WPAR currently
prevents mobility.

Mobility of a Fibre Channel adapter

Use of Fibre Channel adapter in a system WPAR prevents mobility.

```
Global> chwpar -c wpar1
chwpar: 0960-693 Cannot checkpoint a WPAR that has adapter(s).
```

Mobility of a rootvg system WPAR

In order to change the checkpointable flag of a system WPAR, it must be
stopped. Then, providing you get the required optional package mcr.rte being

installed on your system, you can change the checkpoint flag of the WPAR using the `chwpar -c wpar2` command.

A listing of the system WPAR wpar2 states it is checkpointable (Example 3-54).

Example 3-54   Listing of the environment flags of the system WPAR

```
Global> lswpar -G wpar2
================================================================
wpar2 - Defined
================================================================
Type:                   S
RootVG WPAR:            yes
Owner:                  root
Hostname:               wpar2
WPAR-Specific Routing:  no
Directory:              /wpars/wpar2
Start/Stop Script:
Auto:                   no
Private /usr:           no
Checkpointable:         yes
Application:
OStype:                 0
```

But the rootvg system WPAR cannot be checkpointed (Example 3-55).

Example 3-55   Checkpoint WPAR is not allowed with rootvg WPAR

```
/opt/mcr/bin/chkptwpar -d /wpars/wpar2/tmp/chpnt -o
/wpars/wpar2/tmp/ckplog -l debug wpar2
1020-235 chkptwpar is not allowed on rootvg (SAN) WPAR [02.291.0168]
[8650894 29:8:2010 12:23:7]
1020-187 chkptwpar command failed.
```

3.4.15 Debugging log

All events related to WPAR commands are added to the file /var/adm/wpars/event.log.

For example, the last commands being issued, such as `stopwpar` on wpar2 and `chwpar` on wpar3, get appropriate error messages to facilitate debugging (Example 3-56).

Example 3-56   /var/adm/wpars/event.log example

```
Global> tail /var/adm/wpars//event.log
```

```
I 2010-08-29 12:22:04 7929932 runwpar wpar2 Removing work directory
/tmp/.workdir.7077910.7929932_1
V 2010-08-29 12:22:05 7929932 startwpar - COMMAND START, ARGS: -I wpar2
I 2010-08-29 12:22:05 7929932 startwpar wpar2 Removing work directory
/tmp/.workdir.8454242.7929932_1
I 2010-08-29 12:22:05 10289288 startwpar wpar2 Lock released.
I 2010-08-29 12:22:05 10289288 startwpar wpar2 Removing work directory
/tmp/.workdir.8781954.10289288_1
V 2010-08-29 12:22:05 10289288 startwpar wpar2 Return Status = SUCCESS.
E 2010-08-29 12:25:28 7209076 corralinstcmd wpar3
/usr/lib/corrals/corralinstcmd: 0960-231 ATTENTION:
'/usr/lib/corrals/wpardevstop hdisk0' failed with return code 1.
E 2010-08-29 12:25:28 8126600 chwpar wpar3 chwpar: 0960-604 the device
with devname, hdisk4, is still being used in the WPAR.
W 2010-08-29 12:25:28 8126600 chwpar wpar3 chwpar: 0960-018 1 errors
refreshing devices.
W 2010-08-29 12:26:10 8126606 chwpar wpar3 chwpar: 0960-070 Cannot find
a device stanza to remove from /etc/wpars/wpar3.cf where devname=fcs0.
```

3.5 WPAR RAS enhancements

This section discusses how the enhancement introduced with the RAS error
logging mechanism have been propagated to WPARs with AIX 7.1.

This feature first became available in AIX 7.1 and is included in AIX 6.1 TL 06.

3.5.1 Error logging mechanism aspect

The Reliability, Availability, and Serviceability (RAS) kernel services are used to
record the occurrence of hardware or software failures and to capture data about
these failures. The recorded information can be examined using the **errpt** or
trcrpt command.

WPAR mobility commands are integrating AIX messages as well as kernel
services error messages when possible. When an error occurs, these messages
were considered as not descriptive enough for a user.

Since AIX 7.1 is integrating a common error logging and reporting mechanism,
the goal was to propagate that mechanism to WPAR commands as well as for
WPAR mobility commands.

Mobility command error messages are available in the IBM System Director WPAR plug-in or WPAR manager log.

This section describes the message format of the WPAR command error or informative messages.

3.5.2 Goal for these messages

This new messages structure tends to address the following need:

► Have user-level messages as explicit with a resolution statement as possible.

► The messages include errno values when a failure without direct resolution statement occurs.

► When a failure occurs, the message gives information about the cause and the location of that failure to the support team to help debugging.

► Use of formatted messages with component names, component ID and message number enables easy scripting.

3.5.3 Syntax of the messages

The message structure is:

<component name> <component number>-<message number within the component> <message >

In Example 3-57, the `mkwpar` command issues a syntax error if the parameter is invalid, knowing that the following fields are fixed for that command:

► The component is the command name, mkwpar

► The component ID, 0960

► The message number, 077

Example 3-57   mkwpar user command error message

```
Global> mkwpar wpar1
mkwpar: 0960-077 Extra arguments found on the command line.
Usage: mkwpar [-a] [-A] [-b devexportsFile] [-B wparBackupDevice] [-c] [-C]...
```

For the same command, Example 3-58 on page 97, the error type is different. The message number is 299 when the component name and ID remain the same.

Example 3-58   Same command, other message number

```
Global> mkwpar -c -n test
mkwpar: 0960-299 Workload partition name test already exists in /etc/filesystems.
Specify another name.
```

For another WPAR command, such as `rmwpar`, the component remains 0960, but other fields would change (Example 3-59).

Example 3-59   Same component, other command

```
Global> rmwpar  wpar2
rmwpar: 0960-419 Could not find a workload partition called wpar2.
```

In some cases, two messages with different numbers can be displayed for an error—one usually providing resolution advice and one specifying the main error (Example 3-60).

Example 3-60   Multiple messages for a command

```
Global> rmwpar  wpar1
rmwpar: 0960-438 Workload partition wpar1 is running.
rmwpar: 0960-440 Specify -s or -F to stop the workload partition before removing

Global> lswpar -I
lswpar: 0960-568 wpar1 has no user-specified routes.
lswpar: 0960-559 Use the following command to see the
        full routing table for this workload partition:
            netstat -r -@ wpar1
```

As mentioned, WPAR mobility commands follow these rules, as shown in the command line output (Example 3-61).

Example 3-61   WPAR mobility command error messages

```
Global> /opt/mcr/bin/chkptwpar
1020-169 Usage:
To checkpoint an active WPAR:
        chkptwpar [-k | -p] -d /path/to/statefile [-o /path/to/logfile
[-l <debug|error>]] wparName

Global> /opt/mcr/bin/chkptwpar wpar1
1020-054 WPAR wpar1 is not checkpointable [09.211.0449]
1020-187 chkptwpar command failed.
```

These message structures may also apply to informative messages (Example 3-62).

Example 3-62   A few other informative messages

```
Global> mkwpar -c -n test2 -F
....
syncroot: Processing root part installation status.
syncroot: Installp root packages are currently synchronized.
syncroot: RPM root packages are currently synchronized.
syncroot: Root part is currently synchronized.
syncroot: Returns Status = SUCCESS
Workload partition test2 created successfully.
mkwpar: 0960-390 To start the workload partition, execute the following as root:
startwpar [-v] test2

Global> /opt/mcr/bin/chkptwpar -1 debug -o /test2/tmp/L -d /wpars/test2/tmp/D test2
1020-052 WPAR test2 is not active [09.211.0352]
1020-187 chkptwpar command failed.

Global> startwpar test2
Starting workload partition test2.
Mounting all workload partition file systems.
Loading workload partition.
Exporting workload partition devices.
Exporting workload partition kernel extensions.
Starting workload partition subsystem cor_test2.
0513-059 The cor_test2 Subsystem has been started. Subsystem PID is 4456462.
Verifying workload partition startup.

Global> /opt/mcr/bin/chkptwpar -1 debug -o /wpars/test2/tmp/L -d /wpars/test2/tmp/D test2
1020-191 WPAR test2 was checkpointed in /wpars/test2/tmp/D.
1020-186 chkptwpar command succeeded
```

3.6 WPAR migration to AIX Version 7.1

After successfully migrating a global instance running AIX V6.1 to AIX V7.1, all associated Workload Partitions (WPARs) also need to be migrated to the newer version of the operating system. The WPAR shares the same kernel as the global system. System software must be kept at the same level as the global environment in order to avoid unexpected results. There may be unexpected behavior if system calls, functions, or libraries are called from a WPAR that has not been migrated.

Prior to the migration to AIX V7.1, the global instance level of AIX was V6.1. WPARs were created with AIX V6.1. In order for the WPARs to function correctly after the migration to AIX V7.1, they must also be migrated. This is accomplished with the `migwpar` command.

A global instance of AIX is migrated with a normal AIX migration from one release of AIX to another. Refer to the *AIX Installation and Migration Guide*, SC23-6722 for information about migrating AIX, at:

`http://publib.boulder.ibm.com/infocenter/aix/v7r1/topic/com.ibm.aix.`
`install/doc/insgdrf/insgdrf_pdf.pdf`

WPAR migration is separate from a global instance migration. WPARs are not migrated automatically during an AIX migration. Once the global instance has been successfully migrated from AIX V6.1 to AIX V7.1, any associated WPARs must also be migrated to AIX V7.1.

Currently, only system WPARs are supported for migration. Both shared and detached system WPARs are supported. Shared system WPARs are those that do not have their own private `/usr` and `/opt` file systems. They share these file systems from the Global system.

A detached system WPAR (or non-shared system WPAR) has private `/usr` and `/opt` file systems, which are copied from the global environment. In order to migrate a WPAR of this type, the administrator must specify install media as the software source for the migration.

WPAR types that are not supported for migration are:
► Application WPARs
► Versioned WPARs

The `migwpar` command migrates a WPAR that was created in an AIX V6.1 Global instance, to AIX V7.1. Before attempting to use the `migwpar` command, you must ensure that the global system has migrated successfully first. The pre_migration and post_migration scripts can be run in the global instance before and after the migration to determine what software will be removed during the migration, to verify that the migration completed successfully, and identify software that did not migrate.

The pre_migration script is available on the AIX V7.1 media in the following location, /usr/lpp/bos/pre_migration. It can also be found in an AIX V7.1 NIM SPOT, for example, /export/spot/spotaix7100/usr/lpp/bos/pre_migration. The post_migration script is available in the following location, /usr/lpp/bos/post_migration, on an AIX V7.1 system.

Refer to the following website for further information relating to these scripts:

http://publib.boulder.ibm.com/infocenter/aix/v7r1/topic/com.ibm.aix.
install/doc/insgdrf/migration_scripts.htm

Table 3-1 describes the available flags and options for the `migwpar` command.

Table 3-1   migwpar flags and options

Flag	Description
-A	Migrates all migratable WPARs.
-f *wparNameFile*	Migrates the list of WPARs contained in the file `wparNamesFile`, one per line.
-d *software_source*	Installation location used for the detached (private) system WPAR migration.

Only the root user can run the `migwpar` command.

To migrate a single shared system WPAR from AIX V6.1 to AIX V7.1 you would execute this `migwpar` command:

```
# migwpar wpar1
```

A detached system WPAR can be migrated using the following `migwpar` command. The `/images` file system is used as the install source. This file system contains AIX V7.1 packages, copied from the install media.

```
# migwpar -d /images wpar1
```

To migrate all shared system WPARs to AIX V7.1, enter this command:

```
# migwpar -A
```

To migrate all detached WPARs, using /images as the software source, you would enter this command:

```
# migwpar -A -d /images
```

WPAR migration information is logged to the `/var/adm/ras/migwpar.log` file in the global environment. Additional software installation information is logged to the `/wpars/`*wparname*`/var/adm/ras/devinst.log` file for the WPAR, for example, `/wpars/wpar1/var/adm/ras/devinst.log` for wpar1.

> **Note:** If you attempt to run the **syncroot** command after a global instance migration and you have not run the **migwpar** command against the WPAR(s), you will receive the following error message:
>
> ```
> syncroot: Processing root part installation status.
> Your global system is at a higher version than the WPAR.
> Please log out of the WPAR and execute the migwpar command.
> syncroot: Returns Status = FAILURE
> ```
>
> If you run the **syncwpar** command to sync a Version 6 WPAR, on a Version 7 global system, the **syncwpar** command will call the **migwpar** command and will migrate the WPAR. If the SMIT interface to **syncwpar** is used (**smit syncwpar_sys**), the **migwpar** command will be called as required.

In the example that follows, we migrated a global instance of AIX V6.1 to AIX V7.1. We then verified that the migration was successful, before migrating a single shared system WPAR to AIX V7.1.

We performed the following steps to migrate the WPAR:

1. The **syncroot** and **syncwpar** commands should be run prior to migrating the Global instance. This is to verify the system software package integrity of the WPARs before the migration. The **oslevel**, **lslpp**, and **lppchk** commands can also assist in confirming the AIX level and fileset consistency.

2. Stop the WPAR prior to migrating the Global instance.

3. Migrate the Global instance of AIX V6.1 to AIX V7.1. The WPAR is not migrated and remains at AIX V6.1. Verify that the Global system migrates successfully first.

4. Start the WPAR and verify that the WPAR is functioning as expected, after the Global instance migration.

5. Migrate the WPAR to AIX V7.1 with the **migwpar** command.

6. Verify that the WPAR migrated successfully and is functioning as expected.

We confirmed that the WPAR was in an active state (A) prior to the migration, as shown in Example 3-63.

Example 3-63   Confirming the WPAR state is active

```
# lswpar
Name   State  Type  Hostname  Directory    RootVG WPAR
-----------------------------------------------------------
wpar1  A      S     wpar1     /wpars/wpar1 no
```

Prior to migrating the Global instance we first verified the current AIX version and level in both the global system and the WPAR, as shown in Example 3-64.

Example 3-64   Verifying Global and WPAR AIX instances prior to migration

```
# uname -W
0
# syncwpar wpar1
*******************************************************************************
Synchronizing workload partition wpar1 (1 of 1).
*******************************************************************************
Executing /usr/sbin/syncroot in workload partition wpar1.
syncroot: Processing root part installation status.
syncroot: Installp root packages are currently synchronized.
syncroot: RPM root packages are currently synchronized.
syncroot: Root part is currently synchronized.
syncroot: Returns Status = SUCCESS
Workload partition wpar1 synchronized successfully.

Return Status = SUCCESS.

# clogin wpar1
*******************************************************************************
*                                                                             *
*                                                                             *
*   Welcome to AIX Version 6.1!                                               *
*                                                                             *
*                                                                             *
*   Please see the README file in /usr/lpp/bos for information pertinent to   *
*   this release of the AIX Operating System.                                 *
*                                                                             *
*                                                                             *
*******************************************************************************
# uname -W
1
# syncroot
syncroot: Processing root part installation status.
syncroot: Installp root packages are currently synchronized.
syncroot: RPM root packages are currently synchronized.
syncroot: Root part is currently synchronized.
syncroot: Returns Status = SUCCESS
# exit

AIX Version 6
Copyright IBM Corporation, 1982, 2010.
```

```
login: root
root's Password:
*******************************************************************************
*                                                                             *
*                                                                             *
*  Welcome to AIX Version 6.1!                                                 *
*                                                                             *
*                                                                             *
*  Please see the README file in /usr/lpp/bos for information pertinent to     *
*  this release of the AIX Operating System.                                   *
*                                                                             *
*                                                                             *
*******************************************************************************
Last login: Fri Aug 27 17:14:27 CDT 2010 on /dev/vty0

# uname -W
0
# oslevel -s
6100-05-01-1016
# lppchk -m3 -v
#

# clogin wpar1
*******************************************************************************
*                                                                             *
*                                                                             *
*  Welcome to AIX Version 6.1!                                                 *
*                                                                             *
*                                                                             *
*  Please see the README file in /usr/lpp/bos for information pertinent to     *
*  this release of the AIX Operating System.                                   *
*                                                                             *
*                                                                             *
*******************************************************************************
Last login: Fri Aug 27 17:06:56 CDT 2010 on /dev/Global from r2r2m31

# uname -W
1
# oslevel -s
6100-05-01-1016
# lppchk -m3 -v
#
```

Before migrating the Global system, we stopped the WPAR cleanly, as shown in Example 3-65.

> **Note:** The **-F** flag has been specified with the `stopwpar` command to force the WPAR to stop quickly. This should only be performed after all applications in a WPAR have been stopped first.
>
> The **-v** flag has been specified with the `stopwpar` command to produce verbose output. This has been done in order to verify that the WPAR has in fact been stopped successfully. This is confirmed by the `Return Status = SUCCESS` message.
>
> Messages relating to the removal of inter-process communication (IPC) segments and semaphores are also shown, for example `ID=2097153 KEY=0x4107001c UID=0 GID=9 RT=-1`. These messages are generated by the /usr/lib/corrals/removeipc utility, which is called by the `stopwpar` command when stopping a WPAR.

Example 3-65   Clean shutdown of the WPAR

```
# stopwpar -Fv wpar1
Stopping workload partition wpar1.
Stopping workload partition subsystem cor_wpar1.
0513-044 The cor_wpar1 Subsystem was requested to stop.
Shutting down all workload partition processes.
WPAR='wpar1' CID=1
ID=2097153 KEY=0x4107001c UID=0 GID=9 RT=-1
ID=5242897 KEY=0x0100075e UID=0 GID=0 RT=-1
ID=5242898 KEY=0x620002de UID=0 GID=0 RT=-1
ID=9437203 KEY=0xffffffff UID=0 GID=0 RT=-1
wio0 Defined
Unmounting all workload partition file systems.
Umounting /wpars/wpar1/var.
Umounting /wpars/wpar1/usr.
Umounting /wpars/wpar1/tmp.
Umounting /wpars/wpar1/proc.
Umounting /wpars/wpar1/opt.
Umounting /wpars/wpar1/home.
Umounting /wpars/wpar1.
Return Status = SUCCESS.
```

We then migrated the global system from AIX V6.1 to AIX V7.1. This was accomplished with a normal AIX migration, using a virtual SCSI CD drive. Once

the migration completed successfully, we verified that the correct version of AIX was now available in the global environment, as shown in Example 3-66.

> **Note:** AIX V7.1 Technology Level 0, Service Pack 1 must be installed in the global instance prior to running the `migwpar` command.

Example 3-66   AIX Version 7.1 after migration

```
AIX Version 7
Copyright IBM Corporation, 1982, 2010.
login: root
root's Password:
*******************************************************************************
*                                                                             *
*                                                                             *
*  Welcome to AIX Version 7.1!                                                *
*                                                                             *
*                                                                             *
*  Please see the README file in /usr/lpp/bos for information pertinent to    *
*  this release of the AIX Operating System.                                  *
*                                                                             *
*                                                                             *
*******************************************************************************
1 unsuccessful login attempt since last login.
Last unsuccessful login: Tue Aug 31 17:21:56 CDT 2010 on /dev/pts/0 from 10.1.1.99
Last login: Tue Aug 31 17:21:20 CDT 2010 on /dev/vty0

# oslevel
7.1.0.0
# oslevel -s
7100-00-01-1037
# lppchk -m3 -v
#
```

The WPAR was not started and was in a defined (D) state, as shown in Example 3-67.

Example 3-67   WPAR not started after global instance migration to AIX V7.1

```
# lswpar
Name   State  Type  Hostname  Directory      RootVG WPAR
---------------------------------------------------------
wpar1  D      S     wpar1     /wpars/wpar1   no
```

The WPAR was then started successfully, as shown in Example 3-68.

> **Note:** The `-v` flag has been specified with the `startwpar` command to produce verbose output. This has been done in order to verify that the WPAR has in fact been started successfully. This is confimred by the `Return Status = SUCCESS` message.

Example 3-68   Starting the WPAR after global instance migration

```
# startwpar -v wpar1
Starting workload partition wpar1.
Mounting all workload partition file systems.
Mounting /wpars/wpar1
Mounting /wpars/wpar1/home
Mounting /wpars/wpar1/opt
Mounting /wpars/wpar1/proc
Mounting /wpars/wpar1/tmp
Mounting /wpars/wpar1/usr
Mounting /wpars/wpar1/var
Loading workload partition.
Exporting workload partition devices.
Exporting workload partition kernel extensions.
Starting workload partition subsystem cor_wpar1.
0513-059 The cor_wpar1 Subsystem has been started. Subsystem PID is 6619348.
Verifying workload partition startup.
Return Status = SUCCESS.
```

Although the global system was now running AIX V7.1, the WPAR was still running AIX V6.1, as shown in Example 3-69.

Example 3-69   Global instance migrated to Version 7, WPAR still running Version 6

```
# uname -W
0
# lslpp -l -O r bos.rte
  Fileset                      Level  State      Description
  ----------------------------------------------------------------------------
Path: /etc/objrepos
  bos.rte                      7.1.0.0  COMMITTED  Base Operating System Runtime
#
# clogin wpar1 lslpp -l -O r bos.rte
  Fileset                      Level  State      Description
  ----------------------------------------------------------------------------
Path: /etc/objrepos
```

The **migwpar** command was run against the WPAR to migrate it to AIX V7.1, as shown in Example 3-70. Only partial output is shown because the actual migration log is extremely verbose.

Example 3-70   WPAR migration to AIX V7.1 with migwpar

```
# migwpar wpar1

Shared /usr WPAR list:
wpar1
WPAR wpar1 mount point:
/wpars/wpar1
WPAR wpar1 active
MIGWPAR: Saving configuration files for wpar1
MIGWPAR: Removing old bos files for wpar1
MIGWPAR: Replacing bos files for wpar1
MIGWPAR: Merging configuration files for wpar1
0518-307 odmdelete: 1 objects deleted.
0518-307 odmdelete: 0 objects deleted.
0518-307 odmdelete: 2 objects deleted.
....
x ./lib
x ./audit
x ./dev
x ./etc
x ./etc/check_config.files
x ./etc/consdef
x ./etc/cronlog.conf
x ./etc/csh.cshrc
x ./etc/csh.login
x ./etc/dlpi.conf
x ./etc/dumpdates
x ./etc/environment
x ./etc/ewlm
x ./etc/ewlm/limits
x ./etc/ewlm/trc
x ./etc/ewlm/trc/config_schema.xsd
x ./etc/ewlm/trc/output_schema.xsd
x ./etc/filesystems
x ./etc/group
x ./etc/inittab
...
MIGWPAR: Merging configuration files for wpar1
```

```
                        0518-307 odmdelete: 1 objects deleted.
                        MIGWPAR: Running syncroot for wpar1
                        syncroot: Processing root part installation status.
                        syncroot: Synchronizing installp software.
                        syncroot: Processing root part installation status.
                        syncroot: Installp root packages are currently synchronized.
                        syncroot: RPM root packages are currently synchronized.
                        syncroot: Root part is currently synchronized.
                        syncroot: Returns Status = SUCCESS
                        Cleaning up ...
```

We logged into the WPAR using the **clogin** command after the migration to verify
that the WPAR was functioning as expected, as shown in Example 3-71.

Example 3-71   Verifying that WPAR started successfully after migration

```
# clogin wpar1
*******************************************************************************
*                                                                             *
*                                                                             *
*  Welcome to AIX Version 7.1!                                                *
*                                                                             *
*                                                                             *
*  Please see the README file in /usr/lpp/bos for information pertinent to    *
*  this release of the AIX Operating System.                                  *
*                                                                             *
*                                                                             *
*******************************************************************************
Last login: Tue Aug 31 17:32:48 CDT 2010 on /dev/Global from r2r2m31

# oslevel
7.1.0.0
# oslevel -s
7100-00-01-1037
# lppchk -m3 -v
#
# lslpp -l -O u bos.rte
  Fileset                        Level  State      Description
  ----------------------------------------------------------------------------
Path: /usr/lib/objrepos
  bos.rte                        7.1.0.1  COMMITTED  Base Operating System Runtime

# uname -W
1
# df
```

```
Filesystem      512-blocks      Free %Used     Iused %Iused Mounted on
Global             262144    205616   22%       1842     8% /
Global             262144    257320    2%          5     1% /home
Global             786432    377888   52%       8696    18% /opt
Global                  -         -    -           -     - /proc
Global             262144    252456    4%         15     1% /tmp
Global            3932160    321192   92%      39631    51% /usr
Global             262144     94672   64%       4419    29% /var
```

Both the global system and the shared system WPAR have been successfully migrated to AIX V7.1.

In Example 3-72, a detached WPAR is migrated to AIX V7.1. Prior to migrating the WPAR, the global instance was migrated from AIX V6.1 to AIX V7.1.

> **Note:** After the global instance migration to AIX V7.1, the detached Version 6 WPAR (wpar0) is unable to start because it must be migrated first.

The `migwpar` command is called with the **-d /images** flag and option. The /images directory is an NFS mounted file system that resides on a NIM master. The file system contains an AIX V7.1 LPP source on the NIM master.

Once the `migwpar` command has completed successfully, we started the WPAR and confirmed that it had migrated to AIX V7.1. Only partial output from the `migwpar` command is shown because the actual migration log is extremely verbose.

Example 3-72   Migrating a detached WPAR to AIX V7.1

```
# uname -W
0
# oslevel -s
7100-00-01-1037
# lswpar
Name   State Type  Hostname  Directory     RootVG WPAR
----------------------------------------------------------
wpar0  D     S     wpar0     /wpars/wpar0  no

# startwpar -v wpar0
Starting workload partition wpar0.
Mounting all workload partition file systems.
Mounting /wpars/wpar0
Mounting /wpars/wpar0/home
Mounting /wpars/wpar0/opt
Mounting /wpars/wpar0/proc
```

```
Mounting /wpars/wpar0/tmp
Mounting /wpars/wpar0/usr
Mounting /wpars/wpar0/var
startwpar: 0960-667 The operating system level within the workload partition is not
supported.
Unmounting all workload partition file systems.
Umounting /wpars/wpar0/var.
Umounting /wpars/wpar0/usr.
Umounting /wpars/wpar0/tmp.
Umounting /wpars/wpar0/proc.
Umounting /wpars/wpar0/opt.
Umounting /wpars/wpar0/home.
Umounting /wpars/wpar0.
Return Status = FAILURE.
#
# mount 7502lp01:/export/lppsrc/aix7101 /images
# df /images
Filesystem     512-blocks      Free %Used     Iused %Iused Mounted on
7502lp01:/export/lppsrc/aix7101   29425664   4204400   86%      3384      1% /images

# ls -ltr /images
total 0
drwxr-xr-x    3 root      system            256 Sep 09 09:31 RPMS
drwxr-xr-x    3 root      system            256 Sep 09 09:31 usr
drwxr-xr-x    3 root      system            256 Sep 09 09:31 installp

# migwpar -d /images wpar0

Detached WPAR list:
wpar0
WPAR wpar0 mount point:
/wpars/wpar0
Mounting all workload partition file systems.
Loading workload partition.
Saving system configuration files.

Checking for initial required migration space.
Setting up for base operating system restore.
/

Restoring base operating system.
Merging system configuration files.
......
Installing and migrating software.
Updating install utilities.
```

```
........
FILESET STATISTICS
------------------
  725  Selected to be installed, of which:
      720  Passed pre-installation verification
        5  Already installed (directly or via superseding filesets)
    2  Additional requisites to be automatically installed
  ----
  722  Total to be installed


+-----------------------------------------------------------------------------+
                       Installing Software...
+-----------------------------------------------------------------------------+

installp:  APPLYING software for:
        xlC.aix61.rte 11.1.0.1



. . . . . << Copyright notice for xlC.aix61 >> . . . . . . .
 Licensed Materials - Property of IBM

 5724X1301
   Copyright IBM Corp. 1991, 2010.
   Copyright AT&T 1984, 1985, 1986, 1987, 1988, 1989.
   Copyright Unix System Labs, Inc., a subsidiary of Novell, Inc. 1993.
 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 xlC.aix61 >>. . . .


Filesets processed:  1 of 722  (Total time:  4 secs).

installp:  APPLYING software for:
        wio.vscsi 7.1.0.0
.....
Restoring device ODM database.
Shutting down all workload partition processes.
Unloading workload partition.
Unmounting all workload partition file systems.

Cleaning up ...

# startwpar -v wpar0
Starting workload partition wpar0.
Mounting all workload partition file systems.
```

```
Mounting /wpars/wpar0
Mounting /wpars/wpar0/home
Mounting /wpars/wpar0/opt
Mounting /wpars/wpar0/proc
Mounting /wpars/wpar0/tmp
Mounting /wpars/wpar0/usr
Mounting /wpars/wpar0/var
Loading workload partition.
Exporting workload partition devices.
Exporting workload partition kernel extensions.
Starting workload partition subsystem cor_wpar0.
0513-059 The cor_wpar0 Subsystem has been started. Subsystem PID is 7995618.
Verifying workload partition startup.
Return Status = SUCCESS.
#
# clogin wpar0
*******************************************************************************
*                                                                             *
*                                                                             *
*  Welcome to AIX Version 7.1!                                                *
*                                                                             *
*                                                                             *
*  Please see the README file in /usr/lpp/bos for information pertinent to    *
*  this release of the AIX Operating System.                                  *
*                                                                             *
*                                                                             *
*******************************************************************************
Last login: Mon Sep 13 22:19:20 CDT 2010 on /dev/Global from 75021p03

# oslevel -s
7100-00-01-1037
```

4

Continuous availability

This chapter discusses the topics related to continuous availability:

4.1 Firmware-assisted dump

This section discusses the differences in the firmware-assisted dump in AIX V7.1.

4.1.1 Default installation configuration

The introduction of the POWER6® processor-based systems allowed system dumps to be firmware assisted. When performing a firmware-assisted dump, system memory is frozen and the partition rebooted, which allows a new instance of the operating system to complete the dump.

Firmware-assisted dump is now the default dump type in AIX V7.1, when the hardware platform supports firmware-assisted dump.

The traditional dump remains the default dump type for AIX V6.1, even when the hardware platform supports firmware-assisted dump.

Firmware-assisted dump offers improved reliability over the traditional dump type, by rebooting the partition and using a new kernel to dump data from the previous kernel crash.

Firmware-assisted dump requires:

► A POWER6 processor-based or later hardware platform.

► The LPAR must have a minimum of 1.5 GB memory.

► The dump logical volume must be in the root volume group.

► Paging space cannot be defined as the dump logical volume.

In the unlikely event that a firmware-assisted system may encounter a problem with execution, the firmware-assisted dump will be substituted by a traditional dump for this instance.

Example 4-1 shows the `sysdumpdev -l` command output from an AIX V6.1 LPAR. The system dump type has not been modified from the default installation setting. The field `type of dump` displays `traditional`. This shows that the partition default dump type is traditional and not a firmware-assisted dump.

Example 4-1   The sysdumpdev -l output in AIX V6.1

```
# oslevel -s
6100-00-03-0808
# sysdumpdev -l
primary              /dev/lg_dumplv
```

```
secondary              /dev/sysdumpnull
copy directory         /var/adm/ras
forced copy flag       TRUE
always allow dump      FALSE
dump compression       ON
type of dump           traditional
#
```

Example 4-2 shows the **sysdumpdev -l** command output from an AIX V7.1 LPAR. The system dump type has not been modified from the default installation setting. The field `type of dump` displays `fw-assisted`. This shows that the AIX V7.1 partition default dump type is firmware assisted and not traditional.

Example 4-2   The sysdumpdev -l output in AIX V7.1

```
# oslevel -s
7100-00-00-0000
# sysdumpdev -l
primary                /dev/lg_dumplv
secondary              /dev/sysdumpnull
copy directory         /var/adm/ras
forced copy flag       TRUE
always allow dump      FALSE
dump compression       ON
type of dump           fw-assisted
full memory dump       disallow
#
```

4.1.2 Full memory dump options

When firmware-assisted dump is enabled, the **sysdumpdev -l** command displays the `full memory dump` option. The `full memory dump` option can be set with the **sysdumpdev -f** command. This option will only be displayed when the dump type is firmware-assisted dump.

The `full memory dump` option specifies the mode in which the firmware-assisted dump will operate. The administrator can configure firmware-assisted dump to allow, disallow, or require the dump of the full system memory.

Table 4-1 on page 116 lists the `full memory dump` options available with the **sysdumpdev -f** command.

Table 4-1 Full memory dump options available with the sysdumpdev -f command

Option	Description
disallow	Selective memory dump only. A full memory system dump is not allowed. This is the default.
allow | allow_full	The full memory system dump mode is allowed but is performed only when the operating system cannot properly handle the dump request.
require | require_full	The full memory system dump mode is allowed and is always performed.

In Example 4-3 the `full memory dump` option is changed from `disallow` to `require` with the **sysdumpdev -f** command. When modifying the `full memory dump` option from `disallow` to `require`, the next firmware-assisted dump will always perform a full system memory dump.

Example 4-3 Setting the full memory dump option with the sysdumpdev -f command

```
# sysdumpdev -l
primary              /dev/lg_dumplv
secondary            /dev/sysdumpnull
copy directory       /var/adm/ras
forced copy flag     TRUE
always allow dump    FALSE
dump compression     ON
type of dump         fw-assisted
full memory dump     disallow
# sysdumpdev -f require
# sysdumpdev -l
primary              /dev/lg_dumplv
secondary            /dev/sysdumpnull
copy directory       /var/adm/ras
forced copy flag     TRUE
always allow dump    FALSE
dump compression     ON
type of dump         fw-assisted
full memory dump     require
#
```

4.1.3 Changing the dump type on AIX V7.1

The firmware-assisted dump may be changed to traditional dump with the **sysdumpdev -t** command. Using the traditional dump functionality will not allow

the full memory dump options in Table 4-1 on page 116 to be executed, because these options are only available with firmware-assisted dump.

Changing from firmware-assisted to traditional dump will take effect immediately and does not require a reboot of the partition. Example 4-4 shows the **sysdumpdev -t** command being used to change the dump type from firmware-assisted to traditional dump.

Example 4-4   Changing to the traditional dump on AIX V7.1

```
# sysdumpdev -l
primary             /dev/lg_dumplv
secondary           /dev/sysdumpnull
copy directory      /var/adm/ras
forced copy flag    TRUE
always allow dump   FALSE
dump compression    ON
type of dump        fw-assisted
full memory dump    require
# sysdumpdev -t traditional
# sysdumpdev -l
primary             /dev/lg_dumplv
secondary           /dev/sysdumpnull
copy directory      /var/adm/ras
forced copy flag    TRUE
always allow dump   FALSE
dump compression    ON
type of dump        traditional
```

Note: When reverting to traditional dump, the full memory dump options are no longer available because these are options only available with firmware-assisted dump.

A partition configured to use the traditional dump may have the dump type changed to firmware-assisted. If the partition had previously been configured to use firmware-assisted dump, any full memory dump options will be preserved and defined when firmware-assisted dump is reinstated.

Changing from traditional to firmware-assisted dump requires a reboot of the partition for the dump changes to take effect.

> **Note:** Firmware-assisted dump may be configured on POWER5™ or earlier based hardware, but all system dumps will operate as traditional dump. POWER6 is the minimum hardware platform required to support firmware-assisted dump.

Example 4-5 shows the `sysdumpdev -t` command being used to reinstate firmware-assisted dump on a server configured to use the traditional dump.

Example 4-5   Reinstating firmware-assisted dump with the sysdumpdev -t command

```
# sysdumpdev -l
primary              /dev/lg_dumplv
secondary            /dev/sysdumpnull
copy directory       /var/adm/ras
forced copy flag     TRUE
always allow dump    FALSE
dump compression     ON
type of dump         traditional
# sysdumpdev -t fw-assisted
Attention: the firmware-assisted system dump will be configured at the
next reboot.
# sysdumpdev -l
primary              /dev/lg_dumplv
secondary            /dev/sysdumpnull
copy directory       /var/adm/ras
forced copy flag     TRUE
always allow dump    FALSE
dump compression     ON
type of dump         traditional
```

In Example 4-5 the message `Attention: the firmware-assisted system dump will be configured at the next reboot` is displayed once the `sysdumpdev -t fw-assisted` command has completed.

When a partition configured for firmware-assisted dump is booted, a portion of memory known as the *scratch area* is allocated to be used by the firmware-assisted dump functionality. For this reason, a partition configured to use the traditional system dump requires a reboot to allocate the *scratch area* memory that is required for a fimware-assisted dump to be initiated.

If the partition is not rebooted, firmware-assisted dump will not be activated until such a time as the partition reboot is completed.

> **Note:** When an administrator attempts to switch from a traditional to firmware-assisted system dump, system memory is checked against the firmware-assisted system dump memory requirements. If these memory requirements are not met, then the **sysdumpdev -t** command output reports the required minimum system memory to allow for firmware-assisted dump to be configured.

Example 4-6 shows the partition reboot to allow for memory allocation and activation of firmware-assisted dump. Though firmware-assisted dump has been enabled, the **sysdumpdev -l** command displays the dump type as traditional because the partition has not yet been rebooted after the change to firmware-assisted dump.

Example 4-6   Partition reboot to activate firmware-assisted dump

```
# sysdumpdev -l
primary             /dev/lg_dumplv
secondary           /dev/sysdumpnull
copy directory      /var/adm/ras
forced copy flag    TRUE
always allow dump   FALSE
dump compression    ON
type of dump        traditional
# shutdown -Fr

SHUTDOWN PROGRAM
...
...
Stopping The LWI Nonstop Profile...
Waiting for The LWI Nonstop Profile to exit...
Stopped The LWI Nonstop Profile.
0513-044 The sshd Subsystem was requested to stop.

Wait for 'Rebooting...' before stopping.
Error reporting has stopped.
Advanced Accounting has stopped...
Process accounting has stopped.
```

Example 4-7 on page 120 shows the partition after the reboot. The type of dump is displayed with the **sysdumpdev -l** command, showing that the dump type is now set to fw-assisted.

Because this is the same partition that we previously modified the `full memory dump` option to `require`, then changed the type of dump to traditional, the `full memory dump` option is reinstated once the dump type is reverted to firmware-assisted.

Example 4-7   The sysdumpdev -l command after partition reboot

```
# uptime
  06:15PM   up 1 min,  1 user,  load average: 1.12, 0.33, 0.12
# sysdumpdev -l
primary             /dev/lg_dumplv
secondary           /dev/sysdumpnull
copy directory      /var/adm/ras
forced copy flag    TRUE
always allow dump   FALSE
dump compression    ON
type of dump        fw-assisted
full memory dump    require
#
```

4.1.4 Firmware-assisted dump on POWER5 and earlier hardware

The minimum supported hardware platform for firmware-assisted dump is the POWER6 processor based system.

In Example 4-8 we see a typical message output when attempting to enable firmware-assisted dump on a pre-POWER6 processor-based system. In this example the AIX V7.1 is operating on a POWER5 model p550 system.

Example 4-8   Attempting to enable firmware-assisted dump on a POWER5

```
# oslevel -s
7100-00-00-0000
# uname -M
IBM,9113-550
# lsattr -El proc0
frequency   1654344000     Processor Speed       False
smt_enabled true           Processor SMT enabled False
smt_threads 2              Processor SMT threads False
state       enable         Processor state       False
type        PowerPC_POWER5 Processor type        False
# sysdumpdev -l
primary             /dev/hd6
secondary           /dev/sysdumpnull
copy directory      /var/adm/ras
```

```
forced copy flag      TRUE
always allow dump     FALSE
dump compression      ON
type of dump          traditional
# sysdumpdev -t fw-assisted
Cannot set the dump force_system_dump attribute.
        An attempt was made to set an attribute to an unsupported
value.
Firmware-assisted system dump is not supported on this platform.
# sysdumpdev -l
primary               /dev/hd6
secondary             /dev/sysdumpnull
copy directory        /var/adm/ras
forced copy flag      TRUE
always allow dump     FALSE
dump compression      ON
type of dump          traditional
#
```

In Example 4-8 on page 120, even though AIX V7.1 supports firmware-assisted dump as the default dump type, the POWER5 hardware platform does not support firmware-assisted dump, so the dump type at AIX V7.1 installation was set to traditional.

When the dump type was changed to firmware-assisted with the `sysdumpdev -t` command, the message `Firmware-assisted system dump is not supported on this platform` was displayed and the dump type remained set to traditional.

4.1.5 Firmware-assisted dump support for non-boot iSCSI device

The release of AIX Version 6.1 with the 6100-01 Technology Level introduced support for an iSCSI device to be configured as a dump device for firmware-assisted system dump.

The `sysdumpdev` command could be used to configure an iSCSI logical volume as a dump device. In AIX V6.1, it was mandatory that this dump device be located on an iSCSI boot device.

With the release of AIX V7.1, firmware-assisted dump also supports dump devices located on arbitrary non-boot iSCSI disks. This allows diskless servers to dump to remote iSCSI disks using firmware-assisted dump. The iSCSI disks must be members of the root volume group.

4.2 User key enhancements

AIX 7.1 allows for configuring the number of user storage keys. It also allows a mode where all hardware keys are dedicated to user keys. This helps in developing large applications to use more user keys for application-specific needs.

> **Note:** By dedicating all of the hardware keys to user keys, kernel storage keys will get disabled. However, we do *not* recommend this, because the kernel storage keys will not be able to help debug the kernel memory problems any more if they are disabled.

Table 4-2 lists the maximum number of supported hardware keys on different hardware platforms.

Table 4-2   Number of storage keys supported

Power hardware platform	Maximized supported hardware keys on AIX
P5++	4
P6	8
P6+	15
P7	31

The `skctl` command is used to configure storage keys. Example 4-9 shows the usage of this command. It also shows how to view the existing settings and how to modify them.

The `smitty skctl` fastpath can also be used to configure storage keys. So one can use either the `skctl` command or the `smitty skctl` interface for configuration.

Example 4-9   Configuring storage keys

```
# skctl -?
skctl: Not a recognized flag: ?
skctl: usage error
Usage: skctl [-D]
       skctl [-u <nukeys>/off] [-k on/off/default]
       skctl [-v [now|default|boot]

             where:
                       -u <nukeys>   # number of user keys (2 - max.
no. of hardware keys)
```

```
                        -u off      # disable user keys
                        -k on/off   # enable/disable kernel keys
                        -k default  # set default kernel key state
                        -D          # use defaults
                        -v now      # view current settings
                        -v default  # view defaults
                        -v boot     # view settings for next boot
# skctl -v default
Default values for Storage Key attributes:

        Max. number of hardware keys      = 31
        Number of hardware keys enabled   = 31
        Number of user keys               = 7
        Kernel keys state                 = enabled
# skctl -v now
Storage Key attributes for current boot session:

        Max. number of hardware keys      = 31
        Number of hardware keys enabled   = 31
        Number of user keys               = 12
        Kernel keys state                 = enabled
# skctl -u 15
# skctl -v boot
Storage Key attributes for next boot session:

        Max. number of hardware keys      = default
        Number of hardware keys enabled   = default
        Number of user keys               = 15
        Kernel keys state                 = default
```

4.3 Cluster Data Aggregation Tool

First Failure Data Capture (FFDC) is a technique that ensures that when a fault is
detected in a system (through error checkers or other types of detection
methods), the root cause of the fault is captured without the need to recreate the
problem or run any sort of extended tracing or diagnostics program. Further
information about FFDC can be found in *IBM AIX Continuous Availability
Features*, REDP-4367.

FFDC has been enhanced to provide capabilities for quick analysis and root
cause identification for problems that arise in workloads that span multiple

systems. FFDC data will be collected on each of the configured nodes by the Cluster Data Aggregation Tool.

The Cluster Data Aggregation Tool environment consists of a central node and remote nodes. The central node is where the Cluster Data Aggregation Tool is installed and executed from. It hosts the data collection repository, which is a new file system that contains collection of data from multiple remote nodes. The remote nodes are where FFDC data is collected, which is AIX LPARs (AIX 6.1 TL3), VIOS (2.1.1.0 based on AIX 6.1 TL3), or HMC (V7 R 3.4.2). The central node must be able to connect as an administrator user on the remote nodes. There is no need to install the Cluster Data Aggregation Tool on these remote nodes. For making a secure connection, the SSH package should be installed on these nodes.

The Cluster Data Aggregation Tool is known by the `cdat` command. It is divided into several subcommands. The subcommands are `init`, `show`, `check`, `delete`, `discover-nodes`, `list-nodes`, `access`, `collect`, `list-types`, and `archive`. Only the `init` subcommand needs to be executed by the privileged user (root). The `init` subcommand creates the data infrastructure and defines the user used to run all other subcommands. It initializes the Cluster Data Aggregation repository.

> **Note:** To prevent concurrent accesses to the Cluster Data Aggregation Tool configuration files, running multiple instances of the `cdat` command is forbidden and the repository is protected by a lock file.

The `smitty cdat` fastpath can also be used to configure the Cluster Data Aggregation Tool. So one can use either the `cdat` command or the `smitty cdat` interface for configuration.

Example 4-10 shows usage of the `cdat` command in configuring the Cluster Data Aggregation Tool.

Example 4-10   Configuring Cluster Data Aggregation Tool

```
# cdat -?
0965-030: Unknown sub-command: '-?'.

Usage: cdat sub-command [options]
Available sub-commands:
        init              Initialize the repository
        show              Display the content of the repository
        check             Check consistency of the repository
        delete            Remove collects from the repository
        discover-nodes    Find LPARs or WPARs from a list of HMCs or
LPARs
```

```
            list-nodes          Display the list of configured nodes
            access              Manage remote nodes authentication
            collect             Collect data from remote nodes
            list-types          Display the list of supported collect types
            archive             Create a compressed archive of collects

# cdat init
Checking user cdat...Creating missing user.
Changing password for "cdat"
cdat's New password:
Enter the new password again:
Checking for SSH...found
Checking for SSH keys...generated
Checking directory /cdat...created
Checking XML file...created
Done.

# cdat show
Repository: /cdat
Local user: cdat

# cdat check
Repository is valid.

# cdat discover-nodes -?
Unknown option: ?
Usage: cdat discover-nodes -h
       cdat discover-nodes [-a|-w] [-f File] -n Type:[User@]Node ...

# cdat discover-nodes -n HMC:hscroot@192.168.100.111
Discovering nodes managed by hscroot@192.168.100.111...
The authenticity of host '192.168.100.111 (192.168.100.111)' can't be
established.
RSA key fingerprint is ee:5e:55:37:df:31:b6:78:1f:01:6d:f5:d1:67:d6:4f.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.100.111' (RSA) to the list of known
hosts.
Password:
Done.

# cat /cdat/nodes.txt
HMC:192.168.100.111
# LPARs of managed system 750_1-8233-E8B-061AA6P
LPAR:750_1_LP01
LPAR:750_1_LP02
```

```
LPAR:750_1_LP03
LPAR:750_1_LP04
VIOS:750_1_VIO_1
# Could not retrieve LPARs of managed system 750_2-8233-E8B-061AB2P
# HSCL0237 This operation is not allowed when the managed system is in
the No Connection state.  After you have established a connection from
the HMC to the managed system and have entered a valid HMC access
password, try the operation again.

# cdat list-nodes
HMC  192.168.100.111
LPAR 750_1_LP01
LPAR 750_1_LP02
LPAR 750_1_LP03
LPAR 750_1_LP04
VIOS 750_1_VIO_1

# cdat list-types
List of available collect types:

perfpmr (/usr/lib/cdat/types/perfpmr):
    Retrieves the result of the perfpmr command from nodes of type
LPAR.

psrasgrab (/usr/lib/cdat/types/psrasgrab):
    Harvests logs from a Centralized RAS Repository.

psrasinit (/usr/lib/cdat/types/psrasinit):
    Configures Centralized RAS pureScale clients.

psrasremove (/usr/lib/cdat/types/psrasremove):
    Unconfigures Centralized RAS pureScale clients.

snap (/usr/lib/cdat/types/snap):
    Gathers system configuration information from nodes of type LPAR or
VIOS.

trace (/usr/lib/cdat/types/trace):
    Records selected system events from nodes of type LPAR or VIOS.

# cdat access -?
Unknown option: ?
Usage: cdat access -h
        cdat access [-dF] [-u User] -n Type:[User@]Node ...
        cdat access [-dF] [-u User] -f File ...
```

```
# cdat access -n LPAR:root@192.168.101.13 -n LPAR:root@192.168.101.11
The collect user will be created with the same password on all nodes.
Please enter a password for the collect user:
Re-enter the collect user password:
Initializing access to 'root' on host '192.168.101.13'...
Trying 'ssh'...found
The authenticity of host '192.168.101.13 (192.168.101.13)' can't be
established.
RSA key fingerprint is de:7d:f9:ec:8f:ee:e6:1e:8c:aa:18:b3:54:a9:d4:e0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.101.13' (RSA) to the list of known
hosts.
root@192.168.101.13's password:
Initializing access to 'root' on host '192.168.101.11'...
Trying 'ssh'...found
The authenticity of host '192.168.101.11 (192.168.101.11)' can't be
established.
RSA key fingerprint is 28:98:b8:d5:97:ec:86:84:d5:9e:06:ac:3b:b4:c6:5c.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.101.11' (RSA) to the list of known
hosts.
root@192.168.101.11's password:
Done.

# cdat collect -t trace -n LPAR:root@192.168.101.13 -n
LPAR:root@192.168.101.11
Is the collect for IBM support? (y/n) [y]: y
Please enter a PMR number: 12345,678,123
See file /cdat/00000003/logs.txt for detailed status.
Starting collect type "trace"
Collect type "trace" done, see results in "/cdat/00000003/trace/".
==============
Status report:
==============
192.168.101.11: SUCCEEDED
192.168.101.13: SUCCEEDED

# find /cdat/00000003/trace/
/cdat/00000003/trace/
/cdat/00000003/trace/192.168.101.11
/cdat/00000003/trace/192.168.101.11/logs.txt
/cdat/00000003/trace/192.168.101.11/trcfile
/cdat/00000003/trace/192.168.101.11/trcfmt
/cdat/00000003/trace/192.168.101.13
```

```
/cdat/00000003/trace/192.168.101.13/logs.txt
/cdat/00000003/trace/192.168.101.13/trcfile
/cdat/00000003/trace/192.168.101.13/trcfmt

# cdat show -v
Repository: /cdat
Local user: cdat

1: 2010-08-31T12:39:29

    PMR: 12345,123,123
    Location: /cdat/00000001/

2: 2010-08-31T12:40:24

    PMR: 12345,123,123
    Location: /cdat/00000002/

3: 2010-08-31T12:58:31

    PMR: 12345,678,123
    Location: /cdat/00000003/

    192.168.101.11:
        type      : LPAR
        user      : root
        machine id : 00F61AA64C00
        lpar id   : 2
        timezone  : EDT

    192.168.101.13:
        type      : LPAR
        user      : root
        machine id : 00F61AA64C00
        lpar id   : 4
        timezone  : EDT

# cdat archive -p 12345,678,123 -f archive
Compressed archive successfully created at archive.tar.Z.
```

It is possible to schedule periodic data collections using the **crontab** command.
For instance, to run the snap collect type every day at midnight:

```
# crontab -e cdat
```

```
0 0 * * * /usr/bin/cdat collect -q -t snap -f /cdat/nodes.txt
```

With this configuration, **cdat** creates a new directory under /cdat (and a new collect ID) every day at midnight that will contain the snap data for each node present in /cdat/nodes.txt.

Scheduled collects can also be managed transparently using the **smitty cdat_schedule** fastpath.

4.4 Cluster Aware AIX

The Cluster Aware AIX (CAA) services help in creating and managing a cluster of AIX nodes to build a highly available and ideal architectural solution for a data center. IBM cluster products such as Reliable Scalable Cluster Technology (RSCT) and PowerHA use these services. CAA services can assist in the management and monitoring of an arbitrary set of nodes or in running a third-party cluster software.

The rest of this section discusses additional details about each of these services together with examples using commands to configure and manage the cluster.

CAA services are basically a set of commands and services that the cluster software can exploit to provide high availability and disaster recovery support to external applications. The CAA services are broadly classified into the following:

Clusterwide event management
> The AIX Event Infrastructure (5.12, "AIX Event Infrastructure" on page 202) allows event propagation across the cluster so that applications can monitor events from any node in the cluster.

Clusterwide storage naming service
> When a cluster is defined or modified, the AIX interfaces automatically create a consistent shared device view across the cluster. A global device name, such as *cldisk1*, would refer to the same physical disk from any node in the cluster.

Clusterwide command distribution
> The **clcmd** command provides a facility to distribute a command to a set of nodes that are members of a cluster. For example, the command **clcmd date** returns the output of the **date** command from each of the nodes in the cluster.

Clusterwide communication
> Communication between nodes within the cluster is achieved using multicasting over the IP-based network and also using storage interface communication through Fibre Channel and

SAS adapters. A new socket family (AF_CLUST) has been provided for reliable, in-order communication between nodes. When all network interfaces are lost, applications using these interfaces can still run.

The nodes that are part of the cluster should have common storage devices, either through the Storage Attached Network (SAN) or through the Serial-Attached SCSI (SAS) subsystems.

4.4.1 Cluster configuration

This section describes the commands used to create and manage clusters. A sample cluster is created to explain the usage of these commands. Table 4-3 lists them with a brief description.

Table 4-3   Cluster commands

Command	Description
`mkcluster`	Used to create a cluster.
`chcluster`	Used to change a cluster configuration.
`rmcluster`	Used to remove a cluster configuration.
`lscluster`	Used to list cluster configuration information.
`clcmd`	Used to distribute a command to a set of nodes that are members of a cluster.

The following is a sample of creating a cluster on one of the nodes, nodeA. Before creating the cluster the `lscluster` command is used to make sure that no cluster already exists. The list of physical disks is displayed using the `lspv` command to help determine which disks to choose. Note the names of the disks that will be used for the shared cluster disks, hdisk4, hdisk5, hdisk6 and hdisk7. Example 4-11 shows the output of the commands used to determine the information needed before creating the cluster.

Example 4-11   Before creating a cluster

```
# hostname
nodeA
# lscluster -m
Cluster services are not active.
# lspv
hdisk0          00cad74fd6d58ac1                        rootvg          active
hdisk1          00cad74fa9d3b7e1                        None
hdisk2          00cad74fa9d3b8de                        None
```

```
hdisk3          00cad74f3964114a                          None
hdisk4          00cad74f3963c575                          None
hdisk5          00cad74f3963c671                          None
hdisk6          00cad74f3963c6fa                          None
hdisk7          00cad74f3963c775                          None
hdisk8          00cad74f3963c7f7                          None
hdisk9          00cad74f3963c873                          None
hdisk10         00cad74f3963ca13                          None
hdisk11         00cad74f3963caa9                          None
hdisk12         00cad74f3963cb29                          None
hdisk13         00cad74f3963cba4                          None
```

The **mkcluster** command is used to create the cluster. Example 4-12 shows the use of the **mkcluster** command.

The **-r** option is used to specify the repository disk used for storing cluster configuration information.

The **-d** option is used to specify cluster disks, each of which will be renamed to a new name beginning with cldisk*. Each of these cluster disks can be referenced by the new name from any of the nodes in the cluster. These new disk names refer to the same physical disk.

The **-s** option is used to specify the multicast address that is used for communication between the nodes in the cluster.

The **-m** option is used to specify the nodes which will be part of the cluster. Nodes are identified by the fully qualified hostnames as defined in DNS or with the local /etc/hosts file configuration.

The **lscluster** command is used to verify the creation of a cluster. The **lspv** command shows the new names of the cluster disks.

Example 4-12   Creating the cluster

```
# mkcluster -r hdisk3 -d hdisk4,hdisk5,hdisk6,hdisk7 -s 227.1.1.211 -m
nodeA,nodeB,nodeC
Preserving 23812 bytes of symbol table [/usr/lib/drivers/ahafs.ext]
Preserving 19979 bytes of symbol table [/usr/lib/drivers/dpcomdd]
mkcluster: Cluster shared disks are automatically renamed to names such as
        cldisk1, [cldisk2, ...] on all cluster nodes.  However, this cannot
        take place while a disk is busy or on a node which is down or not
        reachable.  If any disks cannot be renamed now, they will be renamed
        later by the clconfd daemon, when the node is available and the disks
        are not busy.
```

```
# lscluster -m
Calling node query for all nodes
Node query number of nodes examined: 3

        Node name: nodeC
        Cluster shorthand id for node: 1
        uuid for node: 40752a9c-b687-11df-94d4-4eb040029002
        State of node:  UP
        Smoothed rtt to node: 7
        Mean Deviation in network rtt to node: 3
        Number of zones this node is a member in: 0
        Number of clusters node is a member in: 1
        CLUSTER NAME        TYPE   SHID    UUID
        SIRCOL_nodeA local          89320f66-ba9c-11df-8d0c-001125bfc896

        Number of points_of_contact for node: 1
        Point-of-contact interface & contact state
         en0   UP

------------------------------

        Node name: nodeB
        Cluster shorthand id for node: 2
        uuid for node: 4001694a-b687-11df-80ec-000255d3926b
        State of node:  UP
        Smoothed rtt to node: 7
        Mean Deviation in network rtt to node: 3
        Number of zones this node is a member in: 0
        Number of clusters node is a member in: 1
        CLUSTER NAME        TYPE   SHID    UUID
        SIRCOL_nodeA local          89320f66-ba9c-11df-8d0c-001125bfc896

        Number of points_of_contact for node: 1
        Point-of-contact interface & contact state
         en0   UP

------------------------------

        Node name: nodeA
        Cluster shorthand id for node: 3
        uuid for node: 21f1756c-b687-11df-80c9-001125bfc896
        State of node:  UP   NODE_LOCAL
        Smoothed rtt to node: 0
        Mean Deviation in network rtt to node: 0
        Number of zones this node is a member in: 0
```

```
Number of clusters node is a member in: 1
CLUSTER NAME         TYPE   SHID    UUID
SIRCOL_nodeA local           89320f66-ba9c-11df-8d0c-001125bfc896

Number of points_of_contact for node: 0
Point-of-contact interface & contact state
 n/a
```

```
# lspv
hdisk0          00cad74fd6d58ac1                         rootvg          active
hdisk1          00cad74fa9d3b7e1                         None
hdisk2          00cad74fa9d3b8de                         None
caa_private0    00cad74f3964114a                         caavg_private   active
cldisk4         00cad74f3963c575                         None
cldisk3         00cad74f3963c671                         None
cldisk2         00cad74f3963c6fa                         None
cldisk1         00cad74f3963c775                         None
hdisk8          00cad74f3963c7f7                         None
hdisk9          00cad74f3963c873                         None
hdisk10         00cad74f3963ca13                         None
hdisk11         00cad74f3963caa9                         None
hdisk12         00cad74f3963cb29                         None
hdisk13         00cad74f3963cba4                         None
```

> **Note:** The **-n** option of the **mkcluster** command can be used to specify an explicit name for the cluster. For a detailed explanation of these options, refer to the manpages.

As soon as the cluster has been created, other active nodes of the cluster configure and join into the cluster. The **lscluster** command is executed from one of the other nodes in the cluster to verify the cluster configuration. Example 4-13 shows the output from the **lscluster** command from the node nodeB. Observe the State of node field in the **lscluster** command. It gives you the latest status of the node as seen from the node where the **lscluster** command is executed. A value of NODE_LOCAL indicates that this node is the local node where the **lscluster** command is executed.

Example 4-13   Verifying the cluster from another node

```
# hostname
nodeB
# lscluster -m
Calling node query for all nodes
Node query number of nodes examined: 3
```

```
              Node name: nodeC
              Cluster shorthand id for node: 1
              uuid for node: 40752a9c-b687-11df-94d4-4eb040029002
              State of node:  UP
              Smoothed rtt to node: 7
              Mean Deviation in network rtt to node: 3
              Number of zones this node is a member in: 0
              Number of clusters node is a member in: 1
             CLUSTER NAME        TYPE  SHID   UUID
              SIRCOL_nodeA local          89320f66-ba9c-11df-8d0c-001125bfc896

              Number of points_of_contact for node: 1
              Point-of-contact interface & contact state
               en0  UP
        ------------------------------

              Node name: nodeB
              Cluster shorthand id for node: 2
              uuid for node: 4001694a-b687-11df-80ec-000255d3926b
              State of node:  UP  NODE_LOCAL
              Smoothed rtt to node: 0
              Mean Deviation in network rtt to node: 0
              Number of zones this node is a member in: 0
              Number of clusters node is a member in: 1
             CLUSTER NAME        TYPE  SHID   UUID
              SIRCOL_nodeA local          89320f66-ba9c-11df-8d0c-001125bfc896

              Number of points_of_contact for node: 0
              Point-of-contact interface & contact state
               n/a

        ------------------------------

              Node name: nodeA
              Cluster shorthand id for node: 3
              uuid for node: 21f1756c-b687-11df-80c9-001125bfc896
              State of node:  UP
              Smoothed rtt to node: 7
              Mean Deviation in network rtt to node: 3
              Number of zones this node is a member in: 0
              Number of clusters node is a member in: 1
             CLUSTER NAME        TYPE  SHID   UUID
              SIRCOL_nodeA local          89320f66-ba9c-11df-8d0c-001125bfc896

              Number of points_of_contact for node: 1
```

```
          Point-of-contact interface & contact state
            en0   UP
```

Example 4-14 shows the output from the **lscluster -c** command to display basic cluster configuration information. The cluster name is `SIRCOL_nodeA`. An explicit cluster name can also be specified using the **-n** option to the **mkcluster** command. A unique `Cluster uuid` is generated for the cluster. Each of the nodes is assigned a unique `Cluster id`.

Example 4-14   Displaying a basic cluster configuration

```
# lscluster -c
Cluster query for cluster SIRCOL_nodeA returns:
Cluster uuid: 89320f66-ba9c-11df-8d0c-001125bfc896
Number of nodes in cluster = 3
        Cluster id for node nodeC is 1
        Primary IP address for node nodeC is 9.126.85.51
        Cluster id for node nodeB is 2
        Primary IP address for node nodeB is 9.126.85.14
        Cluster id for node nodeA is 3
        Primary IP address for node nodeA is 9.126.85.13
Number of disks in cluster = 4
         for disk cldisk4 UUID = 60050763-05ff-c02b-0000-000000001114
cluster_major = 0 cluster_minor = 4
         for disk cldisk3 UUID = 60050763-05ff-c02b-0000-000000001115
cluster_major = 0 cluster_minor = 3
         for disk cldisk2 UUID = 60050763-05ff-c02b-0000-000000001116
cluster_major = 0 cluster_minor = 2
         for disk cldisk1 UUID = 60050763-05ff-c02b-0000-000000001117
cluster_major = 0 cluster_minor = 1
Multicast address for cluster is 227.1.1.211
```

Example 4-15 shows the output from the **lscluster -d** command displaying cluster storage interfaces. Observe the `state` field for each of the disks, which gives the latest state of the corresponding disk. The `type` field is used to represent whether it is a cluster disk or a repository disk.

Example 4-15   Displaying cluster storage interfaces

```
# lscluster -d
Storage Interface Query

Cluster Name:  SIRCOL_nodeA
Cluster uuid:  89320f66-ba9c-11df-8d0c-001125bfc896
Number of nodes reporting = 3
```

```
Number of nodes expected = 3
Node nodeA
Node uuid = 21f1756c-b687-11df-80c9-001125bfc896
Number of disk discovered = 5
        cldisk4
           state : UP
           uDid  : 200B75CWLN1111407210790003IBMfcp
           uUid  : 60050763-05ff-c02b-0000-000000001114
           type  : CLUSDISK
        cldisk3
           state : UP
           uDid  : 200B75CWLN1111507210790003IBMfcp
uUid  : 60050763-05ff-c02b-0000-000000001115
           type  : CLUSDISK
        cldisk2
           state : UP
           uDid  : 200B75CWLN1111607210790003IBMfcp
           uUid  : 60050763-05ff-c02b-0000-000000001116
           type  : CLUSDISK
        cldisk1
           state : UP
           uDid  : 200B75CWLN1111707210790003IBMfcp
           uUid  : 60050763-05ff-c02b-0000-000000001117
           type  : CLUSDISK
caa_private0
           state : UP
           uDid  :
           uUid  : 60050763-05ff-c02b-0000-000000001113
           type  : REPDISK
Node
Node uuid = 00000000-0000-0000-0000-000000000000
Number of disk discovered = 0
Node
Node uuid = 00000000-0000-0000-0000-000000000000
Number of disk discovered = 0
```

Example 4-16 shows the output from the `lscluster -s` command displaying cluster network statistics on the local node. The command gives statistical information regarding the type and amount of packets received or sent to other nodes within the cluster.

Example 4-16   Displaying cluster network statistics

```
# lscluster -s
Cluster Statistics:
```

```
                Cluster Network Statistics:

                pkts seen:71843                    pkts passed:39429
                IP pkts:33775                      UDP pkts:32414
                gossip pkts sent:16558             gossip pkts recv:24296
                cluster address pkts:0             CP pkts:32414
                bad transmits:0                    bad posts:0
                short pkts:0                        multicast pkts:32414
                cluster wide errors:0              bad pkts:0
                dup pkts:1                         pkt fragments:0
                fragments queued:0                 fragments freed:0
                requests dropped:0                 pkts routed:0
                pkts pulled:0                      no memory:0
                rxmit requests recv:7              requests found:4
                requests missed:0                  ooo pkts:0
                requests reset sent:0              reset  recv:0
                requests lnk reset send :0         reset  lnk recv:0
                rxmit requests sent:3
                alive pkts sent:3                  alive pkts recv:0
                ahafs pkts sent:4                  ahafs pkts recv:1
                nodedown pkts sent:8               nodedown pkts recv:3
                socket pkts sent:294               socket pkts recv:75
                cwide pkts sent:33                 cwide pkts recv:45
                socket pkts no space:0             pkts recv notforhere:1918
                stale pkts recv:0                  other cluster pkts:0
                storage pkts sent:1                storage pkts recv:1
                out-of-range pkts recv:0
```

Example 4-17 shows the output from the `lscluster -i` command listing cluster configuration interfaces on the local node. The Interface state gives the latest state of the corresponding interfaces of each of the nodes.

Example 4-17   Displaying cluster configuration interfaces

```
# lscluster -i
Network/Storage Interface Query

Cluster Name:  SIRCOL_nodeA
Cluster uuid:  89320f66-ba9c-11df-8d0c-001125bfc896
Number of nodes reporting = 3
Number of nodes expected = 3
Node nodeA
Node uuid = 21f1756c-b687-11df-80c9-001125bfc896
Number of interfaces discovered = 2
```

```
        Interface number 1 en0
                ifnet type = 6 ndd type = 7
                Mac address length = 6
                Mac address =  0.11.25.bf.c8.96
                Smoothed rrt across interface = 7
                Mean Deviation in network rrt across interface = 3
                Probe interval for interface = 100 ms
                ifnet flags for interface = 0x5e080863
                ndd flags for interface = 0x63081b
                Interface state  UP
                Number of regular addresses configured on interface = 1
                IPV4 ADDRESS: 9.126.85.13  broadcast 9.126.85.255  netmask
255.255.255.0
                Number of cluster multicast addresses configured on interface = 1
                IPV4 MULTICAST ADDRESS: 227.1.1.211  broadcast 0.0.0.0  netmask
0.0.0.0
        Interface number 2 dpcom
                ifnet type = 0 ndd type = 305
                Mac address length = 0
                Mac address =  0.0.0.0.0.0
                Smoothed rrt across interface = 750
                Mean Deviation in network rrt across interface = 1500
                Probe interval for interface = 22500 ms
                ifnet flags for interface = 0x0
                ndd flags for interface = 0x9
                Interface state  UP  RESTRICTED  AIX_CONTROLLED
Node nodeC
Node uuid = 40752a9c-b687-11df-94d4-4eb040029002
Number of interfaces discovered = 2
        Interface number 1 en0
                ifnet type = 6 ndd type = 7
                Mac address length = 6
                Mac address =  4e.b0.40.2.90.2
                Smoothed rrt across interface = 8
                Mean Deviation in network rrt across interface = 3
                Probe interval for interface = 110 ms
                ifnet flags for interface = 0x1e080863
                ndd flags for interface = 0x21081b
                Interface state  UP
                Number of regular addresses configured on interface = 1
                IPV4 ADDRESS: 9.126.85.51  broadcast 9.126.85.255  netmask
255.255.255.0
                Number of cluster multicast addresses configured on interface = 1
                IPV4 MULTICAST ADDRESS: 227.1.1.211  broadcast 0.0.0.0  netmask
0.0.0.0
```

```
        Interface number 2 dpcom
                ifnet type = 0 ndd type = 305
                Mac address length = 0
                Mac address =  0.0.0.0.0.0
                Smoothed rrt across interface = 750
Mean Deviation in network rrt across interface = 1500
                Probe interval for interface = 22500 ms
                ifnet flags for interface = 0x0
                ndd flags for interface = 0x9
                Interface state  UP  RESTRICTED  AIX_CONTROLLED
Node nodeB
Node uuid = 4001694a-b687-11df-80ec-000255d3926b
Number of interfaces discovered = 2
        Interface number 1 en0
                ifnet type = 6 ndd type = 7
                Mac address length = 6
                Mac address =  0.2.55.d3.92.6b
                Smoothed rrt across interface = 7
                Mean Deviation in network rrt across interface = 3
                Probe interval for interface = 100 ms
                ifnet flags for interface = 0x5e080863
                ndd flags for interface = 0x63081b
                Interface state  UP
                Number of regular addresses configured on interface = 1
                IPV4 ADDRESS: 9.126.85.14  broadcast 9.126.85.255  netmask
255.255.255.0
                Number of cluster multicast addresses configured on interface = 1
                IPV4 MULTICAST ADDRESS: 227.1.1.211  broadcast 0.0.0.0  netmask
0.0.0.0
        Interface number 2 dpcom
                ifnet type = 0 ndd type = 305
                Mac address length = 0
                Mac address =  0.0.0.0.0.0
                Smoothed rrt across interface = 750
                Mean Deviation in network rrt across interface = 1500
                Probe interval for interface = 22500 ms
                ifnet flags for interface = 0x0
                ndd flags for interface = 0x9
                Interface state  UP  RESTRICTED  AIX_CONTROLLED
```

Cluster configuration can be modified using the `chcluster` command.
Example 4-18 on page 140 shows the use of the `chcluster` command. Here, the
node nodeC is removed from the cluster. The `lscluster` command is used to
verify the removal of nodeC from the cluster.

Example 4-18   Deletion of a node from a cluster

```
# chcluster -n SIRCOL_nodeA -m -nodeC
# lscluster -m
Calling node query for all nodes
Node query number of nodes examined: 2

        Node name: nodeB
        Cluster shorthand id for node: 2
        uuid for node: 4001694a-b687-11df-80ec-000255d3926b
        State of node:  UP
        Smoothed rtt to node: 7
        Mean Deviation in network rtt to node: 3
        Number of zones this node is a member in: 0
        Number of clusters node is a member in: 1
       CLUSTER NAME        TYPE  SHID   UUID
SIRCOL_nodeA local         c5ea0c7a-bab9-11df-a75b-001125bfc896

        Number of points_of_contact for node: 1
        Point-of-contact interface & contact state
         en0  UP


------------------------------

        Node name: nodeA
        Cluster shorthand id for node: 3
        uuid for node: 21f1756c-b687-11df-80c9-001125bfc896
        State of node:  UP  NODE_LOCAL
        Smoothed rtt to node: 0
        Mean Deviation in network rtt to node: 0
        Number of zones this node is a member in: 0
        Number of clusters node is a member in: 1
       CLUSTER NAME        TYPE  SHID   UUID
        SIRCOL_nodeA local         c5ea0c7a-bab9-11df-a75b-001125bfc896

        Number of points_of_contact for node: 0
        Point-of-contact interface & contact state
         n/a
```

Similarly, Example 4-19 shows the removal of cluster disk cldisk3 from the cluster.

Example 4-19   Deletion of a cluster disk from a cluster

```
# lspv |grep cldisk3
cldisk3        00cad74f3963c6fa                        None
```

```
# chcluster -n SIRCOL_nodeA -d -cldisk3
chcluster: Removed cluster shared disks are automatically renamed to names such
        as hdisk10, [hdisk11, ...] on all cluster nodes.  However, this cannot
        take place while a disk is busy or on a node which is down or not
        reachable.  If any disks cannot be renamed now, you must manually
        rename them by removing them from the ODM database and then running
        the cfgmgr command to recreate them with default names.  For example:
        rmdev -l cldisk1 -d
        rmdev -l cldisk2 -d
        cfgmgr
# lspv |grep cldisk3
# lspv |grep cldisk*
cldisk1         00cad74f3963c575                        None
cldisk4         00cad74f3963c671                        None
cldisk2         00cad74f3963c775                        None
```

Example 4-20 is another example showing addition of a new disk, hdisk9, as a cluster disk. Notice that hdisk9 is renamed to cldisk5 after executing the **chcluster** command.

Example 4-20   Addition of a disk to the cluster

```
# chcluster -n SIRCOL_nodeA -d +hdisk9
chcluster: Cluster shared disks are automatically renamed to names such as
        cldisk1, [cldisk2, ...] on all cluster nodes.  However, this cannot
        take place while a disk is busy or on a node which is down or not
        reachable.  If any disks cannot be renamed now, they will be renamed
        later by the clconfd daemon, when the node is available and the disks
        are not busy.
# lspv |grep cldisk*
cldisk1         00cad74f3963c575                        None
cldisk4         00cad74f3963c671                        None
cldisk2         00cad74f3963c775                        None
cldisk5         00cad74f3963c873                        None
```

Example 4-21 shows use of the **rmcluster** command to remove the cluster configuration. Note the output from the **lscluster** and **lspv** commands after the removal of the cluster.

Example 4-21   Removal of a cluster

```
# rmcluster -n SIRCOL_nodeA
rmcluster: Removed cluster shared disks are automatically renamed to names such
        as hdisk10, [hdisk11, ...] on all cluster nodes.  However, this cannot
        take place while a disk is busy or on a node which is down or not
```

```
        reachable.  If any disks cannot be renamed now, you must manually
        rename them by removing them from the ODM database and then running
        the cfgmgr command to recreate them with default names.  For example:
        rmdev -l cldisk1 -d
        rmdev -l cldisk2 -d
        cfgmgr
# lscluster -m
Cluster services are not active.
# lspv |grep cldisk*
```

The **clcmd** command is used to distribute commands to one or more nodes that are part of the cluster. In Example 4-22, the **clcmd** command executes the **date** command on each of the nodes in the cluster and returns with their outputs.

Example 4-22   Usage of the **clcmd** *command*

```
# clcmd -n SIRCOL_nodeA date
-------------------------------
NODE nodeA
-------------------------------
Wed Sep  8 02:13:58 PAKDT 2010
-------------------------------
NODE nodeB
-------------------------------
Wed Sep  8 02:14:00 PAKDT 2010
-------------------------------
NODE nodeC
-------------------------------
Wed Sep  8 02:13:58 PAKDT 2010
```

4.4.2 Cluster system architecture flow

When a cluster is created, various subsystems get configured. The following list describes the process of the clustering subsystem:

► The cluster is created using the **mkcluster** command.

► The cluster configuration is written to the raw section of one of the shared disks designated as the cluster repository disk.

► Primary and secondary database nodes are selected from the list of candidate nodes in the mkcluster command. For the primary or secondary database failure, an alternate node is started to perform the role of a new primary or new secondary database node.

- ► Special volume groups and logical volumes are created on the cluster repository disk.

- ► Cluster file systems are created on the special volume group.

- ► The cluster repository database is created on both primary and secondary nodes.

- ► The cluster repository database is started.

- ► Cluster services are made available to other functions in the operating system, such as Reliable Scalable Cluster Technology (RSCT) and PowerHA SystemMirror.

- ► Storage framework register lists are created on the cluster repository disk.

- ► A global device namespace is created and interaction with LVM starts for handling associated volume group events.

- ► A clusterwide multicast address is established.

- ► The node discovers all of the available communication interfaces.

- ► The cluster interface monitoring starts.

- ► The cluster interacts with AIX Event Infrastructure for clusterwide event distribution.

- ► The cluster exports cluster messaging and cluster socket services to other functions in the operating system, such as Reliable Scalable Cluster Technology (RSCT) and PowerHA SystemMirror.

4.4.3 Cluster event management

The AIX event infrastructure is used for event management on AIX. For a detailed description, refer to 5.12, "AIX Event Infrastructure" on page 202. Table 4-4 lists the cluster-specific events.

Table 4-4   Cluster events

Cluster events	Description
nodeList	Monitors changes in cluster membership.
clDiskList	Monitors changes in cluster disk membership.
nodeContact	Monitors the last contact status of the node in a cluster.
nodeState	Monitors the state of the node in the cluster.
nodeAddress	Alias is added or removed from a network interface.
networkAdapterState	Monitors the network interface of a node in the cluster.

Cluster events	Description
clDiskState	Monitors clustered disks.
repDiskState	Monitors the repository disk.
diskState	Monitors the local disk changes.
vgState	Verifies the status of the volume group on a disk.

These events are propagated to all nodes in the cluster so that event monitoring applications are notified as and when an event happens on any node in the cluster.

4.4.4 Cluster socket programming

Cluster communications can operate over the traditional networking interfaces (IP-based) or using the storage interfaces (Fibre Channel or SAS).

When cluster communications is configured over both transports, the redundancy and high availability of the underlying cluster node software and hardware configuration can be maximized by using all the paths for communications. In case of network interface failures, you can use the storage framework (Fibre Channel or SAS) to maintain communication between the cluster nodes. Cluster communications is achieved by exploiting the multicast capabilities of the networking and storage subsystems.

Example 4-23 on page 144 provides a sample cluster family socket server and client program that is used to communicate between two nodes in the cluster.

The server will define port 29 to be used for communications.

Node A is identified as node 3 (the shorthand ID for node from the `lscluster -m` output).

Node B is identified as node 2 (the shorthand ID for node from the `lscluster -m` output).

Example 4-23   Cluster messaging example

```
# hostname
nodeA
# ./server 29

Server Waiting for client on port 29
From cluster node: 2
Message: this is test message
```

```
# hostname
nodeB
# ./client 3 29 "this is test message"

->cat server.c
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <stdio.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <stdlib.h>
#include <sys/cluster.h>
#include <cluster/cluster_var.h>

int
main(int argc, char *argv[])
{
        int             sock;
        unsigned long int addr_len, bytes_read;
        char            recv_data[1024];
        struct sockaddr_clust server_addr, client_addr;
        int             port;

        if (argc != 2) {
fprintf(stdout, "Usage: ./server <port num>\n");
                exit(1);
        }
        if ((sock = socket(AF_CLUST, SOCK_DGRAM, 0)) == -1) {
                perror("Socket");
                exit(1);
        }
        port = atoi(argv[1]);
        bzero((char *) &server_addr, sizeof(server_addr));
        server_addr.sclust_family = AF_CLUST;
        server_addr.sclust_port = port;
        server_addr.sclust_cluster_id = WWID_LOCAL_CLUSTER;
        server_addr.sclust_addr = get_clusterid();
        if (bind(sock, (struct sockaddr *) & server_addr, sizeof(struct
sockaddr_clust)) == -1) {
                perror("Bind");
                exit(1);
```

```
        }
        addr_len = sizeof(struct sockaddr_clust);
        fprintf(stdout, "\nServer Waiting for client on port %d",
port);
        fflush(stdout);
        while (1) {
                bytes_read = recvfrom(sock, recv_data, 1024, 0, (struct
sockaddr *) & client_addr, &addr_len);
recv_data[bytes_read] = '\0';
                fprintf(stdout, "\nFrom cluster node: %d",
client_addr.sclust_addr);
                fprintf(stdout, "\nMessage: %s\n", recv_data);
        }

        return 0;
}

->cat client.c
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <netdb.h>
#include <stdio.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <stdlib.h>
#include <sys/cluster.h>
#include <cluster/cluster_var.h>

#define MAX_MSG 100
int
main(int argc, char *argv[])
{
        int             sock, rc, i;
        struct sockaddr_clust sclust;
        struct hostent *host;
        char            send_data[1024];

        if (argc <= 3) {
                fprintf(stdout, "Usage: ./client <cluster ID of server>
<port> < MSG >");
                exit(1);
```

```
        }
        if ((sock = socket(AF_CLUST, SOCK_DGRAM, 0)) == -1) {
                perror("socket");
                exit(1);
        }
        bzero((char *) &sclust.sclust_len, sizeof(struct
sockaddr_clust));
        sclust.sclust_addr = atoi(argv[1]);
        sclust.sclust_len = sizeof(struct sockaddr_clust);
        sclust.sclust_family = AF_CLUST;
sclust.sclust_cluster_id = WWID_LOCAL_CLUSTER;
        sclust.sclust_port = atoi(argv[2]);


        rc = bind(sock, (struct sockaddr *) & sclust, sizeof(sclust));
        if (rc < 0) {
                printf("%s: cannot bind port\n", argv[0]);
                exit(1);
        }
        /* send data */
        for (i = 3; i < argc; i++) {
                rc = sendto(sock, argv[i], strlen(argv[i]) + 1, 0,
(struct sockaddr *) & sclust, sizeof(sclust));
                if (rc < 0) {
                        printf("%s: cannot send data %d \n", argv[0], i
- 1);
                        close(sock);
                        exit(1);
                }
        }
        return 1;
}
```

4.4.5 Cluster storage communication configuration

In order to be able to communicate using storage communication interfaces for
high availability and redundancy of communication paths between nodes in the
cluster, the storage adapters need to be configured.

The following information only applies to Fibre Channel adapters. No setup is necessary for SAS adapters. The following Fibre Channel adapters are supported:

► 4 GB Single-Port Fibre Channel PCI-X 2.0 DDR Adapter (FC 1905; CCIN 1910)

► 4 GB Single-Port Fibre Channel PCI-X 2.0 DDR Adapter (FC 5758; CCIN 280D)

► 4 GB Single-Port Fibre Channel PCI-X Adapter (FC 5773; CCIN 5773)

► 4 GB Dual-Port Fibre Channel PCI-X Adapter (FC 5774; CCIN 5774)

► 4 Gb Dual-Port Fibre Channel PCI-X 2.0 DDR Adapter (FC 1910; CCIN 1910)

► 4 Gb Dual-Port Fibre Channel PCI-X 2.0 DDR Adapter (FC 5759; CCIN 5759)

► 8 Gb PCI Express Dual Port Fibre Channel Adapter (FC 5735; CCIN 577D)

► 8 Gb PCI Express Dual Port Fibre Channel Adapter 1Xe Blade (FC 2B3A; CCIN 2607)

► 3 Gb Dual-Port SAS Adapter PCI-X DDR External (FC 5900 and 5912; CCIN 572A)

> **Note:** For the most current list of supported Fibre Channel adapters, contact your IBM representative.

To configure the Fibre Channel adapters that will be used for cluster storage communications, complete the following steps (the output shown in Example 4-24 on page 149):

> **Note:** In the following steps the X in fcsX represents the number of your Fibre Channel adapters, for example, fcs1, fsc2, or fcs3.

1. Run the following command:

```
rmdev -Rl fcsX
```

> **Note:** If you booted from the Fibre Channel adapter, you do not need to complete this step.

2. Run the following command:

```
chdev -l fcsX -a tme=yes
```

> **Note:** If you booted from the Fibre Channel adapter, add the -P flag.

3. Run the following command:

```
chdev -l fscsiX -a dyntrk=yes -a fc_err_recov=fast_fail
```

4. Run the **cfgmgr** command.

> **Note:** If you booted from the Fibre Channel adapter and used the -P flag, you must reboot.

5. Verify the configuration changes by running the following command:

```
lsdev -C | grep sfwcom
```

After you create the cluster, you can list the cluster interfaces and view the storage interfaces by running the following command:

```
lscluster -i
```

Example 4-24   Cluster storage communication configuration

```
# rmdev -Rl fcs0
fcnet0 Defined
hdisk1 Defined
hdisk2 Defined
hdisk3 Defined
hdisk4 Defined
hdisk5 Defined
hdisk6 Defined
hdisk7 Defined
hdisk8 Defined
hdisk9 Defined
hdisk10 Defined
sfwcomm0 Defined
fscsi0 Defined
fcs0 Defined
# chdev -l fcs0 -a tme=yes
fcs0 changed
# chdev -l fscsi0 -a dyntrk=yes -a fc_err_recov=fast_fail
fscsi0 changed
# cfgmgr >cfg.out 2>&1
# lsdev -C | grep sfwcom
sfwcomm0   Defined   00-00-02-FF Fiber Channel Storage Framework Comm
sfwcomm1   Available 00-01-02-FF Fiber Channel Storage Framework Comm
```

Note: Configure cluster storage interfaces. The above set of commands used to configure the storage interfaces should be executed on all the nodes that are part of the cluster. The cluster should be created after configuring the interfaces on all the nodes.

4.5 SCTP component trace and RTEC adoption

The AIX enterprise Reliability Availability Serviceability (eRAS) infrastructure defines a component definition framework. This framework supports three distinct domains:

► Runtime Error Checking (RTEC)

► Component Trace (CT)

► Component Dump (CD)

The Stream Control Transmission Protocol (SCTP) implementation in AIX V7.1 and AIX V6.1 TL 6100-06 significantly enhances the adoption of the RAS component framework for the RTEC and CT domains. To that extent the following two new trace hooks are defined:

► Event ID 6590 (0x659) with event label SCTP

► Event ID 65a0 (0x65a) with event label SCTP_ERR

The previously existing base component sctp of the CT and RTEC component tree is complemented by an additional subcomponent, sctp_err.

The integration into the component trace framework enables both the memory trace mode (private memory trace) and the user trace mode (system trace) for the base component and its new subcomponent.

The CT SCTP component hierarchy of a given AIX configuration and the current settings for the memory trace mode and the user trace mode can be listed by the `ctctrl` command, which also allows you to modify the component trace-related configuration parameters. The `ctctrl` command output in Example 4-25 on page 151 shows the default component trace configuration for the SCTP component just after the SCTP kernel extension has been loaded with the `sctpctrl load` command. As you can see, the memory trace is set to normal (level=3) and the system trace level to detailed (level=7) for the SCTP

component, and for the sctp.sctp_err subcomponent the memory trace level is set to minimal (level=1) and the system trace level to detailed (level=7).

Example 4-25   ctctrl command output

```
7501lp01:/> ctctrl -c sctp -q -r
-----------------------------+-------+-------+-------+--------------
                             | Have  |Mem Trc|Sys Trc| Buffer size
       Component name        | alias | /level| /level| /Allocated
-----------------------------+-------+-------+-------+--------------
sctp                         |  NO   | ON/3  | ON/7  |     40960/YES
     .sctp_err               |  NO   | ON/1  | ON/7  |     10240/YES
```

The RTEC SCTP component hierarchy of a given AIX configuration and the current settings for error checking level, disposition for low-severity errors, and disposition for medium-severity errors can be listed by the **errctrl** command. The **errctrl** command also allows you to modify the runtime error checking related configuration parameters. The **errctrl** command output in Example 4-26 shows that the default error checking level for all SCTP components is normal (level=3), and that low-severity errors (LowSevDis=64), and medium-severity errors (MedSevDisp=64) are logged (collect service data and continue).

Example 4-26   errctrl command output

```
7501lp01:/> errctrl -c sctp -q -r
-------------------------------------+-------+-------+-------+--------
                             | Have  |ErrChk |LowSev |MedSev
       Component name        | alias | /level| Disp  | Disp
-------------------------------------+-------+-------+-------+--------
sctp                         |  NO   | ON/3  |  64   |  64
     .sctp_err               |  NO   | ON/3  |  64   |  64
```

The AIX SCTP implementation is intentionally not integrated with the AIX enterprise RAS Component Dump domain. A component dump temporarily suspends execution and the Stream Control Transmission Protocol may react negatively by false time-outs and failovers being perceived by peer nodes. However, a functionality similar to the component dump is delivered through the **dump** parameter of the **sctpctrl** command. This command has also been enhanced in AIX V7.1 and AIX V6.1 TL 6100-06 to provide improved formatting of the command output.

4.6 Cluster aware perfstat library interfaces

IBM PowerHA is a high availability solution for AIX that provides automated failure detection, diagnosis, application recovery, and node reintegration.

It consists of two components:

High availability   The process of ensuring an application is available for use through the use of duplicated and/or shared resources.

Cluster multiprocessing

    Multiple applications running on the same nodes with shared or concurrent access to the data.

This high availability solution demands two very important capabilities from the performance monitoring perspective:

► The ability to collect and analyze the performance data of the entire cluster at the aggregate level (from any node in the cluster).

► The ability to collect and analyze the performance data of an individual node in the cluster (from any node in the cluster).

The `perfstat` application programming interface (API) is a collection of C programming language subroutines that execute in the user space and use the perfstat kernel extension to extract various AIX performance metrics.

Beginning with AIX V7.1 and AIX 6.1 TL06, the existing `perfstat` library is enhanced to support performance data collection and analysis for a single node or multiple nodes in a cluster. The enhanced perfstat library provides APIs to obtain performance metrics related to processor, memory, I/O, and others to provide performance statistics about a node in a cluster.

The `perfstat` library is also updated with a new interface called perfstat_cluster_total (similar to the perfstat_partion_total interface) that provides cluster level aggregate data.

A separate interface called perfstat_node_list is also added to retrieve the list of nodes available in the cluster.

New APIs (NODE interfaces) are available that return usage metrics related to a set of components or individual components specific to a remote node in a cluster.

> **Note:** The perfstat_config (PERFSTAT_ENABLE |
> PERFSTAT_CLUSTER_STATS, NULL) must be used to enable the remote
> node statistics collection (available only in a cluster environment).
>
> Once node-related performance data is collected, perfstat_config
> (PERFSTAT_DISABLE | PERFSTAT_CLUSTER_STATS, NULL) must be used
> to disable collection of node or cluster statistics.

Here are the node interfaces that are added:

```
perfstat_<subsystem>_node Subroutines
```

Purpose

Retrieve a remote node's performance statistics of subsystem type. The
subroutines are as follows:

- ► perfstat_cpu_total_node
- ► perfstat_disk_node
- ► perfstat_disk_total_node
- ► perfstat_diskadapter_node
- ► perfstat_diskpath_node
- ► perfstat_logicalvolume_node
- ► perfstat_memory_page_node
- ► perfstat_memory_total_node
- ► perfstat_netbuffer_node
- ► perfstat_netinterface_node
- ► perfstat_netinterface_total_node
- ► perfstat_pagingspace_node
- ► perfstat_partition_total_node
- ► perfstat_protocol_node
- ► perfstat_tape_node
- ► perfstat_tape_total_node
- ► perfstat_volumegroup_node

Library

Perfstat library (libperfstat.a)

Syntax

#include <libperfstat.h>

```
int perfstat_cpu_node ( name, userbuff, sizeof_userbuff, desired_number
)
perfstat_id_node_t *name;
perfstat_cpu_t *userbuff;
int sizeof_userbuff;
int desired_number;

int perfstat_cpu_total_node ( name, userbuff, sizeof_userbuff,
desired_number )
perfstat_id_node_t *name;
perfstat_cpu_total_t *userbuff;
int sizeof_userbuff;
int desired_number;

int perfstat_disk_node ( name, userbuff, sizeof_userbuff,
desired_number )
perfstat_id_node_t *name;
perfstat_disk_t *userbuff;
int sizeof_userbuff;
int desired_number;

int perfstat_disk_total_node ( name, userbuff, sizeof_userbuff,
desired_number )
perfstat_id_node_t *name;
perfstat_disk_total_t *userbuff;
int sizeof_userbuff;
int desired_number;

int perfstat_diskadapter_node ( name, userbuff, sizeof_userbuff,
desired_number )
perfstat_id_node_t *name;
perfstat_diskadapter_t *userbuff;
int sizeof_userbuff;
int desired_number;

int perfstat_diskpath_node ( name, userbuff, sizeof_userbuff,
desired_number )
perfstat_id_node_t *name;
perfstat_diskpath_t *userbuff;
int sizeof_userbuff;
int desired_number;
```

```
int perfstat_logicalvolume_node ( name, userbuff, sizeof_userbuff,
desired_number )
perfstat_id_node_t *name;
perfstat_logicalvolume_t *userbuff;
int sizeof_userbuff;
int desired_number;

int perfstat_memory_page_node ( name, psize, userbuff, sizeof_userbuff,
desired_number )
perfstat_id_node_t *name;
perfstat_psize_t *psize;
perfstat_memory_page_t *userbuff;
int sizeof_userbuff;
int desired_number;

int perfstat_memory_total_node ( name, userbuff, sizeof_userbuff,
desired_number )
perfstat_id_node_t *name;
perfstat_memory_total_t *userbuff;
int sizeof_userbuff;
int desired_number;

int perfstat_netbuffer_node ( name, userbuff, sizeof_userbuff,
desired_number )
perfstat_id_node_t *name;
perfstat_netbuffer_t *userbuff;
int sizeof_userbuff;
int desired_number;

int perfstat_netinterface_node ( name, userbuff, sizeof_userbuff,
desired_number )
perfstat_id_node_t *name;
perfstat_netinterface_t *userbuff;
int sizeof_userbuff;
int desired_number;

int perfstat_netinterface_total_node ( name, userbuff, sizeof_userbuff,
desired_number )
perfstat_id_node_t *name;
perfstat_netinterface_total_t *userbuff;
int sizeof_userbuff;
int desired_number;

int perfstat_pagingspace_node ( name, userbuff, sizeof_userbuff,
desired_number )
```

```
perfstat_id_node_t *name;
perfstat_pagingspace_t *userbuff;
int sizeof_userbuff;
int desired_number;

int perfstat_partition_total_node ( name, userbuff, sizeof_userbuff,
desired_number )
perfstat_id_node_t *name;
perfstat_partition_total_t *userbuff;
int sizeof_userbuff;
int desired_number;

int perfstat_protocol_node ( name, userbuff, sizeof_userbuff,
desired_number )
perfstat_id_node_t *name;
perfstat_protocol_t *userbuff;
int sizeof_userbuff;
int desired_number;

int perfstat_tape_node ( name, userbuff, sizeof_userbuff,
desired_number )
perfstat_id_node_t *name;
perfstat_tape_t *userbuff;
int sizeof_userbuff;
int desired_number;

int perfstat_tape_total_node ( name, userbuff, sizeof_userbuff,
desired_number )
perfstat_id_node_t *name;
perfstat_tape_total_t *userbuff;
int sizeof_userbuff;
int desired_number;

int perfstat_volumegroup_node ( name, userbuff, sizeof_userbuff,
desired_number )
perfstat_id_node_t *name;
perfstat_volumegroup_t *userbuff;
int sizeof_userbuff;
int desired_number;
```

Description

These subroutines return a remote node's performance statistics in their
corresponding perfstat_<subsystem>_t structure.

To get statistics from any particular node in a cluster, the Node ID or the Node name must be specified in the name parameter. The userbuff parameter must be allocated and the desired_number parameter must be set.

> **Note:** The remote node should belong to one of the clusters in which the current node (the perfstat API call is run) is participating.

Refer to the AIX Version 7.1 technical references for additional details at:

    http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=/c
    om.ibm.aix.doc/doc/base/technicalreferences.htm

5

System management

In this chapter, the following system management enhancements are discussed:

5.1 Processor interrupt disablement

AIX 6.1 TL6 and 7.1 provide a facility to quiesce external I/O interrupts on a given set of logical processors. This helps reduce interrupt jitter that affects application performance.

When co-scheduling Parallel Operation Environment (POE) jobs or even in a non-POE commercial environment, administrators can control the process scheduling and interrupt handling across all the processors. It is desirable to quiesce interrupts on the SMT threads that are running POE jobs to avoid interrupting the jobs. By doing so, your applications can run on a given set of processors without being affected by any external interrupts.

The CPU interrupt disablement function can be configured using the following kernel service, system call, or user command:

Kernel service   `k_cpuextintr_ctl()`

System call   `cpuextintr_ctl()`

Command line   `cpuextintr_ctl`

This functionality is supported on POWER5, POWER6, and POWER7 and any future System p hardware. It is supported in both dedicated or shared processor logical partitions.

Example 5-1 shows the output of the **cpuextintr_ctl** command used to disable external interrupts on CPU 1 on a system that has two processors.

> **Note:** The changes are reflected dynamically without requiring a reboot of the system. Also, the changes are *not* persistent across reboots of the system.

Example 5-1   Disabling interrupts

```
# bindprocessor -q
The available processors are:  0 1

# cpuextintr_ctl  -Q
The CPUs that have external interrupt enabled:

   0    1


The CPUs that have external interrupt disabled:

# cpuextintr_ctl -C 1 -i disable
```

```
# cpuextintr_ctl  -Q
The CPUs that have external interrupt enabled:

   0

The CPUs that have external interrupt disabled:

   1
```

Note:

- ► When the request for external interrupt is `disable`, only external interrupt priority more favored than INTCLASS0 may be delivered to the controlled processor, which includes the Environmental and POwer Warning (EPOW) interrupt and IPI (MPC) interrupt.

- ► Even though the external interrupt has been disabled using these interfaces, the processor can still be interrupted by an IPI/MPC or EPOW interrupt or any priority registered at INTMAX.

- ► CPU interrupt disablement works with CPU DR add/removal (dynamic LPAR operation). Once a CPU DR is added to the partition, the external interrupt will be enabled by default.

- ► CPU interrupt disablement works with CPU Intelligent folding.

- ► It guarantees that at least one of the processors on the system will have external interrupt enabled.

5.2 Distributed System Management

Starting with AIX 6.1 TL3 a new package is shipped with the base media called Distributed System Management (DSM). In AIX 7.1 this new DSM package replaces the Cluster Systems Management package (CSM), which is no longer available on AIX 7.1. Commands such as **dcp** and **dsh** are not available on AIX 7.1 without installing the DSM package, which is not installed by default but is on the base installation media. The DSM package is in the filesets `dsm.core` and `dsm.dsh`.

Selecting the DSM package from the install media installs the components shown in Table 5-1 on page 162.

Table 5-1   DSM components

dsm.core	Distributed Systems Management Core
dsm.dsh	Distributed Systems Management Dsh

The new DSM programs found in the fileset dsm.core are:

dpasswd   Creates an encrypted password file for an access point.

dkeyexch   Exchanges default ssh keys with an access point.

dgetmacs   Collects MAC address information from a machine.

dconsole   Opens a remote console to a machine.

5.2.1 The dpasswd command

The `dpasswd` command is used to create the DSM password file. The password
file contains a user ID and associated encrypted password. The command
generates an AES key and writes it to the file `/etc/ibm/sysmgt/dsm/config/.key`,
if this file does not already exist. The default key size will be 128 bits. The
command can generate a 256-bit key if the unrestricted Java security files have
been installed. For more information on these policy files, refer to the Java
Security Guide, which ships with the Java Runtime package.

The key is used to encrypt the password before writing it to the file. It is also used
by the other DSM programs to decrypt the password. If the key file is removed, it
will be recreated with a new key the next time the command is run.

> **Note:** If the key file is removed, password files created with that key cannot be
> decrypted. If the key file is removed, the existing password files must be
> recreated with the `dpasswd` command.

If the password file name is given with no path information, it is written to the
`/etc/ibm/sysmgt/dsm/config` directory.

Run the `dpasswd -h` command to view the command syntax.

Example 5-2 shows the use of the `dpasswd` command to create the password file.

Example 5-2   Creating a password file

```
# dpasswd -f my_password_file -U userID
Password file is /etc/ibm/sysmgt/dsm/config/my_password_file
Password:
```

```
Re-enter password:
Password file created.
#
```

5.2.2 The dkeyexch command

The **dkeyexch** command is used to exchange ssh keys between the NIM master and a client access point. The command requires the encrypted password file created by the **dpasswd** command. The information in the password file is used to exchange ssh keys with the access points specified in the command.

This command exchanges the default ssh RSA and DSA keys located in the user's $HOME/.ssh directory as generated by the **ssh-keygen** command. It will exchange keys stored in user-named files.

> **Note:** openssl (openss.base) and openssh (openssh.base) must be installed.

The command can also be used to remove keys from an access point.

> **Note:** BladeCenter® currently limits the number of installed keys to 12. When adding keys to a BladeCenter, the command verifies that there are keyslots available for the new keys. If only one slot is available, only the DSA key is exchanged.

Run the **dkeyexch -h** command to see the command syntax.

Example 5-3 shows a key exchange between the NIM master and an HMC. The password file must exit and contain a valid user ID and encrypted password for this HMC. Following the key exchange, an ssh session can be established with no password prompt.

Example 5-3 Key exchange between NIM and an HMC

```
# dkeyexch -f /etc/ibm/sysmgt/dsm/config/hmc_password_file -I hmc -H
hmc01.clusters.com
# ssh hscroot@hmc01.clusters.com
Last login: Tue Dec 23 11:57:55 2008 from nim_master.clusters.com
hscroot@hmc01:~>
```

5.2.3 The dgetmacs command

The **dgetmacs** command is used to query a client node for its network adapter information. This information is gathered even if the node has no operating system on it or is powered off. This command requires AIX 7.1 SP 1.

> **Note:** When the open_firmware mode is used (either when specified on the command line or if the dsh and arp modes failed), the command causes the client node to be rebooted into a special state so that the adapter information can be obtained. This only applies to client nodes managed by an HMC or an IVM. Ensure that the client node is not in use before running this command.

Run the **dgetmacs -h** command to view the command syntax.

Example 5-4 shows an example that uses the dsh method.

Example 5-4   Using the dsh method

```
# dgetmacs -m dsh -n canif3_obj -C NIM
Using an adapter type of "ent".
Attempting to use dsh method to collect MAC addresses.
#
Node::adapter_type::interface_name::MAC_address::location::media_speed::adapter_
duplex::UNUSED::install_gateway::ping_status::machine_type::netaddr::subnet_mask
canif3_obj::ent_v::en0::001A644486E1::::1000::full::::172.16.143.250::::secondar
y::172.16.128.91::255.255.240.0
canif3_obj::ent_v::en1::1E9E18F60404::::::::::172.16.143.250::::secondary::::
```

Additional examples can be found in the tech note document located at /opt/ibm/sysmgt/dsm/doc/dsm_tech_note.pdf.

5.2.4 The dconsole command

The **dconsole** command is used to open a remote console to a client node. The command operates in both the DEFAULT and NIM contexts. It supports read-only consoles and console logging.

The command is supported by a daemon program that is launched when the **dconsole** command is invoked for the first time. This console daemon remains running as long as there are consoles open. When the last console is closed, the console daemon terminates. By default, the daemon listens on TCP port number 9085, which has been reserved from IANA for this purpose. The port number may be changed by overriding the dconsole_Port_Number entry in the DSM properties file.

Run the `dconsole -h` command to view the syntax.

The dconsole display modes

The command operates in one of two display modes, default and text.

In the *default* display mode, the command uses an `xterm` window to display the console. In this mode, consoles to multiple client nodes can be opened from a single command. A separate window is opened for each node. The default display mode requires that the DISPLAY environment variable be set before the `dconsole` command is invoked. The variable must be set to the address of an X-Windows server where the console will be displayed. By default, the console window is launched using the fixed font.

The remote console session is closed by closing the xterm window. Issuing Ctrl-x within the console window also closes the console session.

The *text* display mode is invoked by adding the -t flag to the command line. In this mode, no X-Windows server is required. The console is opened in the current session. The text mode console session is closed with Ctrl-x.

DSM offers the ability to log remote console sessions on client nodes. By default, logging is disabled. It may be enabled on a console-by-console basis by issuing the `dconsole` command with the `-l` (lower-case L) flag. It may also be enabled globally by overriding the n entry in the DSM properties file (setting the value to Yes enables global console logging). When logging is enabled, any data that is visible on the console will also be written to a log file. The console must be open for logging to take place.

> **Note:** Changing the global setting has no impact on console sessions that were already open when the setting was changed. Any open consoles must be closed and reopened for the updated setting to take effect.

By default, console log files are written to the `/var/ibm/sysmgt/dsm/log/console` directory. Both the log directory and console log subdirectory may be changed by overriding the dconsole_Log_File_Subdirectory entry in the DSM properties file.

By default, these files will rotate. The maximum file size is about 256 kilobytes, and up to four files are kept for each console log. The number of rotations may be changed by overriding the Log_File_Rotation entry in the DSM properties file. Setting the value to zero disables log rotation and allows the logs to grow in size up to the available file system space.

Example 5-5 on page 166 shows the `dconsole` command starting in text mode with logging enabled.

Example 5-5   Starting dconsole in text mode with logging

```
# dconsole -n  9.47.93.94 -t -l
Starting console daemon
[read-write session]

 Open in progress

 Open Completed.



AIX Version 6
Copyright IBM Corporation, 1982, 2009.
Console login:
```

For Example 5-5, an entry was made in the node info file to define the target system and access point information. The node info file is found in the /etc/ibm/sysmgt/dsm directory.

Example 5-6 shows the format of the fnode info file used in Example 5-5.

Example 5-6   Contents of the node info file

```
# cat /etc/ibm/sysmgt/dsm/nodeinfo
9.47.93.94|hmc|9.47.91.240|TargetHWTypeModel=9117-570:TargetHWSerialNum
=1038FEA:TargetLPARID=11|/etc/ibm/sysmgt/dsm/config/hsc_password
#
```

Additional options and usages of the console command along with information about using DSM and NIM to install new clients can be found in the DSM tech note. This tech note document is located at /opt/ibm/sysmgt/dsm/doc/dsm_tech_note.pdf.

5.2.5 The dcp command

The **dcp** command works the same as it did in AIX 6.1. It copies files to or from multiple nodes. The node list is not the same as the DSM node info file.

Example 5-7 shows the use of the **dcp** command to copy the testdata.log file to a new file on the nodes listed in the node list file.

Example 5-7   Example use of the dcp command

```
# dcp /tmp/testdata.log /tmp/testdata_copy4.log
```

For Example 5-7 the location of the node list was specified in an environment variable, shown in Example 5-8.

Example 5-8   Checking dsh environment variables

```
# env | grep -i dsh
DSH_REMOTE_CMD=/usr/bin/ssh
DSH_NODE_LIST=/etc/ibm/sysmgt/dsm/nodelist
DSH_NODE_RSH=/usr/bin/ssh
#
```

The nodelist of the **dcp** command was a simple list of target IP addresses as seen in Example 5-9.

Example 5-9   Sample node list

```
# cat /etc/ibm/sysmgt/dsm/nodelist
9.47.93.94
9.47.93.60

#
```

5.2.6 The dsh command

The **dsh** command works the same as it did in AIX 6.1. It runs commands concurrently on multiple nodes. The node list is not the same as the DSM node info file.

Example 5-10 shows the use of the **dsh** command to run the **date** command on the nodes listed in the node list file.

Example 5-10   Example using the dsh command

```
# dsh -a date
e19-93-60.ent.beaverton.ibm.com: Tue Sep 14 16:07:51 PDT 2010
e19-93-94.ent.beaverton.ibm.com: Tue Sep 14 16:08:02 PDT 2010
```

For Example 5-10 the location of the node list was specified in an environment variable, shown in Example 5-11.

Example 5-11   Setting up the environment variables

```
# env | grep -i dsh
DSH_REMOTE_CMD=/usr/bin/ssh
DSH_NODE_LIST=/etc/ibm/sysmgt/dsm/nodelist
DSH_NODE_RSH=/usr/bin/ssh
```

```
#
```

The node list for the **dsh** command was a simple list of target IP addresses, as seen in Example 5-12.

Example 5-12   Sample node list

```
# cat /etc/ibm/sysmgt/dsm/nodelist
9.47.93.94
9.47.93.60

#
```

5.2.7 Using DSM and NIM

The AIX Network Installation Manager (NIM) has been enhanced to work with the Distributed System Management (DSM) commands. This integration enables the automatic installation of new AIX systems that are either currently powered on or off.

The example that follows demonstrates this functionality. We follow a sequence of steps to use NIM to install the AIX operating system onto a new NIM client LPAR, using DSM. We will be installing AIX onto an HMC-controlled LPAR.

The steps are as follows:

1. Collect information for console access points, such as the IP address or hostname of the HMC, and the HMC administrator user ID and password.

2. Collect information relating to the new NIM client LPAR, such as the hostname, IP address, hardware type-model, serial number of the system, and LPAR ID.

3. Run the **dpasswd** command to generate the password file for the HMC access point. Run the **dkeyexch** command to exchange the NIM master SSH key with the HMC.

4. Define a new NIM HMC and management object for the HMC and the CEC, specifying the password file that was created in the previous step.

5. Obtain the MAC address for the network adapter of the new LPAR using the **dgetmacs** command.

6. Define a new NIM machine object for the new NIM client LPAR.

7. Perform a NIM bos_inst operation on the NIM client to install the AIX operating system.

8. From the NIM master, open a console window with the **dconsole** command and monitor the NIM installation.

9. The final step is to verify that AIX has installed successfully.

In this scenario, the HMC IP address is 10.52.52.98 and its hostname is hmc5. The system type, model, and serial number information is collected from the HMC, as shown in Example 5-13

Example 5-13   Collecting the system type, model and serial number from HMC

```
hscroot@hmc5:~> lssyscfg -r sys -F name,type_model,serial_num
750_2-8233-E8B-061AB2P,8233-E8B,061AB2P
```

The LPAR ID is also collected from the HMC, as shown in Example 5-14.

Example 5-14   Collecting the LPAR ID information from the HMC

```
hscroot@hmc5:~> lssyscfg -r lpar -m 750_2-8233-E8B-061AB2P -F
name,lpar_id
750_2_LP04,5
750_2_LP03,4
750_2_LP02,3
750_2_LP01,2
750_2_VIO_1,1
orion,6
```

The HMC admin user ID is hscroot and the password is abc123. The **dpasswd** command is run to store the user password. The NIM master SSH key is generated and exchanged with the HMC with the **dkeyexch** command. We confirmed that we could ssh to the HMC without being prompted for a password, as shown in Example 5-15.

Example 5-15   Configuring ssh access to the HMC from the NIM master

```
# dpasswd -f my_password_file -U hscroot
# dkeyexch -f /etc/ibm/sysmgt/dsm/config/my_password_file -I hmc -H 10.52.52.98
# ssh hscroot@hmc5
Last login: Fri Sep 10 09:46:03 2010 from 10.52.52.101
hscroot@hmc5:~>
```

The new NIM client LPAR IP address is 10.52.52.200 and the hostname is orion. The LPAR ID is 6. This information and the hardware type-model and serial number of the target Power System were recorded in the /etc/ibm/sysmgt/dsm/nodeinfo file, as shown in Example 5-16.

Example 5-16   Entry in the nodeinfo file for the new host, Power System and HMC

```
# cat /etc/ibm/sysmgt/dsm/nodeinfo
7502lp01|hmc|10.52.52.98|TargetHWTypeModel=8233-E8B:TargetHWSerialNum=061AB2P:TargetLPARID=2|/e
tc/ibm/sysmgt/dsm/config/my_password_file
7502lp02|hmc|10.52.52.98|TargetHWTypeModel=8233-E8B:TargetHWSerialNum=061AB2P:TargetLPARID=3|/e
tc/ibm/sysmgt/dsm/config/my_password_file
7502lp03|hmc|10.52.52.98|TargetHWTypeModel=8233-E8B:TargetHWSerialNum=061AB2P:TargetLPARID=4|/e
tc/ibm/sysmgt/dsm/config/my_password_file
7502lp04|hmc|10.52.52.98|TargetHWTypeModel=8233-E8B:TargetHWSerialNum=061AB2P:TargetLPARID=5|/e
tc/ibm/sysmgt/dsm/config/my_password_file
orion|hmc|10.52.52.98|TargetHWTypeModel=8233-E8B:TargetHWSerialNum=061AB2P:TargetLPARID=6|/etc/
ibm/sysmgt/dsm/config/my_password_file
```

We defined a new NIM HMC and management object for the HMC and the CEC, as shown in Example 5-17.

Example 5-17   Defining the HMC and CEC NIM objects

```
# nim -o define -t hmc -a if1="find_net hmc5 0" -a
passwd_file="/etc/ibm/sysmgt/dsm/config/my_password_file" hmc5

# lsnim -Fl hmc5
hmc5:
   id          = 1284061389
   class       = management
   type        = hmc
   if1         = net_10_52_52 hmc5 0
   Cstate      = ready for a NIM operation
   prev_state  =
   Mstate      = currently running
   manages     = cec0
   passwd_file = /etc/ibm/sysmgt/dsm/config/my_password_file

# nim -o define -t cec -a hw_type=8233 -a hw_model=E8B -a hw_serial=061AB2P -a
mgmt_source=hmc5 cec0

# lsnim -Fl cec0
cec0:
   id          = 1284061538
   class       = management
   type        = cec
   Cstate      = ready for a NIM operation
   prev_state =
   manages     = 7502lp02
   manages     = orion
```

```
hmc        = hmc5
serial     = 8233-E8B*061AB2P
```

We obtained the MAC address for the virtual network adapter in the new LPAR. The **dgetmacs** command is used to obtain this information. This command will power on the LPAR in *Open Firmware* mode to query the network adapter MAC address information. The LPAR in this example was in a *Not Activated* state prior to running the **dgetmacs** command.

> **Note:** If the MAC address of the network adapter is unknown, you can define the client with a MAC address of 0 and use the **dgetmacs** command to retrieve it. Once the MAC address is identified, the NIM standalone object if1 attribute can be changed with the **nim –o change** command.
>
> This MAC address is required for the bos_inst NIM operation for clients that cannot be reached.
>
> If the LPAR is in a *Running* state, it is be powered down and restarted in *Open Firmware* mode. Once the MAC address has been acquired, the LPAR is powered down again.

Example 5-18   Obtaining the MAC address for the LPARs virtual network adapter

```
# dgetmacs -n orion
Using an adapter type of "ent".
Could not dsh to node orion.
Attempting to use openfirmware method to collect MAC addresses.
Acquiring adapter information from Open Firmware for node orion.

#
Node::adapter_type::interface_name::MAC_address::location::media_speed::adapter_duplex::UNUSED:
:install_gateway::ping_status::machine_type::netaddr::subnet_mask

orion::ent_v::::6E8DD877B814::U8233.E8B.061AB2P-V6-C20-T1::auto::auto::::::n/a::secondary::::
```

We defined a new NIM machine object for the new LPAR, as shown in Example 5-19.

Example 5-19   Defining a new NIM machine object with HMC, LPAR, and CEC options

```
# nim -o define -t standalone -a if1="net_10_52_52 orion 6E8DD877B814" -a
net_settings1="auto auto" -a mgmt_profile1="hmc5 6 cec0" orion
# lsnim -Fl orion
orion:
   id              = 1284075145
```

```
   class          = machines
   type           = standalone
   connect        = nimsh
   platform       = chrp
   netboot_kernel = 64
   if1            = net_10_52_52 orion 6E8DD877B814
   net_settings1  = auto auto
   cable_type1    = N/A
   mgmt_profile1  = hmc5 6 cec0
   Cstate         = ready for a NIM operation
   prev_state     = not running
   Mstate         = currently running
   cpuid          = 00F61AB24C00
   Cstate_result  = success
   default_profile =
type=hmc,ip=10.52.52.98,passwd_file=/etc/ibm/sysmgt/dsm/config/my_password_file:type=
lpar,identity=6:type=cec,serial=8233-E8B*061AB2P:
```

The LPAR was in a *Not Activated* state. We enabled the NIM client for BOS installation as shown in Example 5-20. This initiated a network boot of the LPAR.

Example 5-20   Displaying LPAR state and enabling NIM bos_inst on the NIM client

```
# ssh hscroot@hmc5
Last login: Fri Sep 10 15:57:24 2010 from 10.52.52.101
hscroot@hmc5:~> vtmenu

-------------------------------------------------------------
  Partitions On Managed System:  750_2-8233-E8B-061AB2P
  OS/400 Partitions not listed
 -------------------------------------------------------------
   1)    750_2_LP01                              Running
   2)    750_2_LP02                              Running
   3)    750_2_LP03                              Running
   4)    750_2_LP04                              Running
   5)    750_2_VIO_1                             Running
   6)    orion                                   Not Activated


 Enter Number of Running Partition (q to quit): q
hscroot@hmc5:~> exit
exit
Connection to hmc5 closed.
#
```

```
# nim -o bos_inst -a bosinst_data=noprompt_bosinst -a source=rte -a
installp_flags=agX -a accept_licenses=yes -a spot=spotaix7100 -a lpp_source=aix7100
orion
dnetboot Status: Invoking /opt/ibm/sysmgt/dsm/dsmbin/lpar_netboot orion
dnetboot Status: Was successful network booting node orion.
#
```

We opened a console window (in read-only mode with session logging enabled)
using the **dconsole** command to monitor the NIM installation, as shown in
Example 5-21. Only partial output is shown because the actual log is extremely
verbose.

Example 5-21   Monitoring the NIM installation with the dconsole command

```
# dconsole -n orion -t -l -r
Starting console daemon
[read only session, user input discarded]

 Open in progress

 Open Completed.
IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM
IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM
IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM
IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM

          1 = SMS Menu                          5 = Default Boot List
          8 = Open Firmware Prompt              6 = Stored Boot List


     Memory       Keyboard      Network      SCSI       Speaker
..........
10.52.52.200:    24  bytes from 10.52.52.101:  icmp_seq=9  ttl=? time=11  ms

10.52.52.200:    24  bytes from 10.52.52.101:  icmp_seq=10  ttl=? time=11  ms

PING SUCCESS.
 ok
0 > 0 to my-self  ok
0 > boot
/vdevice/l-lan@30000014:speed=auto,duplex=auto,bootp,10.52.52.101,,10.52.52.200,10.52
.52.101

.........
```

```
TFTP BOOT ---------------------------------------------------
Server IP.....................10.52.52.101
Client IP.....................10.52.52.200
Gateway IP....................10.52.52.101
Subnet Mask...................255.255.254.0
( 1  ) Filename................./tftpboot/orion
TFTP Retries.................5
Block Size...................512
PACKET COUNT = 12900


..........
                        Installing Base Operating System




        Please wait...




        Approximate     Elapsed time
      % tasks complete   (in minutes)
```

On the NIM master, the NIM client status during the installation was monitored, as shown in Example 5-22.

Example 5-22   Monitoring the NIM client installation status from the NIM master

```
# lsnim -Fl orion
orion:
   id             = 1284075145
   class          = machines
   type           = standalone
   connect        = nimsh
   platform       = chrp
   netboot_kernel = 64
   if1            = net_10_52_52 orion 6E8DD877B814
   net_settings1  = auto auto
   cable_type1    = N/A
   mgmt_profile1  = hmc5 6 cec0
   Cstate         = Base Operating System installation is being performed
   prev_state     = BOS installation has been enabled
   Mstate         = in the process of booting
   info           = BOS install 21% complete : Installing additional software.
   boot           = boot
```

```
    bosinst_data    = noprompt_bosinst
    lpp_source      = aix7100
    nim_script      = nim_script
    spot            = spotaix7100
    exported        = /export/lppsrc/aix7100
    exported        = /export/nim/scripts/orion.script
    exported        = /export/spot/spotaix7100/usr
    exported        = /tmp/cg/bosinst.data
    cpuid           = 00F61AB24C00
    control         = master
    Cstate_result   = success
    boot_info       = -aip=10.52.52.200 -aha=6E8DD877B814 -agw=10.52.52.101
-asm=255.255.254.0 -asa=10.52.52.101
    trans1          = 86 1 6 master /usr/sbin/nim -o deallocate -F -asubclass=all
-aasync=yes orion
    trans2          = 86 14 1 master /usr/lpp/bos.sysmgt/nim/methods/m_destroy_res
-aforce=yes -aignore_state=yes -a ignore_lock=yes orion
    default_profile =
type=hmc,ip=10.52.52.98,passwd_file=/etc/ibm/sysmgt/dsm/config/my_password_file:type=
lpar,identity=6:type=cec,serial=8233-E8B*061AB2P:
```

On the NIM master, the DSM network boot output is logged to
/var/ibm/sysmgt/dsm/log/dnetboot.*name*.log.*XXX,* where *name* is the node
name and *XXX* is the log sequence number; see Example 5-23.

Example 5-23   DSM network boot log file output

```
# cd /var/ibm/sysmgt/dsm/log/
# cat dnetboot.orion.log.253
Output log for dnetboot is being written to
/var/ibm/sysmgt/dsm/log//dnetboot.orion.log.253.
----------------------------------------------------------------------
dnetboot: Logging started Fri Sep 10 16:03:21 EDT 2010.
----------------------------------------------------------------------

dnetboot Status: Invoking /opt/ibm/sysmgt/dsm/dsmbin/lpar_netboot orion
16:3:21  dnetboot Status: Invoking /opt/ibm/sysmgt/dsm/dsmbin/lpar_netboot orion
----------------------------------------------------------------------
dnetboot: Logging stopped Fri Sep 10 16:03:21 EDT 2010.
----------------------------------------------------------------------
dnetboot Status: Invoking /opt/ibm/sysmgt/dsm/dsmbin/lpar_netboot -i -t ent -D  -S
10.52.52.101 -G 10.52.52.101 -C 10.52.52.200 -m 6E8DD877B814 -s auto -d auto -F
/etc/ibm/sysmgt/dsm/config/my_password_file -j hmc -J 10.52.52.98 6 061AB2P 8233-E8B
# Connected
# Checking for OF prompt.
```

```
# Timeout waiting for OF prompt; rebooting.
# Checking for power off.
# Client IP address is 10.52.52.200.
# Server IP address is 10.52.52.101.
# Gateway IP address is 10.52.52.101.
# Getting adapter location codes.
# /vdevice/l-lan@30000014 ping successful.
# Network booting install adapter.
# bootp sent over network.
# Network boot proceeding, lpar_netboot is exiting.
# Finished.
16:4:41  dnetboot Status: Was successful network booting node orion.
```

The **dconsole** command can log session output if called with the **-l** flag. The log file is located on the NIM master, in the /var/ibm/sysmgt/dsm/log/console/name.X file, where name is the node name and X is the log sequence number. This file can be monitored using the **tail** command, as shown in Example 5-24.

Example 5-24   DSM dconsole log file

```
# cd /var/ibm/sysmgt/dsm/log/console/
# ls -ltr
total 1664
-rw-r--r--    1 root      system          1464 Sep 09 15:39 7502lp01.0
-rw-r--r--    1 root      system         34118 Sep 09 19:27 7502lp02.0
-rw-r--r--    1 root      system        262553 Sep 10 12:12 orion.3
-rw-r--r--    1 root      system        262202 Sep 10 12:46 orion.2
-rw-r--r--    1 root      system             0 Sep 10 16:01 orion.0.lck
-rw-r--r--    1 root      system        262282 Sep 10 16:09 orion.1
-rw-r--r--    1 root      system         11708 Sep 10 16:09 orion.0
# tail -f orion.0


 5724X1301
   Copyright IBM Corp. 1991, 2010.
   Copyright AT&T 1984, 1985, 1986, 1987, 1988, 1989.
   Copyright Unix System Labs, Inc., a subsidiary of Novell, Inc. 1993.
 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 xlC.rte >>. . . .


Filesets processed:  344 of 591
System Installation Time: 5 minutes        Tasks Complete: 61%
```

```
installp: APPLYING software for:
        xlC.msg.en_US.rte 11.1.0.1


. . . . . << Copyright notice for xlC.msg.en_US >> . . . . . . .
 Licensed Materials - Property of IBM

 5724X1301
   Copyright IBM Corp. 1991, 2010.
   Copyright AT&T 1984, 1985, 1986, 1987, 1988, 1989.
   Copyright Unix System Labs, Inc., a subsidiary of Novell, Inc. 1993.
 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 xlC.msg.en_US >>. . . .
```

Another log file, related to network boot, is also available on the NIM master. It contains extended network book information and is located in /tmp/lpar_netboot.*PID*.exec.log, where *PID* is the process ID of the lpar_netboot process, as shown in Example 5-25. Only partial output is shown because the actual log file is extremely verbose.

Example 5-25   lpar_netboot log file

```
# cd /tmp
# cat lpar_netboot.16056500.exec.log
lpar_netboot Status: node = 6, profile = 061AB2P, manage = 8233-E8B
lpar_netboot Status: process id is 16056500
lpar_netboot Status: -t List only ent adapters
lpar_netboot Status: -D (discovery) flag detected
lpar_netboot Status: -i (force immediate shutdown) flag detected
lpar_netboot Status: using adapter speed of auto
lpar_netboot Status: using adapter duplex of auto
lpar_netboot Status: using server IP address of 10.52.52.101
lpar_netboot Status: using client IP address of 10.52.52.200
lpar_netboot Status: using gateway IP address of 10.52.52.101
lpar_netboot Status: using macaddress of 6E8DD877B814
lpar_netboot Status: ck_args start
lpar_netboot Status: node 6
lpar_netboot Status: managed system 8233-E8B
lpar_netboot Status: username
lpar_netboot Status: password_file /etc/ibm/sysmgt/dsm/config/my_password_file
lpar_netboot Status: password
lpar_netboot Status: hmc-controlled node detected
lpar_netboot Status: node type is hmc
lpar_netboot Status: open port
```

```
lpar_netboot Status: open S1 port
lpar_netboot Status: console command is /opt/ibm/sysmgt/dsm/bin//dconsole -c -f -t -n
....
lpar_netboot Status: power reported as off, checking power state
lpar_netboot Status: power state is 6 Not Activated
lpar_netboot Status: power off complete
lpar_netboot Status: power on the node to Open Firmware
lpar_netboot Status: wait for power on
lpar_netboot Status: power on complete
lpar_netboot Status: waiting for RS/6000 logo
lpar_netboot Status: at RS/6000 logo
lpar_netboot Status: Check for active console.
.......
lpar_netboot Status: ping_server start
lpar_netboot Status: full_path_name : /vdevice/l-lan@30000014
lpar_netboot Status: phandle       : 0000021cf420
lpar_netboot Status : get_adap_prop start
lpar_netboot Status: get_adap_prop start
lpar_netboot Status: get_adap_prop command is " supported-network-types" 0000021cf420
....
lpar_netboot Status: ping_server command is ping
/vdevice/l-lan@30000014:10.52.52.101,10.52.52.200,10.52.52.101
send_command start:ping /vdevice/l-lan@30000014:10.52.52.101,10.52.52.200,10.52.52.101
ping /vdevice/l-lan@30000014:10.52.52.101,10.52.52.200,10.52.52.101
ping /vdevice/l-lan@30000014:10.52.52.101,10.52.52.200,10.52.52.101
10.52.52.200:    24  bytes from 10.52.52.101:  icmp_seq=1  ttl=? time=10  ms

10.52.52.200:    24  bytes from 10.52.52.101:  icmp_seq=2  ttl=? time=10  ms

10.52.52.200:    24  bytes from 10.52.52.101:  icmp_seq=3  ttl=? time=10  ms

10.52.52.200:    24  bytes from 10.52.52.101:  icmp_seq=4  ttl=? time=11  ms
....
PING SUCCESS.
 ok
....

TFTPlpar_netboot Status: network boot initiated
/usr/bin/dspmsg -s 1 /usr/lib/nls/msg/en_US/IBMhsc.netboot.cat 55 '# bootp sent over network.
.....
FINAL PACKET COUNT = 34702 lUNT = 17700
FINAL FILE SIZE = 17766912  BYTES

Elapsed time since release of system processors: 15840 mins 39 secs

--------------------------------------------------------------------------------
                             Welcome to AIX.
                     boot image timestamp: 15:00 09/09
                 The current time and date: 20:04:40 09/10/2010
```

```
        processor count: 2;  memory size: 2048MB;  kernel size: 35060743
boot device:
/vdevice/l-lan@30000014:speed=auto,duplex=auto,bootp,10.52.52.101,,10.52.52.200,10.52.52.101
/usr/bin/dspmsg -s 1 /usr/lib/nls/msg/en_US/IBMhsc.netboot.cat 56 '# Finished.
```

Once the AIX installation is complete, a login prompt is displayed in the console window. We then logged into the LPAR and confirmed that AIX was installed as expected. We started a read-write console session with the `dconsole` command, as shown in Example 5-26.

Example 5-26   Verifying AIX installed successfully from a dconsole session

```
# dconsole -n orion -t -l
Starting console daemon
[read-write session]

 Open in progress

 Open Completed.

AIX Version 7
Copyright IBM Corporation, 1982, 2010.
Console login: root
*******************************************************************************
*                                                                             *
*                                                                             *
*  Welcome to AIX Version 7.1!                                                *
*                                                                             *
*                                                                             *
*  Please see the README file in /usr/lpp/bos for information pertinent to    *
*  this release of the AIX Operating System.                                  *
*                                                                             *
*                                                                             *
*******************************************************************************

# oslevel -s
7100-00-00-0000
```

5.3 AIX system configuration structure expansion

New hardware and operating system capabilities required enhancements of the system configuration structure defined on AIX in `/usr/include/sys/systemcfg.h`.

Therefore, a new kernel service called kgetsystemcfg() and a new library function called getsystemcfg() have been implemented.

This new facility should be used in place of the existing __system_configuration structure that is accessible through memory because this new facility will be used for new configuration information in the future that will not be accessible using the __system_configuration structure.

The new facility, however, gives access to all the data in __system_configuration plus new (future) configuration data.

5.3.1 The kgetsystemcfg kernel service

This kernel service manpage provides the following information (Example 5-27).

Example 5-27   kgetsystemcfg manpage header

```
Purpose
   Displays the system configuration information.
Syntax
   #include <systemcfg.h>
   uint64_t kgetsystemcfg ( int name)
Description
   Displays the system configuration information.
Parameters
   name
      Specifies the system variable setting to be returned. Valid
      values for the name parameter are defined in the systemcfg.h
      file.
Return value
   EINVAL
      The value of the name parameter is invalid.
```

5.3.2 The getsystemcfg subroutine

This libc subroutine manpage provides the information shown in Example 5-28.

Example 5-28   getsystemcfg libc subroutine manpage header

```
Purpose
      Displays the system configuration information.
Syntax
      #include <systemcfg.h>
      uint64_t getsystemcfg ( int name)
Parameters
```

name

> Specifies the system variable setting to be returned. Valid
> values for the name parameter are defined in the systemcfg.h
> file.

Return value

> **EINVAL**
>
> > The value of the name parameter is invalid.

5.4 AIX Runtime Expert

AIX 6.1 TL4 includes a tool called AIX Runtime Expert. It provides the ability to collect, apply and verify the runtime environment for one or more AIX instances. This can be a valuable tool if a system needs to be cloned or if a comparison is needed between the tunables of different AIX instances. With this tool you can create a configuration profile (in XML format) capturing several settings and customizations done to an AIX instance.

With this AIX configuration profile, the system administrator can apply it to new AIX servers or compare it to other configuration servers in order to track any change. From deploying a medium to a large server infrastructure or to maintain server farms in a timely fashion, AIX Runtime Expert is the preferred tool for an efficient system administration with its *one-button* approach to managing and configuring numerous AIX instances.

AIX 6.1 TL6 and AIX 7.1 extends the tool with two new capabilities:

► Consolidating the management of AIX configuration profiles into a single control template.
► Easing the creation of a configuration template that can be deployed across a network of AIX OS instances in a scale-out configuration.

Example 5-29 lists the AIX Runtime Expert filesets for AIX 7.1.

Example 5-29   AIX 7.1 AIX Runtime Expert filesets

```
# lslpp -l | grep -i artex
  artex.base.agent          7.1.0.0  COMMITTED  AIX Runtime Expert CAS agent
  artex.base.rte            7.1.0.0  COMMITTED  AIX Runtime Expert
  artex.base.samples        7.1.0.0  COMMITTED  AIX Runtime Expert sample
```

5.4.1 AIX Runtime Expert overview

AIX components and subsystems provide a diversity of control points to manage runtime behavior. These control points can be configuration files, and command line and environment variables. They are independent of each other and are managed separately. AIX Runtime Expert is a tool to help manage these control points.

AIX Runtime Expert uses an XML file called a profile to manage these control points. You can create one or multiple profile files depending on the desired results. You can create a unique profile to suit your needs. These profiles can be created, edited and used to tune a second AIX instance to match an existing AIX instance. The AIX Runtime Expert can also compare two profiles or compare a profile to a running system to see the differences.

You create these profiles using the AIX Runtime Expert tool along with two types of read-only files that are used to build the profiles. These two types of files are called *profile templates* and *catalogs*.

AIX Runtime Expert profile templates

AIX Runtime Expert profile templates are XML files that include a list of tunable parameters. Each XML profile template is used to control any changeable tunable of a system. For example, the `vmoProfile.xml` file is used for the **vmo** system tuning. The `iooProfile.xm` file is used for I/O system tuning.

There are many profile templates. They can be found in the `/etc/security/artex/samples` directory. They are read-only files. The templates are not meant to be edited. It is also possible to see a list of all available profile templates using the **artexlist** command, as shown in Example 5-30.

Example 5-30   AIX Runtime Expert profile template listing

```
# artexlist
/etc/security/artex/samples/acctctlProfile.xml
/etc/security/artex/samples/aixpertProfile.xml
/etc/security/artex/samples/all.xml
/etc/security/artex/samples/alogProfile.xml
/etc/security/artex/samples/authProfile.xml
...
/etc/security/artex/samples/sysdumpdevProfile.xml
/etc/security/artex/samples/trcctlProfile.xml
/etc/security/artex/samples/trustchkProfile.xml
/etc/security/artex/samples/tsdProfile.xml
```

```
/etc/security/artex/samples/viosdevattrProfile.xml
/etc/security/artex/samples/vmoProfile.xml
```

These profile templates do not have any parameter values. They are used as templates to extract the current system values and create a new profile you may edit.

As new configuration options become available, new templates can be added to expand the value of the AIX Runtime Expert capabilities.

AIX Runtime Expert catalog

The AIX Runtime Expert catalogs are read-only files located in the /etc/security/artex/catalogs directory. They define how to map configuration profile values to parameters that run commands and configuration actions. They also identify values that can be modified.

Each catalog contains parameters for one component. However, some catalogs can contain parameters for multiple closely related components. To list all the catalogs, use the `artexlist -c` command as shown in Example 5-31.

Example 5-31   AIX Runtime Expert catalog listing

```
# artexlist -c
/etc/security/artex/catalogs/acctctlParam.xml
/etc/security/artex/catalogs/aixpertParam.xml
/etc/security/artex/catalogs/alogParam.xml
/etc/security/artex/catalogs/authParam.xml
...
/etc/security/artex/catalogs/trcctlParam.xml
/etc/security/artex/catalogs/trustchkParam.xml
/etc/security/artex/catalogs/tsdParam.xml
/etc/security/artex/catalogs/viosdevattrParam.xml
/etc/security/artex/catalogs/vmoParam.xml
#
```

The names of the catalogs describe the components that are contained in the catalog. The example of a catalog named schedoParam.xml in Example 5-32 gives the command name **schedo** and the short description **schedo** parameters. It allows **schedo** command subparameter configuration.

In each file the <description>.xml element provides a description of the catalog.

Example 5-32   Catalog file schedoParam.xml

```
# head /etc/security/artex/catalogs/schedoParam.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<Catalog id="schedoParam" version="2.0">
<ShortDescription><NLSCatalog catalog="artexcat.cat" setNum="41" msgNum="1">schedo
parameters</NLSCatalog></ShortDescription>
        <Description><NLSCatalog catalog="artexcat.cat" setNum="41"
msgNum="2">Parameter definition for the schedo command</NLSCatalog></Description>
<CfgMethod id="schedo">
            <Get type="current">
                    <Command>/usr/sbin/schedo -a</Command>
                    <Filter>/usr/bin/grep -v '= n/a$'</Filter>
...
```

The profiles file may reference one or multiple catalogs. For example, the
schedoProfile.xml profile only references the schedoParam catalog. The all.xml
profile file references all catalogs since it wants to contain all the system
tunables. Beginnings of these two files are listed in Example 5-33.

Example 5-33   Profiles file referencing catalogs

```
# head /etc/security/artex/samples/schedoProfile.xml
<?xml version="1.0" encoding="UTF-8"?>
<Profile origin="reference" readOnly="true" version="2.0.0">
        <Catalog id="schedoParam" version="2.0">
                <Parameter name="affinity_lim"/>
                <Parameter name="big_tick_size"/>
                <Parameter name="ded_cpu_donate_thresh"/>
                <Parameter name="fixed_pri_global"/>
...

# head /etc/security/artex/sampels/all.xml
<?xml version="1.0" encoding="UTF-8"?>
<Profile origin="merge: acctctlProfile.xml, aixpertProfile.xml,
alogProfile.xml, authProfile.xml, authentProfile.xml,
chconsProfile.xml, chdevProfile.xml, chlicenseProfile.xml,
chservicesProfile.xml, chssysProfile.xml, chsubserverProfile.xml,
chuserProfile.xml, classProfile.xml, coreProfile.xml,
dumpctrlProfile.xml, envProfile.xml, errdemonProfile.xml,
ewlmProfile.xml, ffdcProfile.xml, filterProfile.xml,
gencopyProfile.xml, iooProfile.xml, krecoveryProfile.xml,
login.cfgProfile.xml, lvmoProfile.xml, mktcpipProfile.xml,
mkuser.defaultProfile.xml, namerslvProfile.xml, nfsProfile.xml,
nfsoProfile.xml, nisProfile.xml, noProfile.xml, probevueProfile.xml,
rasoProfile.xml, roleProfile.xml, ruserProfile.xml, schedoProfile.xml,
secattrProfile.xml, shconfProfile.xml, smtctlProfile.xml,
syscorepathProfile.xml, sysdumpdevProfile.xml, trcctlProfile.xml,
```

```
trustchkProfile.xml, tsdProfile.xml, vmoProfile.xml" version="2.0.0"
date="2010-08-20T01:11:26Z" readOnly="true">
 <Catalog id="acctctlParam" version="2.0">
  <Parameter name="turacct"/>
  <Parameter name="agarm"/>
  <Parameter name="agke"/>
  <Parameter name="agproc"/>
  <Parameter name="isystem"/>
  <Parameter name="iprocess"/>
  <Parameter name="email_addr"/>
....
```

As new tunable parameters become available, new catalogs can be created to expand the value of the AIX Runtime Expert capabilities.

AIX Runtime Expert commands

The current commands available in AIX Runtime Expert to manipulate profiles and use catalogs are:

artexget   Extract configuration and tuning parameter information from a running system or from a specified configuration profile.

artexset   Set values on a system from a profile to take effect immediately or after system restart.

artexdiff   Compare values between a running system and a profile, or compare between two profiles.

artexmerge   Combine the contents of two or more profiles into a single profile.

artexlist   List configuration profiles or catalogs that exist on a local system or on the LDAP server.

The **artexget** command output can be in the following formats:

► The txt variable specifies plain text format.

► The csv variable specifies comma-separated values format.

► The xml format specifies xml format. This is the default format.

The **artexset** command dynamically sets the specified tunables if none of them are restricted. It can also specify that it must be applied at each boot of the system. By default, this command also creates a rollback profile that allows you to undo a profile change if needed.

For detailed parameters, see the manpages or info center at:

http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=/c
om.ibm.aix.baseadmn/doc/baseadmndita/aix_ev.htm

Building an AIX Runtime Expert profile

The following steps create a profile on a system:

1. Create a profile from the running system based on the default profile and catalog using the **artexget** command. The result of that command is an XML file that can be modified with any XML editor or any text editor.

2. Profiles you created can be customized by changing the values of the parameters or by removing some of the parameters that are not required.

3. Verify that the profile changes have been saved correctly by comparing them against the current system settings using the **artexdiff** command. It displays the parameters that were modified. The <FirstValue> displays the value of the profile, and the <SecondValue> displays the value of the current system.

4. Use the **artexset** command to set a system with the parameters from the new profile. With this command you can specify when the new parameters are to take effect—immediately, at the next boot, or at each system restart.

> **Note:** When the **-t** option is specified, the **artexset** command tests the correctness of the profile. It checks whether the profile has the correct XML format. Also, it checks whether the parameters defined in the profile are valid and supported by AIX Runtime Expert.

The following sections cover two examples of the use of the AIX Runtime Expert commands.

5.4.2 Changing mkuser defaults example

In this example the desire is to change the following default parameters when creating users:

► The user home directory to be located in the /userhome directory.
► Set the shell to /usr/bin/ksh93.

Using AIX Runtime Expert, a new profile can be created with the desired changes. It is also possible to return to the default system (rollback) without knowing which system config file needs to be modified.

Listing of current environment settings

To get the default environment setting for the mkuser setting, the **artexget** command is used with the profile called `mkuser.defaultProfile.xml` as shown in Example 5-34.

Example 5-34   Default mkuser profile

```
# cd /etc/security/artex/samples
# artexget -r mkuser.defaultProfile.xml
<?xml version="1.0" encoding="UTF-8"?>
<Profile origin="get" version="2.0.1" date="2010-09-07T20:43:32Z">
 <Catalog id="mkuser.default.adminParam" version="2.0">
  <Parameter name="account_locked" value=""/>
...
  <Parameter name="home" value="/home/$USER"/>
...
   <Parameter name="shell" value="/usr/bin/ksh"/>
...
 </Catalog>
</Profile>
```

Note that the default home is `/home/$USER` and the default shell is `/usr/bin/ksh`. Creating the user `user1` with that default profile would result in an entry in `/etc/passwd`, as shown in Example 5-35.

Example 5-35   Default user creation

```
# grep user1 /etc/passwd
user1:*:204:1::/home/user1:/usr/bin/ksh
```

Modify current settings

The **artexget** command is used to create a new profile based on the system defaults, and then the new profile is edited with the desired changes. Example 5-36 shows these steps.

Example 5-36   Building a new profile based on the system defaults

```
# cd /etc/security/artex/samples
# artexget -r mkuser.defaultProfile.xml > /tmp/mkuser1.xml
vi /tmp/mkuser1.xml
```

> **Note:** For this particular example the `mkuser.defaultProfile.xml` file has two sets of parameters, one for the admin user and the other for an ordinary user. The home directory and shell changes were only made to the parameters for the ordinary user.

After updating the new profile with new values for the home directory and shell, the **artexdiff -c -r** command is used to check the changes. Example 5-37 shows the results of this command.

Example 5-37   XLM output of the new profile and running system differences

```
# artexdiff -c -r  /tmp/mkuser1.xml
<?xml version="1.0" encoding="UTF-8"?>
<DifferenceData>
 <Parameter name="shell" catalogName="mkuser.default.userParam"
result="value">
  <FirstValue>/usr/bin/ksh93</FirstValue>
  <SecondValue>/usr/bin/ksh</SecondValue>
 </Parameter>
 <Parameter name="home" catalogName="mkuser.default.userParam"
result="value">
  <FirstValue>/userhome/$USER</FirstValue>
  <SecondValue>/home/$USER</SecondValue>
 </Parameter>
</DifferenceData>
```

A summary listing is available with the **artexdif -c -r -f txt** command as shown in Example 5-38.

Example 5-38   Text output of the new profile and the running system differences

```
# artexdiff -c -r  -f txt /tmp/mkuser1.xml
/tmp/mkuser1.xml | System Values
mkuser.default.userParam:shell /usr/bin/ksh93 | /usr/bin/ksh
mkuser.default.userParam:home /userhome/$USER | /home/$USER
```

Apply the new profile and check the result

Use the **artexset** command with the new profile to change the system defaults as shown in Example 5-39.

Example 5-39   Applying the new profile

```
# artexset /tmp/mkuser1.xml
```

Now any user created will use the new defaults, as shown in Example 5-40.

Example 5-40   Creating a new user with the new defaults

```
# mkuser user3
# grep user3 /etc/passwd
user3:*:206:1::/userhome/user3:/usr/bin/ksh93
```

Note that the new user is now using the `/userhome` directory instead of the `/home` directory and is also using the ksh93 shell.

Profile rollback

In case there is a need to remove the new configuration from the system, the `artexset -u` command will restore parameter values to the value of the last applied profile. The `artexdiff` command can be used to verify the result.

5.4.3 Schedo and ioo profile merging example

In this example it is desired to configure the two tunables that are in different profiles. First is the affinity_lim tunable and the second is posix_aio_maxservers. These values are described in the `/etc/security/artex/samples` default profile directory in multiple profile files:

► `all.xml`

► `default.xml`

► `iooProfile.xml` for `posix_aio_maxservers`

► `schedoProfile.xml` for `affinity_lim`

It is possible to get the current values for `all.xml` or `default.xml` and remove all non-needed entries, but it is easier to create a new profile file using the profile templates `iioProfile.xml` and `schedoPorfile.xml` and then merging them. The steps are:

► Get the runtime values for the `ioo` command.

► Get the runtime values for the `schedo` command.

► Create a merge profile.

► Edit the profile to remove all <Parameter name= > entries not needed. But do not remove the catalog entries.

► Check the profile for correctness using the `artexset -t` command.

► Check the current system values with the `artexget -r -f txt` command.

- ► Check to see if actions would be required, such as a system restart, when these parameters are changed with the **artexset -p** command.
- ► Check the running system values with the new profile using the **artexdiff -r -c -f txt** command.

Example 5-41 shows the execution of these steps. In this example, affinity_lim is changed from 7 to 6 and posix_aio_maxservers is changed from 30 to 60 using the **vi** editor.

Example 5-41   Creating a new merged profile

```
# cd /etc/security/artex/samples
# artexget -r iooProfile.xml > /tmp/1.xml
# artexget -r schedoProfile.xml  > /tmp/2.xml
# artexmerge /tmp/1.xml /tmp/2.xml > /tmp/3.x>
# vi /tmp/3.xml

# cat /tmp/3.xml
<?xml version="1.0" encoding="UTF-8"?>
<Profile origin="merge: /tmp/1.xml, /tmp/2.xml" version="2.0.0"
date="2010-09-09T04:45:19Z">
 <Catalog id="iooParam" version="2.0">
  <Parameter name="posix_aio_maxservers" value="60"/>
 </Catalog>
 <Catalog id="schedoParam" version="2.0">
  <Parameter name="affinity_lim" value="6"/>
 </Catalog>
</Profile>

# artexset -t /tmp/3.xml
Profile correctness check successful.

# artexget  -r -f txt  /tmp/3.xml
Parameter name      Parameter value
-------------------  ------------------
##Begin: schedoParam
affinity_lim        7
posix_aio_maxservers 30
##End:  iooParam

# artexset -p /tmp/3.xml
#Parameter name:Parameter value:Profile apply type:Catalog apply type:Additional
Action
affinity_lim:6:now_perm:now_perm:
posix_aio_maxservers:60:now_perm:now_perm:
```

```
# artexdiff -r -c -f txt /tmp/3.xml
/tmp/3.xml | System Values

schedoParam:affinity_lim 6 | 7
iooParam:posix_aio_maxservers 60 | 30
```

5.4.4 Latest enhancements

With AIX 6.1 TL 6, new enhancements to AIX Runtime Expert are:

- ► LDAP support to distribute files across the network
- ► NIM server remote setting
- ► Capability to do profile versioning, meaning that output profiles can have customized version numbers (`artexget -V` option
- ► Adding a custom profile description to the profile output by using the `artexget -m` command option
- ► Prioritization of parameters and catalogs for set operation
- ► Snap command updates
- ► Director plug-in enablement (see fileset `artex.base.agent`)

The Director plug-in is also known as AIX Profile Manager (APM), which makes possible views and runtime configuration profile management over groups of systems across the data center.

It uses LDAP for distributing files across the network. See the `mksecldap`, `secldapcintd` and `ldapadd` commands. The configuration LDAP file is found as `/etc/security/ldap/ldap.cfg`.

Use of APM allows retrieval, copy, modification and delete of profiles in an easy GUI way, such as using check box style over AIX Runtime Expert templates.

See Director plug-in documentation for more information in the System Director Information Center.

On a NIM server `artexremset` provides the ability to execute `artexset` commands on each client with a designated profile provided by the server or a profile stored on an LDAP server. The command syntax would be similar to:

```
artexremset -L ldap://profile1.xml client1 client2
```

To retrieve a profile on an LDAP server you can use the command:

```
artexget ldap://profile1.xml
```

5.5 Removal of CSM

Starting with AIX V7.1, the Cluster Systems Management (CSM) software will no longer ship with AIX media. CSM will not be supported with AIX V7.1. Table 5-2 lists the filesets that have been removed.

Table 5-2   Removed CSM fileset packages

Fileset	Description
csm.bluegene	CSM support on Blue Gene®
csm.client	Cluster Systems Management Client
csm.core	Cluster Systems Management Core
csm.deploy	Cluster Systems Management Deployment Component
csm.diagnostics	Cluster Systems Management Probe Manager / Diagnostics
csm.dsh	Cluster Systems Management Dsh
csm.essl	Cluster Systems Management ESSL Solution Pack
csm.gpfs	Cluster Systems Management GPFS™ Solution Pack
csm.gui.dcem	Distributed Command Execution Manager Runtime Environment
csm.gui.websm	CSM Graphical User Interface.
csm.hams	Cluster Systems Management HA
csm.hc_utils	Cluster Systems Management Hardware Control Utilities
csm.hpsnm	IBM Switch Network Manager
csm.ll	Cluster Systems Management LoadLeveler® Solution Pack
csm.msg.*	CSM Core Function Messages
csm.pe	Cluster Systems Management PE Solution Pack
csm.pessl	CSM Parallel Engineering Scientific Subroutines Library
csm.server	Cluster Systems Management Server

IBM is shifting to a dual-prong strategy for the system management of IBM server clusters. The strategy and plans have diverged to meet the unique requirements

of High Performance Computing (HPC) customers as compared to those of general computing customers.

High Performance Computing

For HPC customers, the Extreme Cloud Administration Toolkit (xCAT), an open source tool originally developed for IBM System x clusters, has been enhanced to support all of the HPC capabilities of CSM on all of the platforms that CSM currently supports. Clients can begin planning to transition to this strategic cluster system management tool for HPC. IBM will continue to enhance xCAT to meet the needs of the HPC client set.

xCAT provides some improvements over CSM. These include:

► Better scalability, including hierarchical management

► Support for a broader range of hardware and operating systems

► iSCSI support

► Automatic setup of additional services: DNS, syslog, NTP, and LDAP

► Automatic node definition through the discovery process

Refer to the following publication for detailed information relating to xCAT:

xCAT 2 Guide for the CSM System Administrator, REDP-4437 at:

   http://www.redbooks.ibm.com/redpapers/pdfs/redp4437.pdf

General computing

For general computing clients who operate non-HPC clustering infrastructures, IBM Systems Director and its family of products are the IBM strategic cross-platform system management solution.

IBM Systems Director helps clients achieve the full benefits of virtualization in their data center by reducing the complexity of systems management. IBM Systems Director VMControl™ Image Manager V2.2, a plug-in to IBM Systems Director, provides support to manage and automate the deployment of virtual appliances from a centralized location.

Together, IBM Systems Director and VMControl provide many cluster management capabilities found in CSM, such as systems discovery, node inventory, node groups, event monitoring, firmware flashing, and automated responses. They also provide many cluster management capabilities such as CSM's distributed command execution and remote console, NIM-based AIX `mksysb` installation for HMC and IVM-managed LPARs, and the deployment of one or many AIX and/or Linux® virtual server images. IBM Systems Director

includes a command line interface (CLI) for scripting most cluster management functions.

For more information relating to IBM Systems Director, refer to the following websites:

> http://www.ibm.com/systems/management/director/
> http://www.ibm.com/power/software/management/

Other functions of CSM have been ported to the Distributed Systems Management (DSM) package. For example, commands such as **dsh** and **dcp** are located in this package. This component is required in an IBM Systems Director environment. The dsm.core package was first shipped with AIX V6.1 with the 6100-03 Technology Level. Documentation relating to configuration and usage is located in the /opt/ibm/sysmgt/dsm/doc/dsm_tech_note.pdf file from the dsm.core fileset. Refer to the following websites for install and usage information relating to this fileset:

> http://publib.boulder.ibm.com/infocenter/director/v6r2x/index.jsp?to pic=/com.ibm.director.install.helps.doc/fqm0_t_preparing_to_install_ ibm_director_on_aix.html
> http://publib.boulder.ibm.com/infocenter/director/v6r2x/index.jsp?to pic=/com.ibm.director.cli.helps.doc/fqm0_r_cli_remote_access_cmds.ht ml

Functionality relating to Dynamic Logical Partitioning (DLPAR), previously provided by CSM, has been ported to Reliable Scalable Cluster Technology (RSCT). Previous releases of AIX required that the csm.core fileset be installed in order to support DLPAR functions. This functionality is now provided by the rsct.core.rmc fileset, which is automatically installed by default.

5.6 Removal of IBM Text-to-Speech

The IBM Text-to-Speech (TTS) package is a speech engine that allows applications to produce speech. Starting with AIX V7.1, the IBM TTS will no longer ship with the AIX Expansion Pack. The contents of the Expansion Pack vary over time. New software products can be added, changed, or removed. Changes to the content of the AIX Version 7.1 Expansion Pack are announced either as part of an AIX announcement or independently of the release announcement.

TTS is installed in the /usr/opt/ibmtts directory. The following filesets will no longer be included with this media:

tts_access.base – IBM TTS runtime base

`tts_access.base.en_US` - IBM TTS runtime (U.S. English)

Refer to the following website for the latest information relating to the contents of the AIX Expansion Pack:

http://www.ibm.com/systems/power/software/aix/expansionpack/

5.7 AIX device renaming

Devices can be renamed in AIX 6.1 TL6 and 7.1 with the **rendev** command. One of the use cases would be to rename a group of disks on which application data may reside, to be able to distinguish them from other disks on the system.

Once the device is renamed using **rendev**, the device entry under /dev/ corresponding to the old name will go away. A new entry under /dev/ will be seen corresponding to the new name. Applications should refer to the device using the new name.

> **Note:** Certain devices such as /dev/console, /dev/mem, /dev/null, and others that are identified only with /dev special files cannot be renamed. These devices typically do not have any entry in the ODM configuration database.
>
> Some devices may have special requirements on their names in order for other devices or applications to use them. Using the **rendev** command to rename such a device may result in the device being unusable.
>
> Devices that are in use cannot be renamed.

Example 5-42 shows how the disk hdisk11 is renamed to testdisk1.

Example 5-42   Renaming device

```
# lspv
hdisk0          00cad74f7904d234                    rootvg          active
hdisk1          00cad74fa9d4a6c2                    None
hdisk2          00cad74fa9d3b8de                    None
hdisk3          00cad74f3964114a                    None
hdisk4          00cad74f3963c575                    None
hdisk5          00cad74f3963c671                    None
hdisk6          00cad74f3963c6fa                    None
hdisk7          00cad74f3963c775                    None
hdisk8          00cad74f3963c7f7                    None
hdisk9          00cad74f3963c873                    None
hdisk10         00cad74f3963ca13                    None
```

```
hdisk11          00cad74f3963caa9                      None
hdisk12          00cad74f3963cb29                      None
hdisk13          00cad74f3963cba4                      None
# rendev -l hdisk11 -n testdisk1
# lspv
hdisk0           00cad74f7904d234                      rootvg          active
hdisk1           00cad74fa9d4a6c2                      None
hdisk2           00cad74fa9d3b8de                      None
hdisk3           00cad74f3964114a                      None
hdisk4           00cad74f3963c575                      None
hdisk5           00cad74f3963c671                      None
hdisk6           00cad74f3963c6fa                      None
hdisk7           00cad74f3963c775                      None
hdisk8           00cad74f3963c7f7                      None
hdisk9           00cad74f3963c873                      None
hdisk10          00cad74f3963ca13                      None
testdisk1        00cad74f3963caa9                      None
hdisk12          00cad74f3963cb29                      None
hdisk13          00cad74f3963cba4                      None
```

5.8 1024 Hardware thread enablement

AIX 7.1 provides support to run the partition with up to 1024 logical CPUs, both in dedicated and shared processor modes. This has been tested on the IBM 9119-FHB system. The earlier limit on the number of supported processors was 256 on AIX 6.1 TL4 on POWER 7 technology-based systems.

Example 5-43 shows sample output from a few commands executed on the Power 795 system giving details about the system configuration. The `lsattr` command gives information such as modelname. Processor and memory information is seen under the `lparstat` command output. Scheduler Resource Allocation Domains (SRAD) information is seen under the `lssrad` command output.

Example 5-43   Power 795 system configuration

```
# lsattr -El sys0
SW_dist_intr    false               Enable SW distribution of interrupts          True
autorestart     true                Automatically REBOOT OS after a crash         True
boottype        disk                N/A                                           False
capacity_inc    1.00                Processor capacity increment                  False
capped          true                Partition is capped                           False
conslogin       enable              System Console Login                          False
cpuguard        enable              CPU Guard                                     True
```

```
dedicated        true              Partition is dedicated                        False
enhanced_RBAC    true              Enhanced RBAC Mode                            True
ent_capacity     256.00            Entitled processor capacity                   False
frequency        6400000000        System Bus Frequency                          False
fullcore         true              Enable full CORE dump                         True
fwversion        IBM,ZH720_054     Firmware version and revision levels          False
ghostdev         0                 Recreate devices in ODM on system change      True
id_to_partition  0X80000D2F7C100002 Partition ID                                 False
id_to_system     0X80000D2F7C100000 System ID                                    False
iostat           false             Continuously maintain DISK I/O history        True
keylock          normal            State of system keylock at boot time          False
log_pg_dealloc   true              Log predictive memory page deallocation events True
max_capacity     256.00            Maximum potential processor capacity          False
max_logname      9                 Maximum login name length at boot time        True
maxbuf           20                Maximum number of pages in block I/O BUFFER CACHE True
maxmbuf          0                 Maximum Kbytes of real memory allowed for MBUFS True
maxpout          8193              HIGH water mark for pending write I/Os per file True
maxuproc         64000             Maximum number of PROCESSES allowed per user  True
min_capacity     1.00              Minimum potential processor capacity          False
minpout          4096              LOW water mark for pending write I/Os per file True
modelname        IBM,9119-FHB      Machine name                                  False
ncargs           256               ARG/ENV list size in 4K byte blocks           True
nfs4_acl_compat  secure            NFS4 ACL Compatibility Mode                   True
ngroups_allowed  128               Number of Groups Allowed                      True
pre430core       false             Use pre-430 style CORE dump                   True
pre520tune       disable           Pre-520 tuning compatibility mode             True
realmem          4219994112        Amount of usable physical memory in Kbytes    False
rtasversion      1                 Open Firmware RTAS version                     False
sed_config       select            Stack Execution Disable (SED) Mode            True
systemid         IBM,020288C75     Hardware system identifier                    False
variable_weight  0                 Variable processor capacity weight            False
# lparstat -i
Node Name                                 : test1
Partition Name                            : test1new
Partition Number                          : 2
Type                                      : Dedicated
Mode                                      : Capped
Entitled Capacity                         : 256.00
Partition Group-ID                        : 32770
Shared Pool ID                            : -
Online Virtual CPUs                       : 256
Maximum Virtual CPUs                      : 256
Minimum Virtual CPUs                      : 1
Online Memory                             : 4121088 MB
Maximum Memory                            : 4194304 MB
Minimum Memory                            : 256 MB
Variable Capacity Weight                  : -
Minimum Capacity                          : 1.00
Maximum Capacity                          : 256.00
```

```
Capacity Increment                          : 1.00
Maximum Physical CPUs in system             : 256
Active Physical CPUs in system              : 256
Active CPUs in Pool                         : -
Shared Physical CPUs in system              : 0
Maximum Capacity of Pool                    : 0
Entitled Capacity of Pool                   : 0
Unallocated Capacity                        : -
Physical CPU Percentage                     : 100.00%
Unallocated Weight                          : -
Memory Mode                                 : Dedicated
Total I/O Memory Entitlement                : -
Variable Memory Capacity Weight             : -
Memory Pool ID                              : -
Physical Memory in the Pool                 : -
Hypervisor Page Size                        : -
Unallocated Variable Memory Capacity Weight: -
Unallocated I/O Memory entitlement          : -
Memory Group ID of LPAR                     : -
Desired Virtual CPUs                        : 256
Desired Memory                              : 4121088 MB
Desired Variable Capacity Weight            : -
Desired Capacity                            : 256.00
Target Memory Expansion Factor              : -
Target Memory Expansion Size                : -
Power Saving Mode                           : Disabled
# lssrad -av
REF1   SRAD      MEM      CPU
0
          0   94341.00      0 4 8 12 16 20 24 28
          1   94711.00     32 36 40 44 48 52 56 60
          2   94711.00     64 68 72 76 80 84 88 92
          3   94711.00     96 100 104 108 112 116 120 124
1
          4   94711.00     128 132 136 140 144 148 152 156
          5   94695.00     160 164 168 172 176 180 184 188
          6   94695.00     192 196 200 204 208 212 216 220
          7   94695.00     224 228 232 236 240 244 248 252
2
          8   94695.00     256 260 264 268 272 276 280 284
          9   94695.00     288 292 296 300 304 308 312 316
         10   94695.00     320 324 328 332 336 340 344 348
         11   94695.00     352 356 360 364 368 372 376 380
3
         12   94695.00     384 388 392 396 400 404 408 412
         13   94695.00     416 420 424 428 432 436 440 444
         14   94695.00     448 452 456 460 464 468 472 476
         15   94695.00     480 484 488 492 496 500 504 508
4
```

```
16    93970.94      512 516 520 524 528 532 536 540
17    45421.00      544 548 552 556 560 564 568 572
18    94695.00      576 580 584 588 592 596 600 604
19    94695.00      608 612 616 620 624 628 632 636
5
20    94695.00      640 644 648 652 656 660 664 668
21    94695.00      672 676 680 684 688 692 696 700
22    94695.00      704 708 712 716 720 724 728 732
23    94695.00      736 740 744 748 752 756 760 764
6
24    94695.00      768 772 776 780 784 788 792 796
25    94695.00      800 804 808 812 816 820 824 828
26    94695.00      832 836 840 844 848 852 856 860
27    94864.00      864 868 872 876 880 884 888 892
7
28    94896.00      896 900 904 908 912 916 920 924
29    94880.00      928 932 936 940 944 948 952 956
30    94896.00      960 964 968 972 976 980 984 988
31    94309.00      992 996 1000 1004 1008 1012 1016 1020
```

5.9 Kernel memory pinning

AIX 6.1 TL6 and 7.1 provide a facility to keep AIX kernel and kernel extension data in physical memory for as long as possible. This feature is referred to as Kernel Memory Pinning or Locking. On systems running with sufficiently large amounts of memory, locking avoids unnecessary kernel page faults, thereby providing improved performance.

Kernel memory locking differs from traditional pinning of memory in the following ways:

► Pining is an explicit operation performed using kernel services such as pin(), ltpin(), xlate_pin(), and others. A pinned page is never unpinned until it is explicitly unpinned using the kernel services. Kernel locking is an implicit operation. There are no kernel services to lock and unlock a page.

► Pinned memory is never eligible for stealing by the Least Recently Used (LRU) page replacement demon. Locked memory, on the other hand, is eligible for stealing when no other pages are available for stealing. The real advantage of locked memory is that it is not stolen until no other option is left. Because of this, there are more chances of retaining kernel data in memory for a longer period.

► Pinned memory has a hard limit. Once the limit is reached, the pin service can fail with ENOMEM. Locking enforces a soft limit in the sense that if a page frame can be allocated for the kernel data, it is automatically locked. It cannot

happen that a page frame is not locked due to some locking limit, because there is no such limit.

► User memory can be pinned using the mlock() system call. User memory cannot be locked.

The following are considered as kernel memory that is eligible for locking:

► A kernel segment where the kernel itself resides

► All global kernel space such as kernel heaps, message buffer (mbuf) heaps, Ldata heaps, mtrace buffers, scb pool, and others.

► All kernel space private to a process such as Process private segments for 64-bit processes, kernel thread segments, loader overflow segments, and others.

The following are *not* considered as kernel memory and are *not* locked:

► Process text and data (heaps and user-space stacks)

► Shared library text and data

► Shared memory segments, mmapped segments

► File cache segments

► And a few others

The following Virtual Memory Management (VMM) tunables were added or modified to support kernel memory locking.

► vmm_klock_mode - New tunable to enable and disable kernel memory locking.

► maxpin - Kernel's locked memory is treated like pinned memory. Therefore, the default maxpin% is raised from 80% to 90% if kernel locking is enabled.

Example 5-44 shows how to configure kernel memory locking using the `vmo` tunable.

Example 5-44   Configuring kernel memory locking

```
# vmo -h vmm_klock_mode
Help for tunable vmm_klock_mode:
Purpose:
Select the kernel memory locking mode.
Values:
        Default: 2
        Range: 0 - 3
        Type: Bosboot
        Unit: numeric
Tuning:
```

Kernel locking prevents paging out kernel data. This improves system performance in many cases.
If set to 0, kernel locking is disabled. If set to 1, kernel locking is enabled automatically
if Active Memory Expansion (AME) feature is also enabled. In this mode, only a subset of kernel
memory is locked. If set to 2, kernel locking is enabled regardless of AME and all of kernel
data is eligible for locking. If set to 3, only the kernel stacks of processes are locked in
memory. Enabling kernel locking has the most positive impact on performance of systems that do
paging but not enough to page out kernel data or on systems that do not do paging activity at
all. Note that 1, 2, and 3 are only advisory. If a system runs low on free memory and performs
extensive paging activity, kernel locking is rendered ineffective by paging out kernel data.
Kernel locking only impacts pageable page-sizes in the system.

```
# vmo -L vmm_klock_mode
NAME                       CUR   DEF   BOOT   MIN   MAX   UNIT          TYPE
    DEPENDENCIES
--------------------------------------------------------------------------------
vmm_klock_mode              2     2     2      0     3    numeric         B
--------------------------------------------------------------------------------

# vmo -o vmm_klock_mode
vmm_klock_mode = 2

# vmo -r -o vmm_klock_mode=1
Modification to restricted tunable vmm_klock_mode, confirmation required yes/no yes
Setting vmm_klock_mode to 1 in nextboot file
Warning: some changes will take effect only after a bosboot and a reboot
Run bosboot now? yes/no yes

bosboot: Boot image is 45651 512 byte blocks.
Warning: changes will take effect only at next reboot

# vmo -L vmm_klock_mode
NAME                       CUR   DEF   BOOT   MIN   MAX   UNIT          TYPE
    DEPENDENCIES
--------------------------------------------------------------------------------
vmm_klock_mode              2     2     1      0     3    numeric         B
--------------------------------------------------------------------------------
```

The following are a few guidelines for setting the vmm_klock_mode tunable:

► Setting vmm_klock_mode to value 2 or 3 is an appropriate value for those systems where applications are sensitive to page-faults inside the kernel.

► Value 2 is used for systems where no page-faults of any kind are expected, because kernel is already locked in memory. However, by setting value 2 the system is better prepared for future optimizations in the kernel that require a fully-pinned kernel.

► For systems where value 2 results in excessive paging of user-space data, value 3 is used.

> ► Systems that see their paging spaces getting filled up such that the overall usage does not decrease much even when no applications are running may benefit from using value 3. This is because a nearly full paging space whose usage does not seem to track the usage by applications is most likely experiencing heavy paging of kernel data. For such systems, value 2 is also worth a try; however, the risk of excessive paging of user-space data may be greatly increased.

5.10 ksh93 enhancements

In addition to the default system Korn Shell (/usr/bin/ksh), AIX provides an enhanced version available as Korn Shell (/usr/bin/ksh93) shipped as a 32-bit binary. This enhanced version is mostly upward compatible with the current default version, and includes additional features that are not available in /usr/bin/ksh.

Starting in AIX 7.1, ksh93 is shipped as a 64-bit binary (Version M 93t+ 2009-05-05). This 64-bit binary is built from a more recent code base to include additional features.

For a complete list of information on ksh93, refer to /usr/bin/ksh93 man pages.

5.11 DWARF

AIX V7.1 adds support for the standard DWARF debugging format, which is a modern standard for specifying the format of debugging information in executables. It is used by a wide variety of operating systems and provides greater extensibility and compactness. The widespread use of DWARF also increases the portability of software for developers of compilers and other debugging tools between AIX and other operating systems.

Detailed DWARF debugging information format can be found at:

    http://www.dwarfstd.org

5.12 AIX Event Infrastructure

This AIX Event Infrastructure feature has been enhanced in AIX 6.1 TL 06.

AIX Event Infrastructure is an event monitoring framework for monitoring predefined and user-defined events.

In the context of the AIX Event Infrastructure, an event is defined as:

► Any change of state that can be detected by the kernel or a kernel extension at the exact moment when (or an approximation) the change occurs.

► Any change of value that can be detected by the kernel or a kernel extension at the exact moment when (or an approximation) the change occurs.

In both the change of state and change of value, the events that may be monitored are represented as a pseudo file system.

5.12.1 Some advantages of AIX Event Infrastructure

Advantages of the AIX Event Infrastructure include:

► No need for constant polling. Users monitoring the events are notified when those events occur.

► Detailed information about an event (such as stack trace and user and process information) is provided to the user monitoring the event.

► Existing file system interfaces are used so that there is no need for a new API.

► Control is handed to the AIX Event Infrastructure at the exact time the event occurs.

For further information on the AIX Event Infrastructure, visit:

    http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=/c
    om.ibm.aix.baseadmn/doc/baseadmndita/aix_ev.htm

5.12.2 Configuring the AIX Event Infrastructure

The following procedure outlines the activities required to configure the AIX Event Infrastructure:

1. Install the `bos.ahafs` fileset (available in AIX 6.1 TL 6 and later).

   The AIX V7.1 `bos.ahafs` package description is listed with the **lslpp -l** command in Example 5-45.

Example 5-45   The lslpp -f bos.ahafs package listing

```
# lslpp -l bos.ahafs
  Fileset                 File
-----------------------------------------------------------------------
Fileset                        Level  State      Description
```

```
      ----------------------------------------------------------------------
Path: /usr/lib/objrepos
  bos.ahafs                      7.1.0.0  COMMITTED  Aha File System

Path: /etc/objrepos
  bos.ahafs                      7.1.0.0  COMMITTED  Aha File System
```

2. Create the directory for the desired mount point using the **mkdir** command:

   `mkdir /aha`

3. Run the **mount** command for the file system of type `ahafs` on the desired mount point in order to load the AIX Event Infrastructure kernel extension and create the file system structure needed by the AIX Event Infrastructure environment, as shown in Example 5-46.

Example 5-46   Mounting the file system

```
# mount -v ahafs /aha /aha
# df | grep aha
/aha                     -       -    -        15    1% /aha
# genkex | grep aha
f1000000c033c000    19000 /usr/lib/drivers/ahafs.ext
```

Note1: Only one instance of an AIX Event Infrastructure file system may be mounted at a time.

An AIX Event Infrastructure file system may be mounted on any regular directory, but it is suggested that you use the `/aha` mount point.

Note2: Currently, all directories in the AIX Event Infrastructure file system have a mode of 01777 and all files have a mode of 0666. These modes cannot be changed, but ownership of files and directories may be changed.

Access control for monitoring events is done at the event producer level.

Creation and modification times are not maintained in the AIX Event Infrastructure file system and are always returned as the current time when issuing **stat()** on a file. Any attempt to modify these times will return an error.

5.12.3 Use of monitoring samples

For our purpose we will use an event monitoring called evMon with a C program called eventcatch, shown in Example 5-47 on page 205.

Example 5-47   Source code of simple example eventcatch

```
#cat eventcatch.c
/*                                                                  */
/* Licensed Materials - Property of IBM                             */
/*                                                                  */
/* Restricted Materials of IBM                                      */
/*                                                                  */
/* COPYRIGHT International Business Machines Corp. 2010              */
/* All Rights Reserved                                              */
/*                                                                  */
/* US Government Users Restricted Rights - Use, duplication or      */
/* disclosure restricted by GSA ADP Schedule Contract with IBM Corp. */
/*                                                                  */
/* IBM_PROLOG_END_TAG
 * PURPOSE:
 *     Sample C program to test monitoring an AHA event represented by an
 *      AHA file with suffix ".mon".
 *     It simply waits for an event to happen on the .mon file
 *     Using select() syscall
 * SYNTAX:
 *     mon_wait <aha-monitor-file> [<key1>=<value1>[;<key2>=<value2>;...]]
 *        e.g. mon_wait  /aha/fs/utilFs.monFactory/tmp.mon "THRESH_HI=45"
 *        waits for the file system /tmp usage to reach a thresold value of 45
 * CHANGELOG:
 *     2010/09    Inspired from AIX 6.1 TL04 sample
 */
#include <stdio.h>
#include <string.h>
#include <fcntl.h>
#include <errno.h>
#include <sys/time.h>
#include <sys/select.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <libgen.h>
#include <usersec.h>

#define MAX_WRITE_STR_LEN    255


char      *monFile;
/* --------------------------------------------------------------------------
*/
/* Syntax of user command
*/
void syntax(char *prog)
{
  printf("\nSYNTAX: %s <aha-monitor-file>
[<key1>=<value1>[;<key2>=<value2>;...]] \n",prog);
```

```
            printf(" where: \n");
            printf("  <aha-monitor-file> : Pathname of an AHA file with suffix
        \".mon\".\n");
            printf("  The possible keys and their values are:\n");
            printf("   ----------------------------------------------------------------
        \n");
            printf("         Keys  |        values          |        comments
        \n");
            printf("   ================================================================
        \n");
            printf("    CHANGED | YES (default)          | monitors state-change.
        \n");
            printf("            | or not-YES             | It cannot be used with
        \n");
            printf("            |                        |  THRESH_HI.
        \n");
            printf("   -----------|------------------------|-----------------------
        \n");
            printf("    THRESH_HI | positive integer       | monitors high
        threshold.\n");
            printf("
        ----------------------------------------------------------------\n\n");
            printf("Examples: \n");
            printf("   1: %s /aha/fs/utilFs.monFactory/var.mon  \"THRESH_HI=95\"\n",
        prog);
            printf("   2: %s /aha/fs/modFile.monFactory/etc/passwd.mon \"CHANGED=YES\"
        \n", prog);
            printf("   3: %s /aha/mem/vmo.monFactory/npskill.mon  \n", prog);
            printf("   4: %s /aha/cpu/waitTmCPU.monFactory/waitTmCPU.mon \n", prog);
            printf("                        \"THRESH_HI=50\" \n");
            exit (1);
        }

        /* ----------------------------------------------------------------------------
         * NAME:    checkValideMonFile()
         * PURPOSE: To check whether the file provided is an AHA monitor file.
         */
        int checkValideMonFile(char *str)
        {
            char cwd[PATH_MAX];
        int len1=strlen(str), len2=strlen(".mon");
            int rc = 0;
            struct stat     sbuf;

            /* Make sure /aha is mounted. */
            if ((stat("/aha", &sbuf) < 0) ||
                (sbuf.st_flag != FS_MOUNT))
            {
                printf("ERROR: The filesystem /aha is not mounted!\n");
```

```
            return (rc);
        }

    /* Make sure the path has .mon as a suffix. */
    if ((len1 <= len2) ||
        (strcmp ( (str + len1 - len2), ".mon"))
        )
        goto end;

    if (! strncmp (str, "/aha",4))  /* The given path starts with /aha */
        rc = 1;
    else    /* It could be a relative path */
    {
        getcwd (cwd, PATH_MAX);
        if ((str[0] != '/' )  &&           /* Relative path and */
            (! strncmp (cwd, "/aha",4))  /* cwd starts with /aha . */
            )
            rc = 1;
    }
 end:
    if (!rc)
        printf("ERROR:  %s is not an AHA monitor file !\n", str);
    return (rc);
}
/*------------------------------------------------------------------
 * NAME: read_data
 * PURPOSE: To parse and print the data received at the occurrence
 *          of the event.
 */
void
read_data (int fd)
{
  #define READ_BUF_SIZE  4096
    char   data[READ_BUF_SIZE];
    char   *p, *line;
    char   cmd[64];
    time_t sec, nsec;
    pid_t  pid;
    uid_t  uid, gid;
    gid_t  luid;
    char   curTm[64];
    int    n, seqnum;
    int    stackInfo = 0;
    char   uname[64], lname[64], gname[64];

    bzero((char *)data, READ_BUF_SIZE);

    /* Read the info from the beginning of the file. */
    n=pread(fd, data,READ_BUF_SIZE, 0);
```

```
    p = data;
    line=strsep(&p, "\n");
    while (line)
    {
        if( (!stackInfo) &&
            (sscanf(line,"TIME_tvsec=%ld",&sec) == 1))
        {
            ctime_r(&sec, curTm);
            if (sscanf(p,

"TIME_tvnsec=%ld\nSEQUENCE_NUM=%d\nPID=%ld\nUID=%ld\nUID_LOGIN=%ld\nGID=%ld\nPR
OG_NAME=%s\n",
                        &nsec, &seqnum, &pid, &uid, &luid, &gid, cmd) == 7)
            {
                strcpy(uname, IDtouser(uid));
                strcpy(lname, IDtouser(luid));
                strcpy(gname, IDtogroup(gid));

                printf("Time         : %s",curTm);
                printf("Sequence Num : %d\n",++seqnum);
                printf("Process ID   : %d\n", pid);
                printf("User Info    : userName=%s, loginName=%s,
groupName=%s\n",
                    uname, lname, gname);
                printf("Program Name : %s\n", cmd);
            }
            else if (sscanf(p,
            "TIME_tvnsec=%ld\nSEQUENCE_NUM=%d\n",
                        &nsec, &seqnum) == 2)
            {
                printf("Time         : %s",curTm);
                printf("Sequence Num : %d\n",++seqnum);
        }
            stackInfo=1;
        }
        if (!stackInfo)
        printf ("%s\n", line);
    else if ((!strncmp(line, "RC_FROM_EVPROD",14)) ||
        (!strncmp(line, "CURRENT_VALUE",13)))
    {
        printf("%s\n%s\n", line, p);
        goto out;
    }

        line=strsep(&p, "\n");
    };
 out:
    return;
```

```
}
/*
-----------------------------------------------------------------------
--- */
/* This funtion requires 2 arguments
    . Monitor file name
    . Even thresold parameter
*/
int main (int argc, char *argv[])
{
    char    parameterString[MAX_WRITE_STR_LEN+1];
    char   *dirp;
    char    s[PATH_MAX];
    struct stat buf;
    int     rc=0;
    int     fd;
    fd_set readfds;

    if (argc < 2)
       syntax( argv[0]);

    /* Get .mon file name and check it is valid  */
    /* Checking the /aha structure is also valid */
    monFile = argv[1];
    if ( ! checkValideMonFile(monFile) )
       syntax( argv[0]);

    /* Create intermediate directories of the .mon file if not exist */
    dirp = dirname(monFile);
    if (stat(dirp, &buf) != 0)
    {
        sprintf(s, "/usr/bin/mkdir -p %s", dirp);
        rc = system(s);
        if (rc)
        {
                fprintf (stderr,
                "Could not create intermediate directories of the file %s !\n",
monFile);
                return(-1);
        }
}
    printf("Monitor file name in /aha file system : %s\n", monFile);

    /* Get parameter string or default it to CHANGED=YES */
    if (argc >= 3)
        sprintf (parameterString, "%s", argv[2]);
    else
        sprintf (parameterString, "CHANGED=YES");
```

```
        printf("Monitoring String action : %s\n\n", parameterString);

        /* Open monitoring file name with CREATE mode */
        fd = open (monFile, O_CREAT|O_RDWR);
        if (fd < 0)
        {
            fprintf (stderr,"Could not open the file %s; errno = %d\n",
monFile,errno);
            exit (1);
        }

        /* Write the monitoring string action to the file */
        rc=write(fd, parameterString, strlen(parameterString));
        if (rc < 0)
        {
            perror ("write: ");
            fprintf (stderr, "Failed writing to monFile %s !\n", monFile);
            return(-1);
        }

        FD_ZERO(&readfds);
        FD_SET(fd, &readfds);

        printf("Entering select() to wait till the event corresponding to the AHA
node \n %s occurs.\n", monFile);

        printf("Please issue a command from another window to trigger this
event.\n\n");
        rc = select (fd+1, &readfds, NULL, NULL, NULL);
        printf("\nThe select() completed. \n");
if (rc <= 0) /* No event occurred or an error was found. */
        {
            fprintf (stderr, "The select() returned %d.\n", rc);
            perror ("select: ");
            return (-1);
        }

        if(! FD_ISSET(fd, &readfds))
            goto end;

        printf("The event corresponding to the AHA node %s has occurred.\n\n",
monFile);

        read_data(fd);
 end:
        close(fd);
}
```

The eventcatch monitor is used to monitor a single event only.

Once the monitor is triggered and the event is reported, the eventcatch monitor exits. Any new monitor will need to be reinitiated.

Example 5-48   The syntax output from the eventcatch C program

```
# ./eventcatch

SYNTAX: ./eventcatch <aha-monitor-file> [<key1>=<value1>[;<key2>=<value2>;...]]
 where:
  <aha-monitor-file> : Pathname of an AHA file with suffix ".mon".
  The possible keys and their values are:
    ---------------------------------------------------------------
        Keys  |        values            |         comments
    ===============================================================
     WAIT_TYPE | WAIT_IN_SELECT (default) | uses select() to wait.
               | WAIT_IN_READ             | uses read() to wait.
    -----------|--------------------------|-----------------------
     CHANGED   | YES (default)            | monitors state-change.
               | or not-YES               | It cannot be used with
               |                          |  THRESH_HI.
    -----------|--------------------------|-----------------------
     THRESH_HI | positive integer         | monitors high threshold.
    ---------------------------------------------------------------

Examples:
   1: ./eventcatch /aha/fs/utilFs.monFactory/var.mon  "THRESH_HI=95"
   2: ./eventcatch /aha/fs/modFile.monFactory/etc/passwd.mon "CHANGED=YES"
   3: ./eventcatch /aha/mem/vmo.monFactory/npskill.mon
   4: ./eventcatch /aha/cpu/waitTmCPU.monFactory/waitTmCPU.mon
                     "THRESH_HI=50"
```

Creating the monitor file

Before monitoring an event, the monitor file corresponding to the event must be created. The AIX Event Infrastructure file system does support open() with the O_CREAT flag.

Example 5-49 on page 212 shows the steps required to monitor the /tmp file system for a threshold utilization of 45%.

In Example 5-49, the following definitions are used:

► The eventcatch C program has been used to open the monitor file.

► The monitor file is the /aha/fs/utilFs.monFactory/tmp.mon file.

► The monitor event is the value THRESH_HI=45.

Generally, the necessary subdirectories may need to be created when the mount point is not the / file system. In this example, /tmp is a subdirectory of /, so there is no need to create any subdirectories.

Next, create the monitoring file tmp.mon for the /tmp file system.

> **Note:** Monitoring the root file system would require the creation of a monitor called .mon in /aha/fs/utilFs.monFactory.

Example 5-49   Creating and monitoring the event

```
# df /tmp
Filesystem     512-blocks      Free %Used     Iused %Iused Mounted on
/dev/hd3         262144     255648    3%        42     1% /tmp
# ls /aha/fs/utilFs.monFactory/tmp.mon
/aha/fs/utilFs.monFactory/tmp.mon
# cat /aha/fs/utilFs.monFactory/tmp.mon
# ./eventcatch /aha/fs/utilFs.monFactory/tmp.mon "THRESH_HI=45"
Monitor file name in /aha file system : /aha/fs/utilFs.monFactory/tmp.mon
Monitoring Write Action : THRESH_HI=45

Entering select() to wait till the event corresponding to the AHA node
/aha/fs/utilFs.monFactory/tmp.mon occurs.
Please issue a command from another window to trigger this event.
```

At this stage, the console in Example 5-49 is paused awaiting the event to trigger.

On another window we issue the **dd** command to create the /tmp/TEST file. By doing this, the /tmp file system utilization increases to 29%.

Example 5-50 shows the **dd** command being used to create the /tmp/TEST file.

Example 5-50   Using the dd command to increase /tmp file system utilization

```
# dd if=unix of=/tmp/TEST
68478+1 records in.
68478+1 records out.
# df /tmp
Filesystem     512-blocks      Free %Used     Iused %Iused Mounted on
/dev/hd3         262144     187168   29%        43     1% /tmp
```

Because the /tmp file system did not reach the 45% threshold limit defined by the THRESH_HI value, no activity or response was seen on the initial window.

In Example 5-51, a second **dd** command is used to create the /tmp/TEST2 file.

Example 5-51   Increase of /tmp file system utilization to 55%

```
# df /tmp
Filesystem     512-blocks     Free %Used     Iused %Iused Mounted on
/dev/hd3          262144    187168   29%        43     1% /tmp
# dd if=unix of=/tmp/TEST2
68478+1 records in.
68478+1 records out.
# df /tmp
Filesystem     512-blocks     Free %Used     Iused %Iused Mounted on
/dev/hd3          262144    118688   55%        44     1% /tmp
#
```

In Example 5-51, the /tmp file system utilization has now reached 55%, which is above the 45% trigger defined in the value THRESH_HI, in Example 5-49 on page 212.

The eventcatch C program will now complete and the initial window will display the response seen in Example 5-52.

Example 5-52   The THRESH_HI threshold is reached or exceeded

```
The select() completed.
The event corresponding to the AHA node
/aha/fs/utilFs.monFactory/testfs.mon has occurred.

BEGIN_EVENT_INFO
Time          : Mon Nov  8 09:03:39 2010
Sequence Num  : 3
CURRENT_VALUE=40
RC_FROM_EVPROD=1000
END_EVENT_INFO
```

To summarize, once a successful write has been performed to the monitor file /aha/fs/utilFs.monFactory/tmp.mon, the monitor waits on the event in select().

The select() call will return indicating that the event has occurred. Monitors waiting in select() will need to perform a separate read() to obtain the event data.

Once the event occurs, it will no longer be monitored by the monitor process (This is only true if you are not using continuous monitoring (NOTIFY_CNT=-1)).

If another monitoring of the event is required, another monitor needs to be initiated to again specify how and when to notify of the alert process.

> **Note:** Writing information to the monitor file only prepares the AIX Event Infrastructure file system for a subsequent select() or blocking read(). Monitoring does not start until a select() or blocking read() is done.
>
> To prevent multiple threads from overwriting each other's data, if a process already has a thread waiting in a select() or read() call, another thread's write to the file will return EBUSY.

Available predefined event producers

A set of predefined event producers is available in the system. They are modFile, modDir, utilFs, waitTmCPU, waitersFreePg, waitTmPgInOut, vmo, schedo, pidProcessMon, and processMon.

When the system is part of an active cluster, more predefined event producers are available such as nodeList, clDiskList, linkedCl, nodeContact, nodeState, nodeAddress, networkAdapterState, clDiskState, repDiskState, diskState, and vgState.

5.13 Olson time zone support in libc

Beginning with AIX V6.1 the operating system recognizes and processes the Olson time zone naming conventions to facilitate support for a comprehensive set of time zones. This feature offers an alternative to the industry standard time zone convention based on the POSIX time zone specification. To implement the Olson time zone feature, AIX V6.1 used the International Components for Unicode (ICU) library APIs that are shipped in the `ICU4C.rte` fileset.

In AIX V7.1 the implementation of the Olson time zone support has been enhanced in the following ways:

► Olson time zone support is provided as integrated component of the native libc standard AIX C library through the implementation of the public domain code developed and distributed by Arthur David Olson. The source code is available through the government website of the National Institute of Health (NIH):

`ftp://elsie.nci.nih.gov/pub/`

This enhancement streamlines the Olson functionality by removing the dependency on an additional external library, thus reducing some execution and memory overhead.

- The Olson tz database, also known as zoneinfo database
  `/usr/share/lib/zoneinfo`, is updated with the latest time zone binaries.

- The time zone compiler `zic` command and the command to dump the time
  zone information, `zdump`, are modified to work with the updated time zone data
  files.

- The undocumented `/usr/lib/nls/lstz` command makes use of the updated
  zoneinfo database. The Systems Management Interface Tool (SMIT), for
  example, utilizes the `lstz` command to produce a list of available countries
  and regions to choose from. Note that undocumented commands and
  features are not officially supported for client use, are not covered by the AIX
  compatibility statement, and may be subject to change without notice.

As indicated above, you can rely on SMIT to configure the server time zone by
using system-defined values for the TZ environment variable. The SMIT fast path
chtz_date will directly open the Change/Show Date and Time panel from where
you can access the Change Time Zone Using System Defined Values menu.

5.14 Withdrawal of the Web-based System Manager

The initial technology release of the Web-based System Manager was provided
with AIX V4.3 in October 1997 and about half a year later in April 1998 AIX
V4.3.1 delivered the first full functional version. Web-based System Manager was
implemented as a Java-based client-server system management application and
received many enhancements over the past years. However, with the introduction
of the IBM Systems Director cross-platform management suite and the IBM
Systems Director Console for AIX (pConsole), more modern and more powerful
system administration tools are available today.

The Web-based System Manager is no longer supported in AIX V7.1 and later
releases. The withdrawal of support has the following impact on Web-based
System Manager components:

- The Web-based System Manager server component is no longer included
  with AIX V7.1.

- AIX V7.1 systems cannot be managed by existing Web-based System
  Manager clients.

- The Web-based System Manager Remote Clients for Windows® and Linux
  operating system environments are no longer delivered with the AIX V7.1
  product.

Table 5-3 lists the filesets that are removed during a base operating system migration installation from previous AIX releases to AIX V7.1.

Table 5-3   Web-based System Manager related obsolete filesets

Fileset name	Fileset description
bos.aixpert.websm	AIX Security Hardening WebSM
bos.net.ipsec.websm	IP Security WebSM
invscout.websm	Inventory Scout WebSM Firmware Management GUI
sysmgt.sguide.rte	TaskGuide Runtime Environment
sysmgt.websm.accessibility	WebSM Accessibility Support
sysmgt.websm.apps	Web-based System Manager Applications
sysmgt.websm.diag	Web-based System Manager Diagnostic Applications
sysmgt.websm.diskarray.fc	Web-based System Manager FC SCSI Disk Array Application
sysmgt.websm.framework	Web-based System Manager Client/Server Support
sysmgt.websm.icons	Web-based System Manager Icons
sysmgt.websm.rte	Web-based System Manager Runtime Environment
sysmgt.websm.webaccess	WebSM Web Access Enablement
sysmgt.websm.security	Web-based System Manager base security function (AIX Expansion Pack)
sysmgt.websm.security-us	Web-based System Manager stronger encryption capabilities for the US and other selected countries (AIX Expansion Pack)
sysmgt.pconsole.apps.websm	System P Console - Web-Based System Manager LIC
sysmgt.help.$LL.websm[a]	WebSM Extended Helps
sysmgt.help.msg.$LL.websm[a]	WebSM Context Helps
sysmgt.msg.$LL.sguide.rte[a]	TaskGuide Viewer Messages
sysmgt.msg.$LL.websm.apps[a]	WebSM Client Apps. Messages

a. $LL designates the installation specific locals

6

Performance management

The performance of a computer system is evaluated based on client expectations and the ability of the system to fulfill these expectations. The objective of performance management is to balance between appropriate expectations and optimizing the available system resources.

Many performance-related issues can be traced back to operations performed by a person with limited experience and knowledge who unintentionally restricts some vital logical or physical resource of the system. Most of these actions may at first be initiated to optimize the satisfaction level of some users, but in the end, they degrade the overall satisfaction of other users.

This chapter discusses the following performance management enhancements:

6.1 Support for Active Memory Expansion

Active Memory™ Expansion (AME) is a technology available on IBM POWER7™ processor-based systems. It provides the capability for expanding a system's effective memory capacity. AME employs memory compression technology to transparently compress in-memory data, allowing more data to be placed into memory. This has the positive effect of expanding the memory capacity for a given system. Refer to the following website for detailed information relating to AME:

http://www.ibm.com/systems/power/hardware/whitepapers/am_exp.html

With the introduction of AME a tool was required to monitor, report, and plan for an AME environment. To assist in planning the deployment of a workload in an AME environment, a tool known as the Active Memory Expansion Planning and Advisory Tool (`amepat`) has been introduced. Several existing AIX performance tools have been modified to monitor AME statistics. This section discusses the performance monitoring tools related to AME monitoring and reporting.

6.1.1 The amepat command

This tool is available in AIX V7.1 and in AIX V6.1 with the 6100-04 Technology Level, Service Pack 2. The utility is able to monitor global memory usage for an individual LPAR. The `amepat` command serves two key functions:

► Workload Planning

The `amepat` command can be run to determine whether a workload would benefit from AME, and also to provide a list of possible AME configurations for a particular workload.

► Monitoring

When AME is enabled, the `amepat` command can be used to monitor the workload and AME performance statistics.

The tool can be invoked in two different modes:

► Recording

In this mode `amepat` records system configurations and various performance statistics into a user-specified recording file.

► Reporting

In this mode the `amepat` command analyzes the system configuration and performance statistics, collected in real time or from the user-specified recording file, to generate workload utilization and planning reports.

When considering using AME for an existing workload, the `amepat` command can be used to provide guidance on possible AME configurations. You can run the `amepat` command on an existing system that is not currently using AME. The tool will monitor the memory usage, memory reference patterns, and data compressibility over a (user-configurable) period of time. A report is generated with a list of possible AME configurations for the given workload. Estimated processor utilization impacts for the different AME configurations are also shown.

The tool can be run on all versions of IBM Power Systems supported by AIX V6.1 and AIX V7.1. This includes POWER4™, POWER5, POWER6, and POWER7 processors.

Two key considerations when running the `amepat` command, when planning for a given workload, are time and duration.

► Time

  The time at which to run the tool. To get the best results from the tool, it must be run during a period of peak utilization on the system. This ensures that the tool captures peak utilization of memory for the specific workload.

► Duration

  The duration to run the tool. A monitoring duration must be specified when starting the `amepat` command. For the best results from the tool, it must be run for the duration of peak utilization on the system.

The tool can also be used on AME-enabled systems to provide a report of other possible AME configurations for a workload.

The `amepat` command requires privileged access to run in *Workload Planning* mode. If the tool is invoked without the necessary privilege, then the planning capability is disabled (the -N flag is turned on implicitly), as shown in Example 6-1.

Example 6-1   Running amepat without privileged access

```
$ amepat
WARNING: Running in no modeling mode.

Command Invoked            : amepat

Date/Time of invocation    : Mon Aug 30 17:21:25 EDT 2010
Total Monitored time       : NA
Total Samples Collected    : NA

System Configuration:
---------------------
```

```
Partition Name              : 75021p01
Processor Implementation Mode : POWER7
Number Of Logical CPUs      : 16
Processor Entitled Capacity : 1.00
Processor Max. Capacity     : 4.00
True Memory                 : 8.00 GB
SMT Threads                 : 4
Shared Processor Mode       : Enabled-Uncapped
Active Memory Sharing       : Disabled
Active Memory Expansion     : Enabled
Target Expanded Memory Size : 16.00 GB
Target Memory Expansion factor : 2.00


System Resource Statistics:          Current
---------------------------       ----------------
CPU Util (Phys. Processors)          0.10 [  2%]
Virtual Memory Size (MB)             1697 [ 10%]
True Memory In-Use (MB)              1621 [ 20%]
Pinned Memory (MB)                   1400 [ 17%]
File Cache Size (MB)                   30 [  0%]
Available Memory (MB)               14608 [ 89%]

AME Statistics:                      Current
---------------                   ----------------
AME CPU Usage (Phy. Proc Units)      0.00 [  0%]
Compressed Memory (MB)                203 [  1%]
Compression Ratio                    2.35
Deficit Memory Size (MB)               74 [  0%]
```

This tool can also be used to monitor processor and memory usage statistics only. In this mode, the `amepat` command will gather processor and memory utilization statistics but will not provide any workload planning data or reports. If it is invoked without any duration or interval, the `amepat` command provides a snapshot report of the LPAR's memory and processor utilization, as shown in Example 6-2.

Example 6-2   Processor and memory utilization snapshot from amepat

```
# amepat

Command Invoked         : amepat

Date/Time of invocation : Mon Aug 30 17:37:58 EDT 2010
Total Monitored time    : NA
```

```
            Total Samples Collected        : NA

            System Configuration:
            ---------------------
            Partition Name             : 7502lp01
            Processor Implementation Mode : POWER7
            Number Of Logical CPUs     : 16
            Processor Entitled Capacity : 1.00
            Processor Max. Capacity     : 4.00
            True Memory                : 8.00 GB
            SMT Threads                : 4
            Shared Processor Mode      : Enabled-Uncapped
            Active Memory Sharing      : Disabled
            Active Memory Expansion    : Enabled
            Target Expanded Memory Size : 16.00 GB
            Target Memory Expansion factor : 2.00


            System Resource Statistics:            Current
            ---------------------------        ----------------
            CPU Util (Phys. Processors)            0.45 [ 11%]
            Virtual Memory Size (MB)               1706 [ 10%]
            True Memory In-Use (MB)                1590 [ 19%]
            Pinned Memory (MB)                     1405 [ 17%]
            File Cache Size (MB)                      11 [  0%]
            Available Memory (MB)                 13994 [ 85%]

            AME Statistics:                        Current
            ---------------                    ----------------
            AME CPU Usage (Phy. Proc Units)        0.02 [  1%]
            Compressed Memory (MB)                  237 [  1%]
            Compression Ratio                      2.25
            Deficit Memory Size (MB)                700 [  4%]
```

Example 6-3 demonstrates how to generate a report with a list of possible AME configurations for a workload. The tool includes an estimate of the processor utilization impacts for the different AME configurations.

Example 6-3   List possible AME configurations for an LPAR with amepat

```
# amepat 1

Command Invoked            : amepat 1

Date/Time of invocation    : Tue Aug 31 12:35:17 EDT 2010
```

```
Total Monitored time          : 1 mins 51 secs
Total Samples Collected        : 1


System Configuration:
---------------------
Partition Name                 : 7502lp02
Processor Implementation Mode  : POWER7
Number Of Logical CPUs         : 16
Processor Entitled Capacity    : 1.00
Processor Max. Capacity        : 4.00
True Memory                    : 8.00 GB
SMT Threads                    : 4
Shared Processor Mode          : Enabled-Uncapped
Active Memory Sharing          : Disabled
Active Memory Expansion        : Disabled


System Resource Statistics:              Current
---------------------------          ----------------
CPU Util (Phys. Processors)           1.74 [ 44%]
Virtual Memory Size (MB)              5041 [ 62%]
True Memory In-Use (MB)               5237 [ 64%]
Pinned Memory (MB)                    1448 [ 18%]
File Cache Size (MB)                    180 [  2%]
Available Memory (MB)                 2939 [ 36%]


Active Memory Expansion Modeled Statistics         :
---------------------------------------------
Modeled Expanded Memory Size :   8.00 GB
Achievable Compression ratio   :2.12


Expansion     Modeled True     Modeled                CPU Usage
Factor        Memory Size      Memory Gain            Estimate
---------     -------------    ------------------     -----------
    1.00          8.00 GB        0.00 KB [  0%]       0.00 [  0%]
    1.11          7.25 GB      768.00 MB [ 10%]       0.00 [  0%]
    1.19          6.75 GB        1.25 GB [ 19%]       0.00 [  0%]
    1.34          6.00 GB        2.00 GB [ 33%]       0.00 [  0%]
    1.40          5.75 GB        2.25 GB [ 39%]       0.00 [  0%]
    1.53          5.25 GB        2.75 GB [ 52%]       0.00 [  0%]
    1.60          5.00 GB        3.00 GB [ 60%]       0.00 [  0%]


Active Memory Expansion Recommendation:
---------------------------------------
The recommended AME configuration for this workload is to configure the LPAR
with a memory size of 5.00 GB and to configure a memory expansion factor
```

```
of 1.60.  This will result in a memory gain of 60%. With this
configuration, the estimated CPU usage due to AME is approximately 0.00
physical processors, and the estimated overall peak CPU resource required for
the LPAR is 1.74 physical processors.

NOTE: amepat's recommendations are based on the workload's utilization level
during the monitored period. If there is a change in the workload's utilization
level or a change in workload itself, amepat should be run again.

The modeled Active Memory Expansion CPU usage reported by amepat is just an
estimate.  The actual CPU usage used for Active Memory Expansion may be lower
or higher depending on the workload.
```

The `amepat` report consists of six different sections, discussed here.

Command Information

This section provides details about the arguments passed to the tool, such as time of invocation, total time the system was monitored and the number of samples collected.

System Configuration

In this section, details relating to the system's configuration are shown. The details are listed in Table 6-1.

Table 6-1   System Configuration details reported by amepat

System Configuration Detail	Description
Partition Name	The node name from where the `amepat` command is invoked.
Processor Implementation Mode	The processor mode. The mode can be POWER4, POWER5, POWER6, and POWER7.
Number of Logical CPUs	The total number of logical processors configured and active in the partition.
Processor Entitled Capacity	Capacity Entitlement of the partition, represented in physical processor units. **Note:** The physical processor units can be expressed in fractions of CPU, for example, 0.5 of a physical processor.

Processor Max. Capacity	Maximum Capacity this partition can have, represented in physical processor units. **Note:** The physical processor unit can be expressed in fractions of CPU, for example, 0.5 of a physical processor.
True Memory	The true memory represents real physical or logical memory configured for this LPAR.
SMT Threads	Number of SMT threads configured in the partition. This can be 1, 2, or 4.
Shared Processor Mode	Indicates whether the Shared Processor Mode is configured for this partition. Possible values are: **Disabled** - Shared Processor Mode is not configured. **Enabled-Capped** - Shared Processor Mode is enabled and running in capped mode. **Enabled-Uncapped** - Shared Processor Mode is enabled and running in uncapped mode.
Active Memory Sharing	Indicates whether Active Memory Sharing is **Enabled** or **Disabled**.
Active Memory Expansion	Indicates whether Active Memory Expansion is **Enabled** or **Disabled**.
Target Expanded Memory Size	Indicates the target expanded memory size in megabytes for the LPAR. The Target Expanded Memory Size is the True Memory Size multiplied by the Target Memory Expansion Factor. **Note:** This is displayed only when AME is enabled.
Target Memory Expansion Factor	Indicates the target expansion factor configured for the LPAR. **Note:** This is displayed only when AME is enabled.

System Resource Statistics

In this section, details relating to the system resource utilization, from a processor and memory perspective, are displayed.

Table 6-2   System resource statistics reported by amepat

System Resource	Description
CPU Util	The Partition's processor utilization in units of number of physical processors. The percentage of utilization against the Maximum Capacity is also reported. **Note:** If AME is enabled, the processor utilization due to memory compression or decompression is also included.
Virtual Memory Size	The Active Virtual Memory size in megabytes. The percentage against the True Memory size is also reported.
True Memory In-Use	This is the amount of the LPAR's real physical (or logical) memory in megabytes. The percentage against the True Memory size is also reported.
Pinned Memory	This represents the pinned memory size in megabytes. The percentage against the True Memory size is also reported.
File Cache Size	This represents the non-computational file cache size in megabytes. The percentage against the True Memory size is also reported.
Available Memory	This represents the size of the memory available, in megabytes, for application usage. The percentage against the True Memory Size is also reported.

Note: If `amepat` is run with a duration and interval, then Average, Minimum and Maximum utilization metrics are displayed.

Active Memory Expansion statistics

If AME is enabled, then AME usage statistics are displayed in this section. Table 6-3 describes the various statistics that are reported.

Table 6-3   AME statistics reported using amepat

Statistic	Description
AME CPU Usage	The processor utilization for AME activity in units of physical processors. It indicates the amount of processing capacity used for memory compression activity. The percentage of utilization against the Maximum Capacity is also reported.
Compressed Memory	The total amount of virtual memory that is compressed. This is measured in megabytes. The percentage against the Target Expanded Memory Size is also reported.
Compression Ratio	This represents how well the data is compressed in memory. A higher compression ratio indicates that the data compresses to a smaller size. For example, if 4 kilobytes of data can be compressed down to 1 kilobyte, then the compression ration is 4.0.
Deficit Memory Size	The size of the expanded memory, in megabytes, deficit for the LPAR. This is only displayed if the LPAR has a memory deficit. The percentage against the Target Expanded Memory Size is also reported.

Note: The AME statistics section is only displayed when the tool is invoked on an AME-enabled machine. It also displays the Average, Minimum, and Maximum values when run with a duration and interval.

Active Memory Expansion modeled statistics

This section provides details for the modeled statistics for AME. Table 6-4 describes the information relating to modeled statistics.

Table 6-4   AME modeled statistics

Modeled Expanded Memory Size	Represents the expanded memory size that is used to produce the modeled statistics.

Average Compression Ratio	Represents the average compression ratio of the in-memory data of the workload. This compression ratio is used to produce the modeled statistics.
Modeled Expansion Factor	Represents the modeled target memory expansion factor.
Modeled True Memory Size	Represents the modeled true memory size (real physical or logical memory).
Modeled Memory Gain	Represents the amount of memory the partition can gain by enabling AME for the reported modeled expansion factor.
AME CPU Usage Estimate	Represents an estimate of the processor that would be used for memory compression activity. The processor usage is reported in units of physical processors. The percentage of utilization against the Maximum Capacity is also reported. **Note:** This is an estimate and should only be used as a guide. The actual usage can be higher or lower depending on the workload.

Considerations

This section provides information relating to optimal AME configurations and the benefits they may provide to the currently running workload. These considerations are based on the behavior of the system during the monitoring period. They can be used for guidance when choosing an optimal AME configuration for the system. Actual statistics can vary based on the real time behavior of the workload. AME statistics and considerations are used for workload planning.

Note: Only one instance of the `amepat` command is allowed to run, in *Workload Planning* mode, at a time. If you attempt to run two instances of the tool in this mode, the following message will be displayed:

```
amepat: Only one instance of amepat is allowed to run at a time.
```

The tool can also be invoked using the smit fast path, `smit amepat`.

The command is restricted in a WPAR environment. If you attempt to run the tool from a WPAR, an error message is displayed, as shown in Example 6-4.

Example 6-4   Running amepat within a WPAR

```
# amepat
amepat: amepat cannot be run inside a WPAR
```

The optional `amepat` command line flags and their descriptions are listed in Table 6-5.

Table 6-5   Optional command line flags of amepat

Flag	Description
-a	Specifies to auto-tune the expanded memory size for AME modeled statistics. When this option is selected, the Modeled Expanded Memory Size is estimated based on the current memory usage of the workload (excludes the available memory size). **Note:** `-a -t` are mutually exclusive.
-c *max_ame_cpuusage%*	Specifies the maximum AME processor usage in terms of percentage to be used for producing the modeled statistics and uses. **Note:** The default maximum used is 15%. The **-C** and **-c** options cannot be specified together. The **-c** and **-e** options are mutually exclusive.
-C *max_ame_cpuusage%*	Specifies the maximum AME processor usage in terms of number of physical processors to be used for producing the modeled statistics and uses. **Note:** The **-C** and **-c** option cannot be specified together. The **-C** and **-e** options are mutually exclusive.

Flag	Description
-e *startexpfactor:stopexpfactor:incexpfactor*	Specifies the range of expansion factors to be reported in the AME Modeled Statistics section. *Startexpfactor* - Starting expansion factor. This field is mandatory if **-e** is used. *Stopexpfactor* - Stop expansion factor. If not specified, the modeled statistics are generated for the start expansion factor alone. *incexpfactor* - Incremental expansion factor. Allowed range is 0.01-1.0. Default is 0.5. Stop expansion factor needs to be specified in order to specify the incremental expansion factor. **Note:** The **-e** option cannot be combined with **-C** or **-c** options.
-m *min_mem_gain*	Specifies the Minimum Memory Gain. This value is specified in megabytes. This value is used in determining the various possible expansion factors reported in the modeled statistics and also influences the produced uses.
-n *num_entries*	Specifies the number of entries that need to be displayed in the modeled statistics. **Note:** When the **-e** option is used with incexpfactor, the **-n** value is ignored.
-N	Disable AME modeling (Workload Planning Capability).
-P *recfile*	Process the specified recording file and generate a report.
-R *recfile*	Record the active memory expansion data in the specified recording file. The recorded data can be post-processed later using the **-P** option. **Note:** Only the **-N** option can be combined with **-R**.
-t *tgt_expmem_size*	Specifies the Modeled Target Expanded Memory Size. This causes the tool to use the user-specified size for modeling instead of the calculated one. **Note**: The **-t** and **-a** options are mutually exclusive.

Flag	Description
-u *minuncompressdpoolsize*	Specifies the minimum uncompressed pool size in megabytes. This value overrides the tool-calculated value for producing modeled statistics. **Note:** This flag can be used only when AME is disabled.
-v	Enables verbose logging. When specified, a verbose log file is generated, named as amepat_yyyymmddhmm.log, where yyyymmddhmm represents the time of invocation. **Note:** The verbose log also contains detailed information about various samples collected and hence the file will be larger than the output generated by the tool.
Duration	Duration represents the amount of total time the tool required to monitor the system before generating any reports. **Note:** When duration is specified, interval and samples cannot be specified. The interval and samples are determined by the tool automatically. The actual monitoring time can be higher than the duration specified based on the memory usage and access patterns of the workload.
Interval <Samples>	Interval represents the amount of sampling time. <Samples> represents the number of samples that need to be collected. **Note:** When interval samples are specified, duration is calculated automatically as (interval x Samples). The actual monitoring time can be higher than the duration specified, based on the memory usage and access patterns of the workload.

To display the AME monitoring report, run the `amepat` command without any flags or options, as shown in Example 6-5.

Example 6-5   Displaying the amepat monitoring report

```
# amepat
```

```
Command Invoked              : amepat

Date/Time of invocation      : Mon Aug 30 17:22:00 EDT 2010
Total Monitored time         : NA
Total Samples Collected      : NA

System Configuration:
---------------------
Partition Name               : 75021p01
Processor Implementation Mode : POWER7
Number Of Logical CPUs       : 16
Processor Entitled Capacity  : 1.00
Processor Max. Capacity      : 4.00
True Memory                  : 8.00 GB
SMT Threads                  : 4
Shared Processor Mode        : Enabled-Uncapped
Active Memory Sharing        : Disabled
Active Memory Expansion      : Enabled
Target Expanded Memory Size  : 16.00 GB
Target Memory Expansion factor : 2.00


System Resource Statistics:            Current
---------------------------        ----------------
CPU Util (Phys. Processors)          0.10 [  2%]
Virtual Memory Size (MB)             1697 [ 10%]
True Memory In-Use (MB)              1620 [ 20%]
Pinned Memory (MB)                   1400 [ 17%]
File Cache Size (MB)                   30 [  0%]
Available Memory (MB)               14608 [ 89%]

AME Statistics:                        Current
---------------                    ----------------
AME CPU Usage (Phy. Proc Units)      0.00 [  0%]
Compressed Memory (MB)                203 [  1%]
Compression Ratio                    2.35
Deficit Memory Size (MB)               74 [  0%]
```

In Example 6-6 the `amepat` command monitors the workload on a system for a
duration of 10 minutes with 5 minute sampling intervals and 2 samples.

Example 6-6   Monitoring the workload on a system with amepat for 10 minutes

```
# amepat 5 2
```

```
Command Invoked            : amepat 5 2

Date/Time of invocation    : Mon Aug 30 17:26:20 EDT 2010
Total Monitored time       : 10 mins 48 secs
Total Samples Collected    : 2

System Configuration:
---------------------
Partition Name                 : 7502lp01
Processor Implementation Mode  : POWER7
Number Of Logical CPUs         : 16
Processor Entitled Capacity    : 1.00
Processor Max. Capacity        : 4.00
True Memory                    : 8.00 GB
SMT Threads                    : 4
Shared Processor Mode          : Enabled-Uncapped
Active Memory Sharing          : Disabled
Active Memory Expansion        : Enabled
Target Expanded Memory Size    : 16.00 GB
Target Memory Expansion factor : 2.00


System Resource Statistics:            Average          Min             Max
---------------------------           -----------     -----------     -----------
CPU Util (Phys. Processors)           2.39 [ 60%]     1.94 [ 48%]     2.84 [ 71%]
Virtual Memory Size (MB)              1704 [ 10%]     1703 [ 10%]     1706 [ 10%]
True Memory In-Use (MB)               1589 [ 19%]     1589 [ 19%]     1590 [ 19%]
Pinned Memory (MB)                    1411 [ 17%]     1405 [ 17%]     1418 [ 17%]
File Cache Size (MB)                     10 [  0%]       10 [  0%]       11 [  0%]
Available Memory (MB)                14057 [ 86%]    13994 [ 85%]    14121 [ 86%]

AME Statistics:                        Average          Min             Max
---------------                       -----------     -----------     -----------
AME CPU Usage (Phy. Proc Units)       0.11 [  3%]     0.02 [  1%]     0.21 [  5%]
Compressed Memory (MB)                 234 [  1%]      230 [  1%]      238 [  1%]
Compression Ratio                     2.25            2.25            2.26
Deficit Memory Size (MB)               701 [  4%]      701 [  4%]      702 [  4%]

Active Memory Expansion Modeled Statistics        :
-------------------------------------------
Modeled Expanded Memory Size :  16.00 GB
Achievable Compression ratio :2.25

Expansion    Modeled True    Modeled            CPU Usage
Factor       Memory Size     Memory Gain        Estimate
---------    -------------   ------------------ -----------
    1.02         15.75 GB     256.00 MB [  2%]  0.00 [  0%]
    1.17         13.75 GB       2.25 GB [ 16%]  0.00 [  0%]
```

```
        1.31          12.25 GB          3.75 GB [ 31%]   0.00 [  0%]
        1.46          11.00 GB          5.00 GB [ 45%]   0.75 [ 19%]
        1.60          10.00 GB          6.00 GB [ 60%]   1.54 [ 39%]
        1.73           9.25 GB          6.75 GB [ 73%]   2.14 [ 53%]
        1.89           8.50 GB          7.50 GB [ 88%]   2.73 [ 68%]

Active Memory Expansion Recommendation:
---------------------------------------
WARNING: This LPAR currently has a memory deficit of 701 MB.
A memory deficit is caused by a memory expansion factor that is too
high for the current workload.  It is recommended that you reconfigure
the LPAR to eliminate this memory deficit.  Reconfiguring the LPAR
with one of the recommended configurations in the above table should
eliminate this memory deficit.

The recommended AME configuration for this workload is to configure the LPAR
with a memory size of 12.25 GB and to configure a memory expansion factor
of 1.31.  This will result in a memory gain of 31%. With this
configuration, the estimated CPU usage due to AME is approximately 0.00
physical processors, and the estimated overall peak CPU resource required for
the LPAR is 2.64 physical processors.

NOTE: amepat's recommendations are based on the workload's utilization level
during the monitored period. If there is a change in the workload's utilization
level or a change in workload itself, amepat should be run again.

The modeled Active Memory Expansion CPU usage reported by amepat is just an
estimate.  The actual CPU usage used for Active Memory Expansion may be lower
or higher depending on the workload.
```

To cap AME processor usage to 30%, when capturing Workload Planning data for 5 minutes, you would enter the command shown in Example 6-7.

Example 6-7   Capping AME processor usage to 30%

```
# amepat -c 30 5

Command Invoked              : amepat -c 30 5

Date/Time of invocation      : Mon Aug 30 17:43:28 EDT 2010
Total Monitored time         : 6 mins 7 secs
Total Samples Collected      : 3

System Configuration:
---------------------
Partition Name               : 7502lp01
Processor Implementation Mode : POWER7
Number Of Logical CPUs       : 16
```

```
Processor Entitled Capacity   : 1.00
Processor Max. Capacity       : 4.00
True Memory                   : 8.00 GB
SMT Threads                   : 4
Shared Processor Mode         : Enabled-Uncapped
Active Memory Sharing         : Disabled
Active Memory Expansion       : Enabled
Target Expanded Memory Size   : 16.00 GB
Target Memory Expansion factor : 2.00


System Resource Statistics:              Average          Min              Max
---------------------------            -----------      -----------      -----------
CPU Util (Phys. Processors)            0.02 [  0%]      0.01 [  0%]      0.02 [  1%]
Virtual Memory Size (MB)               1780 [ 11%]      1780 [ 11%]      1781 [ 11%]
True Memory In-Use (MB)                1799 [ 22%]      1796 [ 22%]      1801 [ 22%]
Pinned Memory (MB)                     1448 [ 18%]      1448 [ 18%]      1448 [ 18%]
File Cache Size (MB)                      83 [  1%]        83 [  1%]        84 [  1%]
Available Memory (MB)                 14405 [ 88%]     14405 [ 88%]     14407 [ 88%]

AME Statistics:                          Average          Min              Max
---------------                        -----------      -----------      -----------
AME CPU Usage (Phy. Proc Units)        0.00 [  0%]      0.00 [  0%]      0.00 [  0%]
Compressed Memory (MB)                  198 [  1%]       198 [  1%]       199 [  1%]
Compression Ratio                      2.35             2.35             2.36
Deficit Memory Size (MB)                116 [  1%]       116 [  1%]       116 [  1%]

Active Memory Expansion Modeled Statistics        :
-------------------------------------------
Modeled Expanded Memory Size :  16.00 GB
Achievable Compression ratio   :2.35

Expansion    Modeled True     Modeled            CPU Usage
Factor       Memory Size      Memory Gain        Estimate
---------    -------------    -----------------  -----------
    1.02         15.75 GB      256.00 MB [  2%]  0.00 [  0%]
    1.17         13.75 GB        2.25 GB [ 16%]  0.00 [  0%]
    1.34         12.00 GB        4.00 GB [ 33%]  0.00 [  0%]
    1.49         10.75 GB        5.25 GB [ 49%]  0.00 [  0%]
    1.65          9.75 GB        6.25 GB [ 64%]  0.00 [  0%]
    1.78          9.00 GB        7.00 GB [ 78%]  0.00 [  0%]
    1.94          8.25 GB        7.75 GB [ 94%]  0.00 [  0%]

Active Memory Expansion Recommendation:
---------------------------------------
WARNING: This LPAR currently has a memory deficit of 116 MB.
A memory deficit is caused by a memory expansion factor that is too
high for the current workload.  It is recommended that you reconfigure
the LPAR to eliminate this memory deficit.  Reconfiguring the LPAR
```

with one of the recommended configurations in the above table should
eliminate this memory deficit.

The recommended AME configuration for this workload is to configure the LPAR
with a memory size of 8.25 GB and to configure a memory expansion factor
of 1.94. This will result in a memory gain of 94%. With this
configuration, the estimated CPU usage due to AME is approximately 0.00
physical processors, and the estimated overall peak CPU resource required for
the LPAR is 0.02 physical processors.

NOTE: amepat's recommendations are based on the workload's utilization level
during the monitored period. If there is a change in the workload's utilization
level or a change in workload itself, amepat should be run again.

The modeled Active Memory Expansion CPU usage reported by amepat is just an
estimate. The actual CPU usage used for Active Memory Expansion may be lower
or higher depending on the workload.

To start modeling a memory gain of 1000 MB for a duration of 5 minutes and
generate an AME Workload Planning report, you would enter the command
shown in Example 6-8.

Example 6-8   AME modeling memory gain of 1000 MB

```
# amepat -m 1000 5

Command Invoked             : amepat -m 1000 5

Date/Time of invocation     : Mon Aug 30 18:42:46 EDT 2010
Total Monitored time        : 6 mins 9 secs
Total Samples Collected     : 3

System Configuration:
---------------------
Partition Name              : 7502lp01
Processor Implementation Mode : POWER7
Number Of Logical CPUs      : 16
Processor Entitled Capacity : 1.00
Processor Max. Capacity     : 4.00
True Memory                 : 8.00 GB
SMT Threads                 : 4
Shared Processor Mode       : Enabled-Uncapped
Active Memory Sharing       : Disabled
Active Memory Expansion     : Enabled
Target Expanded Memory Size : 16.00 GB
Target Memory Expansion factor : 2.00
```

```
System Resource Statistics:                  Average           Min              Max
---------------------------                  -----------       -----------      -----------
CPU Util (Phys. Processors)                  0.02 [  0%]       0.01 [  0%]      0.02 [  1%]
Virtual Memory Size (MB)                     1659 [ 10%]       1658 [ 10%]      1661 [ 10%]
True Memory In-Use (MB)                       1862 [ 23%]       1861 [ 23%]      1864 [ 23%]
Pinned Memory (MB)                            1362 [ 17%]       1362 [ 17%]      1363 [ 17%]
File Cache Size (MB)                           163 [  2%]        163 [  2%]       163 [  2%]
Available Memory (MB)                        14538 [ 89%]      14536 [ 89%]     14539 [ 89%]

AME Statistics:                              Average           Min              Max
---------------                              -----------       -----------      -----------
AME CPU Usage (Phy. Proc Units)              0.00 [  0%]       0.00 [  0%]      0.00 [  0%]
Compressed Memory (MB)                          0 [  0%]          0 [  0%]         0 [  0%]
Compression Ratio                            N/A

Active Memory Expansion Modeled Statistics           :
-------------------------------------------
Modeled Expanded Memory Size :   16.00 GB
Achievable Compression ratio    :0.00


Active Memory Expansion Recommendation:
---------------------------------------
The amount of compressible memory for this workload is small. Only
1.81% of the current memory size is compressible.  This tool analyzes
compressible memory in order to make recommendations.  Due to the small
amount of compressible memory, this tool cannot make a recommendation
for the current workload.

This small amount of compressible memory is likely due to the large amount
of free memory.  38.63% of memory is free (unused).  This may
indicate the load was very light when this tool was run.  If so, please
increase the load and run this tool again.
```

To start modeling a minimum uncompressed pool size of 2000 MB for a duration of 5 minutes and generate an AME Workload Planning report, you would enter the command shown in Example 6-9.

Note: This command can only be run on a system with AME disabled. If you attempt to run it on an AME-enabled system, you will see the following message: `amepat: -u option is not allowed when AME is ON`.

Example 6-9   Modeling a minimum uncompressed pool size of 2000 MB

```
# amepat -u 2000 5

Command Invoked             : amepat -u 2000 5
```

```
Date/Time of invocation      : Mon Aug 30 18:51:46 EDT 2010
Total Monitored time         : 6 mins 8 secs
Total Samples Collected      : 3

System Configuration:
---------------------
Partition Name               : 7502lp02
Processor Implementation Mode : POWER7
Number Of Logical CPUs       : 16
Processor Entitled Capacity  : 1.00
Processor Max. Capacity      : 4.00
True Memory                  : 8.00 GB
SMT Threads                  : 4
Shared Processor Mode        : Enabled-Uncapped
Active Memory Sharing        : Disabled
Active Memory Expansion      : Disabled

System Resource Statistics:               Average          Min              Max
---------------------------            -----------      -----------      -----------
CPU Util (Phys. Processors)            0.01 [  0%]      0.01 [  0%]      0.02 [  0%]
Virtual Memory Size (MB)               1756 [ 21%]      1756 [ 21%]      1756 [ 21%]
True Memory In-Use (MB)                1949 [ 24%]      1949 [ 24%]      1949 [ 24%]
Pinned Memory (MB)                     1446 [ 18%]      1446 [ 18%]      1446 [ 18%]
File Cache Size (MB)                     178 [  2%]       178 [  2%]       178 [  2%]
Available Memory (MB)                   6227 [ 76%]      6227 [ 76%]      6227 [ 76%]

Active Memory Expansion Modeled Statistics      :
-------------------------------------------
Modeled Expanded Memory Size :   8.00 GB
Achievable Compression ratio   :2.13

Expansion    Modeled True      Modeled            CPU Usage
Factor       Memory Size       Memory Gain        Estimate
---------    -------------    ------------------  -----------
    1.00          8.00 GB        0.00 KB [  0%]   0.00 [  0%]
    1.07          7.50 GB      512.00 MB [  7%]   0.00 [  0%]
    1.15          7.00 GB        1.00 GB [ 14%]   0.00 [  0%]
    1.19          6.75 GB        1.25 GB [ 19%]   0.00 [  0%]
    1.28          6.25 GB        1.75 GB [ 28%]   0.00 [  0%]
    1.34          6.00 GB        2.00 GB [ 33%]   0.00 [  0%]
    1.40          5.75 GB        2.25 GB [ 39%]   0.00 [  0%]

Active Memory Expansion Recommendation:
---------------------------------------
The recommended AME configuration for this workload is to configure the LPAR
with a memory size of 5.75 GB and to configure a memory expansion factor
of 1.40.  This will result in a memory gain of 39%. With this
configuration, the estimated CPU usage due to AME is approximately 0.00
```

```
physical processors, and the estimated overall peak CPU resource required for
the LPAR is 0.02 physical processors.

NOTE: amepat's recommendations are based on the workload's utilization level
during the monitored period. If there is a change in the workload's utilization
level or a change in workload itself, amepat should be run again.

The modeled Active Memory Expansion CPU usage reported by amepat is just an
estimate.  The actual CPU usage used for Active Memory Expansion may be lower
or higher depending on the workload.
```

To use the **amepat** recording mode to generate a recording file and report, you would enter the command shown in Example 6-10 (this will start recording for a duration of 60 minutes).

> **Note:** This will invoke the tool as a background process.

Example 6-10   Starting amepat in recording mode

```
# amepat -R /tmp/myrecord_amepat 60
Continuing Recording through background process...

# ps -ef | grep amepat
    root  5898374 12976300    0 11:14:36  pts/0  0:00 grep amepat
    root 20119654         1    0 10:42:14  pts/0  0:21 amepat -R /tmp/myrecord_amepat 60

# ls -ltr /tmp/myrecord_amepat
total 208
-rw-r--r--    1 root      system         22706 Aug 31 11:13 myrecord_amepat
```

In Example 6-11 the **amepat** command will generate a report, for workload planning purposes, using a previously generated recording file.

Example 6-11   Generating an amepat report using an existing recording file

```
# amepat -P /tmp/myrecord_amepat

Command Invoked              : amepat -P /tmp/myrecord_amepat

Date/Time of invocation      : Mon Aug 30 18:59:25 EDT 2010
Total Monitored time         : 1 hrs 3 mins 23 secs
Total Samples Collected      : 9

System Configuration:
---------------------
Partition Name               : 7502lp01
Processor Implementation Mode : POWER7
```

```
Number Of Logical CPUs        : 16
Processor Entitled Capacity   : 1.00
Processor Max. Capacity       : 4.00
True Memory                   : 8.00 GB
SMT Threads                   : 4
Shared Processor Mode         : Enabled-Uncapped
Active Memory Sharing         : Disabled
Active Memory Expansion       : Enabled
Target Expanded Memory Size   : 16.00 GB
Target Memory Expansion factor : 2.00


System Resource Statistics:            Average          Min              Max
---------------------------          -----------      -----------      -----------
CPU Util (Phys. Processors)          0.01 [  0%]      0.01 [  0%]      0.01 [  0%]
Virtual Memory Size (MB)             1653 [ 10%]      1653 [ 10%]      1656 [ 10%]
True Memory In-Use (MB)              1856 [ 23%]      1856 [ 23%]      1859 [ 23%]
Pinned Memory (MB)                   1362 [ 17%]      1362 [ 17%]      1362 [ 17%]
File Cache Size (MB)                   163 [  2%]       163 [  2%]       163 [  2%]
Available Memory (MB)               14542 [ 89%]     14541 [ 89%]     14543 [ 89%]

AME Statistics:                        Average          Min              Max
---------------                      -----------      -----------      -----------
AME CPU Usage (Phy. Proc Units)      0.00 [  0%]      0.00 [  0%]      0.00 [  0%]
Compressed Memory (MB)                  0 [  0%]         0 [  0%]         0 [  0%]
Compression Ratio                    N/A

Active Memory Expansion Modeled Statistics      :
-------------------------------------------
Modeled Expanded Memory Size :  16.00 GB
Achievable Compression ratio    :0.00


Active Memory Expansion Recommendation:
----------------------------------------
The amount of compressible memory for this workload is small. Only
1.78% of the current memory size is compressible.  This tool analyzes
compressible memory in order to make recommendations.  Due to the small
amount of compressible memory, this tool cannot make a recommendation
for the current workload.

This small amount of compressible memory is likely due to the large amount
of free memory.  38.66% of memory is free (unused).  This may
indicate the load was very light when this tool was run.  If so, please
increase the load and run this tool again.
```

Example 6-12 generates a report for workload planning, with the modeled memory expansion factors ranging between 2 to 4 with a 0.5 delta factor.

Example 6-12   Modeled expansion factor report from a recorded file

```
# amepat -e 2.0:4.0:0.5 -P /tmp/myrecord_amepat

Command Invoked            : amepat -e 2.0:4.0:0.5 -P /tmp/myrecord_amepat

Date/Time of invocation    : Mon Aug 30 18:59:25 EDT 2010
Total Monitored time       : 1 hrs 3 mins 23 secs
Total Samples Collected    : 9

System Configuration:
---------------------
Partition Name                : 7502lp01
Processor Implementation Mode : POWER7
Number Of Logical CPUs        : 16
Processor Entitled Capacity   : 1.00
Processor Max. Capacity       : 4.00
True Memory                   : 8.00 GB
SMT Threads                   : 4
Shared Processor Mode         : Enabled-Uncapped
Active Memory Sharing         : Disabled
Active Memory Expansion       : Enabled
Target Expanded Memory Size   : 16.00 GB
Target Memory Expansion factor : 2.00

System Resource Statistics:              Average            Min            Max
---------------------------            -----------    -----------    -----------
CPU Util (Phys. Processors)            0.01 [  0%]    0.01 [  0%]    0.01 [  0%]
Virtual Memory Size (MB)               1653 [ 10%]    1653 [ 10%]    1656 [ 10%]
True Memory In-Use (MB)                1856 [ 23%]    1856 [ 23%]    1859 [ 23%]
Pinned Memory (MB)                     1362 [ 17%]    1362 [ 17%]    1362 [ 17%]
File Cache Size (MB)                     163 [  2%]     163 [  2%]     163 [  2%]
Available Memory (MB)                 14542 [ 89%]   14541 [ 89%]   14543 [ 89%]

AME Statistics:                          Average            Min            Max
---------------                        -----------    -----------    -----------
AME CPU Usage (Phy. Proc Units)        0.00 [  0%]    0.00 [  0%]    0.00 [  0%]
Compressed Memory (MB)                    0 [  0%]       0 [  0%]       0 [  0%]
Compression Ratio                      N/A

Active Memory Expansion Modeled Statistics       :
-------------------------------------------
Modeled Expanded Memory Size :  16.00 GB
Achievable Compression ratio   :0.00


Active Memory Expansion Recommendation:
---------------------------------------
The amount of compressible memory for this workload is small. Only
```

1.78% of the current memory size is compressible. This tool analyzes
compressible memory in order to make recommendations. Due to the small
amount of compressible memory, this tool cannot make a recommendation
for the current workload.

This small amount of compressible memory is likely due to the large amount
of free memory. 38.66% of memory is free (unused). This may
indicate the load was very light when this tool was run. If so, please
increase the load and run this tool again.

To generate an AME monitoring only report from a previously recorded file, you
would enter the command shown in Example 6-13.

Example 6-13   AME monitoring report from a recorded file

```
# amepat -N -P /tmp/myrecord_amepat
WARNING: Running in no modeling mode.


Command Invoked                    : amepat -N -P /tmp/myrecord_amepat

Date/Time of invocation            : Mon Aug 30 18:59:25 EDT 2010
Total Monitored time               : 1 hrs 3 mins 23 secs
Total Samples Collected            : 9

System Configuration:
---------------------
Partition Name                     : 7502lp01
Processor Implementation Mode  : POWER7
Number Of Logical CPUs             : 16
Processor Entitled Capacity    : 1.00
Processor Max. Capacity            : 4.00
True Memory                        : 8.00 GB
SMT Threads                        : 4
Shared Processor Mode              : Enabled-Uncapped
Active Memory Sharing              : Disabled
Active Memory Expansion            : Enabled
Target Expanded Memory Size    : 16.00 GB
Target Memory Expansion factor : 2.00


System Resource Statistics:                  Average          Min            Max
---------------------------              -----------     -----------     -----------
CPU Util (Phys. Processors)              0.01 [  0%]     0.01 [  0%]     0.01 [  0%]
Virtual Memory Size (MB)                 1653 [ 10%]     1653 [ 10%]     1656 [ 10%]
True Memory In-Use (MB)                  1856 [ 23%]     1856 [ 23%]     1859 [ 23%]
Pinned Memory (MB)                       1362 [ 17%]     1362 [ 17%]     1362 [ 17%]
File Cache Size (MB)                       163 [  2%]      163 [  2%]      163 [  2%]
Available Memory (MB)                   14542 [ 89%]    14541 [ 89%]    14543 [ 89%]
```

```
AME Statistics:                       Average          Min            Max
---------------                     -----------    -----------    -----------
AME CPU Usage (Phy. Proc Units)     0.00 [  0%]    0.00 [  0%]    0.00 [  0%]
Compressed Memory (MB)                 0 [  0%]       0 [  0%]       0 [  0%]
Compression Ratio                   N/A
```

Example 6-14 will disable the Workload Planning capability and only monitor system utilization for 5 minutes.

Example 6-14   Disable workload planning and only monitor system utilization

```
# amepat -N 5
WARNING: Running in no modeling mode.


Command Invoked           : amepat -N 5

Date/Time of invocation   : Tue Aug 31 11:20:41 EDT 2010
Total Monitored time      : 6 mins 0 secs
Total Samples Collected   : 3

System Configuration:
---------------------
Partition Name                : 7502lp01
Processor Implementation Mode : POWER7
Number Of Logical CPUs        : 16
Processor Entitled Capacity   : 1.00
Processor Max. Capacity       : 4.00
True Memory                   : 8.00 GB
SMT Threads                   : 4
Shared Processor Mode         : Enabled-Uncapped
Active Memory Sharing         : Disabled
Active Memory Expansion       : Enabled
Target Expanded Memory Size   : 16.00 GB
Target Memory Expansion factor : 2.00


System Resource Statistics:            Average          Min            Max
---------------------------          -----------    -----------    -----------
CPU Util (Phys. Processors)          0.01 [  0%]    0.01 [  0%]    0.01 [  0%]
Virtual Memory Size (MB)             1759 [ 11%]    1759 [ 11%]    1759 [ 11%]
True Memory In-Use (MB)              1656 [ 20%]    1654 [ 20%]    1657 [ 20%]
Pinned Memory (MB)                   1461 [ 18%]    1461 [ 18%]    1461 [ 18%]
File Cache Size (MB)                     9 [  0%]       9 [  0%]      10 [  0%]
Available Memory (MB)               14092 [ 86%]   14092 [ 86%]   14094 [ 86%]

AME Statistics:                        Average          Min            Max
---------------                      -----------    -----------    -----------
AME CPU Usage (Phy. Proc Units)      0.00 [  0%]    0.00 [  0%]    0.00 [  0%]
```

```
Compressed Memory (MB)                                          220 [  1%]        220 [  1%]        221 [  1%]
Compression Ratio                                               2.27              2.27              2.28
Deficit Memory Size (MB)                                        550 [  3%]        550 [  3%]        550 [  3%]
```

6.1.2 Enhanced AIX performance monitoring tools for AME

Several AIX performance tools can be used to monitor AME statistics and gather information relating to AME. The following AIX tools have been enhanced to support AME monitoring and reporting:

► vmstat

► lparstat

► topas

► topas_nmon

► svmon

The additional options for each tool are summarized in Table 6-6.

Table 6-6   AIX performance tool enhancements for AME

Tool	Option	Description
vmstat	-c	Provides compression, decompression, and deficit statistics.
lparstat	-c	Provides an indication of the processor utilization for AME compression and decompression activity. Also provides memory deficit information.
svmon	-O summary=ame	Provides a summary view of memory usage broken down into compressed and uncompressed pages.
topas		The default topas panel displays the memory compression statistics when it is run in the AME environment.

The `vmstat` command can be used with the `-c` flag to display AME statistics, as shown in Example 6-15.

Example 6-15   Using vmstat to display AME statistics

```
# vmstat -wc 1 5

System configuration: lcpu=16 mem=16384MB tmem=8192MB ent=1.00 mmode=dedicated-E

 kthr                      memory                             page                  faults            cpu
------- --------------------------------------------------- --------------------- ----------------- -----------------------
```

```
  r  b      avm       fre       csz       cfr       dxm  ci  co  pi  po  in   sy   cs us sy id wa   pc    ec
 51  0  1287384   2854257     35650     13550     61379   0   0   0   0   3  470 1712 99  0  0  0  3.99 399.4
 53  0  1287384   2854264     35650     13567     61379  30   0   0   0   2   45 1721 99  0  0  0  3.99 399.2
 51  0  1287384   2854264     35650     13567     61379   0   0   0   0   1   40 1712 99  0  0  0  3.99 399.2
  0  0  1287384   2854264     35650     13567     61379   0   0   0   0   3   45 1713 99  0  0  0  3.99 399.2
 51  0  1287384   2854264     35650     13567     61379   0   0   0   0   2   38 1716 99  0  0  0  3.99 399.2
```

In the output from Example 6-15, the following memory compression statistics are provided:

- ► Expanded memory size (mem) of the LPAR is 16384 MB.

- ► True memory size (tmem) is 8192 MB.

- ► The memory mode (mmode) of the LPAR is AME enabled, Dedicated-Expanded.

- ► Compressed Pool size (csz) is 35650 4 KB pages.

- ► Amount of free memory (cfr) in the compressed pool is 13567 4 KB pages.

- ► Size of the expanded memory deficit (dxm) is 61379 4 KB pages.

- ► Number of compression operations or page-outs to the compressed pool per second (co) is 0.

- ► Number of decompression operations or page-ins from the compressed pool per second (ci) is 0.

The **lparstat** command can be used, with the **-c** flag, to display AME statistics, as shown in Example 6-16.

Example 6-16   Using lparstat to display AME statistics

```
# lparstat -c 1 5

System configuration: type=Shared mode=Uncapped mmode=Ded-E smt=4 lcpu=16 mem=16384MB
tmem=8192MB psize=16 ent=1.00

%user  %sys  %wait  %idle physc %entc  lbusy  vcsw phint  %xcpu xphysc    dxm
-----  ----- ------ ------ ----- -----  ------ ----- ----- ------ ------ ------
 91.9   8.1    0.0    0.0  3.99 399.3  100.0  1600    1    9.7 0.3861   2417
 89.1  10.9    0.0    0.0  3.99 398.7  100.0  1585    0   15.0 0.5965   2418
 85.5  14.5    0.0    0.0  3.99 399.2  100.0  1599    4   16.9 0.6736   2418
 87.6  12.4    0.0    0.0  3.99 399.2  100.0  1600   16   16.7 0.6664   2418
 82.7  17.3    0.0    0.0  3.99 399.4  100.0  1615   12   17.3 0.6908    742
```

In the output in Example 6-16, the following memory compression statistics are provided:

- ► Memory mode (mmode) of the LPAR is AME enabled, Dedicated-Expanded.

- ► Expanded memory size (mem) of the LPAR is 16384 MB.

- ▶ True memory size (tmem) of the LPAR is 8192 MB.

- ▶ Percentage of processor utilized for AME activity (%xcpu) is 17.3.

- ▶ Size of expanded memory deficit (dxm) in megabytes is 742.

Example 6-17 displays output from **lparstat -i** showing configuration information relating to LPAR memory mode and AME settings.

Example 6-17   Using lparstat to view AME configuration details

```
# lparstat -i | grep -i memory | grep -i ex
Memory Mode                          : Dedicated-Expanded
Target Memory Expansion Factor       : 2.00
Target Memory Expansion Size         : 16384 MB
```

The LPAR's memory mode is Dedicated-Expanded, the target memory expansion factor is 2.0 and the target memory expansion size is 16384 MB.

The main panel of the **topas** command has been modified to display AME compression statistics. The data is displayed under an additional subsection called AME, as shown in Example 6-18.

Example 6-18   Additional topas subsection for AME

```
Topas Monitor for host:750_2_LP01              EVENTS/QUEUES    FILE/TTY
Tue Aug 31 11:04:22 2010    Interval:FROZEN    Cswitch    210   Readch      361
                                               Syscall    120   Writech     697
CPU     User% Kern% Wait% Idle%   Physc  Entc% Reads        0   Rawin         0
Total    0.0   0.7   0.0  99.3     0.01   1.26 Writes       0   Ttyout      335
                                               Forks        0   Igets         0
Network   BPS  I-Pkts  O-Pkts    B-In   B-Out  Execs        0   Namei         1
Total   462.0   0.50    1.00    46.00   416.0  Runqueue     0   Dirblk        0
                                               Waitqueue  0.0
Disk    Busy%     BPS     TPS  B-Read  B-Writ                   MEMORY
Total    0.0        0       0       0       0   PAGING         Real,MB   16384
                                               Faults       0   % Comp       14
FileSystem        BPS     TPS  B-Read  B-Writ  Steals       0   % Noncomp     0
Total           336.0    0.50   336.0       0   PgspIn       0   % Client      0
                                               PgspOut      0
Name          PID  CPU%  PgSp Owner           PageIn       0   PAGING SPACE
vmmd        393228  0.3  188K root            PageOut      0   Size,MB     512
xmgc        851994  0.2 60.0K root            Sios         0   % Used        3
topas     18939976  0.1 2.35M root                            % Free       97
getty       6160394  0.0  580K root           AME
java        6095084  0.0 48.8M pconsole       TMEM,MB     8192 WPAR Activ    0
gil         1966140  0.0  124K root           CMEM,MB   139.82 WPAR Total    1
```

```
sshd        6619376   0.0 1.18M root          EF[T/A] 2.00/1.04 Press: "h"-help
clcomd      5767348   0.0 1.75M root          CI:    0.0 CO: 0.0 "q"-quit
java        5177386   0.0 73.7M root
rpc.lock    5243052   0.0  204K root
rmcd        5832906   0.0 6.54M root
netm        1900602   0.0 60.0K root
cmemd        655380   0.0  180K root
lrud         262152   0.0 92.0K root
topasrec    5701812   0.0 1.07M root
amepat     20119654   0.0  328K root
syncd       2949304   0.0  604K root
random      3670206   0.0 60.0K root
j2pg        2424912   0.0 1.17M root
lvmbb       2490464   0.0 60.0K root
```

In Example 6-18 on page 245, the following memory compression statistics are provided from the `topas` command:

► True memory size (`TMEM,MB`) of the LPAR is 8192 MB.

► Compressed pool size (`CMEM,MB`) is 139.82 MB.

► `EF[T/A]` - The Target Expansion Factor is 2.00 and the Achieved Expansion Factor is 1.04.

► Rate of compressions (`co`) and decompressions (`ci`) per second are 0.0 and 0.0 pages, respectively.

The `topas` command CEC view has been enhanced to report AME status across all of the LPARs on a server. The memory mode for an LPAR is displayed in the CEC view. The possible memory modes shown by the `topas -C` command are shown in Table 6-7.

Table 6-7   topas -C memory mode values for an LPAR

Value	Description
M	In shared memory mode (shared LPARs only), and AME is disabled
-	Not in shared memory mode, and AME is disabled
E	In shared memory mode and, AME is enabled
e	Not in shared memory mode, and AME is enabled.

Example 6-19 provides output from the **topas -C** command for a system with six AME-enabled LPARs.

Example 6-19   topas CEC view with AME-enabled LPARs

```
Topas CEC Monitor                Interval:  10              Thu Sep 16 10:19:22 2010
Partitions Memory (GB)           Processors
Shr:  6    Mon:46.0  InUse:18.0  Shr:4.3  PSz: 16   Don: 0.0 Shr_PhysB  0.65
Ded:  0    Avl:   -              Ded:  0  APP: 15.3 Stl: 0.0 Ded_PhysB  0.00


Host       OS  Mod Mem InU Lp  Us Sy Wa Id  PhysB  Vcsw Ent  %EntC PhI  pmem
-----------------------------------shared-----------------------------------
75021p03   A71 Ued 8.0 3.2 16   8 27  0 64  0.57     0 1.00  56.5  0    -
75021p01   A71 Ued  16 8.0 16   0  1  0 98  0.02   286 1.00   2.4  0    -
75021p06   A71 Ued 2.0 2.0  8   0  5  0 94  0.02   336 0.20  10.6  1    -
75021p05   A71 Ued 4.0 1.0  4   0  7  0 92  0.02     0 0.10  16.9  0    -
75021p04   A71 Ued 8.0 2.2 16   0  0  0 99  0.02     0 1.00   1.5  0    -
75021p02   A71 Ued 8.0 1.7 16   0  0  0 99  0.01   276 1.00   1.2  0    -
```

The second character under the mode column (Mod) for each LPAR is e, which indicates Active Memory Sharing is disabled and AME is enabled.

The **topas_nmon** command supports AME statistics reporting in the nmon recording file. The MEM tag reports the size of the compressed pool in MB, the size of true memory in MB, the expanded memory size in MB and the size of the uncompressed pool in MB. The MEMNEW tag shows the compressed pool percentage. The PAGE tag displays the compressed pool page-ins and the compressed pool page-outs.

The **svmon** command can provide a detailed view of AME usage on an LPAR, as shown in Example 6-20.

Example 6-20   AME statistics displayed using the svmon command

```
# svmon -G -O summary=ame,pgsz=on,unit=MB
Unit: MB
--------------------------------------------------------------------------------
            size       inuse        free         pin     virtual  available  mmode
memory   16384.00     1725.00    14114.61     1453.91     1752.57   14107.11  Ded-E
 ucomprsd      -       1497.54        -
 comprsd       -        227.46        -
pg space   512.00        14.4

            work        pers        clnt        other
pin        937.25          0           0       516.66
in use    1715.52          0        9.47
 ucomprsd 1488.07
```

```
 comprsd    227.46
--------------------------------------------------------------------------------------
True Memory: 8192.00

             CurSz      %Cur      TgtSz      %Tgt      MaxSz      %Max     CRatio
ucomprsd   8052.18     98.29    1531.84     18.70         -         -          -
comprsd     139.82      1.71    6660.16     81.30    6213.15     75.84       2.28

               txf       cxf        dxf        dxm
AME           2.00      1.93       0.07     549.83
```

The following memory compression statistics are displayed from the **svmon** command in Example 6-20:

► Memory mode (mmode) of the LPAR is AME-enabled, Dedicated-Expanded.

► Out of a total of 1725.00 MB in use, uncompressed pages (ucomprsd) constitute 1497.54 MB and compressed pages (comprsd) constitute the remaining 227.46 MB.

► Out of a total of 1715.52 MB of working pages in use, uncompressed pages (ucomprsd) constitute 1488.07 MB and compressed pages (comprsd) constitute 227.46 MB.

► Expanded memory size (memory) of the LPAR is 16384 MB.

► True memory size (True Memory) of the LPAR is 8192 MB.

► Current size of the uncompressed pool (ucomprsd CurSz) is 8052.18 MB (98.29% of the total true memory size of the LPAR, %Cur).

► Current size of the compressed pool (comprsd CurSz) is 139.82 MB (1.71% of the total true memory size of the LPAR, %Cur).

► The target size of the compressed pool (comprsd TgtSz) required to achieve the target memory expansion factor (txf) of 2.00 is 1531.84 MB (18.70% of the total true memory size of the LPAR, %Tgt).

► The size of the uncompressed pool (ucomprsd TgtSz) in that case becomes 6660.16 MB (81.30% of the total true memory size of the LPAR, %Tgt).

► The maximum size of the compressed pool (comprsd MaxSz) is 6213.15 MB (75.84% of the total true memory size of the LPAR, %Max).

► The current compression ratio (CRatio) is 2.28 and the current expansion factor (cxf) is 1.93.

► The amount of expanded memory deficit (dxm) is 549.83 MB and the deficit expansion factor (dxf) is 0.07

The -O summary=longname option provides a summary of memory compression details, from the **svmon** command, as shown in Example 6-21.

Example 6-21   Viewing AME summary usage information with svmon

```
# svmon -G -O summary=longame,unit=MB
Unit: MB
-----------------------------------------------------------------------------------------------------------------------
                                                Active Memory Expansion
-----------------------------------------------------------------------------------------------------------------------
        Size      Inuse       Free      DXMSz   UCMInuse    CMInuse        TMSz       TMFr       CPSz       CPFr    txf    cxf     CR
    16384.00    1725.35   14114.02     550.07    1498.47     226.88     8192.00    6553.71     139.82       40.5   2.00   1.93   2.28
```

In the output, the following memory compression statistics are provided:

▶ Out of the total expanded memory size (`Size`) of 16384 MB, 1725.35 MB is in use (`Inuse`) and 14114.02 MB is free (`Free`). The deficit in expanded memory size (`DXMSz`) is 550.07 MB.

▶ Out of the total in use memory (`Inuse`) of 1725.35 MB, uncompressed pages (`UCMInuse`) constitute 1498.47 MB, and the compressed pages (`CMInuse`) constitute the remaining 226.88 MB.

▶ Out of the true memory size (`TMSz`) of 8192 MB, only 6535.71 MB of True Free memory (`TMFr`) is available.

▶ Out of the compressed pool size (`CPSz`) of 139.82 MB, only 40.5 MB of free memory (`CPFr`) is available in the compressed pool.

▶ Whereas the target expansion factor (`txf`) is 2.00, the current expansion factor (`cxf`) achieved is 1.93.

▶ The compression ratio (`CR`) is 2.28.

6.2 Hot Files Detection and filemon

An enhancement to the **filemon** command allows for the detection of *hot* files in a file system. The introduction of flash storage or Solid-State Disk (SSD) has necessitated the need for a method to determine the most active files in a file system. These files can then be located on or relocated to the fastest storage available. The enhancement is available in AIX V7.1, AIX V6.1 with Technology Level 4 and AIX V5.3 with Technology Level 11.

For a file to be considered "hot" it must be one that is read from, or written to frequently, or read from, or written to in large chunks of data. The **filemon** command can assist in determining which files are hot, and produces a report highlighting which files are the best candidates for SSD storage.

Using the **-O** hot option with the **filemon** command, administrators can generate reports that will assist with the placement of data on SSDs. The reports contain

statistics for I/O operations of hot files, logical volumes and physical volumes. This data guides an administrator in determining which files and/or logical volumes are the ideal candidates for migration to SSDs. The *hotness* of a file and/or logical volume is based on the number of read operations, average number of bytes read per read operation, number of read sequences and the average sequence length.

The report generated by the `filemon` command consists of three main sections. The first section contains information relating to the system type, the `filemon` command and the `trace` command. The second section is a summary that displays the total number of read/write operations, the total time taken, the total data read/written and the processor utilization. The third section contains the hot data reports. There are three hot data reports in this section:

► Hot Files Report
► Hot Logical Volumes Report
► Hot Physical Volumes Report

Table 6-8 describes the information collected in the Hot Files Report section.

Table 6-8   Hot Files Report description

Column	Description
Name	The name of the file.
Size	The size of the file. The default unit is MB. The default unit is overridden by the unit specified by the **-0** unit option.
CAP_ACC	The capacity accessed. This is the unique data accessed in the file. The default unit is MB. The default unit is overridden by the unit specified by the **-0** unit option.
IOP/#	The number of I/O operations per unit of data accessed. The unit of data is taken from the **-0** unit option. The default is MB. Other units could be K for KB, M for MB, G for GB and T for TB. For example, 0.000/K, 0.256/M, 256/G, 2560/T.
LV	The name of the logical volume where the file is located. If this information cannot be obtained, a "-" is reported.
#ROP	Total number of read operations for the file.
#WOP	Total number of write operations for the file.
B/ROP	The minimum, average, and maximum number of bytes read per read operation.
B/WOP	The minimum, average, and maximum number of bytes write per read operation.
RTIME	The minimum, average, and maximum time taken per read operation in milliseconds.
WTIME	The minimum, average, and maximum time taken per write operation in milliseconds.

Column	Description
Seqlen	The minimum, average, and maximum length of read sequences.
#Seq	Number of read sequences. A sequence is a string of 4 K pages that are read (paged in) consecutively. The number of read sequences is an indicator of the amount of sequential access.

Table 6-9 describes the information collected in the Hot Logical Volumes Report.

Table 6-9   Hot Logical Volumes Report description

Column	Description
Name	The name of the logical volume.
Size	The size of the logical volume. The default unit is MB. The default unit is overridden by the unit specified by the **-0** unit option.
CAP_ACC	The capacity accessed. This is the unique data accessed in the logical volume. The default unit is MB. The default unit is overridden by the unit specified by the **-0** unit option.
IOP/#	The number of I/O operations per unit of data accessed. The unit of data is taken from the **-0** unit option. The default is MB. Other units could be K for KB, M for MB, G for GB and T for TB. For example, 0.000/K, 0.256/M, 256/G, 2560/T.
#Files	Number of files accessed in this logical volume.
#ROP	Total number of read operations for the logical volume.
#WOP	Total number of write operations for the logical volume.
B/ROP	The minimum, average, and maximum number of bytes read per read operation.
B/WOP	The minimum, average, and maximum number of bytes written per write operation.
RTIME	The minimum, average, and maximum time taken per read operation in milliseconds.
WTIME	The minimum, average, and maximum time taken per write operation in milliseconds.
Seqlen	The minimum, average, and maximum length of read sequences.
#Seq	Number of read sequences. A sequence is a string of 4 K pages that are read (paged in) consecutively. The number of read sequences is an indicator of the amount of sequential access.

Table 6-10 describes the information collected in the Hot Physical Volumes Report.

Table 6-10   Hot Physical Volumes Report description

Column	Description
Name	The name of the physical volume.
Size	The size of the physical volume. The default unit is MB. The default unit is overridden by the unit specified by the **-0** unit option.
CAP_ACC	The capacity accessed. This is the unique data accessed for the physical volume. The default unit is MB. The default unit is overridden by the unit specified by the **-0** unit option.
IOP/#	The number of I/O operations per unit of data accessed. The unit of data is taken from the **-0** unit option. The default is MB. Other units could be K for KB, M for MB, G for GB and T for TB. For example, 0.000/K, 0.256/M, 256/G, 2560/T.
#ROP	Total number of read operations for the physical volume.
#WOP	Total number of write operations for the physical volume.
B/ROP	The minimum, average, and maximum number of bytes read per read operation.
B/WOP	The minimum, average, and maximum number of bytes written per write operation.
RTIME	The minimum, average, and maximum time taken per read operation in milliseconds.
WTIME	The minimum, average, and maximum time taken per write operation in milliseconds.
Seqlen	The minimum, average, and maximum length for read sequences.
#Seq	Number of read sequences. A sequence is a string of 512-byte blocks that are read consecutively. The number of read sequences is an indicator of the amount of sequential access.

Each of the hot reports are also sorted by capacity accessed. The data contained in the hot reports can be customized by specifying different options to the **-0** hot flag, as shown in Table 6-11.

Table 6-11   filemon -O hot flag options

Flag	Description
-O hot=r	Generates reports based on read operations only.
-O hot=w	Generates reports based on write operations only.

If the administrator specifies the **-O** `hot=r` option, then only read operation-based reports are generated. If the administrator specifies the **-O** `hot=w` option, then only write operation-based reports are captured.

The use of the **-O** hot option with the `filemon` command is only supported in automated offline mode. If you attempt to run the command in real-time mode you will receive an error message, as shown in Example 6-22:

Example 6-22   filemon -O hot is not supported in real-time mode

```
# filemon -O hot -o fmon.out
hot option not supported in realtime mode
Usage: filemon [-i file -n file] [-o file] [-d] [-v] [-u] [-O opt [-w][-I count:interval]] [-P] [-T
num] [-@ [WparList | ALL ]] [-r RootString [-A -x "<User Command>"]]
        -i file:  offline filemon - open trace file
        -n file:  offline filemon - open gensyms file
                     **Use gensyms -F to get the gensyms file
        -o file:  open output file (default is stdout)
        -d:       deferred trace (until 'trcon')
        -T num:   set trace kernel buf sz (default 32000 bytes)
        -P:       pin monitor process in memory
        -v:       verbose mode (print extra details)
        -u:       print unnamed file activity via pid
        -O opt:   other monitor-specific options
        -@ wparlist|ALL:
                  output one report per WPAR in the list
        -@:       output additionnal WPAR information
        -A:       Enable Automated Offline Mode
        -x:       Provide the user command to execute in double quotes if you provide argument to
the command
        -r:       Root String for trace and gennames filenames
        -w:       prints the hotness report in wide format(Valid only with -O hot option)
        -I count:interval :      Used to specify multiple snapshots of trace collection (Valid only
with -O hot option)

valid -O options: [[detailed,]lf[=num],vm[=num],lv[=num],pv[=num],pr[=num],th[=num],all[=num]] |
abbreviated | collated | hot[={r|w}]lf=[num],lv=[num],pv=[num],sz=num,unit={KB|MB|GB|TB}
        lf[=num]:  monitor logical file   I/O and display first num records where num > 0
        vm[=num]:  monitor virtual memory  I/O and display first num records where num > 0
        lv[=num]:  monitor logical volume  I/O and display first num records where num > 0
        pv[=num]:  monitor physical volume I/O and display first num records where num > 0
        pr[=num]:  display data process-wise and display first num records where num > 0
        th[=num]:  display data thread-wise and display first num records where num > 0
        all[=num]: short for lf,vm,lv,pv,pr,th and display first num records where num > 0
        detailed: display detailed information other than summary report
        abbreviated: Abbreviated mode (transactions). Supported only in offline mode
        collated: Collated mode (transactions).  Supported only in offline mode
        hot[={r|w}]: Generates hotness report(Not supported in realtime mode)
       sz=num: specifies the size of data accessed to be reported in the hotness report(valid only
with -O hot and in automated offline mode.
                Unit for this value is specified through -O unit option. Default is MB.)
```

```
        unit={KB|MB|GB|TB}: unit for CAP_ACC and Size values in hotness report and unit for value
specified by -O sz option
```

Example 6-23 starts the `filemon` command in automated offline mode with the **-A** and **-x** flags, captures hot file data with the **-O** hot flag, specifies that trace data is stored in fmon (.trc is appended to the file name automatically) with the **-r** flag and writes I/O activity to the fmon.out file with the **-o** flag. A user-specified command is placed after the **-x** flag. The trace is collected until this command completes its work. A typical example of a user command is **sleep 60**.

Example 6-23   Generating filemon hot file report in automated offline mode

```
# filemon -O hot,unit=MB -r fmon -o fmon.out -A -x "sleep 60"
```

The contents of the fmon.out file are displayed in the examples that follow. Only the first few lines of each section of the report are displayed, because the report contains a large amount of data. However, the data shown provides an introduction to the typical detail that is reported.

Example 6-24 shows the information and summary sections of the report.

Example 6-24   Information and summary sections of the hot file report

```
Thu Sep  2 19:32:27 2010
System: AIX 7.1 Node: 7502lp04 Machine: 00F61AB24C00

Filemon Command used: filemon -O hot,unit=MB -A -x sleep 60 -r fmon -o fmon.out
Trace command used: /usr/bin/trace -ad -L 2031364915 -T 1000000 -j
00A,001,002,003,38F,005,006,139,465,102,10C,106,4B0,419,107,101,104,10D,15B,12E,130,1
63,19C,154,3D3,137,1BA,1BE,1BC,10B,AB2,221,232,1C9,2A2,
2A1,222,228,45B,5D8,3C4,3B9,223, -o fmon.trc

Summary Section
---------------
Total monitored time: 60.012 seconds
Cpu utilization:   5.4%
Cpu allocation:  100.0%
Total no. of files monitored: 11
Total No. of I/O Operations: 126 ( Read: 126, write: 0 )
Total bytes transferred: 0.427 MB( Read: 0.427 MB, write: 0.000 MB )
Total IOP per unit: 295/MB
Total time taken for I/O operations(in miliseconds): 0.338 ( Read: 0.338, write:
0.000 )
```

The Hot Files Report section is shown in Example 6-25.

Example 6-25   Hot Files Report

```
Hot Files Report
---------------------------------------------------------------------------------
NAME                                 Size      CAP_ACC    IOP/#       LV
B/ROP                  B/WOP              RTIME                 WTIME
#ROP                   #WOP               Seqlen                #Seq
---------------------------------------------------------------------------------
/unix                                33.437M   0.141M     256/M       /dev/hd2
4096,4096,4096         0,0,0              0.002,0.003,0.008   0.000,0.000,0.000
97                     0                  1,1,1                 97
---------------------------------------------------------------------------------
/etc/security/user                   0.011M    0.012M     256/M       /dev/hd4
4096,4096,4096         0,0,0              0.003,0.004,0.008   0.000,0.000,0.000
5                      0                  1,1,1                 5
---------------------------------------------------------------------------------
/etc/security/group                  0.000M    0.012M     256/M       /dev/hd4
4096,4096,4096         0,0,0              0.001,0.003,0.004   0.000,0.000,0.000
4                      0                  1,1,1                 4
---------------------------------------------------------------------------------
```

The Hot Logical Volume Report is shown in Example 6-26.

Example 6-26   Hot Logical Volume Report

```
Hot Logical Volume Report
---------------------------------------------------------------------------------
NAME                   Size           CAP_ACC        IOP/#           #Files
B/ROP                  B/WOP          RTIME                  WTIME
#ROP                   #WOP           Seqlen                 #Seq
---------------------------------------------------------------------------------
/dev/loglv00           64.000M        0.000M         256/M           0
0,0,0                  8,8,8          0.000,0.000,0.000   0.362,0.362,0.362
0                      1              0,0,0                  0
---------------------------------------------------------------------------------
/dev/hd8               64.000M        0.070M         256/M           0
0,0,0                  8,8,8          0.000,0.000,0.000   3.596,11.490,99.599
0                      25             0,0,0                  0
---------------------------------------------------------------------------------
/dev/hd4               1984.000M      154.812M       256/M           4
0,0,0                  8,8,8          0.000,0.000,0.000   3.962,93.807,141.121
0                      21             0,0,0                  0
---------------------------------------------------------------------------------
```

The Hot Physical Volume Report is shown in Example 6-27.

Example 6-27   Hot Physical Volume Report

```
Hot Physical Volume Report
--------------------------------------------------------------------------------
NAME                     Size         CAP_ACC       IOP/#
B/ROP                    B/WOP        RTIME                  WTIME
#ROP                     #WOP         Seqlen                 #Seq
--------------------------------------------------------------------------------
/dev/hdisk0                35840.000M 17442.406M    52/M
0,0,0                    8,40,512     0.000,0.000,0.000   1.176,6.358,28.029
0                        132          0,0,0                  0
--------------------------------------------------------------------------------
/dev/hdisk1                51200.000M 11528.816M    256/M
0,0,0                    8,8,8        0.000,0.000,0.000   0.351,0.351,0.351
0                        1            0,0,0                  0
--------------------------------------------------------------------------------
```

The Hot File Report, sorted by capacity accessed section is shown in
Example 6-28:

Example 6-28   Hot Files sorted by capacity accessed

```
Hot Files Report(sorted by CAP_ACC)
--------------------------------------------------------------------------------
NAME                            Size     CAP_ACC   IOP/#     LV
B/ROP                    B/WOP           RTIME               WTIME
#ROP                     #WOP            Seqlen              #Seq
--------------------------------------------------------------------------------
MYFILE3                         100.000M 100.000M  1024/M    /dev/hd3
0,0,0                    4096,1024,4096   0.000,0.000,0.000   0.010,0.006,159.054
0                        102400          0,0,0               0
--------------------------------------------------------------------------------
MYFILE2                         100.000M 100.000M  1024/M    /dev/hd3
0,0,0                    4096,1024,4096   0.000,0.000,0.000   0.010,0.016,888.224
0                        102400          0,0,0               0
--------------------------------------------------------------------------------
MYFILE1                         100.000M 100.000M  1024/M    /dev/hd3
0,0,0                    4096,1024,4096   0.000,0.000,0.000   0.009,0.012,341.280
0                        102400          0,0,0               0
--------------------------------------------------------------------------------
```

The Hot Logical Volume Report, sorted by capacity accessed section is
displayed in Example 6-29.

Example 6-29   Hot Logical Volumes

```
Hot Logical Volume Report(sorted by CAP_ACC)
--------------------------------------------------------------------------------
NAME                    Size          CAP_ACC      IOP/#         #Files
B/ROP               B/WOP             RTIME                WTIME
#ROP                #WOP              Seqlen               #Seq
--------------------------------------------------------------------------------
/dev/hd2                1984.000M     1581.219M    256/M           3
0,0,0               8,8,8             0.000,0.000,0.000   11.756,42.800,81.619
0                   12                0,0,0                0
--------------------------------------------------------------------------------
/dev/hd3                4224.000M     459.812M     8/M             3
0,0,0               8,263,512         0.000,0.000,0.000   3.720,339.170,1359.117
0                   10364             0,0,0                0
--------------------------------------------------------------------------------
/dev/hd9var             384.000M      302.699M     256/M           2
0,0,0               8,8,8             0.000,0.000,0.000   3.935,50.324,103.397
0                   15                0,0,0                0
--------------------------------------------------------------------------------
```

The Hot Physical Volume Report, sorted by capacity accessed section is displayed in Example 6-30.

Example 6-30   Hot Physical Volumes

```
Hot Physical Volume Report(sorted by CAP_ACC)
--------------------------------------------------------------------------------
NAME                    Size          CAP_ACC      IOP/#
B/ROP               B/WOP             RTIME                WTIME
#ROP                #WOP              Seqlen               #Seq
--------------------------------------------------------------------------------
/dev/hdisk0             35840.000M    17998.020M   8/M
0,0,0               8,262,512         0.000,0.000,0.000   0.984,3.001,59.713
0                   10400             0,0,0                0
--------------------------------------------------------------------------------
```

The Hot Files Report, sorted by IOP/# is shown in Example 6-31.

Example 6-31   Hot Files sorted by IOP

```
Hot Files Report(sorted by IOP/#)
--------------------------------------------------------------------------------
NAME                            Size      CAP_ACC    IOP/#     LV
B/ROP               B/WOP             RTIME                WTIME
#ROP                #WOP              Seqlen               #Seq
```

```
--------------------------------------------------------------------------------
/etc/objrepos/SWservAt.vc           0.016M    0.000M    52429/M    /dev/hd4
40,20,40                0,0,0                   0.002,0.001,0.003  0.000,0.000,0.000
4                       0                       1,1,1                      1
--------------------------------------------------------------------------------
/var/adm/cron/log                   0.596M    0.000M    14075/M    /dev/hd9var
0,0,0                   39,74,110               0.000,0.000,0.000  0.009,0.015,0.021
0                       2                       0,0,0                      0
--------------------------------------------------------------------------------
/etc/objrepos/SWservAt              0.012M    0.000M    5269/M     /dev/hd4
328,199,468             0,0,0                   0.002,0.001,0.004  0.000,0.000,0.000
4                       0                       1,1,1                      1
--------------------------------------------------------------------------------
```

The Hot Logical Volume report, sorted by IOP/# is shown in Example 6-32.

Example 6-32   Hot Logical Volumes sorted by IOP

```
Hot Logical Volume Report(sorted by IOP/#)
--------------------------------------------------------------------------------
NAME                    Size            CAP_ACC        IOP/#          #Files
B/ROP                   B/WOP           RTIME          WTIME
#ROP                    #WOP            Seqlen         #Seq
--------------------------------------------------------------------------------
/dev/fslv00             128.000M        0.000M         256/M          0
0,0,0                   8,8,8           0.000,0.000,0.000  59.731,59.731,59.731
0                       1               0,0,0              0
--------------------------------------------------------------------------------
/dev/fslv01             64.000M         0.000M         256/M          0
0,0,0                   8,8,8           0.000,0.000,0.000  3.854,3.854,3.854
0                       1               0,0,0              0
--------------------------------------------------------------------------------
/dev/fslv02             128.000M        0.000M         256/M          0
0,0,0                   8,8,8           0.000,0.000,0.000  4.108,4.108,4.108
0                       1               0,0,0              0
--------------------------------------------------------------------------------
```

The Hot Physical Volume Report, sorted by IOP/# is shown in Example 6-33.

Example 6-33   Hot Physical Volumes sorted by IOP

```
Hot Physical Volume Report(sorted by IOP/#)
--------------------------------------------------------------------------------
NAME                    Size            CAP_ACC        IOP/#
B/ROP                   B/WOP           RTIME          WTIME
```

```
#ROP                  #WOP                Seqlen            #Seq
--------------------------------------------------------------------------------
/dev/hdisk0                    35840.000M  17998.020M    8/M
0,0,0                 8,262,512           0.000,0.000,0.000   0.984,3.001,59.713
0                     10400               0,0,0             0
--------------------------------------------------------------------------------
```

The Hot Files Report, sorted by #ROP is shown in Example 6-34.

Example 6-34   Hot Files sorted by #ROP

```
Hot Files Report(sorted by #ROP)
--------------------------------------------------------------------------------
NAME                              Size    CAP_ACC   IOP/#      LV
B/ROP                 B/WOP               RTIME              WTIME
#ROP                  #WOP                Seqlen            #Seq
--------------------------------------------------------------------------------
/unix                             33.437M   0.141M   256/M     /dev/hd2
4096,4096,4096        0,0,0               0.002,0.003,0.008   0.000,0.000,0.000
97                    0                   1,1,1             97
--------------------------------------------------------------------------------
/usr/lib/nls/msg/en_US/ksh.cat    0.006M    0.008M   4352/M    /dev/hd2
4096,241,4096         0,0,0               0.003,0.000,0.004   0.000,0.000,0.000
68                    0                   1,2,2             2
--------------------------------------------------------------------------------
/etc/security/user                0.011M    0.012M   256/M     /dev/hd4
4096,4096,4096        0,0,0               0.003,0.004,0.008   0.000,0.000,0.000
5                     0                   1,1,1             5
--------------------------------------------------------------------------------
```

The Hot Logical Volume Report, sorted by #ROP is shown in Example 6-35.

Example 6-35   Hot Logical Volumes sorted by #ROP

```
Hot Logical Volume Report(sorted by #ROP)
--------------------------------------------------------------------------------
NAME                      Size      CAP_ACC      IOP/#        #Files
B/ROP                 B/WOP         RTIME                WTIME
#ROP                  #WOP          Seqlen               #Seq
--------------------------------------------------------------------------------
/dev/hd3                  4224.000M   459.812M     8/M            3
0,0,0                 8,263,512     0.000,0.000,0.000   3.720,339.170,1359.117
0                     10364         0,0,0                0
--------------------------------------------------------------------------------
/dev/hd2                  1984.000M   1581.219M    256/M          3
```

```
0,0,0                    8,8,8              0.000,0.000,0.000  11.756,42.800,81.619
0                        12                 0,0,0                  0
--------------------------------------------------------------------------------
/dev/hd9var                     384.000M    302.699M      256/M          2
0,0,0                    8,8,8              0.000,0.000,0.000  3.935,50.324,103.397
0                        15                 0,0,0                  0
--------------------------------------------------------------------------------
```

The Hot Physical Volumes Report sorted by #ROP is shown in Example 6-36.

Example 6-36   Hot Physical Volumes Report sorted by #ROP

```
Hot Physical Volume Report(sorted by #ROP)
--------------------------------------------------------------------------------
NAME                            Size        CAP_ACC       IOP/#
B/ROP                    B/WOP              RTIME             WTIME
#ROP                     #WOP               Seqlen            #Seq
--------------------------------------------------------------------------------
/dev/hdisk0                     35840.000M  17998.020M    8/M
0,0,0                    8,262,512          0.000,0.000,0.000  0.984,3.001,59.713
0                        10400              0,0,0                  0
--------------------------------------------------------------------------------
```

The Hot Files Report, sorted by #WOP, is shown in Example 6-37.

Example 6-37   Hot Files sorted by #WOP

```
Hot Files Report(sorted by #WOP)
--------------------------------------------------------------------------------
NAME                             Size     CAP_ACC   IOP/#     LV
B/ROP                    B/WOP             RTIME             WTIME
#ROP                     #WOP              Seqlen            #Seq
--------------------------------------------------------------------------------
1                                100.000M  100.000M  1024/M    /dev/hd3
0,0,0                    4096,1024,4096     0.000,0.000,0.000  0.009,0.012,341.280
0                        102400             0,0,0                  0
--------------------------------------------------------------------------------
2                                100.000M  100.000M  1024/M    /dev/hd3
0,0,0                    4096,1024,4096     0.000,0.000,0.000  0.010,0.016,888.224
0                        102400             0,0,0                  0
--------------------------------------------------------------------------------
3                                100.000M  100.000M  1024/M    /dev/hd3
0,0,0                    4096,1024,4096     0.000,0.000,0.000  0.010,0.006,159.054
0                        102400             0,0,0                  0
--------------------------------------------------------------------------------
```

The Hot Logical Volume Report, sorted by #WOP, is shown in Example 6-38.

Example 6-38   Hot Logical Volumes sorted by #WOP

```
Hot Logical Volume Report(sorted by #WOP)
--------------------------------------------------------------------------------
NAME                         Size          CAP_ACC       IOP/#         #Files
B/ROP                 B/WOP               RTIME                 WTIME
#ROP                  #WOP                Seqlen                #Seq
--------------------------------------------------------------------------------
/dev/hd3                     4224.000M     459.812M      8/M             3
0,0,0                 8,263,512           0.000,0.000,0.000   3.720,339.170,1359.117
0                     10364               0,0,0                 0
--------------------------------------------------------------------------------
/dev/hd8                     64.000M       0.090M        256/M           0
0,0,0                 8,8,8               0.000,0.000,0.000   1.010,75.709,1046.734
0                     61                  0,0,0                 0
--------------------------------------------------------------------------------
/dev/hd4                     192.000M      154.934M      256/M           12
0,0,0                 8,8,8               0.000,0.000,0.000   1.907,27.166,74.692
0                     16                  0,0,0                 0
--------------------------------------------------------------------------------
```

The Hot Physical Volume Report, sorted by #WOP, is shown in Example 6-39.

Example 6-39   Hot Physical Volumes sorted by #WOP

```
Hot Physical Volume Report(sorted by #WOP)
--------------------------------------------------------------------------------
NAME                         Size          CAP_ACC       IOP/#
B/ROP                 B/WOP               RTIME                 WTIME
#ROP                  #WOP                Seqlen                #Seq
--------------------------------------------------------------------------------
/dev/hdisk0                  35840.000M    17998.020M    8/M
0,0,0                 8,262,512           0.000,0.000,0.000   0.984,3.001,59.713
0                     10400               0,0,0                 0
--------------------------------------------------------------------------------
```

The Hot Files Report, sorted by RTIME, is shown in Example 6-40.

Example 6-40   Hot Files sorted by RTIME

```
Hot Files Report(sorted by RTIME)
--------------------------------------------------------------------------------
NAME                               Size      CAP_ACC   IOP/#     LV
B/ROP                 B/WOP                 RTIME               WTIME
#ROP                  #WOP                  Seqlen              #Seq
```

```
--------------------------------------------------------------------------------
/etc/vfs                             0.002M    0.008M    256/M      /dev/hd4
4096,4096,4096       0,0,0                    0.002,0.006,0.010   0.000,0.000,0.000
2                    0                        2,2,2                  1
--------------------------------------------------------------------------------
/etc/security/user                   0.011M    0.012M    256/M      /dev/hd4
4096,4096,4096       0,0,0                    0.003,0.004,0.008   0.000,0.000,0.000
5                    0                        1,1,1                  5
--------------------------------------------------------------------------------
/usr/lib/nls/msg/en_US/cmdtrace.cat 0.064M    0.004M    256/M      /dev/hd2
4096,4096,4096       0,0,0                    0.004,0.004,0.004   0.000,0.000,0.000
2                    0                        1,1,1                  2
--------------------------------------------------------------------------------
```

The Hot Logical Volume Report, sorted by RTIME, is shown in Example 6-41.

Example 6-41   Hot Logical Volumes sorted by RTIME

```
Hot Logical Volume Report(sorted by RTIME)
--------------------------------------------------------------------------------
NAME                      Size          CAP_ACC      IOP/#         #Files
B/ROP                     B/WOP         RTIME                WTIME
#ROP                      #WOP          Seqlen               #Seq
--------------------------------------------------------------------------------
/dev/fslv02               128.000M      0.000M       256/M           0
0,0,0                     8,8,8         0.000,0.000,0.000  4.108,4.108,4.108
0                         1             0,0,0                0
--------------------------------------------------------------------------------
/dev/fslv01               64.000M       0.000M       256/M           0
0,0,0                     8,8,8         0.000,0.000,0.000  3.854,3.854,3.854
0                         1             0,0,0                0
--------------------------------------------------------------------------------
/dev/fslv00               128.000M      0.000M       256/M           0
0,0,0                     8,8,8         0.000,0.000,0.000  59.731,59.731,59.731
0                         1             0,0,0                0
--------------------------------------------------------------------------------
```

The Hot Physical Volume Report, sorted by RTIME, is shown in Example 6-42.

Example 6-42   Hot Physical Volumes sorted by RTIME

```
Hot Physical Volume Report(sorted by RTIME)
--------------------------------------------------------------------------------
NAME                      Size          CAP_ACC      IOP/#
B/ROP                     B/WOP         RTIME                WTIME
```

```
#ROP                    #WOP            Seqlen              #Seq
----------------------------------------------------------------------------
/dev/hdisk0                35840.000M   17998.020M   8/M
0,0,0                   8,262,512       0.000,0.000,0.000   0.984,3.001,59.713
0                       10400           0,0,0               0
----------------------------------------------------------------------------
```

The Hot Files Report, sorted by WTIME, is shown in Example 6-43.

Example 6-43   Hot Files sorted by WTIME

```
Hot Files Report(sorted by WTIME)
----------------------------------------------------------------------------
NAME                            Size      CAP_ACC   IOP/#     LV
B/ROP                   B/WOP           RTIME               WTIME
#ROP                    #WOP            Seqlen              #Seq
----------------------------------------------------------------------------
2                               100.000M  100.000M  1024/M    /dev/hd3
0,0,0                   4096,1024,4096    0.000,0.000,0.000   0.010,0.016,888.224
0                       102400            0,0,0               0
----------------------------------------------------------------------------
/var/adm/cron/log               0.596M    0.000M    14075/M   /dev/hd9var
0,0,0                   39,74,110         0.000,0.000,0.000   0.009,0.015,0.021
0                       2                 0,0,0               0
----------------------------------------------------------------------------
1                               100.000M  100.000M  1024/M    /dev/hd3
0,0,0                   4096,1024,4096    0.000,0.000,0.000   0.009,0.012,341.280
0                       102400            0,0,0               0
----------------------------------------------------------------------------
3                               100.000M  100.000M  1024/M    /dev/hd3
0,0,0                   4096,1024,4096    0.000,0.000,0.000   0.010,0.006,159.054
0                       102400            0,0,0               0
----------------------------------------------------------------------------
```

The Hot Logical Volume Report, sorted by WTIME, is shown in Example 6-44.

Example 6-44   Hot Logical Volumes sorted by WTIME

```
Hot Logical Volume Report(sorted by WTIME)
----------------------------------------------------------------------------
NAME                     Size      CAP_ACC     IOP/#          #Files
B/ROP                   B/WOP       RTIME                WTIME
#ROP                    #WOP        Seqlen               #Seq
----------------------------------------------------------------------------
/dev/hd3                 4224.000M   459.812M    8/M            3
```

```
0,0,0                   8,263,512           0.000,0.000,0.000   3.720,339.170,1359.117
0                       10364               0,0,0               0
--------------------------------------------------------------------------------
/dev/hd8                        64.000M     0.090M      256/M           0
0,0,0                   8,8,8               0.000,0.000,0.000   1.010,75.709,1046.734
0                       61                  0,0,0               0
--------------------------------------------------------------------------------
/dev/fslv00                     128.000M    0.000M      256/M           0
0,0,0                   8,8,8               0.000,0.000,0.000   59.731,59.731,59.731
0                       1                   0,0,0               0
--------------------------------------------------------------------------------
```

The Hot Physical Volume Report, sorted by WTIME, is shown in Example 6-45.

Example 6-45   Hot Physical Volume Report sorted by WTIME

```
Hot Physical Volume Report(sorted by WTIME)
--------------------------------------------------------------------------------
NAME                    Size            CAP_ACC     IOP/#
B/ROP                   B/WOP               RTIME               WTIME
#ROP                    #WOP                Seqlen              #Seq
--------------------------------------------------------------------------------
/dev/hdisk0                     35840.000M  17998.020M  8/M
0,0,0                   8,262,512           0.000,0.000,0.000   0.984,3.001,59.713
0                       10400               0,0,0               0
--------------------------------------------------------------------------------
```

6.3 Memory affinity API enhancements

AIX 7.1 allows an application to request that a thread have a *strict* attachment to an SRAD for purposes of memory affinity. The new form of attachment is similar to the current SRAD attachment APIs except that the thread is not moved to a different SRAD for purposes of load balancing by the dispatcher.

The following is a comparison between a new strict attachment API and the existing advisory attachment API.

► When a thread has an *advisory* SRAD attachment, the AIX thread dispatcher is free to ignore the attachment if the distribution of load across various SRADs justifies migration of the thread to another SRAD. The new strict attachment will override any load balancing efforts of the dispatcher.

► The current advisory SRAD attachment APIs allow SRAD attachments to R_PROCESS, R_THREAD, R_SHM, R_FILDES, and R_PROCMEM

resource types. The new strict SRAD attachment only allows SRAD attachment to R_THREAD resource types. Any other use of strict SRAD attachment results in an EINVAL error code.

► The pthread_attr_setsrad_np API is modified to accept a new *flag* parameter that indicates whether the SRAD attachment is strict or advisory.

The following is a list of functionalities that are not changed from advisory SRAD attachments. They are mentioned here for completeness.

► If a strict attachment is sought for an SRAD that has only folded processors at the time of the attachment request, the request is processed normally. The threads are placed temporarily on the node global run queue. The expectation is that folding is a temporary situation and the threads will get runtime when the processors are unfolded.

► Unauthorized applications can make strict SRAD attachments. root authority or CAP_NUMA_ATTACH capability is not a requirement. This is the same behavior as in advisory SRAD attachment APIs.

► If a strict attachment is attempted to an SRAD that has only exclusive processors, the attachment succeeds and the thread is marked as permanently borrowed. This is the same behavior as in advisory SRAD attachment APIs.

► DR CPU remove operation will ignore strict SRAD attachments when calculating processor costs that DRM uses to pick the processor to remove. This is the same behavior as in advisory SRAD attachment APIs.

► Advisory attachments are ignored in the event of a DR operation requiring all threads to be migrated off a processor. This holds true for strict attachments as well.

► When a request for an advisory SRAD attachment conflicts with an existing RSET attachment, the SRAD attachment is still processed if there is at least one processor in the intersection between the SRAD and the RSET. This holds true for strict SRAD attachments.

► When an advisory attachment is sought for a thread that already has a previous attachment, the older attachment is overridden by the new one. This behavior is maintained when seeking a strict attachment as well.

6.3.1 API enhancements

This section discusses the new APIs for memory affinity.

A new flag, R_STRICT_SRAD, is added to the flag parameter of the ra_attach, ra_fork and ra_exec APIs.

The R_STRICT_SRAD flag indicates that a thread is attached to an SRAD in a strict manner. It will run in the same SRAD, unaffected by load balancing operations. It will be rehomed to a different SRAD only if a DR operation removes all processors from the current SRAD. It is important to note that when strict SRAD attachments are used, the application must cater to the possibility of uneven loads across SRADs.

> **Note:** ra_detach removes all SRAD attachments, *strict* is used to detach an existing SRAD attachment, and any attachment *strict* or *advisory* will be removed.

6.3.2 The pthread attribute API

There are two existing pthread APIs to set/get an SRAD in the pthread attributes, namely pthread_attr_setsrad_np and pthread_attr_getsrad_np. These are modified to have a flags parameter that will indicate whether the SRAD attachment is strict.

pthread_attr_setsrad_np
The following defines the pthread_attr_setsrap_np API.

Syntax	int pthread_attr_setsrad_np (attr, srad, flags) pthread_attr_t *attr; sradid_t srad; *int flags*;
Description	The *flags* parameter indicates whether the SRAD attachment is strict or advisory.
Parameters	flags: Setting R_STRICT_SRAD indicates that the SRAD is a strictly preferred one.

pthread_attr_getsrad_np
The following defines the pthread_attr_getsrad_np API.

Syntax	int pthread_attr_getsrad_np (attr, sradp, flagsp) pthread_attr_t *attr; sradid_t *srad; *int *flagsp;*
Description	The *flagsp* parameter returns R_STRICT_SRAD if the SRAD attachment, if any, is strict.

flagsp:

Set to R_STRICT_SRAD if SRAD attachment is strict, NULL otherwise.

6.4 Enhancement of the iostat command

Debugging I/O performance and hang issues is a time-consuming and iterative process. To help with the analysis of I/O issues, the **iostat** command has been enhanced in AIX 6.1 TL6 and in AIX 7.1. With this enhancement useful data can be captured to help identify and correct the problem quicker.

The enhancement to the **iostat** command leverages the bufx capabilities in AIX to produce an end-to-end I/O metrics report. It is called the Block I/O Device Utilization Report, which provides statistics per I/O device. The report helps you in analyzing the I/O statistics at VMM or file system, and disk layers of I/O stack. The report also helps you in analyzing the performance of the I/O stack.

A new flag, **-b**, is available for the **iostat** command that will display block I/O device utilization statistics.

Example 6-46 shows an example of the command output when this new flag is used.

Example 6-46   Example of the new iostat output

```
# iostat -b 5

System configuration: lcpu=2 drives=3 vdisks=3
Block Devices :7
device            reads writes    bread   bwrite   rserv   wserv  rerr  werr
hdisk0            0.00   0.00    0.000    0.000    0.00    0.00  0.00  0.00
hd8              0.00   0.00    0.000    0.000    0.00    0.00  0.00  0.00
hd4              0.00   0.00    0.000    0.000    0.00    0.00  0.00  0.00
hd9var           0.00   0.00    0.000    0.000    0.00    0.00  0.00  0.00
hd2              0.00   0.00    0.000    0.000    0.00    0.00  0.00  0.00
hd3              0.00   0.00    0.000    0.000    0.00    0.00  0.00  0.00
hd10opt          0.00   0.00    0.000    0.000    0.00    0.00  0.00  0.00
```

The meaning of the columns is as follows:

device   Indicates the device name

reads   Indicates the number of read requests over the monitoring interval.

writes   Indicates the number of write requests over the monitoring interval.

bread	Indicates the number of bytes read over the monitoring interval.
bwrite	Indicates the number of bytes written over the monitoring interval.
rserv	Indicates the read service time per read over the monitoring interval. The default unit of measure is milliseconds.
wserv	Indicates the write service time per write over the monitoring interval. The default unit of measure is milliseconds.
rerr	Indicates the number of read errors over the monitoring interval.
werr	Indicates the number of write errors over the monitoring interval.

The **raso** command is used to turn the statistic collection on and off. Example 6-47 shows how to use the **raso** command to turn on the statistic collection that the **iostat** command uses.

Example 6-47   Using the raso command to turn on statistic collection

```
# raso -o biostat=1
Setting biostat to 1
#
```

The **raso -L** command shows the current status of statistic collection. Example 6-48 shows the output of the **raso -L** command.

Example 6-48   Using raso -L command to see whether statistic collection is on

```
# raso -L
NAME                     CUR    DEF   BOOT   MIN   MAX    UNIT           TYPE
     DEPENDENCIES
--------------------------------------------------------------------------------
biostat                  1      0     0      0     1      boolean          D
--------------------------------------------------------------------------------
kern_heap_noexec         0      0     0      0     1      boolean          B
--------------------------------------------------------------------------------
kernel_noexec            1      1     1      0     1      boolean          B
--------------------------------------------------------------------------------
mbuf_heap_noexec         0      0     0      0     1      boolean          B
--------------------------------------------------------------------------------
mtrc_commonbufsize       1209   1209  1209   1     16320  4KB pages        D
     mtrc_enabled
     mtrc_rarebufsize
--------------------------------------------------------------------------------
mtrc_enabled             1      1     1      0     1      boolean          B
--------------------------------------------------------------------------------
mtrc_rarebufsize         62     62    62     1     15173  4KB pages        D
     mtrc_enabled
```

```
     mtrc_commonbufsize
--------------------------------------------------------------------------------
tprof_cyc_mult              1    1    1    1    100    numeric         D
--------------------------------------------------------------------------------
tprof_evt_mult              1    1    1    1    10000  numeric         D
--------------------------------------------------------------------------------
tprof_evt_system            0    0    0    0    1      boolean         D
--------------------------------------------------------------------------------
tprof_inst_threshold        1000 1000 1000 1    2G-1   numeric         D
--------------------------------------------------------------------------------

n/a means parameter not supported by the current platform or kernel

Parameter types:
    S = Static: cannot be changed
    D = Dynamic: can be freely changed
    B = Bosboot: can only be changed using bosboot and reboot
    R = Reboot: can only be changed during reboot
    C = Connect: changes are only effective for future socket connections
    M = Mount: changes are only effective for future mountings
    I = Incremental: can only be incremented
    d = deprecated: deprecated and cannot be changed

Value conventions:
    K = Kilo: 2^10     G = Giga: 2^30     P = Peta: 2^50
    M = Mega: 2^20     T = Tera: 2^40     E = Exa: 2^60
#
```

Note: The biostat tuning parameter is dynamic. It does not require a reboot to take effect.

Turning on the statistic collection uses a little more memory but does not have a processor utilization impact.

6.5 The vmo command lru_file_repage setting

In AIX V7, the vmo command lru_file_repage setting has been removed. AIX 7.1 will make the same decisions as AIX 6.1 with lru_file_repage at its default setting of 0.

7

Networking

AIX V7.1 provides many enhancements in the networking area. Described in this chapter, they include:

► 7.1, "Enhancement to IEEE 802.3ad Link Aggregation" on page 272

► 7.2, "Removal of BIND 8 application code" on page 282

► 7.3, "Network Time Protocol version 4" on page 283

271

7.1 Enhancement to IEEE 802.3ad Link Aggregation

This section discusses the enhancement to the Ethernet link aggregation in AIX V7.1.

This feature first became available in AIX V7.1 and is included in AIX 6.1 TL 06.

7.1.1 EtherChannel and Link Aggregation in AIX

EtherChannel and IEEE 802.3ad Link Aggregation are network port aggregation technologies that allow multiple Ethernet adapters to be teamed to form a single pseudo Ethernet device. This teaming of multiple Ethernet adapters to form a single pseudo Ethernet device is known as aggregation.

Conceptually, IEEE 802.3ad Link Aggregation works the same as EtherChannel.

Advantages of using IEEE 802.3ad Link Aggregation over EtherChannel are that IEEE 802.3ad Link Aggregation can create the link aggregations in the switch automatically, and that it allows you to use switches that support the IEEE 802.3ad standard but do not support EtherChannel.

Note: When using IEE 802.3ad Link Aggregation ensure that your Ethernet switch hardware supports the IEEE 802.3ad standard.

With the release of AIX V7.1 and AIX V6.1 TL06, configuring an AIX Ethernet interface to use the 802.3ad mode requires that the Ethernet switch ports also be configured in IEEE 802.3ad mode.

7.1.2 IEEE 802.3ad Link Aggregation functionality

The IEEE 802.3ad Link Aggregation protocol, also known as Link Aggregation Control Protocol (LACP), relies on LACP Data Units (LACPDU) to control the status of link aggregation between two parties, the actor and the partner.

The actor is the IEEE 802.3ad Link Aggregation and the partner is the Ethernet switch port.

The Link Aggregation Control Protocol Data Unit (LACPDU) contains the information about the actor and the actor's view of its partner. Each port in the aggregation acts as an actor and a partner. LACPDU is exchanged at the rate specified by the actor. All ports under the link aggregation are required to participate in LACP activity.

Both the actor and the partner monitor LACPDU in order to ensure that communication is correctly established and that they have the correct view of the other's capability.

The aggregated link is considered to be nonoperational when there is a disagreement between an actor and its partner. When an aggregation is considered nonoperational, that port will not be used to transfer data packets. A port will only be used to transfer data packets if both the actor and the partner have exchanged LACPDU and they agree with each other's view.

7.1.3 AIX V7.1 enhancement to IEEE 802.3ad Link Aggregation

Prior to AIX V7.1, the AIX implementation of the IEEE 802.3ad protocol did not wait for the LACP exchange to complete before using the port for data transmission.

This could result in packet loss if the LACP partner, which may typically be an Ethernet switch, relies on LACP exchange to complete before it uses the port for data transmission. This could result in significant packet loss if the delay between the link status up and the LACP exchange complete is large.

AIX V7.1 includes an enhancement to the LACP implementation to allow ports to exchange LACPDU and agree upon each other's state before they are ready for data transmission.

This enhancement is particularly useful when using stacked Ethernet switches.

Without this enhancement to the AIX implementation of IEEE 802.3ad, stacked Ethernet switches may experience delays between the time that an Ethernet port is activated and an LACPDU transmit occurs when integrating or reintegrating an Ethernet switch into the stacked Ethernet switch configuration.

> **Important:** In previous versions of AIX, the implementation of the IEEE 802.3ad protocol did not require Ethernet switch ports to be configured to use the 802.3ad protocol.
>
> AIX V7.1 and AIX V6.1 TL06 require the corresponding Ethernet switch ports to be configured in IEEE 802.3ad mode when the AIX Ethernet interface is operating in the 802.3ad mode.
>
> When planning to upgrade or migrate to AIX V7.1 or AIX V6.1 TL06, ensure that any Ethernet switch ports in use by an AIX 802.3ad Link Aggregation are configured to support the 802.3ad protocol.

When operating in IEEE 802.3ad mode, the enhanced support allows for up to three LACPDUs to be missed within the interval value. Once three LACPDUs are missed within the interval value, AIX will not use the link for data transmission until such time as a new LACPDU is received.

The interval durations are displayed in Table 7-1.

Table 7-1   The LACP interval duration

Type of interval	Interval duration
Short interval	1 seconds
Long interval	30 seconds

In the following examples we show an IEEE 802.3ad Link Aggregation change from an operational to nonoperational state, then revert to operational status due to a hardware cabling issue.

Our IEEE 802.3ad Link Aggregation pseudo Ethernet device is defined as `ent6` and consists of the two logical Ethernet devices ent2 and ent4. Example 7-1 lists the `lsdev -Cc adapter` command output, displaying the `ent6` pseudo Ethernet device.

> **Note:** The `lsdev` command displays the `ent6` pseudo Ethernet device as an EtherChannel and IEEE 802.3ad Link Aggregation. We discuss later in the example how to determine whether the `ent6` pseudo device is operating as an IEEE 802.3ad Link Aggregation.

Example 7-1   The lsdev -Cc adapter command

```
# lsdev -Cc adapter
ent0    Available       Virtual I/O Ethernet Adapter (l-lan)
ent1    Available       Virtual I/O Ethernet Adapter (l-lan)
ent2    Available 00-08 2-Port 10/100/1000 Base-TX PCI-X Adapter (14108902)
ent3    Available 00-09 2-Port 10/100/1000 Base-TX PCI-X Adapter (14108902)
ent4    Available 01-08 2-Port 10/100/1000 Base-TX PCI-X Adapter (14108902)
ent5    Available 01-09 2-Port 10/100/1000 Base-TX PCI-X Adapter (14108902)
ent6    Available       EtherChannel / IEEE 802.3ad Link Aggregation
vsa0    Available       LPAR Virtual Serial Adapter
vscsi0 Available       Virtual SCSI Client Adapter
#
```

By using the `lsattr -El` command, we can display the logical Ethernet devices that make up the `ent6` pseudo Ethernet device.

The **lsattr -El** command also displays in which mode the pseudo Ethernet device is operating. We can see that the ent6 pseudo Ethernet device is made up of the ent2 and ent4 logical Ethernet devices. Additionally, the ent6 pseudo Ethernet device is operating in IEEE 802.3ad mode and the interval is long.

Example 7-2   Displaying the logical Ethernet devices in the ent6 pseudo Ethernet device

```
# lsattr -El ent6
adapter_names    ent2,ent4          EtherChannel Adapters                       True
alt_addr         0x000000000000 Alternate EtherChannel Address                  True
auto_recovery    yes                Enable automatic recovery after failover    True
backup_adapter   NONE               Adapter used when whole channel fails       True
hash_mode        default            Determines how outgoing adapter is chosen   True
interval         long               Determines interval value for IEEE 802.3ad mode True
mode             8023ad             EtherChannel mode of operation              True
netaddr          0                  Address to ping                             True
noloss_failover  yes                Enable lossless failover after ping failure True
num_retries      3                  Times to retry ping before failing          True
retry_time       1                  Wait time (in seconds) between pings        True
use_alt_addr     no                 Enable Alternate EtherChannel Address       True
use_jumbo_frame  no                 Enable Gigabit Ethernet Jumbo Frames        True
#
```

The ent2 and ent4 devices are each defined on port T1 of a 1-gigabit Ethernet adapter in the AIX V7.1 partition.

Example 7-3 lists the physical hardware locations for the ent2 and ent4 logical Ethernet devices by using the **lsslot -c pci** and **lscfg-vl** commands.

Example 7-3   The lsslot and lscfg commands display the physical Ethernet adapters

```
# lsslot -c pci
# Slot                  Description                        Device(s)
U78A0.001.DNWHZS4-P1-C4  PCI-X capable, 64 bit, 266MHz slot  ent2 ent3
U78A0.001.DNWHZS4-P1-C5  PCI-X capable, 64 bit, 266MHz slot  ent4 ent5

# lscfg -vl ent2
  ent2             U78A0.001.DNWHZS4-P1-C4-T1  2-Port 10/100/1000 Base-TX PCI-X
Adapter (14108902)

      2-Port 10/100/1000 Base-TX PCI-X Adapter:
        Part Number.................03N5297
        FRU Number..................03N5297
        EC Level....................H13845
        Manufacture ID..............YL1021
        Network Address.............00215E8A4072
```

```
        ROM Level.(alterable).......DV0210
        Hardware Location Code......U78A0.001.DNWHZS4-P1-C4-T1

# lscfg -vl ent4
  ent4              U78A0.001.DNWHZS4-P1-C5-T1  2-Port 10/100/1000 Base-TX PCI-X
Adapter (14108902)

      2-Port 10/100/1000 Base-TX PCI-X Adapter:
        Part Number.................03N5297
        FRU Number..................03N5297
        EC Level....................H13845
        Manufacture ID..............YL1021
        Network Address.............00215E8A41B6
        ROM Level.(alterable).......DV0210
        Hardware Location Code......U78A0.001.DNWHZS4-P1-C5-T1
#
```

Example 7-4 shows the **entstat -d** command being used to display the status of the ent6 pseudo Ethernet device.

> **Note:** Due to the large amount of output displayed by the **entstat -d** command, only the fields relevant to this example are shown.

Example 7-4   The entstat -d ent6 output - Link Aggregation operational

```
# entstat -d ent6
-------------------------------------------------------------
ETHERNET STATISTICS (ent6) :
Device Type: IEEE 802.3ad Link Aggregation
Hardware Address: 00:21:5e:8a:40:72
Elapsed Time: 0 days 21 hours 43 minutes 30 seconds
-------------------------------------------------------------
ETHERNET STATISTICS (ent2) :
Device Type: 2-Port 10/100/1000 Base-TX PCI-X Adapter (14108902)
Hardware Address: 00:21:5e:8a:40:72

IEEE 802.3ad Port Statistics:
----------------------------

            Actor State:
                    LACP activity: Active
                    LACP timeout: Long
                    Aggregation: Aggregatable
                    Synchronization: IN_SYNC
```

```
                Collecting: Enabled
                Distributing: **Enabled**
                Defaulted: False
                Expired: **False**

        Partner State:
                LACP activity: Active
                LACP timeout: Long
                Aggregation: Aggregatable
                Synchronization: **IN_SYNC**
                Collecting: Enabled
                Distributing: **Enabled**
                Defaulted: False
                Expired: **False**
--------------------------------------------------------------
ETHERNET STATISTICS (ent4) :
Device Type: 2-Port 10/100/1000 Base-TX PCI-X Adapter (14108902)
Hardware Address: 00:21:5e:8a:40:72

IEEE 802.3ad Port Statistics:
-----------------------------

        Actor State:
                LACP activity: Active
                LACP timeout: Long
                Aggregation: Aggregatable
                Synchronization: **IN_SYNC**
                Collecting: Enabled
                Distributing: **Enabled**
                Defaulted: False
                Expired: **False**

        Partner State:
                LACP activity: Active
                LACP timeout: Long
                Aggregation: Aggregatable
                Synchronization: **IN_SYNC**
                Collecting: Enabled
                Distributing: **Enabled**
                Defaulted: False
                Expired: **False**
#
```

In Example 7-4 on page 276, the `Actor State` for both the `ent2` and `ent4` logical Ethernet devices shows the `Distributing` state as `Enabled` and the `Expired` state as `False`. The `Synchronization` state is `IN_SYNC`.

Additionally, the `Partner State` for both the `ent2` and `ent4` logical Ethernet devices shows the `Distributing` state as `Enabled` and the `Expired` state as `False`. The `Synchronization` state is `IN_SYNC`.

This is the normal status mode for an operational IEEE 802.3a Link Aggregation.

The administrator is alerted of a connectivity issue by an error in the AIX error report. By using the **entstat -d** command the administrator discovers that the `ent4` logical Ethernet device is no longer operational.

Example 7-5 lists the output from the **entstat -d** command. In this example, the `Actor State` and `Partner State` values for the `ent4` logical Ethernet device status have changed. The `ent2` logical Ethernet device status remains unchanged.

> **Note:** Due to the large amount of output displayed by the **entstat -d** command, only the fields relevant to this example are shown.

Example 7-5   The entstat -d ent6 output - Link Aggregation nonoperational

```
# errpt
ECOBCCD4   0825110510 T H ent4             ETHERNET DOWN
A6DF45AA   0820181410 I O RMCdaemon        The daemon is started.
# entstat -d ent6
-------------------------------------------------------------
ETHERNET STATISTICS (ent6) :
Device Type: IEEE 802.3ad Link Aggregation
Hardware Address: 00:21:5e:8a:40:72
Elapsed Time: 0 days 22 hours 12 minutes 19 seconds
-------------------------------------------------------------
ETHERNET STATISTICS (ent2) :
Device Type: 2-Port 10/100/1000 Base-TX PCI-X Adapter (14108902)
Hardware Address: 00:21:5e:8a:40:72

IEEE 802.3ad Port Statistics:
----------------------------

        Actor State:
                LACP activity: Active
                LACP timeout: Long
                Aggregation: Aggregatable
```

```
                Synchronization: IN_SYNC
                Collecting: Enabled
                Distributing: **Enabled**
                Defaulted: False
                Expired: **False**

        Partner State:
                LACP activity: Active
                LACP timeout: Long
                Aggregation: Aggregatable
                Synchronization: IN_SYNC
                Collecting: Enabled
                Distributing: **Enabled**
                Defaulted: False
                Expired: **False**


----------------------------------------------------------------
ETHERNET STATISTICS (ent4) :
Device Type: 2-Port 10/100/1000 Base-TX PCI-X Adapter (14108902)
Hardware Address: 00:21:5e:8a:40:72
------------------------------------------------------------------------

IEEE 802.3ad Port Statistics:
-----------------------------

        Actor State:
                LACP activity: Active
                LACP timeout: Long
                Aggregation: Aggregatable
                Synchronization: IN_SYNC
                Collecting: Enabled
                Distributing: **Disabled**
                Defaulted: False
                Expired: **True**

        Partner State:
                LACP activity: Active
                LACP timeout: Long
                Aggregation: Aggregatable
                Synchronization: **OUT_OF_SYNC**
                Collecting: Enabled
                Distributing: Enabled
                Defaulted: False
                Expired: False
```

In Example 7-5 on page 278, the `Actor State` for the `ent4` logical Ethernet device shows the `Distributing` state as `Disabled` and the `Expired` state as `True`. The Synchronization state is IN_SYNC.

Additionally, the `Partner State` for the `ent4` logical Ethernet device shows the `Distributing` state as `Enabled` and the `Expired` state as `False`. The Synchronization state is `OUT_OF_SYNC`.

The `ent2` logical Ethernet adapter status remains unchanged.

From this, the administrator can determine that the ent4 logical Ethernet adapter has disabled its LACPDU sending and has expired its state, because it has failed to receive three LACPDU responses from the Ethernet switch port partner. In turn, the partner is now displayed as OUT_OF_SYNC, as the actor and partner are unable to agree upon their status.

Prior to the IEEE 802.3ad enhancement in AIX V7.1, the **entstat** output may not have reliably displayed the status for devices that do not report their *up/down* state, which could result in significant packet loss.

With the AIX V7.1 enhancement to IEEE 802.3ad Link Aggregation, the actor determines that the partner is not responding to three LACPDU packets and discontinues activity on that logical Ethernet adapter, until such time as it receives an LACPDU packet from the partner.

> **Note:** In this example, the `interval` is set to `long` (30 seconds).
>
> AIX V7.1 still supports *device up/down* status reporting, but if no *device down* status was reported, then the link status would be changed after 90 seconds (3*long interval).
>
> The `interval` may be changed to `short`, which would reduce the link status change to 3 seconds (3*short interval). Such changes should be tested to determine whether long or short interval is suitable for your specific environment.

It was determined that the loss of connectivity was due to a network change that resulted in the network cable connecting the `ent4` logical Ethernet device to the Ethernet switch port being moved to another switch port that was not enabled. Once the cabling was reinstated, the administrator again checked the `ent6` pseudo Ethernet device with the **entstat -d** command.

> **Note:** Due to the large amount of output displayed by the `entstat -d`
> command, only the fields relevant to this example are shown.

Example 7-6   The entstat -d ent6 output - Link Aggregation recovered and operational

```
# entstat -d ent6
---------------------------------------------------------------
ETHERNET STATISTICS (ent6) :
Device Type: IEEE 802.3ad Link Aggregation
Hardware Address: 00:21:5e:8a:40:72
Elapsed Time: 0 days 22 hours 33 minutes 50 seconds
===============================================================
ETHERNET STATISTICS (ent2) :
Device Type: 2-Port 10/100/1000 Base-TX PCI-X Adapter (14108902)
Hardware Address: 00:21:5e:8a:40:72

IEEE 802.3ad Port Statistics:
-----------------------------

        Actor State:
                LACP activity: Active
                LACP timeout: Long
                Aggregation: Aggregatable
                Synchronization: IN_SYNC
                Collecting: Enabled
                Distributing: Enabled
                Defaulted: False
                Expired: False

        Partner State:
                LACP activity: Active
                LACP timeout: Long
                Aggregation: Aggregatable
                Synchronization: IN_SYNC
                Collecting: Enabled
                Distributing: Enabled
                Defaulted: False
                Expired: False

---------------------------------------------------------------
ETHERNET STATISTICS (ent4) :
Device Type: 2-Port 10/100/1000 Base-TX PCI-X Adapter (14108902)
Hardware Address: 00:21:5e:8a:40:72
```

```
IEEE 802.3ad Port Statistics:
-----------------------------

        Actor State:
                LACP activity: Active
                LACP timeout: Long
                Aggregation: Aggregatable
                Synchronization: IN_SYNC
                Collecting: Enabled
                Distributing: Enabled
                Defaulted: False
                Expired: False

        Partner State:
                LACP activity: Active
                LACP timeout: Long
                Aggregation: Aggregatable
                Synchronization: IN_SYNC
                Collecting: Enabled
                Distributing: Enabled
                Defaulted: False
                Expired: False
#
```

In Example 7-6 on page 281 the `Actor State` for the `ent4` logical Ethernet device once more shows the `Distributing` state as `Enabled` and the `Expired` state as `False`. The Synchronization state is IN_SYNC.

Additionally, the `Partner State` for the `ent4` logical Ethernet device shows the `Distributing` state as `Enabled` and the `Expired` state as `False`. The Synchronization state is `IN_SYNC`.

The `ent2` logical Ethernet adapter status remains unchanged.

From this, the administrator can determine that the `ent4` logical Ethernet adapter has received an LACPDU from its Ethernet switch partner and enabled link state. The link state is now synchronized and the IEEE 802.3ad Link Aggregation is again operating normally.

7.2 Removal of BIND 8 application code

Berkeley Internet Name Domain (BIND) is a widely used implementation of the Domain Name System (DNS) protocol, since the general availability of AIX V6.1

Technology Level 2 in November 2008 AIX supports BIND 9 (version 9.4.1). In comparison to the previous version, BIND 8, the majority of the code was redesigned for BIND 9 to effectively exploit the underlying BIND architecture, to introduce many new features and in particular to support the DNS Security Extensions. The Internet System Consortium (ISC `http://www.isc.org`) maintains the BIND code and officially declared the end-of life for BIND 8 in August 2007. Ever since no code updates have been implemented in BIND 8. Also, the ISC only provides support for security-related issues to BIND version 9 or higher.

In consideration of the named facts AIX Version 7.1 only supports BIND version 9 and the BIND 8 application code has been removed from the AIX V7.1 code base and is no longer provided on the product media. However, the complete BIND 8 library code in `/usr/ccs/lib/libbind.a` is retained since many AIX applications are using the provided functionality.

As consequence of the BIND 8 application code removal the following application programs are no longer available with AIX 7:

► /usr/sbin/named8

► /usr/sbin/named8-xfer

On an AIX 7 system the symbolic link of the **named** daemon is defined to point to the BIND 9 application, which provides the server function for the Domain Name Protocol:

```
# cd /usr/sbin
# ls -l named
lrwxrwxrwx    1 root system 16 Aug 19 21:23 named -> /usr/sbin/named9
```

In previous AIX releases `/usr/sbin/named-xfer` is linked to the `/usr/sbin/named8-xfer` BIND 8 binary but because there is no equivalent program in BIND 9, the symbolic link `/usr/sbin/named-xfer` no longer exists on AIX 7 systems.

7.3 Network Time Protocol version 4

The Network Time Protocol (NTP) is an Internet protocol used to synchronize the clocks of computers to some time reference, usually the Coordinated Universal Time (UTC). NTP is an Internet standard protocol originally developed by Professor David L. Mills at the University of Delaware.

The NTP version 3 (NTPv3) Internet draft standard is formalized in the Request for Comments (RFC) 1305 (Network Time Protocol (Version 3) Specification,

Implementation and Analysis). NTP version 4 (NTPv4) is a significant revision of the NTP standard, and is the current development version. NTPv4 has not been formalized but is described in the proposed standard RFC 5905 (Network Time Protocol Version 4: Protocol and Algorithms Specification).

The NTP subnet operates with a hierarchy of levels, where each level is assigned a number called the stratum. Stratum 1 (primary) servers at the lowest level are directly synchronized to national time services. Stratum 2 (secondary) servers at the next higher level are synchronized to stratum 1 servers and so on. Normally, NTP clients and servers with a relatively small number of clients do not synchronize to public primary servers. There are several hundred public secondary servers operating at higher strata and they are the preferred choice.

According to a 1999 survey[1] of the NTP network there were at least 175,000 hosts running NTP on the Internet. Among these there were over 300 valid stratum 1 servers. In addition there were over 20,000 servers at stratum 2, and over 80,000 servers at stratum 3.

Beginning with AIX V7.1 and AIX V6.1 TL 6100-06 the AIX operating system supports NTP version 4 in addition to the older NTP version 3. The AIX NTPv4 implementation is based on the port of the ntp-4.2.4 version of the Internet Systems Consortium (ISC) code and is in full compliance with RFC 2030 (Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI).

Additional information about the Network Time Protocol project, the Internet Systems Consortium, and the Request for Comments can be found at:

    http://www.ntp.org/
    http://www.isc.org/
    http://www.rfcs.org/

As in previous AIX releases, the NTPv3 code is included with the `bos.net.tcp.client` fileset that is provided on the AIX product media and installed by default. The new NTPv4 functionality is delivered via the `ntp.rte` and the `ntp.man.en_US` filesets of the AIX Expansion Pack.

The `ntp.rte` fileset for the NTP runtime environment installs the following NTPv4 programs under the `/usr/sbin/ntp4` directory:

ntptrace4   Perl script that traces a chain of NTP hosts back to their master time source.

sntp4   SNTP client that queries an NTP server and displays the offset time of the system clock with respect to the server clock.

ntpq4   Standard NTP query program.

---

[1] Source: *A Survey of the NTP Network*, found at:
    http://alumni.media.mit.edu/~nelson/research/ntp-survey99

ntp-keygen4   Command that generates public and private keys.

ntpdc4   Special NTP query program.

ntpdate4   Sets the date and time using the NTPv4 protocol.

ntpd4   NTPv4 daemon.

System administrators can use the `lslpp` command to get a full listing of the `ntp.rte` fileset content:

```
7501lp01:sbin/ntp4> lslpp -f ntp.rte
  Fileset              File

  ----------------------------------------------------------------------
Path: /usr/lib/objrepos
  ntp.rte 6.1.6.0      /usr/lib/nls/msg/en_US/ntpdate4.cat
                       /usr/lib/nls/msg/en_US/ntpq4.cat
                       /usr/sbin/ntp4/ntptrace4
                       /usr/sbin/ntp4/sntp4
                       /usr/sbin/ntp4/ntpq4
                       /usr/sbin/ntp4/ntp-keygen4
                       /usr/sbin/ntp4/ntpdc4
                       /usr/sbin/ntp4/ntpdate4
                       /usr/lib/nls/msg/en_US/ntpdc4.cat
                       /usr/lib/nls/msg/en_US/ntpd4.cat
                       /usr/sbin/ntp4
                       /usr/lib/nls/msg/en_US/libntp4.cat
                       /usr/sbin/ntp4/ntpd4
```

The NTPv3 and NTPv4 binaries can coexist on an AIX system. The NTPv3 functionality is installed by default via the `bos.net.tcp.client` fileset and the commands are placed in the `/usr/sbin` subdirectory.

If the system administrator likes to use the NTPv4 services, all the commands will be in the `/usr/sbin/ntp4` directory after the NTPv4 code has been installed from the AIX Expansion Pack. Table 7-2 provides a list of the NTPv4 binaries and the NTPv3 binaries on AIX.

Table 7-2   NTP binaries directory mapping on AIX

NTPv4 binaries in /usr/sbin/ntp4	NTPv3 binaries in /usr/sbin
ntpd4	xntpd
ntpdate4	ntpdate
ntpdc4	xntpdc
ntpq4	ntpq

NTPv4 binaries in /usr/sbin/ntp4	NTPv3 binaries in /usr/sbin
ntp-keygen4	Not available
ntptrace4	ntptrace
sntp4	sntp

In comparison with the NTPv3 protocol, the utilization of NTPv4 offers improved functionality, and many new features and refinements. A comprehensive list that summarizes the differences between the NTPv4 and the NTPv3 versions is provided by the *NTP Version 4 Release Notes,* which can be found at:

http://www.eecis.udel.edu/~mills/ntp/html/release.html

The following list is an extract of the release notes that gives an overview of the new features pertaining to AIX.

► Support for the IPv6 addressing family. If the Basic Socket Interface Extensions for IPv6 (RFC 2553) is detected, support for the IPv6 address family is generated in addition to the default support for the IPv4 address family.

► Most calculations are now done using 64-bit floating double format, rather than 64-bit fixed point format. The motivation for this is to reduce size, improve speed, and avoid messy bounds checking.

► The clock discipline algorithm has been redesigned to improve accuracy, reduce the impact of network jitter and allow increase in poll intervals to 36 hours with only moderate sacrifice in accuracy.

► The clock selection algorithm has been redesigned to reduce *clockhopping* when the choice of servers changes frequently as the result of comparatively insignificant quality changes.

► This release includes support for Autokey public-key cryptography, which is the preferred scheme for authenticating servers to clients.

► The OpenSSL cryptographic library has replaced the library formerly available from RSA Laboratories. All cryptographic routines except a version of the MD5 message digest routine have been removed from the base distribution.

► NTPv4 includes three new server discovery schemes, which in most applications can avoid per-host configuration altogether. Two of these are based on IP multicast technology, while the remaining one is based on crafted DNS lookups.

► This release includes comprehensive packet rate management tools to help reduce the level of spurious network traffic and protect the busiest servers from overload.

- This release includes support for the orphan mode, which replaces the local clock driver for most configurations. Orphan mode provides an automatic, subnet-wide synchronization feature with multiple sources. It can be used in isolated networks or in Internet subnets where the servers or Internet connection have failed.

- There are two new burst mode features available where special conditions apply. One of these is enabled by the `iburst` keyword in the server configuration command. It is intended for cases where it is important to set the clock quickly when an association is first mobilized. The other is enabled by the `burst` keyword in the server configuration command. It is intended for cases where the network attachment requires an initial calling or training procedure.

- The reference clock driver interface is smaller, more rational, and more accurate.

- In all except a very few cases, all timing intervals are randomized, so that the tendency for NTPv3 to self-synchronize and bunch messages, especially with a large number of configured associations, is minimized.

- Several new options have been added for the `ntpd` command line. For the system administrators, several of the more important performance variables can be changed to fit actual or perceived special conditions. In particular, the `tinker` and `tos` commands can be used to adjust thresholds, throw switches and change limits.

- The `ntpd` daemon can be operated in a one-time mode similar to `ntpdate`, which will become obsolete over time.

8

Security, autentication, and authorization

This chapter is dedicated to the latest security topics as they apply to AIX V7.1. Topics include:

► 8.1, "Domain Role Based Access Control" on page 290

► 8.2, "Auditing enhancements" on page 345

► 8.3, "Propolice or Stack Smashing Protection" on page 352

► 8.4, "Security enhancements" on page 353

► 8.5, "Remote Statistic Interface (Rsi) client firewall support" on page 360

► 8.6, "AIX LDAP authentication enhancements" on page 360

► 8.7, "RealSecure Server Sensor" on page 362

8.1 Domain Role Based Access Control

The section discusses domain Role Based Access Control (RBAC).

This feature first became available in AIX V7.1 and is included in AIX 6.1 TL 06.

Domain RBAC is an enhancement to Enhanced Role Based Access Control, introduced in AIX V6.1.

8.1.1 The traditional approach to AIX security

The traditional approach to privileged administration in the AIX operating system has relied on a single system administrator account, named the root user.

The root user account is the superuser. It has the authority to perform all privileged system administration on the AIX system.

Using the root user, the administrator could perform day-to-day activities including, but not limited to, adding user accounts, setting user passwords, removing files, and maintaining system log files.

Reliance on a single superuser for all aspects of system administration raises issues with regard to the separation of administrative duties.

The root user allows the administrator to have a single point of administration when managing the AIX operating system, but in turn allows an individual to have unrestricted access to the operating system and its resources. While this freedom could be a benefit in day-to-day administration, it also has the potential to introduce security exposures.

While a single administrative account may be acceptable in certain business environments, some environments use multiple administrators, each with responsibility for performing different tasks.

Alternatively, in some environments, the superuser role is shared among two or more system administrators. This shared administrative approach may breach business audit guidelines in an environment that requires that all privileged system administration is attributable to a single individual.

Sharing administration functions may create issues from a security perspective.

With each administrator having access to the root user, there was no way to limit the operations that any given administrator could perform.

Since the root user is the most privileged user, the root user could perform operations and also be able to erase any audit log entries designed to keep track of these activities, thereby making the identification to an individual of the administrative actions impossible.

Additionally, if the access to the root user's password were compromised and an unauthorized individual accesses the root user, then that individual could cause significant damage to the systems' integrity.

Role Based Access Control offers the option to define roles to users to perform privileged commands based upon the user's needs.

8.1.2 Enhanced and Legacy Role Based Access Control

In this section we discuss the differences between the two operating modes of RBAC available in AIX, Legacy mode and Enhanced mode.

The release of AIX V6.1 saw the introduction of an enhanced version of Role Based Access Control (RBAC), which added to the version of RBAC already available in AIX since V4.2.1.

To distinguish between the two versions, the following naming conventions are used:

Enhanced RBAC   The enhanced version of RBAC introduced in AIX V6.1

Legacy RBAC   The version of RBAC introduced in AIX V4.2.1

The following is a brief overview of Legacy RBAC and Enhanced RBAC.

For more information on Role Based Access Control, see *AIX V6 Advanced Security Features Introduction and Configuration*, SG24-7430 at:

    http://www.redbooks.ibm.com/abstracts/sg247430.html?Open

Legacy RBAC

Legacy RBAC was introduced in AIX V4.2.1. The AIX security infrastructure began to provide the administrator with the ability to allow a user account other than the root user to perform certain privileged system administration tasks.

Legacy RBAC often requires that the command being controlled by an authorization have *setuid* to the root user in order for an authorized invoker to have the proper privileges to accomplish the operation.

The Legacy RBAC implementation introduced a predefined set of authorizations that can be used to determine access to administrative commands and could be expanded by the administrator.

Legacy RBAC includes a framework of administrative commands and interfaces to create roles, assign authorizations to roles, and assign roles to users.

The functionality of Legacy RBAC was limited because:

► The framework required changes to commands and applications for them to be RBAC enabled.

► The predefined authorizations were not granular.

► Users often required membership in a certain group as well as having a role with a given authorization in order to execute a command.

► A true separation of duties is difficult to implement. If a user account is assigned multiple roles, then all assigned roles are always active. There is no method to activate only a single role without activating all roles assigned to a user.

► The least privilege principle was not adopted in the operating system. Privileged commands must typically be *setuid* to the root user.

Enhanced RBAC

Beginning with AIX V6.1, Enhanced RBAC provides administrators with a method to delegate roles and responsibilities among one or more general user accounts.

These general user accounts may then perform tasks that would traditionally be performed by the root user or through the use of *setuid* or *setgid*.

The Enhanced RBAC integration options use granular privileges and authorizations and give the administrator the ability to configure any command on the system as a privileged command.

The administrator can use Enhanced RBAC to provide for a customized set of authorizations, roles, privileged commands, devices, and files through the Enhanced RBAC security database.

The Enhanced RBAC security database may reside either in the local file system or be managed remotely through LDAP.

Enhanced RBAC consists of the following security database files:

► Authorization Database

► Role Database

► Privileged Command Database

► Privileged Device Database

► Privileged File Database

Enhanced RBAC includes a granular set of system-defined authorizations and enables an administrator to create additional user-defined authorizations as necessary.

Both Legacy RBAC and Enhanced RBAC are supported on AIX V7.1.

Enhanced RBAC is enabled by default in AIX V7.1, but will not be active until the administrator configures the Enhanced RBAC functions.

Role Based Access Control may be configured to operate in either Legacy or Enhanced mode.

There is no specific install package in AIX V7.1 for Legacy or Enhanced mode RBAC because the majority of the Enhanced RBAC commands are included in the `bos.rte.security` fileset.

While Legacy RBAC is supported in AIX V7.1, administrators are encouraged to use Enhanced RBAC over Legacy RBAC.

Enhanced RBAC offers more granular control of authorizations and reduces the reliance upon *setuid* programs.

8.1.3 Domain Role Based Access Control

As discussed earlier, Enhanced RBAC provides administrators with a method to delegate roles and responsibilities to a non-root user, but Enhanced RBAC cannot provide the administrator with a mechanism to further limit those authorized users to specific system resources.

As an example, Enhanced RBAC could be used to authorize a non-root user to use the **chfs** command to extend the size of a JFS2 file system. After authorizing the non-root user, Enhanced RBAC could not limit the authorized non-root user to using the **chfs** command to extend only an individual or selected file system.

Domain RBAC introduces the *domain* into Role Based Access Control, a feature that allows the administrator to further restrict an authorized user to a specific resource.

With the introduction of Enhanced RBAC in AIX V6.1 the administrator was offered a granular approach to managing roles and responsibilities.

With the introduction of Domain RBAC, the granularity is further extended to allow finer control over resources.

Domain RBAC requires that Enhanced RBAC be enabled. Domain RBAC will not operate in the Legacy RBAC framework.

> **Note:** Unless noted, further references to RBAC will refer to Enhanced RBAC, as Domain RBAC does not operate under Legacy RBAC.

Example 8-1 shows the **lsattr** command being used to determine whether Enhanced RBAC is enabled on an AIX V7.1 partition. The enhanced_RBAC true attribute shows that enhanced RBAC is enabled.

Example 8-1   Using the lsattr command to display the enhanced_RBAC status

```
# oslevel -s
7100-00-00-0000
# lsattr -El sys0 -a enhanced_RBAC
enhanced_RBAC true Enhanced RBAC Mode True
#
```

The enhanced_RBAC attribute may be enabled or disabled with the **chdev** command. If Enhanced RBAC is not enabled on your partition, it may be enabled by using the **chdev** command to change the sys0 device.

Example 8-2 shows the **chdev** command being used to change the enhanced_RBAC attribute from false to true.

Example 8-2   Using the chdev command to enable the enhanced_RBAC attribute

```
# lsattr -El sys0 -a enhanced_RBAC
enhanced_RBAC false Enhanced RBAC Mode True
# chdev -l sys0 -a enhanced_RBAC=true
sys0 changed
# lsattr -El sys0 -a enhanced_RBAC
enhanced_RBAC true Enhanced RBAC Mode True
# shutdown -Fr

SHUTDOWN PROGRAM
Thu Sep 16 11:00:50 EDT 2010
Stopping The LWI Nonstop Profile...
Stopped The LWI Nonstop Profile.
0513-044 The sshd Subsystem was requested to stop.

Wait for 'Rebooting...' before stopping.
Error reporting has stopped.
```

> **Note:** Changing the enhanced_RBAC attribute will require a reboot of AIX for the change to take effect.

At the time of publication, Domain RBAC functionality is not available on Workload Partition (WPAR).

Domain RBAC definitions

Domain RBAC introduces new concepts into the RBAC security framework.

Subject   A *subject* is defined as an entity that requires access to another entity. A subject is an initiator of an action. An example of a subject would be a process accessing a file. When the process accesses the file, the process is considered a subject. A user account may also be a subject when the user account has been granted association with a domain.

Object   An *object* is an entity that holds information that can be accessed by another entity. The object is typically accessed by a *subject* and is typically the target of the action. The object may be thought of as the entity on which the action is being performed. As an example, when process 2001 tries to access another process, 2011, to send a signal then process 2001 is the subject and process 2011 is the object.

Domain   A *domain* is defined as a category to which an entity may belong. When an entity belongs to a domain, access control to the entity is governed by a rule set that is known as a *property*. An entity could belong to more than one domain at a time. Each domain has a unique numeric domain identifier. A maximum of 1024 domains are allowed, with the highest possible value of the domain identifier allowed as the number 1024. A user account may belong to a domain. When a user account belongs to a domain, it can be described as having an association with a domain.

Property   A *property* is the rule set that determines whether a subject is granted access to an object.

Conflict Set   A *conflict set* is a domain object attribute that restricts access to a domain based upon the existing domain access that an entity may already have defined. This is further explained when discussing the `setsecattr` command, later in this section.

Security Flag   A *security flag* is a domain object attribute that may restrict access to an object based upon the `FSF_DOM_ANY` or `FSF_DOM ALL` attribute. When the `secflags` attribute is set to `FSF_DOM_ANY` a subject may access the object when it is associated with any of the domains specified in the `domains` attribute. When the `secflags` attribute is `FSF_DOM_ALL`, a subject may access the object only when it is associated with all of the `domains` specified in the attribute. The default `secflags` value is

FSF_DOM_ALL. If no `secflags` attribute value is specified, then the default value of FSF_DOM_ALL is used.

In Example 8-3 we see the **ps** command being used to display the process identifier assigned to the **vi** command. The **vi** command is being used by the root user to edit a file named /tmp/myfile.

Example 8-3   Using the ps command to identify the process editing /tmp/myfile

```
# cd /tmp
# pwd
/tmp
# ls -ltra myfile
-rw-r--r--    1 root      system             15 Sep 02 11:58 myfile
# ps -ef|grep myfile
    root 6226020 6488264   0 11:59:42  pts/1  0:00 vi myfile
# ps -fT 6226020
     UID     PID    PPID   C    STIME    TTY  TIME CMD
    root 6226020 6488264   0 11:59:42  pts/1  0:00 vi myfile
#
```

In Example 8-3 we see an example of the subject and the object.

▶ The *subject* is process id 6226020, which is a process that is executing the **vi** command to edit the file named /tmp/myfile.

▶ The *object* is the file named /tmp/myfile.

8.1.4 Domain RBAC command structure

Domain RBAC introduces four new commands into the RBAC framework.

These are the **mkdom**, **lsdom**, **chdom** and **rmdom** commands.

The mkdom command

The **mkdom** command creates a new RBAC domain.

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

mkdom [Attribute = Value ...] Name

The **mkdom** command creates a new domain in the domain database. The domain attributes can be set during the domain creation phase by using the Attribute = Value parameter.

The domain database is located in the /etc/security/domains file.

The **mkdom** command has the following requirements:

► The system must be operating in the Enhanced Role Based Access Control (RBAC) mode.

► Modifications made to the domain database are not available for use until updated into the Kernel Security Tables (KST) with the **setkst** command.

► The **mkdom** command is a privileged command. Users of this command must have activated a role with the *aix.security.domains.create* authorization or be the root user.

Example 8-4 shows the **mkdom** command being used by the root user to create a new domain named Network with a domain identifier (Domain ID) of 22:

Example 8-4   Using the mkdom command to create the domain Network with a Domain ID of 22

```
# mkdom id=22 Network
# lsdom Network
Network id=22
#
```

> **Note:** The **mkdom** command will not return with text output when a domain is successfully created. The **lsdom** command was used in Example 8-4 to display that the **mkdom** command did successfully create the Network domain. The **lsdom** command is introduced next.

The **mkdom** command contains character usage restrictions. For a full listing of these character restrictions, see the **mkdom** command reference.

The lsdom command

The **lsdom** command displays the domain attributes of an RBAC domain.

The domain database is located in the /etc/security/domains file.

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

lsdom [-C] [-f] [-a Attr [Attr]...] { ALL | Name [, Name] ...}

The **lsdom** command lists the attributes of either all domains or specific domains.

The **lsdom** command lists all domain attributes. To view selected attributes, use the **lsdom -a** command option.

The `lsdom` command can list the domain attributes in the following formats:

► List domain attributes on one line with the attribute information displayed as `Attribute = Value`, each separated by a blank space. This is the default list option.

► To list the domain attributes in stanza format, use the `lsdom -f` command flag.

► To list the information as colon-separated records, use the `lsdom -C` command flag.

The `lsdom` command has the following domain name specification available:

ALL   Indicates that all domains will be listed, including the domain attributes.

Name   Indicates the name of the domain that will have the attributes listed. This may be multiple domain names, comma separated.

The `lsdom` command has the following requirements:

► The system must be operating in the Enhanced Role Based Access Control (RBAC) mode.

► The `lsdom` command is a privileged command. Users of this command must have activated a role with the *aix.security.domains.list* authorization or be the root user.

Example 8-5 shows the `lsdom -f` command being used by the root user to display the `DBA` and `HR` domains in stanza format.

Example 8-5   Using the lsdom command -f to display the DBA and HR domains in stanza format

```
# lsdom -f DBA,HR
DBA:
        id=1

HR:
        id=2

#
```

The chdom command

The `chdom` command modifies attributes of an existing RBAC domain.

The syntax of the `chdom` command is:

```
chdom Attribute = Value ... Name
```

If the specified attribute or attribute value is invalid, the **chdom** command does not modify the domain.

The **chdom** command has the following requirements:

► The system must be operating in Enhanced Role Based Access Control (RBAC) mode.

► Modifications made to the domain database are not available for use until updated into the Kernel Security Tables with the **setkst** command.

► The **chdom** command is a privileged command. Users of this command must have activated a role with the *aix.security.dom.change* authorization or be the root user.

Example 8-6 shows the **chdom** command being used by the root user to change the ID of the Network domain from 22 to 20. The Network domain was created in Example 8-4 on page 297 and has not yet been used and is not associated with any entities.

Example 8-6   Using the chdom command to change the ID attribute of the Network domain

```
# lsdom -f Network
Network:
        id=22

# chdom id=20 Network
# lsdom -f Network
Network:
        id=20

#
```

Note: Modification of the ID attribute of a domain can affect the security aspects of the system, as processes and files might be using the current value of the ID.

Modify the ID of a domain only if the domain has not been used, else the security aspects of the system could be adversly effected.

The rmdom command
The **rmdom** command removes an RBAC domain.

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

```
rmdom Name
```

The **rmdom** command removes the domain that is identified by the Name parameter. It only removes the existing domains from the domain database.

A domain that is referenced by the domain object database cannot be removed until you remove the references to the domain.

The **rmdom** command has the following requirements:

► The system must be operating in Enhanced Role Based Access COntrol (RBAC) mode.

► Modifications made to the domain database are not available for use until updated into the Kernel Security Tables with the **setkst** command.

► The **rmdom** command is a privileged command. Users of this command must have activated a role with the *aix.security.dom.remove* authorization or be the root user.

Example 8-7 shows the **rmdom** command being used by the root user to remove the Network domain. The Network domain has not yet been used and is not with any entities.

By using the **lssecattr -o ALL** command, we can see that there are no domain objects referenced by the Network domain, so the Network domain may be removed.

Example 8-7   Using the rmdom command to remove the Network domain

```
# lsdom -f Network
Network:
        id=22

# lssecattr -o ALL
/home/dba/privatefiles domains=DBA conflictsets=HR objtype=file
secflags=FSF_DOM_ANY
# rmdom Network
# lsdom -f Network
3004-733 Role "Network" does not exist.
# lsdom ALL
DBA id=1
HR id=2
#
```

> **Note:** If a user account belonged to the Network domain, the user account would still see the domains=Network attribute listed from the **lsuser** output. This domains=Network attribute value can be removed with the **chuser** command.

In addition to the `mkdom`, `lsdom`, `chdom`, and `rmdom` commands, domain RBAC introduces enhanced functionality to the existing commands, shown in Table 8-1.

Table 8-1   Domain RBAC enhancements to existing commands

Command	Description	New Functionality
`setsecattr`	Add or modify the domain attributes for objects	-o
`lssecattr`	Display the domain attributes for objects	-o
`rmsecattr`	Remove domain object definitions	-o
`setkst`	Read the security databases and load the information from the databases into the kernel security tables	The option to download the domain and the domain object databases
`lsuser`	List user attributes	The attribute domain is added for users
`lssec`	List user attributes	The attribute domain is added for users
`chuser`	Change user attributes	The attribute domain is added for users
`chsec`	Change user attributes	The attribute domain is added for users

The Domain RBAC enhanced functionality to the commands in Table 8-1 is further explained in the following examples.

The setsecattr command

The `setsecattr` command includes the **-o** flag. It is used to add and modify domain attributes for objects. An example of the `setsecattr` command is shown in Example 8-8.

Example 8-8   The setsecatrr -o command

```
# setsecattr -o domains=DBA conflictsets=HR objtype=file \
secflags=FSF_DOM_ANY /home/dba/privatefiles
#
```

As discussed earlier, domain RBAC introduces the *conflict set* and *security flag* object attributes into the RBAC framework.

The *conflict set* attribute can deny access to an object based upon existing domain association. When used, the `conflictsets` attribute would be set to a domain name other than the domain defined in the `domains` attribute.

In Example 8-8 the `conflictsets` attribute is defined as `HR` and the `domains` attribute as `DBA`. Both `HR` and `DBA` are names of domains defined in the RBAC security database.

Using the `conflictsets` attribute in this manner will restrict access to the `/home/dba/privatefiles` object by entities that have an association with the `HR` domain, regardless of whether these entities have membership to the `DBA` domain.

Example 8-9 shows the **lssecattr** and the **ls -ltra** commands being used to display the attributes of the file named `/home/dba/privatefiles`.

Example 8-9   Using the lssecattr and ls -ltra command to display the file named /home/dba/privatefiles

```
# cd /home/dba
# lssecattr -o privatefiles
/home/dba/privatefiles domains=DBA conflictsets=HR \
objtype=file secflags=FSF_DOM_ANY
# ls -ltra /home/dba/privatefiles
-rw-r--r--    1 dba       staff             33 Sep 03 11:18 privatefiles
# lssec -f /etc/security/user -s dba -a domains
dba domains=DBA
# lssecattr -o /home/dba/example111
"/home/dba/example111" does not exist in the domained object database.
#
```

From the output in Example 8-9 we can determine that:

▶ The **lssecattr** command shows that the file named `/home/dba/privatefiles` is defined as a domain RBAC object. If the file was not defined as a domain RBAC object, the output returned would be similar to the response from the **lssecattr -o /home/dba/example111** command which returned `"/home/dba/example111" does not exist in the domained object database`.

▶ The **lssecattr** command shows that the `domains` attribute is defined as the `DBA` domain and the `conflictsets` attribute is defined as the `HR` domain.

▶ The **lssecattr** command shows `secflags=FSF_DOM_ANY`. In this example, `FSF_DOM_ANY` does not offer any further restriction because the domain RBAC object `/home/dba/privatefiles` is defined with only a single domain.

- The `ls -ltra` command shows that the dba user account has read and write access to the file named /home/dba/privatefiles through Discretionary Access Control (DAC).

- The `lssec` command shows that the dba user account has been granted association to the DBA domain but has not been granted association to the HR domain, because only the DBA domain is returned in the domains=DBA listing.

By using the combination of `conflictsets` and `domains` in Example 8-9 on page 302 the dba user account would be able to access the file named /home/dba/privatefiles.

If the dba user account was to be granted association to the HR domain, then the dba user account would no longer be able to access the file named /home/dba/privatefiles because the HR domain is defined as a *conflict set* to the domain RBAC object /home/dba/privatefiles.

The access to the file named /home/dba/privatefiles would be refused even though the dba user has read and write access to the file via DAC.

The `secflags=FSF_DOM_ANY` attribute sets the behavior of the `domains` attribute of the object. In Example 8-9 on page 302 the object /home/dba/privatefiles is defined with only the DBA domain.

If the object /home/dba/privatefiles had been defined to multiple domains, and the `secflags` attribute been set as `FSF_DOM_ALL`, then the dba user account would have to be associated with all domains defined in the `domains` attribute for the /home/dba/privatefiles object, else access to the /home/dba/privatefiles would be denied.

The lssecattr command

The `lssecattr` command now includes the **-o** flag. It is used to display the domain attributes for *objects*. An example of the `lssecattr` command is shown in Example 8-10.

Example 8-10   The lssecattr -o command

```
# lssecattr -o /home/dba/privatefiles
/home/dba/privatefiles domains=DBA conflictsets=HR objtype=file \
secflags=FSF_DOM_ANY
#
```

The rmsecattr command

The `rmsecattr` command now includes the `-o` flag. It is used to remove domain object definitions from the RBAC security database. An example of the `rmsecattr` command is shown in Example 8-11.

Example 8-11   The rmsecattr -o command

```
# rmsecattr -o /home/dba/privatefiles
#
```

The setkst command

The `setkst` command is used to read the security database and load the security databases into the kernel security tables (KST).

It includes the option to load the domain and the domain object database.

The domain and domain object database are located in the /etc/security directory in the following files:

The domains file	The domain security database. To update the domain security database into the KST, use the `setkst -t dom` command.
The domobj file	The domain object security database. To update the domain object security database into the KST, use the `setkst -t domobj` command.

An example of the `setkst` command is shown in Example 8-12.

Example 8-12   The setkst -t command updating the domain into the KST

```
# setkst -t dom
Successfully updated the Kernel Domains Table.
#
```

Note: Changes made to the RBAC database are not activated into the Kernel Security Table (KST) until such time as the `setkst` command is executed.

The lskst command

The `lskst` command lists the entries in the Kernel Security Tables (KST). It includes the option to list the domain and the domain object database.

An example of the **lskst** command is shown in Example 8-13.

Example 8-13   Listing the kernel security tables with the lskst -t command

```
# lskst -t domobj
/home/dba/privatefiles objtype=FILE domains=DBA \
conflictsets=HR secflags=FSF_DOM_ANY
#
```

The lsuser command

The **lsuser** command includes the option to display the domains to which a user has association. An example of the **lsuser** command is shown in Example 8-14.

Example 8-14   The lsuser -a command - display a user domain access

```
# lsuser -a domains dba
dba domains=DBA
#
```

The lssec command

As with the **lsuser** command, the **lssec** command includes the option to display the domains to which a user has an association. An example of the **lssec** command is shown in Example 8-15.

Example 8-15   The lssec -f command - display a user domain access

```
# lssec -f /etc/security/user -s dba -a domains
dba domains=DBA
#
```

The chuser command

The **chuser** command includes the option to change the domains to which a user has an association. An example of the **chuser** command is shown in Example 8-16.

Example 8-16   The chuser command - change a user domain association

```
# lsuser -a domains dba
dba domains=DBA
# chuser domains=HR dba
# lsuser -a domains dba
dba domains=HR
#
```

To remove all domains to which a user has an association, the **chuser** command can be used without any `domain` attribute, as shown in Example 8-17.

Example 8-17   The chuser command - remove all domain association from a user

```
# lsuser -a domains dba
dba domains=HR
# chuser domains= dba
# lsuser -a domains dba
dba
# lssec -f /etc/security/user -s dba -a domains
dba domains=
#
```

Example 8-17 shows the different outputs returned by the **lssec -f** and **lsuser -a** commands.

The chsec command

As with the **chuser** command, the **chsec** command includes the option to change the domains to which a user has an association. An example of the **chsec** command is shown in Example 8-18.

Example 8-18   The chsec command - adding DBA domain access to the dba user

```
# lssec -f /etc/security/user -s dba -a domains
dba domains=
# chsec -f /etc/security/user -s dba -a domains=DBA
# lssec -f /etc/security/user -s dba -a domains
dba domains=DBA
#
```

8.1.5 LDAP support in Domain RBAC

The Enhanced RBAC security database may reside either in the local file system or be managed remotely through LDAP.

At the time of publication the domain RBAC databases must reside locally in the `/etc/security` directory.

When upgrading an LPAR that is using RBAC with LDAP authentication, the LDAP authentication will remain operational. Any domain RBAC definitions will reside locally in the `/etc/security` directory.

The `/etc/nscontrol.conf` file contains the location and lookup order for the RBAC security database.

Example 8-19 shows the RBAC security database stanza output of the `/etc/nscontrol.conf` file.

The `secorder` attribute describes the location of the security database file. It is possible to store the Enhanced RBAC security database files either in the `/etc/security` directory or on an LDAP server, or a combination of the two.

Domain RBAC security database files are only stored in the `/etc/security` directory, so they will not have a stanza in the `/etc/nscontrol.conf` file.

The options for the `secorder` attribute are:

files   The database file is located in the `/etc/security` directory. This is the default location.

LDAP   The database file is located on an LDAP server.

LDAP, files   The database file is located on the LDAP server and the `/etc/security` directory. The lookup order is LDAP first, followed by the `/etc/security` directory

files, LDAP   The database file is located in the `/etc/security` directory and the LDAP server. The lookup order is the `/etc/security` directory first, followed by the LDAP server.

Example 8-19   The /etc/nscontrol.conf file

```
# more /etc/nscontrol.conf
# IBM_PROLOG_BEGIN_TAG
# This is an automatically generated prolog.
#
output ommitted .....
#
authorizations:
        secorder = files

roles:
        secorder = files

privcmds:
        secorder = files

privdevs:
        secorder = files
```

```
privfiles:
        secorder = files
#
```

Example 8-19 on page 307 shows that the five files in the Enhanced RBAC security database are stored in the `/etc/security` directory and LDAP is not being used for RBAC on this server.

8.1.6 Scenarios

This section introduces four scenarios to describe the usage of the new features available in domain RBAC.

The four scenarios consist of:

Device scenario   Using domain RBAC to control privileged command execution on logical volume devices.

File scenario   Two scenarios. Using domain RBAC to restrict user access and to remove user access to a file.

Network scenario   Use domain RBAC to restrict privileged access to a network interface.

These four scenarios show examples of how domain RBAC may be used to provide additional functionality to the AIX security framework.

The AIX partition used in the scenario:

► Has AIX V7.1 installed.

► Is operating in `Enhanced_RBAC` mode.

► Has no additional or customized RBAC roles or authorizations defined.

► Has no previous domain RBAC customizing defined.

> **Note:** At the time of publication, Domain RBAC may be managed through the command line only. Domain RBAC support is not included in the System Management Interface Tool (SMIT).

Device scenario

Domain RBAC allows the administrator to define devices as domain RBAC objects.

In this scenario, logical volume devices will be defined as domain RBAC objects.

The AIX V7.1 LPAR consists of two volume groups, `rootvg` and `appsvg`.

The `appsvg` group contains application data, which is supported by the application support team by using the `appuser` user account.

The application support team has requested the ability to add/modify and delete the four file systems used by the application.

The application file systems reside exclusively in a volume group named `appsvg`.

The systems administrator will grant the application support team the ability to add/modify/delete the four application file systems in the `appsvg` volume group, but restrict add/modify/delete access to all other file systems on the LPAR.

Enhanced RBAC allows the systems administrator to grant the application support team the privileges to add/modify/delete the four file systems without having to grant access to the root user.

Enhanced RBAC does not allow the systems administrator to restrict access to only those four file systems needed by the application support team.

Domain RBAC will allow such a granular separation of devices and allow the systems administrator to allow add/modify/delete access to only the four application file systems and restrict add/modify/delete access to the remaining file systems.

The system administrator identifies that the application support team requires access to the following AIX privileged commands.

crfs   Create a new file system

chfs   Modify an existing file system

rmfs   Remove an existing file system

mount   Mount a file systems

unmount   Unmount a file system

With the privileged commands identified, the administrator defines an RBAC role to allow the application support team to perform these five privileged commands.

Unless noted otherwise, all commands in the scenario will be run as the root user.

AIX includes predefined RBAC roles, one of which is the FSAdmin role. The FSAdmin role includes commands that may be used to manage file systems and could be used in this situation.

In this scenario the administrator creates a new RBAC role, named apps_fs_manage, using the **mkrole** command.

The benefits in creating the apps_fs_manage role are:

► This introduces an example of using the **mkrole** command used in Enhanced RBAC.

► The apps_fs_manage role includes only a subset of the privileged commands included in the FSAdmin role. This complies with the Least Privilege Principal.

Before using the **mkrole** command to create the apps_fs_manage role, the administrator must determine the access authorizations required by each of the commands that will be included in the apps_fs_manage role.

The **lssecattr** command is used to determine the access authorizations.

Example 8-20 shows the **lssecattr** command being used to determine the access authorizations of each of the five privileged commands that will be included in the apps_fs_manage role.

Example 8-20   Using the lssecattr command to identify command authorizations

```
# id
uid=0(root) gid=0(system)
groups=2(bin),3(sys),7(security),8(cron),10(audit),11(lp)
# lssecattr -c -a accessauths /usr/sbin/crfs
/usr/sbin/crfs accessauths=aix.fs.manage.create
# lssecattr -c -a accessauths /usr/sbin/chfs
/usr/sbin/chfs accessauths=aix.fs.manage.change
# lssecattr -c -a accessauths /usr/sbin/rmfs
/usr/sbin/rmfs accessauths=aix.fs.manage.remove
# lssecattr -c -a accessauths /usr/sbin/mount
/usr/sbin/mount accessauths=aix.fs.manage.mount
# lssecattr -c -a accessauths /usr/sbin/umount
/usr/sbin/umount accessauths=aix.fs.manage.unmount
#
```

Example 8-20 shows that the privileged commands require the following access authorizations:

crfs   Requires the access authorization aix.fs.manage.create.

chfs   Requires the access authorization aix.fs.manage.change.

rmfs   Requires the access authorization aix.fs.manage.remove.

mount   Requires the access authorization aix.fs.manage.mount.

unmount   Requires the access authorization aix.fs.manage.unmount.

At this stage, the administrator has identified the privileged commands required by the application support team, decided on the name of the RBAC role to be created, and determined the access authorizations required for the five privileged commands.

The administrator may now create the apps_fs_manage RBAC role with the **mkrole** command.

Example 8-21 shows the **mkrole** command being used to create the RBAC role named apps_fs_manage.

Example 8-21   Using the mkrole command - create the apps_fs_manage role

```
# id
uid=0(root) gid=0(system) groups=2(bin),3(sys),7(security),8(cron),10(audit),11(lp)
# mkrole authorizations=aix.fs.manage.create,aix.fs.manage.change,/
aix.fs.manage.remove,/aix.fs.manage.mount,aix.fs.manage.unmount/ dfltmsg='Manage apps
filesystems' apps_fs_manage
# lsrole apps_fs_manage
apps_fs_manage authorizations=aix.fs.manage.create,aix.fs.manage.change,/
aix.fs.manage.remove,aix.fs.manage.mount,aix.fs.manage.unmount rolelist= groups= visibility=1
screens=* dfltmsg=Manage apps filesystems msgcat= auth_mode=INVOKER id=11
#
```

> **Note:** The **smitty mkrole** fastpath may also be used to create an RBAC role. Due to the length of the authorization definitions, using the **smitty mkrole** fastpath may be convenient when multiple access authorizations are included in a role.

Once the apps_fs_manage role has been created, the role must be updated into the Kernel Security Tables (KST) with the **setkst** command. The role is not available for use until the **setkst** command updates the changes into the KST.

In Example 8-22 we see the **lsrole** command being used to list the apps_fs_manage role.

The **lsrole** command output shows that the apps_fs_manage role exists in the RBAC database, but when the **swrole** command is used to switch to the role, the role switching is not allowed.

This is because the apps_fs_manage role has not been updated into the KST.

The administrator can verify this by using the **lskst** command.

The **lskst** command lists the KST, whereas the **lsrole** command lists the contents of the RBAC security database in the /etc/security directory.

Example 8-22 shows the usage of the `lsrole`, `swrole` and `lskst` commands.

Example 8-22   Using the lsrole, swrole, and lskst commands

```
# lsrole apps_fs_manage
apps_fs_manage authorizations=aix.fs.manage.create,aix.fs.manage.change,/
aix.fs.manage.remove,aix.fs.manage.mount,aix.fs.manage.unmount rolelist= groups= visibility=1
screens=* dfltmsg=Manage apps filesystems msgcat= auth_mode=INVOKER id=11
# swrole apps_fs_manage
swrole: 1420-050 apps_fs_manage is not a valid role.
# lskst -t role apps_fs_manage
3004-733 Role "apps_fs_manage" does not exist.
#
```

In Example 8-23 we use the `setkst` command to update the KST with the changes made to the RBAC security database.

The `setkst` command may be run without any options or with the `setkst -t` option.

The `setkst -t` command allows the KST to be updated with only a selected RBAC database table or tables.

Example 8-23 shows the `setkst -t` command being used to update the KST with only the RBAC `role` database information.

Example 8-23   The setkst -t command - updating the role database into the KST

```
# lskst -t role apps_fs_manage
3004-733 Role "apps_fs_manage" does not exist.
# setkst -t role
Successfully updated the Kernel Role Table.
# lskst -t role -f apps_fs_manage
apps_fs_manage:

authorizations=aix.fs.manage.change,aix.fs.manage.create,aix.fs.manage.mount,/
aix.fs.manage.remove,aix.fs.manage.unmount
        rolelist=
        groups=
        visibility=1
        screens=*
        dfltmsg=Manage apps filesystems
        msgcat=
        auth_mode=INVOKER
        id=11
#
```

After updating the KST, the `appuser` account must be associated with the `apps_fs_manage` role.

Use the **lsuser** command to display whether any roles have previously been associated with the `appuser` account.

In this case, the `appuser` account has no role associations defined, as can be seen from the **lsuser** command output in Example 8-24.

If the `appuser` account had existing roles associated, the existing roles would need to be included in the **chuser** command along with the new `apps_fs_manage` role.

The **chuser** command is used in Example 8-24 to associate the `appuser` account with the `apps_fs_manage` role.

Example 8-24   The lsuser and chuser commands - assigning the apps_fs_manage role to the appuser account with the chuser command

```
# lsuser -a roles appuser
appuser roles=
# chuser roles=apps_fs_manage appuser
# lsuser -a roles appuser
appuser roles=apps_fs_manage
#
```

At this stage, the administrator has completed the steps required to grant the `appuser` account the ability to use the **crfs**, **chfs**, **rmfs**, **mount** and **unmount** commands. Even though these privileged commands could normally only be executed by the root user, the RBAC framework allows a non-privileged user to execute these commands, once the appropriate access authorizations and roles have been created and associated.

To demonstrate this, the `appuser` account uses the **chfs** and **umount** commands.

Example 8-25 shows the `appuser` account login and uses the **rolelist** command to display to which RBAC roles it has an association with and whether the role is effective.

A role that is active on the user account is known as the effective role.

Example 8-25   Using the rolelist -a and rolelist -e commands

```
$ id
uid=301(appuser) gid=202(appgroup) groups=1(staff)
$ rolelist -a
apps_fs_manage  aix.fs.manage.change
```

```
                aix.fs.manage.create
                aix.fs.manage.mount
                aix.fs.manage.remove
                aix.fs.manage.unmount
$ rolelist -e
rolelist: 1420-062 There is no active role set.
$
```

From the **rolelist -a** and **rolelist -e** output you can determine that the
appuser has been associated with the apps_fs_manage role, but the role is not
currently the effective role.

Use the **swrole** command to switch to the apps_fs_manage role.

Once the **swrole** command is used to switch to the apps_fs_manage role, the role
becomes the effective role, allowing the appuser account to perform the
privileged commands defined in the apps_fs_manage role.

Example 8-26 shows the appuser account using the **swrole** command to switch
to the apps_fs_manage role.

*Example 8-26   The appuser account using the swrole command to switch to the
apps_fs_manage role*

```
$ ps
     PID    TTY  TIME CMD
 7995462  pts/0  0:00 -ksh
 9633860  pts/0  0:00 ps
$ swrole apps_fs_manage
appuser's Password:
$ rolelist -e
apps_fs_manage  Manage apps filesystems
$ ps
     PID    TTY  TIME CMD
 7995462  pts/0  0:00 -ksh
 9044098  pts/0  0:00 ps
 9240642  pts/0  0:00 ksh
$
```

Note: The **swrole** command requires authentication with the user's password
credentials.

The **swrole** command initiates a new shell, which can be seen with the new
PID 940642, displayed in the **ps** command output.

The `appuser` account may now execute the privileged commands in the apps_fs_manage role.

In Example 8-27 the `appuser` account uses the **chfs** command to add 1 GB to the /apps04 file system.

Example 8-27   The appuser account using the chfs command to add 1 GB to the /apps04 file system

```
$ id
uid=301(appuser) gid=202(appgroup) groups=1(staff)
$ df -g /apps04
Filesystem    GB blocks     Free %Used    Iused %Iused Mounted on
/dev/appslv_04    1.25     0.18   86%      15     1% /apps04
$ chfs -a size=+1G /apps04
Filesystem size changed to 4718592
$ df -g /apps04
Filesystem    GB blocks     Free %Used    Iused %Iused Mounted on
/dev/appslv_04    2.25     1.18   48%      15     1% /apps04
$
```

The `appuser` was successful in using the **chfs** command to add 1 GB to the /apps04 file system.

The RBAC role allows the `appuser` account to execute the **chfs** command. This is the expected operation of the RBAC role.

In Example 8-28 the `appuser` account uses the **unmount** command to unmount the /apps01 file system.

Example 8-28   The appuser account using the umount command to unmount the /apps01 file system

```
$ df -g /apps01
Filesystem    GB blocks     Free %Used    Iused %Iused Mounted on
/dev/appslv_01    1.25     0.18   86%      15     1% /apps01
$ unmount /apps01
$ df -g /apps01
Filesystem    GB blocks     Free %Used    Iused %Iused Mounted on
/dev/hd4        0.19     0.01   95%    9845    77% /
$lslv appslv_01
LOGICAL VOLUME:     appslv_01              VOLUME GROUP:   appsvg
LV IDENTIFIER:      00f61aa600004c000000012aee536a63.1 PERMISSION:
read/write
VG STATE:          active/complete        LV STATE:       closed/syncd
TYPE:              jfs2                   WRITE VERIFY:   off
```

```
MAX LPs:              512                  PP SIZE:       64
megabyte(s)
COPIES:               1                    SCHED POLICY:  parallel
LPs:                  36                   PPs:           36
STALE PPs:            0                    BB POLICY:     relocatable
INTER-POLICY:         minimum              RELOCATABLE:   yes
INTRA-POLICY:         middle               UPPER BOUND:   32
MOUNT POINT:          /apps01              LABEL:         /apps01
MIRROR WRITE CONSISTENCY: on/ACTIVE
EACH LP COPY ON A SEPARATE PV ?: yes
Serialize IO ?:       NO
$
```

In Example 8-28, the appuser was successfully able to use the **unmount** command to **unmount** the /apps01 file system. By using the **df** and the **lslv** commands, we can determine that the /apps01 file system has been unmounted.

The RBAC role is allowing the appuser account to execute the **unmount** command. This is the expected operation of the RBAC role.

By using RBAC, the administrator has been able to grant the appuser account access to selected privileged commands. This has satisfied the request requirements of the application support team, because the appuser may now manage the four file systems in the appsvg.

Prior to domain RBAC, there was no RBAC functionality to allow the administrator to grant a user privileged access to only selected devices. For example, if privileged access was granted to the **chfs** command, the privilege could be used to change the attributes of all file systems.

This meant that there was no way to prevent a user-granted privileged access to the **chfs** command from accessing or modifying file systems to which they may not be authorized to access or administer.

The /backup file system was not a file system to which the appuser account requires privileged access, but because the appuser account has been granted privileged access to the **chfs** command, the administrator is unable to use Enhanced RBAC to limit the file systems that the appuser may modify.

In Example 8-29 we see the appuser account using the **chfs** command to add 1 GB to the /backup file system.

Example 8-29   The appuser account using the chfs command to change the /backup file system

```
$ id
```

```
uid=301(appuser) gid=202(appgroup) groups=1(staff)
$ df -g /backup
Filesystem    GB blocks      Free %Used    Iused %Iused Mounted on
/dev/backup_lv    1.25       1.15    8%        5    1% /backup
$ chfs -a size=+1G /backup
Filesystem size changed to 4718592
$ df -g /backup
Filesystem    GB blocks      Free %Used    Iused %Iused Mounted on
/dev/backup_lv    2.25       2.15    5%        5    1% /backup
$
```

The `appuser` account was able to modify the `/backup` file system because the `apps_fs_manage` role includes the access authorization for the **chfs** command.

The RBAC role is functioning correctly, but does not offer the functionality to limit the **chfs** command execution to only selected file systems.

Domain RBAC introduces the domain into Role Based Access Control.

The domain allows the administrator to further granualize the privileged command execution by limiting access to system resources to which a user may be granted privileged command execution.

The administrator will now use domain RBAC to:

1. Create two RBAC domains

2. Create multiple domain RBAC objects

3. Update the Kernel Security Tables (KST)

4. Associate the RBAC domain to the `appuser` account

5. Attempt to use the **chlv** command to change the `/apps04` and `/backup` file systems

Firstly, the administrator creates two RBAC domains:

applvDom   This domain will be used to reference the `/apps01`, `/apps02`, `/apps03` and `/apps04` file systems.

privlvDom   This domain will be used to restrict access to the file systems that the `appuser` may access.

Note: RBAC domain names do have to be in mixed case. Mixed case has been used in this scenario as an example.

Example 8-30 shows the **mkdom** command being used by the root user to create the applvDom and privlvDom domains.

Example 8-30   The mkdom command - creating the applvDom and privlvDom domains

```
# id
uid=0(root) gid=0(system)
groups=2(bin),3(sys),7(security),8(cron),10(audit),11(lp)
# mkdom applvDom
# lsdom applvDom
applvDom id=1
# mkdom privlvDom
# lsdom privlvDom
privlvDom id=2
#
```

The next step is to define the file systems as domain RBAC objects.

The **setsecattr** command is used to define domain RBAC *objects*. In this scenario the administrator wishes to grant privileged access to four file systems and restrict privileged access to the remaining file systems. To do this the administrator needs to define each file system as a domain RBAC object.

The administrator ensures that all file systems on the server are mounted, then uses the **df** command to check the logical volume and file system names.

Example 8-31   The df -kP output - file systems on the AIX V7.1 LPAR

```
# df -kP
Filesystem     1024-blocks      Used Available Capacity Mounted on
/dev/hd4            196608    186300     10308     95% /
/dev/hd2           2031616   1806452    225164     89% /usr
/dev/hd9var         393216    335268     57948     86% /var
/dev/hd3            131072      2184    128888      2% /tmp
/dev/hd1             65536       428     65108      1% /home
/dev/hd11admin      131072       380    130692      1% /admin
/proc                    -         -         -      - /proc
/dev/hd10opt        393216    179492    213724     46% /opt
/dev/livedump       262144       368    261776      1% /var/adm/ras/livedump
/dev/backup_lv     2359296    102272   2257024      5% /backup
/dev/appslv_01     1310720   1117912    192808     86% /apps01
/dev/appslv_02     1310720   1117912    192808     86% /apps02
/dev/appslv_03     1310720   1117912    192808     86% /apps03
/dev/appslv_04     2359296   1118072   1241224     48% /apps04
#
```

The administrator now uses the **setsecattr** command to define each of the four application file systems as domain RBAC objects.

Example 8-32 shows the **setsecattr** command being used by the root user to define the domain RBAC objects for the four appsvg file systems.

> **Note:** When defining a file system object in domain RBAC, the logical volume device name will be used for the domain *object*.

Example 8-32   Using the setsecattr command to define the four application file systems as domain RBAC objects

```
# id
uid=0(root) gid=0(system) groups=2(bin),3(sys),7(security),8(cron),10(audit),11(lp)
# setsecattr -o domains=applvDom objtype=device secflags=FSF_DOM_ANY /dev/appslv_01
# setsecattr -o domains=applvDom objtype=device secflags=FSF_DOM_ANY /dev/appslv_02
# setsecattr -o domains=applvDom objtype=device secflags=FSF_DOM_ANY /dev/appslv_03
# setsecattr -o domains=applvDom objtype=device secflags=FSF_DOM_ANY /dev/appslv_04
# lssecattr -o /dev/appslv_01
/dev/appslv_01 domains=applvDom objtype=device secflags=FSF_DOM_ANY
# lssecattr -o /dev/appslv_02
/dev/appslv_02 domains=applvDom objtype=device secflags=FSF_DOM_ANY
# lssecattr -o /dev/appslv_03
/dev/appslv_03 domains=applvDom objtype=device secflags=FSF_DOM_ANY
# lssecattr -o /dev/appslv_04
/dev/appslv_04 domains=applvDom objtype=device secflags=FSF_DOM_ANY
#
```

In Example 8-32 the following attributes were defined

Domain   The domains attribute is the domain to which the domain
                 RBAC *object* will be associated.

Object Type   This is the type of domain RBAC object. The objtype=device
                 is used for a logical volume.

Security Flags   When the secflags attribute is set to FSF_DOM_ANY a
                 subject may access the *object* when it contains any of the
                 domains specified in the *domains* attribute.

Device Name   This is the full path name to the logical volume
                 corresponding to the file system. As an example,
                 /dev/appslv_01 is the logical volume corresponding to the
                 /apps01 file system.

> **Note:** In domain RBAC, all *objects* with an `objtype=device` must specify the full path name to the device, starting with the */dev* name.
>
> As an example, the rootvg volume group device would be specified to domain RBAC as `objtype=/dev/rootvg`.

The administrator will now use the **setsecattr** command to define the remaining file systems as domain RBAC *objects*.

Example 8-33 shows the **setsecattr** command being used by the root user to define the domain RBAC *objects* for the remaining file systems.

Example 8-33   Using the setsecattr command to define the remaining file systems as domain RBAC objects

```
# id
uid=0(root) gid=0(system)
groups=2(bin),3(sys),7(security),8(cron),10(audit),11(lp)
# setsecattr -o domains=privlvDom conflictsets=applvDom \
objtype=device secflags=FSF_DOM_ANY /dev/hd4
# setsecattr -o domains=privlvDom conflictsets=applvDom \
objtype=device secflags=FSF_DOM_ANY /dev/hd2
# setsecattr -o domains=privlvDom conflictsets=applvDom \
objtype=device secflags=FSF_DOM_ANY /dev/hd9var
# setsecattr -o domains=privlvDom conflictsets=applvDom \
objtype=device secflags=FSF_DOM_ANY /dev/hd3
# setsecattr -o domains=privlvDom conflictsets=applvDom \
objtype=device secflags=FSF_DOM_ANY /dev/hd1
# setsecattr -o domains=privlvDom conflictsets=applvDom \
objtype=device secflags=FSF_DOM_ANY /dev/hd11admin
# setsecattr -o domains=privlvDom conflictsets=applvDom \
objtype=device secflags=FSF_DOM_ANY /dev/proc
# setsecattr -o domains=privlvDom conflictsets=applvDom \
objtype=device secflags=FSF_DOM_ANY /dev/hd10opt
# setsecattr -o domains=privlvDom conflictsets=applvDom \
objtype=device secflags=FSF_DOM_ANY /dev/livedump
# setsecattr -o domains=privlvDom conflictsets=applvDom \
objtype=device secflags=FSF_DOM_ANY /dev/backup_lv
# lssecattr -o /dev/hd4
/dev/hd4 domains=privlvDom conflictsets=applvDom objtype=device \
secflags=FSF_DOM_ANY
#
```

In Example 8-33 on page 320 the following attributes were defined:

Domain   The `domains` attribute is the domain to which the domain
                        RBAC *object* will be associated

Conflict Set   This is an optional attribute. By defining the
                        `conflictsets=applvDom`, this *object* will not be accessible
                        if the entity has an existing association to the `applvDom`
                        domain.

Object Type   This is the type of domain RBAC *object*. The
                        `objtype=device` is used for a logical volume

Security Flags   When the `secflags` attribute is set to FSF_DOM_ANY a
                        subject may access the *object* when it contains any of the
                        domains specified in the domains attribute

Device Name   This is the full path name to the logical volume
                        corresponding to the file system. As an example,
                        `/dev/hd2` is the logical volume corresponding to the `/usr`
                        file system

The administrator will now use the **setkst** command to update the KST with the
changes made with the **setsecattr** and **mkdom** commands.

Example 8-34 shows the **setkst** command being executed from the root user.

Example 8-34   Using the setkst command to update the KST

```
# id
uid=0(root) gid=0(system)
groups=2(bin),3(sys),7(security),8(cron),10(audit),11(lp)
# setkst
Successfully updated the Kernel Authorization Table.
Successfully updated the Kernel Role Table.
Successfully updated the Kernel Command Table.
Successfully updated the Kernel Device Table.
Successfully updated the Kernel Object Domain Table.
Successfully updated the Kernel Domains Table.
#
```

The administrator will now use the **chuser** command to associate the `appuser`
account with the `applvDom` domain.

Example 8-35 shows the **chuser** command being executed by the root user.

Example 8-35   Using the chuser command to associate the appuser account with the applvDom domain

```
# lsuser -a domains appuser
appuser
# chuser domains=applvDom appuser
# lsuser -a domains appuser
appuser domains=applvDom
#
```

The administrator has now completed the domain RBAC configuration. The four application file systems have been defined as domain RBAC *objects* and the appuser has been associated with the applvDom domain.

The administrator has also defined the remaining file systems as domain RBAC *objects*. This restricts privileged access to users only associated with the privlvDom domain, and adds a conflict set to the applvDom domain.

The conflict set ensures that if the appuser account were to be granted an association to the privlvDom domain, the file system objects could not be modified with the privileged commands, because the privlvDom and applvDom domains are in conflict.

In Example 8-36 the appuser account uses the **swrole** command to switch to the apps_fs_manage role.

Example 8-36   The appuser account uses the swrole command to switch to the apps_fs_manage role

```
$ id
uid=301(appuser) gid=202(appgroup) groups=1(staff)
$ rolelist -a
apps_fs_manage  aix.fs.manage.change
                aix.fs.manage.create
                aix.fs.manage.mount
                aix.fs.manage.remove
                aix.fs.manage.unmount
$ swrole apps_fs_manage
appuser's Password:
$
```

The appuser account may now use the privileged commands in the apps_fs_manage role.

In Example 8-37 the appuser uses the **chfs** command to increase the size of the /apps01 file system by 1 GB. This command will successfully complete because the /dev/appslv_01 device was defined as a domain RBAC *object* to which the appuser has been granted an association through the applvDom domain.

Example 8-37 shows the appuser account using the **chfs** command to add 1 GB to the /apps01 file system.

Example 8-37   The appuser account using the chfs command to add 1 GB to the /apps01 file system

```
$ df -g /apps01
Filesystem    GB blocks      Free %Used     Iused %Iused Mounted on
/dev/appslv_01     1.25      0.18   86%        15    1% /apps01
$ chfs -a size=+1G /apps01
Filesystem size changed to 4718592
$ df -g /apps01
Filesystem    GB blocks      Free %Used     Iused %Iused Mounted on
/dev/appslv_01     2.25      1.18   48%        15    1% /apps01
$
```

In Example 8-37 we see that the **chfs** command has been successful.

Next, the appuser uses the **chfs** command to increase the size of the /backup file system by 1 GB.

Example 8-38 shows the appuser account attempting to use the **chfs** command to add 1 GB to the /backup file system.

Example 8-38   The appuser account attempting to use the chfs command to add 1 GB to the /backup file system

```
$ df -g /backup
Filesystem    GB blocks      Free %Used     Iused %Iused Mounted on
/dev/backup_lv     2.25      2.15    5%         5    1% /backup
$ chfs -a size=+1G /backup
/dev/backup_lv: Operation not permitted.
$ df -g /backup
Filesystem    GB blocks      Free %Used     Iused %Iused Mounted on
/dev/backup_lv     2.25      2.15    5%         5    1% /backup
$
```

In Example 8-38, the **chfs** command was not successful.

The **chfs** command was not sucsessful because the /dev/backup_lv device was defined as a domain RBAC object but the appuser account has not been granted association to the privlvDom domain.

Domain RBAC has restricted the appuser account using the **chfs** command to change the /backup file system because the appuser account has no association with the privlvDom domain.

Even though the appuser account has used the **swrole** command to switch to the apps_fs_manage role, the privileged **chfs** command is unsuccessful because domain RBAC has denied the appuser account access based on the domain object attributes of the /backup_lv *object* and the domain association of the appuser account.

By using this methodology, domain RBAC has restricted the appuser to managing only the file systems for which it has direct responsibility, and excluded privileged access to the remaining file systems on the LPAR.

In Example 8-39 the appuser account changes directory to the /tmp file system and uses the **touch appuser_tmp_file** command to show that the appuser account may still access the /tmp file system, but may not execute privileged commands, even though the apps_fs_manage role is effective.

In Example 8-39, the appuser account may also run the **whoami** command which is located in the /usr/bin directory in the /usr file system.

The /usr file system was also defined as a domain RBAC *object*, but is still accessible from the appuser and other user accounts, though the appuser account may not perform privileged operations on the /usr file system as shown when the appuser account attempts to execute the **chfs -a freeze=30 /usr** command.

Example 8-39   The appuser account using the touch and whoami commands

```
$ id
uid=301(appuser) gid=202(appgroup) groups=1(staff)
$ rolelist -e
apps_fs_manage  Manage apps filesystems
$ cd /tmp
$ touch appuser_tmp_file
$ ls -ltra appuser_tmp_file
-rw-r--r--    1 appuser  appgroup            0 Sep 13 19:44 appuser_tmp_file
$ whoami
appuser
$ chfs -a freeze=30 /usr
/dev/hd2: Operation not permitted.
```

The `appuser` and other user accounts may still access the domained file systems, such as the `/tmp` and `/usr` file systems as general users, but the privileged commands available to the `appuser` account in the `apps_fs_manage` role may not be used on file systems other than the `/apps01`, `/apps02`, `/apps03` and `/apps04` file systems.

File scenario - Restrict access

In a default installation of AIX, some files may be installed with DAC permissions that allow the files to be read by non-privileged users. Though the files may only be modified by the root user, these files may contain information that the administrator may not wish to be readable by all users.

By using domain RBAC, the administrator can restrict file access to only those user accounts that are deemed to require access.

In this scenario the administrator has been requested to limit read access of the `/etc/hosts` file to only the `netuser` user account. This can be accomplished by using domain RBAC.

In this scenario we have:

► An AIX V7.1 partition with enhanced RBAC enabled

► A non-privileged user named `netuser`

► A non-privileged user named `appuser`

In Example 8-40, the user `netuser` account uses the **head -15** command to view the first 15 lines of the `/etc/hosts` file.

The **ls -ltra** command output shows that the DAC permissions allow any user account to view the `/etc/hosts` file.

Example 8-40   The netuser account - using the head -15 command to view the first 15 lines of the /etc/hosts file

```
$ id
uid=302(netuser) gid=204(netgroup) groups=1(staff)
$ ls -ltra /etc/hosts
-rw-rw-r--    1 root     system          2052 Aug 22 20:35 /etc/hosts
$ head -15 /etc/hosts
# IBM_PROLOG_BEGIN_TAG
# This is an automatically generated prolog.
#
# bos61D src/bos/usr/sbin/netstart/hosts 1.2
```

```
#
# Licensed Materials - Property of IBM
#
# COPYRIGHT International Business Machines Corp. 1985,1989
# All Rights Reserved
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
# @(#)47        1.2  src/bos/usr/sbin/netstart/hosts, cmdnet, bos61D, d2007_49A2
10/1/07 13:57:52
# IBM_PROLOG_END_TAG
$
```

In Example 8-41, the user `appuser` uses the **head-15** command to view the first
15 lines of the **/etc/hosts file.** Again, the **ls-ltra** command output shows that
the DAC permissions allow any user account to view the /etc/hosts file.

*Example 8-41   The appuser account - using the head -15 command to view the first 15 lines of the
/etc/hosts file*

```
$ id
uid=301(appuser) gid=202(appgroup) groups=1(staff)
$ ls -ltra /etc/hosts
-rw-rw-r--    1 root     system        2052 Aug 22 20:35 /etc/hosts
$ head -15 /etc/hosts
# IBM_PROLOG_BEGIN_TAG
# This is an automatically generated prolog.
#
# bos61D src/bos/usr/sbin/netstart/hosts 1.2
#
# Licensed Materials - Property of IBM
#
# COPYRIGHT International Business Machines Corp. 1985,1989
# All Rights Reserved
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
# @(#)47        1.2  src/bos/usr/sbin/netstart/hosts, cmdnet, bos61D, d2007_49A2
10/1/07 13:57:52
# IBM_PROLOG_END_TAG
$
```

Both the `netuser` and `appuser` accounts are able to view the `/etc/hosts` file, due to the DAC of the `/etc/hosts file`.

By creating an RBAC domain and defining the `/etc/hosts` file as a domain RBAC *object*, access to the `/etc/hosts` file may be restricted, based upon the user account's association with the RBAC domain.

In Example 8-42, the root user logs in and uses the **mkdom** command to create an RBAC domain named `privDom`. The `privDom` domain has a domain ID of 3, which has been automatically system generated because the administrator did not include a domain ID in the the **mkdom** command.

Example 8-42   Using the mkdom command to create the privDom domain

```
# id
uid=0(root) gid=0(system)
groups=2(bin),3(sys),7(security),8(cron),10(audit),11(lp)
# mkdom privDom
# lsdom privDom
privDom id=3
#
```

From the root user, the administrator next defines the `/etc/hosts` file as a domain RBAC *object*.

In Example 8-43, the administrator uses the **setsecattr** command to define the `/etc/hosts` file as a domain RBAC object and assign the RBAC domain as `privDom`. The `objtype` attribute is set as the type `file`.

Example 8-43   Using the setsecattr command to define the /etc/hosts file as a domain RBAC object

```
# setsecattr -o domains=privDom objtype=file secflags=FSF_DOM_ANY /etc/hosts
# lssecattr -o /etc/hosts
/etc/hosts domains=privDom objtype=file secflags=FSF_DOM_ANY
#
```

For these changes to be available for use, the root user must update the KST with the **setkst** command.

Example 8-44 on page 328 shows the **lskst -t** command being used to list the KST prior to the **setkst** command being run.

Once the **setkst** command is run, the `privDom` domain and `/etc/hosts` file are both updated into the KST and are available for use.

Example 8-44   Updating the KST with the setkst command

```
# lskst -t dom privDom
Domain  "privDom" does not exist.
# lskst -t domobj /etc/hosts
Domain object "/etc/hosts" does not exist.
# setkst
Successfully updated the Kernel Authorization Table.
Successfully updated the Kernel Role Table.
Successfully updated the Kernel Command Table.
Successfully updated the Kernel Device Table.
Successfully updated the Kernel Object Domain Table.
Successfully updated the Kernel Domains Table.
# lskst -t dom privDom
privDom id=4
# lskst -t domobj /etc/hosts
/etc/hosts objtype=FILE domains=privDom \
conflictsets= secflags=FSF_DOM_ANY
#
```

At this stage, the `/etc/hosts` file has been defined as domain RBAC *object* and the KST updated.

The `/etc/hosts` file will now operate as a domain RBAC *object* and restrict access to any user accounts that have not been associated with the `privDom` domain.

This can be tested by attempting to access the `/etc/hosts` file from the `netuser` and `appuser` accounts.

> **Note:** The root user is automatically a member of all RBAC domains so does not require any special access to the `privDom` domain.

Example 8-45 and Example 8-46 on page 329 show the `netuser` account using the **head -15** command to read the `/etc/hosts file`.

Example 8-45   The netuser account using the head -15 command to access the /etc/hosts file

```
$ id
uid=302(netuser) gid=204(netgroup) groups=1(staff)
$ ls -ltra /etc/hosts
-rw-rw-r--   1 root     system        2052 Aug 22 20:35 /etc/hosts
$ head -15 /etc/hosts
/etc/hosts: Operation not permitted.
```

```
$
```

Example 8-46   The appuser account using the head -15 command to access the /etc/hosts file

```
$ id
uid=301(appuser) gid=202(appgroup) groups=1(staff)
$ ls -ltra /etc/hosts
-rw-rw-r--    1 root     system          2052 Aug 22 20:35 /etc/hosts
$ head -15 /etc/hosts
/etc/hosts: Operation not permitted.
$
```

The `netuser` and `appuser` accounts are no longer able to access the /etc/hosts file, even though the /etc/hosts file DAC allows for read access by any user. This is because the /etc/hosts file is now a domain RBAC object and access is dependant on the `privDom` domain association.

In Example 8-47, the administrator associates the `netuser` account with the `privDom` domain by using the **chuser** command from the root user.

Example 8-47   Using the chuser command to grant the netuser account association to the privDom domain

```
# lsuser -a domains netuser
netuser
# chuser domains=privDom netuser
# lsuser -a domains netuser
netuser domains=privDom
#
```

Now that the `netuser` account has been associated with the `privDom` domain, the `netuser` account may again access the /etc/hosts file.

> **Note:** Due to the **chuser** attribute change, the `netuser` account must log out and login for the domain=privDom association to take effect.

In Example 8-48 we see the `netuser` account using the **head -15** command to access the /etc/hosts file.

Example 8-48   The netuser account using the head -15 command to access the /etc/hosts file

```
$ id
uid=302(netuser) gid=204(netgroup) groups=1(staff)
$ ls -ltra /etc/hosts
```

```
-rw-rw-r--    1 root     system         2052 Aug 22 20:35 /etc/hosts
$ head -15 /etc/hosts
# IBM_PROLOG_BEGIN_TAG
# This is an automatically generated prolog.
#
# bos61D src/bos/usr/sbin/netstart/hosts 1.2
#
# Licensed Materials - Property of IBM
#
# COPYRIGHT International Business Machines Corp. 1985,1989
# All Rights Reserved
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
# @(#)47        1.2  src/bos/usr/sbin/netstart/hosts, cmdnet, bos61D, d2007_49A2
10/1/07 13:57:52
# IBM_PROLOG_END_TAG
$
```

The `netuser` account is now able to access the `/etc/hosts` file.

Associating the `netuser` account with the `privDom` domain has allowed the `netuser` account to access the *object* and list the contents of the `/etc/hosts` file with the **head -15** command.

Domain RBAC will still honor the DAC for the file object, so the `netuser` account will have only read access to the `/etc/host` file. Domain RBAC does not automatically grant write access to the file, but does allow the administrator to restrict the access to the `/etc/hosts` file without having to change the DAC file permission bits.

The `appuser` account will remain unable to access the `/etc/hosts` file because it has not been associated with the `privDom` domain.

Example 8-49 shows the `appuser` account attempting to access the `/etc/hosts` file by using the **head -15** command.

Example 8-49   The appuser account using the head -15 command to access the /etc/hosts file

```
$ id
uid=301(appuser) gid=202(appgroup) groups=1(staff)
$ ls -ltra /etc/hosts
-rw-rw-r--    1 root     system         2052 Aug 22 20:35 /etc/hosts
$ head -15 /etc/hosts
```

```
/etc/hosts: Operation not permitted.
$
```

The `appuser` account is denied access to the `/etc/hosts` file because it does not have the association with the `privDom` domain.

The administrator has successfully completed the request because the `/etc/hosts` file is now restricted to access by only the `netuser` account.

More than one user can be associated with a domain, so were more users to require access to the `/etc/hosts` file, the administrator need only use the **chuser** command to grant those users association with the `privDom` domain.

The root user is automatically considered a member of all domains, so the root user remains able to access the `/etc/hosts` file.

> **Note:** When restricting access to files, consider the impact to existing AIX commands and functions.
>
> As an example, restricting access to the `/etc/passwd` file would result in non-privileged users being no longer able to successfully execute the **passwd** command to set their own passwords.

File scenario - Remove access

In this scenario we discuss how domain RBAC can be used to remove access to files or non-privileged users.

In a default installation of AIX, some files may be installed with DAC permissions that allow the files to be read by non-privileged users. Though the files may only be modified by the root user, these files may contain information that the administrator may not wish to be readable by all users.

By using domain RBAC, the administrator can remove file access to user accounts that are deemed to not require access to such files.

In this scenario the administrator has chosen to remove read access to the `/etc/ssh/sshd_config` file. This can be accomplished by using domain RBAC.

In this scenario we have:

▶ An AIX V7.1 partition with enhanced RBAC enabled

▶ A non-privileged user named `appuser`

In Example 8-50 on page 332 we see the user `appuser` using the **head -15** command to view the first 15 lines of the `/etc/ssh/sshd_config` file.

We can see from the `ls -ltra` command output that the DAC permissions allow
any user account to view the `/etc/ssh/sshd_config` file.

*Example 8-50   The appuser account - using the head -15 command to view the first 15 lines of the
/etc/ssh/sshd_config file*

```
$ id
uid=301(appuser) gid=202(appgroup) groups=1(staff)
$ ls -ltra /etc/ssh/sshd_config
-rw-r--r--    1 root     system          3173 Aug 19 23:29 /etc/ssh/sshd_config
$ head -15 /etc/ssh/sshd_config
#        $OpenBSD: sshd_config,v 1.81 2009/10/08 14:03:41 markus Exp $

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/bin:/bin:/usr/sbin:/sbin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options change a
# default value.

#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
$
```

As shown in Example 8-50, the `/etc/ssh/sshd_config` file has DAC permissions
that allow all users on the LPAR to read the file.

By creating an RBAC domain and defining the `/etc/ssh/sshd_config` file as a
domain RBAC *object*, the administrator may restrict access to the
`/etc/ssh/sshd_config` to only user accounts with membership to the RBAC
domain.

By not associating the RBAC domain to any user accounts, the RBAC object will
not be accessible to any user accounts other than the root user.

In Example 8-51, the administrator uses the root user to create an RBAC domain
named `lockDom`. The `lockDom` domain has a domain ID of 4, which has been
automatically system generated because no domain ID was specified with the
mkdom command.

Example 8-51   Using the mkdom command to create the lockDom domain

```
# id
uid=0(root) gid=0(system)
groups=2(bin),3(sys),7(security),8(cron),10(audit),11(lp)
# mkdom lockDom
# lsdom lockDom
lockDom id=4
#
```

The administrator next uses the setsecattr command to define the
/etc/ssh/sshd_config file as a domain RBAC object.

In Example 8-52, the root user executes the **setsecattr** command to define the
/etc/ssh/sshd_config file as a domain RBAC object and set the RBAC domain
as lockDom.

*Example 8-52   Using the setsecattr command to define the /etc/ssh/sshd_config file as a
domain RBAC object*

```
# id
uid=0(root) gid=0(system)
groups=2(bin),3(sys),7(security),8(cron),10(audit),11(lp)
# setsecattr -o domains=lockDom objtype=file \
secflags=FSF_DOM_ANY /etc/ssh/sshd_config
# lssecattr -o /etc/ssh/sshd_config
/etc/ssh/sshd_config domains=lockDom objtype=file secflags=FSF_DOM_ANY
#
```

The /etc/ssh/sshd_config file has now been defined as a domain RBAC *object*.

To update the RBAC database change into the KST, the administrator uses the
setkst command.

Example 8-53 shows the root user running the **lskst** command to list the
contents of the KST. The root user then updates the KST by running the **setkst**
command.

*Example 8-53   Using the setkst command to update the KST and the lskst command to
list the KST*

```
# lskst -t dom lockDom
Domain  "lockDom" does not exist.
# lskst -t domobj /etc/ssh/sshd_config
Domain object "/etc/ssh/sshd_config" does not exist.
# setkst
Successfully updated the Kernel Authorization Table.
```

```
Successfully updated the Kernel Role Table.
Successfully updated the Kernel Command Table.
Successfully updated the Kernel Device Table.
Successfully updated the Kernel Object Domain Table.
Successfully updated the Kernel Domains Table.
# lskst -t dom lockDom
lockDom id=4
# lskst -t domobj /etc/ssh/sshd_config
/etc/ssh/sshd_config objtype=FILE domains=lockDom conflictsets=
secflags=FSF_DOM_ANY
#
```

At this stage, the /etc/ssh/sshd_config file is now defined as a domain RBAC
object and the KST updated. Access to the /etc/ssh/sshd_config file is now
restricted to the root user and any user accounts that are associated with the
lockDom domain.

Because no user accounts have an association with the lockDom domain, the
/etc/ssh/sshd_config file is now only accessible by the root user.

Example 8-54 shows the appuser account attempting to access the
/etc/ssh/sshd_config file with the **head**, **more**, **cat**, **pg** and **vi** commands:

*Example 8-54   Using the head, more, cat, pg and vi commands to attempt access to the
/etc/ssh/sshd_config file*

```
$ id
uid=301(appuser) gid=202(appgroup) groups=1(staff)
$ head -15 /etc/ssh/sshd_config
/etc/ssh/sshd_config: Operation not permitted.
$ more /etc/ssh/sshd_config
/etc/ssh/sshd_config: Operation not permitted.
$ cat /etc/ssh/sshd_config
cat: 0652-050 Cannot open /etc/ssh/sshd_config.
$ pg /etc/ssh/sshd_config
/etc/ssh/sshd_config: Operation not permitted.
$ vi /etc/ssh/sshd_config
~

...
...
~

"/etc/ssh/sshd_config" Operation not permitted.
$
```

The appuser account is not able to access the /etc/ssh/sshd_config file.

The only user able to access the `/etc/ssh/sshd_config` file is the root user.

If the `appuser` account were to be associated with the `lockDom` domain, then the `appuser` account would again be able to access the `/etc/ssh/sshd_config` file, based on the file DAC permission.

The benefits of using domain RBAC to restrict file access include:

File modification
There is no requirement to modify the file DAC settings, including ownership and bit permissions.

Quick to reinstate
Reinstating the file access does not require the administrator to modify the file DAC. The administrator can generally reinstate the file access by removing the *object* from the domain RBAC and updating the KST.

Granular control
The administrator may still grant access to the file *object* by associating user accounts with the RBAC domain, if required for temporary or long term access.

> **Note:** When removing access to files consider the impact to existing AIX commands and functions.
>
> As an example, removing access to the `/etc/security/passwd` file would result in non-privileged users no longer being able to execute the **passwd** command to set their own passwords.

Network scenario

In this scenario, domain RBAC will be used to restrict privileged access to an Ethernet network interface.

In domain RBAC, network objects may be either of two object types:

netint
This object type is a network interface. As an example, the en0 Ethernet interface would be an object type of netint.

netport
This object type is a network port. As an example, the TCP port 22 would be an object type of netport.

By using domain RBAC, the administrator can restrict a subject from performing privileged commands upon a netint or netport object.

In this scenario, the AIX V7.1 LPAR has two Ethernet network interfaces configured.

The administrator will use domain RBAC to:

► Allow the netuser account to use the **ifconfig** command on the en2 Ethernet interface.

► Restrict the `appuser` account from using the **ifconfig** command on the `en0` Ethernet interface.

Unless noted otherwise, all commands in the scenario will be run as the root user.

The administrator first uses the **lssecattr** command to determine which access authorizations the **ifconfig** command requires.

Example 8-55 shows the root user using the **lssecattr** command to display the access authorizations required by the **ifconfig** command:

Example 8-55   Using the lssecatr command from the root user to list the access authorizations for the ifconfig command

```
# lssecattr -c -a accessauths /usr/sbin/ifconfig
/usr/sbin/ifconfig accessauths=aix.network.config.tcpip
#
```

The **ifconfig** command requires the `aix.network.config.tcpip` access authorization.

The administrator will now use the **authrpt** command to determine whether there is an existing role that contains the necessary access authorizations required for executing the **ifconfig** command. The **authrpt -r** command limits the output displayed to only the roles associated with an authorization.

Example 8-56 shows the **authrpt -r** command being used to report on the `aix.network.config.tcpip` authorization.

Example 8-56   Using the authrpt command from the root user to determine role association with the aix.network.config.tcpip authorization

```
# authrpt -r aix.network.config.tcpip
authorization:
aix.network.config.tcpip
roles:

#
```

The `roles:` field in Example 8-56 has no value returned, which shows that there is no existing role associated with the `aix.network.config.tcpip` authorization. The administrator must use the **mkrole** command to create a role and associate the `aix.network.config.tcpip` authorization to the role.

Example 8-57 on page 337 shows the administrator using the **mkrole** command to create the `netifconf` role and include the `aix.network.config.tcpip`

authorization as the `accessauths` attribute. The administrator then updates the KST with the **setkst** command.

Example 8-57   Using the mkrole command from the root user to create the netifconf role and associate with the aix.network.config.tcpip authorization

```
# mkrole authorizations=aix.network.config.tcpip \
dfltmsg="Manage net interface" netifconf
# lsrole netifconf
netifconf authorizations=aix.network.config.tcpip rolelist= \
groups= visibility=1 screens=* dfltmsg=Manage net interface \
msgcat= auth_mode=INVOKER id=19
# setkst
Successfully updated the Kernel Authorization Table.
Successfully updated the Kernel Role Table.
Successfully updated the Kernel Command Table.
Successfully updated the Kernel Device Table.
Successfully updated the Kernel Object Domain Table.
Successfully updated the Kernel Domains Table.
#
```

The administrator next uses the **lsuser** command to display the existing roles, if any, that the `netuser` command may have associated to it. The administrator then associates the `netuser` with the `netifconf` role, including any existing roles in the **chuser** command.

Example 8-58 shows the **chuser** command being used to associate the `netuser` account with the `netifconf` role. The **lsuser** command showed that the `netuser` did not have any existing roles.

Example 8-58   Using the chuser command from the root user to associate the netuser account with the netifconf role

```
# lsuser -a roles netuser
netuser roles=
# chuser roles=netifconf netuser
# lsuser -a roles netuser
netuser roles=netifconf
#
```

At this stage, the `netuser` account has been associated with the `netifconf` role and may execute the **ifconfig** privileged command.

The administrator may verify this by using the **authrpt** and **rolerpt** commands.

Example 8-59 shows the **authrpt** command being used to report the
`aix.network.config.tcpip` authorization association with the `netifconf` role.

Example 8-59 also shows the **rolerpt** command being used to report that the
`netifconf` role has an association with the `netuser` account.

Example 8-59   The root user using the authrpt and rolerpt commands

```
# authrpt -r aix.network.config.tcpip
authorization:
aix.network.config.tcpip
roles:
netifconf
# rolerpt -u netifconf
role:
netifconf
users:
netuser
#
```

The administrator now uses domain RBAC to restrict the authority of the `netuser`
account's usage of the **ifconfig** command so that the **ifconfig** command will
only execute successfully when used upon the `en2` Ethernet interface.

The administrator uses domain RBAC to:

1. Create two RBAC domains.
2. Create two domain RBAC objects.
3. Update the Kernel Security Tables (KST).
4. Associate the RBAC domain to the `netuser` account.
5. Attempt to use the **ifconfig** command to change the status of the `en0` and
   `en2` Ethernet interfaces.

In Example 8-60 the administrator uses the **ifconfig -a** command to display the
network interfaces. The `en0` and `en2` Ethernet interfaces are both active, shown
by the `UP` status.

Example 8-60   The ifconfig -a command to display the network inferface status

```
# ifconfig -a
en0:
flags=1e080863,480<UP,BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST,GR
OUPRT,64BIT,CHECKSUM_OFFLOAD(ACTIVE),CHAIN>
        inet 192.168.101.12 netmask 0xffffff00 broadcast
192.168.101.255
```

```
        tcp_sendspace 262144 tcp_recvspace 262144 rfc1323 1
en2:
flags=5e080867,c0<UP,BROADCAST,DEBUG,NOTRAILERS,RUNNING,SIMPLEX,MULTICA
ST,GROUPRT,64BIT,CHECKSUM_OFFLOAD(ACTIVE),PSEG,LARGESEND,CHAIN>
        inet 10.10.100.2 netmask 0xffffff00 broadcast 10.10.100.255
         tcp_sendspace 131072 tcp_recvspace 65536 rfc1323 0
lo0:
flags=e08084b,c0<UP,BROADCAST,LOOPBACK,RUNNING,SIMPLEX,MULTICAST,GROUPR
T,64BIT,LARGESEND,CHAIN>
        inet 127.0.0.1 netmask 0xff000000 broadcast 127.255.255.255
        inet6 ::1%1/0
         tcp_sendspace 131072 tcp_recvspace 131072 rfc1323 1
#
```

After verifying the names of the Ethernet network interfaces in Example 8-60, the administrator now begins the domain RBAC configuration.

in Example 8-61 the root user is used to create the netDom and privNetDom RBAC domains.

Example 8-61   The mkdom command to create the netDom and the privNetDom RBAC domains

```
# mkdom netDom
# lsdom netDom
netDom id=5
# mkdom privNetDom
# lsdom privNetDom
privNetDom id=6
#
```

Next, in Example 8-62 the administrator uses the **setsecattr** command to define the en2 and en0 Ethernet network interfaces as domain RBAC objects. The **setkst** command is then run to update the KST.

Example 8-62   The setsecattr command being used by the root user to define the en0 and en2 domain RBAC objects

```
# setsecattr -o domains=netDom objtype=netint secflags=FSF_DOM_ANY en2
# setsecattr -o domains=privNetDom conflictsets=netDom \
objtype=netint secflags=FSF_DOM_ANY en0
# lssecattr -o en2
en2 domains=netDom objtype=netint secflags=FSF_DOM_ANY
# lssecattr -o en0
en0 domains=privNetDom conflictsets=netDom objtype=netint
secflags=FSF_DOM_ANY
```

```
# setkst
Successfully updated the Kernel Authorization Table.
Successfully updated the Kernel Role Table.
Successfully updated the Kernel Command Table.
Successfully updated the Kernel Device Table.
Successfully updated the Kernel Object Domain Table.
Successfully updated the Kernel Domains Table.
#
```

In Example 8-62 the administrator has included the `conflictsets=netDom`
attribute when defining the `en0` object. This means that if an entity were granted
association with the `privNetDom` and the `netDom`, the entity would not be granted
authorization to perform actions on the `en0` object, because the `privNetDom` and
`netDom` domains are in conflict.

> **Note:** The root user has an automatic association to all domains and objects.
>
> The root user does not honor the `conflictsets` attribute because the root user
> must remain able to access all domain RBAC objects.

The `netuser` next has its domain association extended to include the `netDom`
domain. The `netuser` account is already associated with the `privDom` domain
from a previous scenario. The privDom domain association is included in the
`chuser` command, else access to the `privDom` domain would be removed.

Example 8-63 shows the **chuser** command being used to associate the `netuser`
account with the `netDom` domain.

> **Note:** The `privDom` domain will not be used in this scenario and should not be
> confused with the `privNetDom` domain, which is used in this scenario.

*Example 8-63   Using the chuser command to associate the netuser account with the
netDom domain*

```
# lsuser -a domains netuser
netuser domains=privDom
# chuser domains=privDom,netDom netuser
# lsuser -a domains netuser
netuser domains=privDom,netDom
#
```

The administrator has now completed the domain RBAC configuration tasks.

The netuser account is now used to test the use of the **ifconfig** command and the domain RBAC configuration.

In Example 8-64 the netuser logs into the AIX V7.1 LPAR and uses the **swrole** command to switch to the netifconf role. The **rolelist -e** command shows that the netifconf role becomes the active role.

Example 8-64   The netuser account uses the swrole command to switch to the netifconf role

```
$ id
uid=302(netuser) gid=204(netgroup) groups=1(staff)
$ rolelist -a
netifconf       aix.network.config.tcpip
$ swrole netifconf
netuser's Password:
$ rolelist -e
netifconf       Manage net interface
$
```

In Example 8-65 on page 341 the netuser account uses the **ifconfig** command to display the status of the en2 Ethernet interface, showing that the status is UP. The **ping** command is used to confirm the UP status and has 0% packet loss.

The netuser account then uses the **ifconfig en2 down** command to inactivate the en2 interface. The **ifconfig** command no longer displays the UP status and the **ping** command returns 100% packet loss.

The netuser account has successfully used the **ifconfig** command to deactivate the en2 Ethernet interface.

Example 8-65   The netuser account using the ifconfig command to deactivate the en2 Ethernet interface

```
$ ifconfig en2
en2:
flags=5e080867,c0<UP,BROADCAST,DEBUG,NOTRAILERS,RUNNING,SIMPLEX,MULTICA
ST,GROUPRT,64BIT,CHECKSUM_OFFLOAD(ACTIVE),PSEG,LARGESEND,CHAIN>
        inet 10.10.100.2 netmask 0xffffff00 broadcast 10.10.100.255
         tcp_sendspace 131072 tcp_recvspace 65536 rfc1323 0
$ ping -c2 -w 2 10.10.100.5
PING 10.10.100.5: (10.10.100.5): 56 data bytes
64 bytes from 10.10.100.5: icmp_seq=0 ttl=64 time=1 ms
64 bytes from 10.10.100.5: icmp_seq=1 ttl=64 time=0 ms

----10.10.100.5 PING Statistics----
2 packets transmitted, 2 packets received, 0% packet loss
```

```
round-trip min/avg/max = 0/0/1 ms
$ ifconfig en2 down
$ ifconfig en2
en2:
flags=5e080866,c0<BROADCAST,DEBUG,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST,
GROUPRT,64BIT,CHECKSUM_OFFLOAD(ACTIVE),PSEG,LARGESEND,CHAIN>
        inet 10.10.100.2 netmask 0xffffff00 broadcast 10.10.100.255
         tcp_sendspace 131072 tcp_recvspace 65536 rfc1323 0
$ ping -c2 -w 2 10.10.100.5
PING 10.10.100.5: (10.10.100.5): 56 data bytes
0821-069 ping: sendto: The network is not currently available.
ping: wrote 10.10.100.5 64 chars, ret=-1
0821-069 ping: sendto: The network is not currently available.
ping: wrote 10.10.100.5 64 chars, ret=-1

----10.10.100.5 PING Statistics----
2 packets transmitted, 0 packets received, 100% packet loss
$
```

In Example 8-66, the netuser account then uses the **ifconfig en2 up** command
to reactivate the en2 interface. The **ifconfig** command displays the UP status and
the **ping** command returns 0% packet loss.

The netuser account has successfully used the **ifconfig** command to activate
the en2 Ethernet interface.

*Example 8-66   The netuser account using the ifconfig command to activate the en2
Ethernet interface*

```
$ ifconfig en2 up
$ ifconfig en2
en2:
flags=5e080867,c0<UP,BROADCAST,DEBUG,NOTRAILERS,RUNNING,SIMPLEX,MULTICA
ST,GROUPRT,64BIT,CHECKSUM_OFFLOAD(ACTIVE),PSEG,LARGESEND,CHAIN>
        inet 10.10.100.2 netmask 0xffffff00 broadcast 10.10.100.255
         tcp_sendspace 131072 tcp_recvspace 65536 rfc1323 0
$ ping -c2 -w 2 10.10.100.5
PING 10.10.100.5: (10.10.100.5): 56 data bytes
64 bytes from 10.10.100.5: icmp_seq=0 ttl=64 time=0 ms
64 bytes from 10.10.100.5: icmp_seq=1 ttl=64 time=0 ms

----10.10.100.5 PING Statistics----
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0/0/0 ms
$
```

By using RBAC, the `netuser` account has been able to successfully use the **ifconfig** command to activate and deactivate the en2 Ethernet interface.

In Example 8-67 , domain RBAC is used to restrict the `netuser` account from using the **ifconfig** command to change the status en0 interface. When the `netuser` account uses the i**fconfig en0 down** command, the **ifconfig** command is not successful.

Example 8-67   The netuser account is unsuccessful in using the ifconfig command to inactivate the en0 Ethernet interface

```
$ id
uid=302(netuser) gid=204(netgroup) groups=1(staff)
$ rolelist -e
netifconf       Manage net interface
$ ifconfig en0
en0:
flags=1e080863,480<UP,BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST,GR
OUPRT,64BIT,CHECKSUM_OFFLOAD(ACTIVE),CHAIN>
        inet 192.168.101.12 netmask 0xffffff00 broadcast
192.168.101.255
         tcp_sendspace 262144 tcp_recvspace 262144 rfc1323 1
$ ping -c2 -w 2 192.168.101.11
PING 192.168.101.11: (192.168.101.11): 56 data bytes
64 bytes from 192.168.101.11: icmp_seq=0 ttl=255 time=0 ms
64 bytes from 192.168.101.11: icmp_seq=1 ttl=255 time=0 ms

----192.168.101.11 PING Statistics----
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0/0/0 ms
$ ifconfig en0 down
0821-555 ioctl (SIOCIFATTACH).: The file access permissions do not
allow the specified action.
$ ifconfig en0
en0:
flags=1e080863,480<UP,BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST,GR
OUPRT,64BIT,CHECKSUM_OFFLOAD(ACTIVE),CHAIN>
        inet 192.168.101.12 netmask 0xffffff00 broadcast
192.168.101.255
         tcp_sendspace 262144 tcp_recvspace 262144 rfc1323 1
$ ping -c2 -w 2 192.168.101.11
PING 192.168.101.11: (192.168.101.11): 56 data bytes
64 bytes from 192.168.101.11: icmp_seq=0 ttl=255 time=0 ms
64 bytes from 192.168.101.11: icmp_seq=1 ttl=255 time=0 ms

----192.168.101.11 PING Statistics----
```

```
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0/0/0 ms
$
```

Example 8-67 on page 343 shows the netuser account using the **ifconfig** command to display the status of the en0 Ethernet interface, showing that the status is UP. The **ping** command is used to confirm the UP status and has 0% packet loss.

The netuser account then uses the **ifconfig en0 down** command to inactivate the en0 interface.

Because the netuser account has no association with the privNetDom domain, the **ifconfig** command returns the message:

```
0821-555 ioctl (SIOCIFATTACH).: The file access permissions do not
allow the specified action.
```

The **ifconfig** command is not successful and the status of the en0 Ethernet interface remains UP.

By using this methodology, domain RBAC has restricted the netuser account to using the **ifconfig** command to manage only the en2 network interface, and excluded privileged access to the en0 network interface.

In Example 8-62 on page 339 the administrator chose to use the **setsecattr** command with the optional conflictsets=netDom attribute. The conflictsets=netDom attribute can be used to further increase the security layer within the domain RBAC security framework.

Because the en0 object defines the domain attribute as privNetDom and the conflict set attribute is defined as netDom, the en0 object association will not be granted to an entity if the entity has associations to both the privNetDom and netDom domains.

In Example 8-68, the **chuser** command is used to add the privNetDom association with the netuser account. The existing associations with the privDom and netDom domains are included in the **chuser** command.

Example 8-68   The chuser command used to add the privNetDom association to the netuser account

```
# chuser domains=privDom,netDom,privNetDom netuser
# lsuser -a roles netuser
netuser roles=netifconf
#
```

Because the **chuser** command was used to grant the `netuser` account an association with the `privDom,netDom` and `privNetDom` domains and the `en0` object includes the conflict set between the `privNetDom` and the `netDom` domain, the `netuser` account will not be granted access to the `en0` object.

Example 8-69 shows the `netuser` account attempting to use the **ifconfig** command to deactivate the `en2` and `en0` Ethernet interfaces.

As in Example 8-65 on page 341, the **ifconfig en2 down** command is successful, because the `netuser` account has the `netifconf` role active and the domain RBAC configuration has been configured to allow for the operation of the **ifconfig** command on the `en2` object.

In Example 8-69, the **ifconfig en0 dow**n command is not successful, because the `conflictsets=netDom` attribute does not allow the `netuser` account access to the `en0` device.

Example 8-69   The netuser account using the ifconfig command to deactivate the en0 interface - the conflict set does not allow access to the en0 domain RBAC object

```
$ id
uid=302(netuser) gid=204(netgroup) groups=1(staff)
$ rolelist -a
netifconf        aix.network.config.tcpip
$ swrole netifconf
netuser's Password:
$ ifconfig en2 down
$ ifconfig en0 down
0821-555 ioctl (SIOCIFATTACH).: The file access permissions do not
allow the specified action.
$
```

8.2 Auditing enhancements

The following sections discuss the enhancements for auditing.

8.2.1 Auditing with full pathnames

The AIX audit subsystem allows auditing of objects with full path names for certain events, such as FILE_Open, FILE_Read and FILE_Write. This helps to achieve security compliance and gives complete information about the file that is being audited.

An option is provided to the **audit** command to enable auditing with full pathnames.

```
audit { on [ panic | fullpath ] | off | query | start | shutdown }{-@ wparname ...}
```

Likewise, the **audit** subroutine can also be used to enable full path auditing.

Example 8-70 shows how to enable or disable auditing with full pathnames.

Example 8-70   Configuring auditing with full pathnames

```
# audit query
auditing off
bin processing off
audit events:
        none

audit objects:
        none

# audit start

# audit off
auditing disabled

# audit on fullpath
auditing enabled

# cat newfile1

# auditpr -v < /audit/trail |grep newfile1
        flags: 67109633 mode: 644 fd: 3 filename /tmp/newfile1
        flags: 67108864 mode: 0 fd: 3 filename /tmp/newfile1
        file descriptor = 3 filename = /tmp/newfile1

# audit query
auditing on[fullpath]
audit bin manager is process 7143522
audit events:
        general -
FS_Mkdir,FILE_Unlink,FILE_Rename,FS_Chdir,USER_SU,PASSWORD_Change,FILE_
Link,FS_Chroot,PORT_Locked,PORT_Change,FS_Rmdir
........
........
```

8.2.2 Auditing support for Trusted Execution

Trusted Execution (TE) offers functionalities that are used to verify the integrity of the system and implement advanced security policies, which together can be used to enhance the trust level of the complete system. The functionalities offered can be grouped into the following:

► Managing the Trusted Signature Database

► Auditing integrity of the Trusted Signature Database

► Configuring Security Policies

New auditing events have been added to record security relevant information that can be analyzed to detect potential and actual violations of the system security policy.

Table 8-2 lists the audit events which have been added to audit Trusted Execution events.

Table 8-2   Audit event list

Event	Description
TEAdd_Stnz	This event is logged whenever a new stanza is being added to the /etc/security/tsd/tsd.dat (tsd.dat) database.
TEDel_Stnz	This event is logged whenever a stanza is deleted from the tsd.dat database.
TESwitch_algo	This event is logged when a hashing algorithm is changed for a command present in the tsd.dat database.
TEQuery_Stnz	This event is logged when the tsd.dat database is queried.
TE_Policies	This event is logged when modifying TE policies using the **trustchk** command. The various TE policies are listed below together with the possible values they can take: ► TE ON/OFF ► CHKEXEC ON/OFF ► CHKSHLIB ON/OFF ► CHKSCRIPT ON/OFF ► CHKKERNEXT ON/OFF ► STOP_UNTRUSTD ON/OFF/TROJAN ► STOP_ON_CHKFAIL ON/OFF ► LOCK_KERN_POLICIES ON/OFF ► TSD_FILES_LOCK ON/OFF ► TEP ON/OFF ► TLP ON/OFF
TE_VerifyAttr	This event is logged when the user attribute verification fails.

Event	Description
TE_Untrusted	Reports non-trusted files when they are executed
TE_FileWrite	Reports files that get opened in write mode
TSDTPolicy_Fail	Reports setting/setting of the Trusted Execution policy
TE_PermsChk	Reports when Owner/Group/Mode checks fail in the kernel
TE_HashComp	Reports when crypto hash comparison fails in the kernel

Recycling Audit trail files

Audit-related parameters are configured in the `/etc/security/audit/config` file. When the size of files `/audit/bin1` or `/audit/bin2` reaches the binsize parameter (defined in the config file) it is written to the `/audit/trail` file. The size of the trail file is in turn limited by the size of the / file system. When the file system free space reaches the freespace (defined in the config file) value, it will start logging the error message in the syslog. However, in case there is no space in the / file system, auditing will stop without affecting the functionality of the running system and errors will be logged in syslog.

To overcome this difficulty, tunable parameters have been provided in the `/etc/security/audit/config` file:

backupsize   A backup of the trail file is taken when the size of the trail file reaches this value. The existing trail file will be truncated. Size should be specified in units of 512-byte blocks.

backuppath   A valid full directory path, where a backup of the trail file needs to be taken.

In the `/etc/security/audit/bincmds` file, the **auditcat** command will be invoked in the following ways:

```
auditcat -p -s $backupsize -d $backuppath -o $trail $bin
```

or

```
auditcat -p -s <size value> -d <path value> -o $trail $bin
```

In the first case, it will replace the value of $backupsize and $backuppath from values mentioned in the `/etc/security/audit/config` file. In the later case it will take the actual values as specified at the command line.

The backup trail file name will be in the following format:

trail.YYYYMMDDThhmmss.<random number>

Example 8-71 shows the configuration of recycling of audit trail files.

Example 8-71   Recycling of audit trail files

```
# grep bincmds /etc/security/audit/config
        cmds = /etc/security/audit/bincmds

# cat /etc/security/audit/bincmds
/usr/sbin/auditcat -p -s 16 -d /tmp/audit -o $trail $bin

# audit start

# pwd
/tmp/audit

# ls
trail.20100826T025603.73142
```

> **Note:** If a copy of the trail file to newpath fails due to lack of space or any other reason, it will take the backup of the trail file in the /audit file system (or in the current file system if it is different from /audit, defined in the config file). However, if /audit is full, then it will not take the backup of the trail file and the legacy behavior will prevail, that is, auditing will stop and errors will be logged to syslog.

The `auditmerge` command is used to merge binary audit trails. This is especially useful if there are audit trails from several systems that need to be combined. The `auditmerge` command takes the names of the trails on the command line and sends the merged binary trail to standard output. Example 8-72 shows use of `auditmerge` and `auditpr` commands to read the audit records from the trail files.

Example 8-72   Merging audit trail files

```
auditmerge trail.system1 trail.system2 | auditpr -v -hhelrRtpc
```

8.2.3 Role-based auditing

Auditing has been enhanced to audit events on per role basis. This capability will provide the administrator with more flexibility to monitor the system based on roles.

In role-based auditing, auditing events are assigned to roles that are in turn assigned to users. This can be considered equivalent to assigning the audit

events for all the users having those roles. Auditing events are triggered for all users who are having the role configured for auditing.

As an example, audit events EventA and EventB are assigned to role Role1. The users User1, User2 and User3 have been assigned the role Role1. When auditing is started, events EventA and EventB are audited for all three users: User1, User2 and User3. Figure 8-1 depicts role-based auditing.

Figure 8-1   Illustration of role-based auditing

Example 8-73 shows the usage of role-based auditing.

Example 8-73

```
# mkrole auditclasses=files roleA
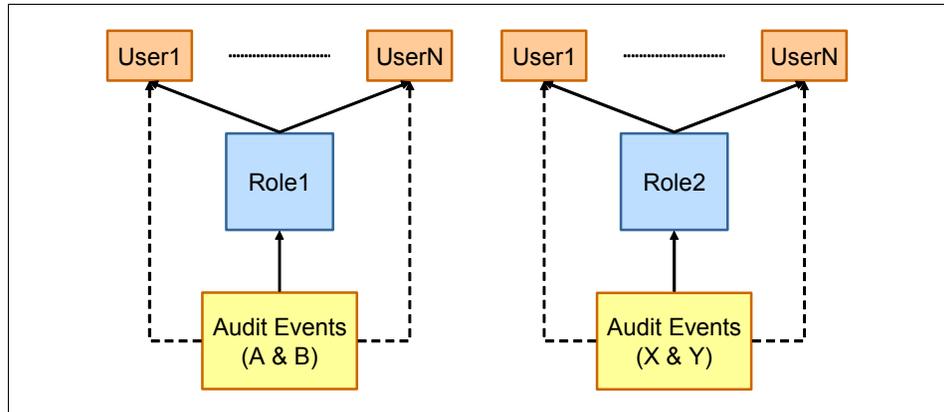
# setkst
Successfully updated the Kernel Authorization Table.
Successfully updated the Kernel Role Table.
Successfully updated the Kernel Command Table.
Successfully updated the Kernel Device Table.
Successfully updated the Kernel Object Domain Table.
Successfully updated the Kernel Domains Table.

# mkuser roles=roleA default_roles=roleA userA

# passwd userA
Changing password for "userA"
userA's New password:
Enter the new password again:

# audit start
```

```
# login userA
userA's Password:
[compat]: 3004-610 You are required to change your password.
        Please choose a new one.
userA's New password:
Enter the new password again:
********************************************************************************
*                                                                              *
*                                                                              *
*  Welcome to AIX Version 7.1!                                                 *
*                                                                              *
*                                                                              *
*  Please see the README file in /usr/lpp/bos for information pertinent to     *
*  this release of the AIX Operating System.                                   *
*                                                                              *
*                                                                              *
********************************************************************************

$ rolelist -e
roleA
$ exit

.....
.....
# id
uid=0(root) gid=0(system) groups=2(bin),3(sys),7(security),8(cron),10(audit),11(lp)

# auditpr -v </audit/trail |grep userA
        userA
FILE_Open      userA    OK           Thu Aug 26 02:11:02 2010 tsm
Global
FILE_Read      userA    OK           Thu Aug 26 02:11:02 2010 tsm
Global
FILE_Close     userA    OK           Thu Aug 26 02:11:02 2010 tsm
Global
....
....
```

8.2.4 Object auditing for NFS mounted files

All of the operations on the auditable objects residing on the NFS mounted file
systems, are logged on the client, provided that there are no operations on those

objects by the NFS server or by the other NFS clients, or fullpath auditing is enabled on the client. If fullpath auditing is not enabled and if the file is modified by the server or by other clients, the consecutive auditing might be undefined. This behavior is corrected by restarting audit on the client.

To illustrate, in the context of the Network File System (NFS), if an inode is reassigned to another file on the server side, the client will not be aware of it. Hence, it will keep track of the wrong files.

As a solution, if a file system is mounted on multiple clients, audit the operations on the server to get the exact log of the events or enable fullpath auditing on the client:

```
# audit on fullpath
```

By enabling fullpath auditing:

► If a file, say xyz, is deleted on the server and recreated with the same name (with the same or different inode), then the client will continue auditing it.

► If the file is deleted on the server and recreated with the same inode (but with a different name), then the client will not audit it.

8.3 Propolice or Stack Smashing Protection

Stack Smashing Protection is supported on AIX since AIX 6.1 TL4 and using XLC compiler Version 11. This feature can be used to minimize the risk of security vulnerabilities such as buffer overflows in AIX.

On AIX 7.1, most of the setuid programs are shipped with this feature enabled automatically and no explicit configuration is required.

For more information regarding the compiler option -qstackprotect, refer to the IBM XLC compiler version 11 documentation.

In Example 8-74, when the test program is compiled with the -qstackprotect option on the XLC v11 compiler and executed onthe AIX 6.1 TL6 or 7.1 system, buffer overflow will be detected, resulting in termination of the process.

Example 8-74   Propolice or Stack Smashing Protection

```
# cat test.c
char largebuffer[34];

main()
{
```

```
            char buffer[31];

            memcpy(buffer, largebuffer, 34);
}

# ./test
*** stack smashing detected ***: program terminated
IOT/Abort trap(coredump)
```

> **Note:** Propolice may not detect all buffer overruns. Its main goal is to prevent buffer overruns from overwriting the stack in a way that could lead to execution of malicious code. So as long as other local variables are overwritten, Propolice may not trigger.

8.4 Security enhancements

The following sections describe additional security enhancements.

8.4.1 ODM directory permissions

The Object Data Manager (ODM) is a data manager used for storing system configuration information. On AIX, the directories and files that make up the ODM are owned by root and are part of the system group. Both owner and group have write permissions. The group write permission opens a security hole by allowing any user in the system group the ability to create and modify files. This puts the system at risk from corruption and the potential to give unauthorized access to system users.

This security vulnerability is resolved by removing the group write permissions on these two directories:

/etc/objrepos

/etc/lib/objrepos

8.4.2 Configurable NGROUPS_MAX

The current hardcoded value for the maximum number of groups a user can be part of is 128. On AIX 7.1, this limit has been increased to 2048 (NGROUPS_MAX). The new kernel parameter ngroups_allowed is introduced,

which can be tuned in the range of 128 >= *ngroups_allowed* <=
NGROUPS_MAX.

The default is 128. This tunable allows administrators to configure the maximum
number of groups users can be members of. NGROUPS_MAX is the max value
that the tunable can be set to.

The **lsattr** command shows the current `ngroups_allowed` value. The **chdev**
command is used to modify the value. The **smitty chgsys** fastpath can also be
used to modify this parameter. Programmatically, the **sys_parm** subroutine with
the `SYSP_V_NGROUPS_ALLOWED` parameter can be used to retrieve the
`ngroups_allowed` value.

Example 8-75 shows configuring the `ngroups_allowed` parameter.

Example 8-75   Modifying ngroups_allowed

```
# lsattr -El sys0 |grep ngroups_allowed
ngroups_allowed 128                 Number of Groups Allowed
True

# chdev -l sys0 -a ngroups_allowed=2048
sys0 changed
```

Note: The system must be rebooted in order for the changes to take effect.

8.4.3 Kerberos client kadmind_timeout option

When using authentication other than the KRB5 load module, such as Single
Sign On (SSO), there can be long delays when the kadmind server is down. This
is because there are multiple kadmind connect calls for each Kerberos task,
which causes multiple tcp timeouts.

To solve this problem, a new option has been introduced in the
`/usr/lib/security/methods.cfg` for the KRB5 load module,
kadmind_timeout=<seconds>. The `kadmind_timeout` option specifies the amount
of time for the KRB5 load module to wait before attempting a kadmind connect
call after a previous timeout. If kadmind_timeout time has not elapsed since the
last timeout, then the KRB5 load module will not attempt to contact the down
server. Therefore, there will only be one timeout within the kadmind_timeout time
frame. The KADMIND_TIMEOUT_FILE will be used to notify all processes that
there was a previous timeout. Whenever a process successfully connects to the
kadmind server, the KADMIND_TIMEOUT_FILE is deleted.

Example 8-76 shows a sample configuration from the
/usr/lib/security/methods.cfg file.

Example 8-76   Kerberos client kadmind_timeout option

```
/usr/lib/security/methods.cfg:

KRB5:
            program = /usr/lib/security/KRB5
            program_64 = /usr/lib/security/KRB5_64
            options = kadmind_timeout=300

KRB5files
            options = db=BUILTIN,auth=KRB5
```

8.4.4 KRB5A load module removal

The KRB5 load module handles both KRB5 and KRB5A Kerberos environments.
Hence the KRB5A load module has been removed from AIX 7.1.

8.4.5 Chpasswd support for LDAP

The `chpasswd` command administers users' passwords. The root user can supply
or change users' passwords specified through standard input. The `chpasswd`
command has been enhanced to set Lightweight Directory Access Protocol
(LDAP) user passwords in an ldap_auth environment by specifying `-R LDAP` and
not specifying the `-e` flag for encrypted format. If you specify the `-e` option for the
encrypted format, the `chpasswd` command-crypted format and LDAP
server-crypted format must match.

8.4.6 AIX password policy enhancements

The following are the major password policy enhancements.

Restricting user name or regular expression in the password
The AIX password policy has been strengthened such that passwords are not
allowed to contain user names or regular expressions.

User name can be disallowed in the password by adding an entry with the key
word $USER in the dictionary files. This key word cannot be part of any word or
regular expression of the entries in dictionary files.

As an example, if root user has the entry $USER in the dictionary file, say `dicfile`, then the root cannot have the following passwords: root, root123, abcRoot, aRooTb, and so forth.

Example 8-77 shows how the password can be strengthened to *not to contain* any user names.

Example 8-77   Disallowing user names in passwords

```
# chsec -f /etc/security/user -s default -a dictionlist=/usr/share/dict/words
# tail /usr/share/dict/words
zoom
Zorn
Zoroaster
Zoroastrian
zounds
z's
zucchini
Zurich
zygote
$USER

$ id
uid=205(tester) gid=1(staff)
$ passwd
Changing password for "tester"
tester's Old password:
tester's New password: (the password entered is "tester")
3004-335 Passwords must not match words in the dictionary.
tester's New password:
Enter the new password again:
```

Passwords can be further strengthened by disallowing regular expressions. This is achieved by including the regular expression in the dictionary file. To differentiate between a word and a regular expression in the dictionary file, a regular expression will be indicated with '*' as first character.

For example, if administrator wishes to disallow any password beginning with "pas", then he can make the following entry in the dictionary file:

`*pas*`

The first * will be used to indicate a regular expression entry and the remaining part will be the regular expression, that is, `pas*`. Example 8-78 on page 357 shows the complete procedure.

Example 8-78   Disallowing regular expressions in passwords

```
# tail /usr/share/dict/words
Zorn
Zoroaster
Zoroastrian
zounds
z's
zucchini
Zurich
zygote
$USER
*pas*

$ id
uid=205(tester) gid=1(staff)
$ passwd
Changing password for "tester"
tester's Old password:
tester's New password: (the password entered is "passw0rd")
3004-335 Passwords must not match words in the dictionary.
tester's New password:
Enter the new password again:
```

Enforcing restrictions on the passwords

Passwords can be strengthened to force users to set passwords to contain the
following character elements:

▶ Uppercase letters: A, B, C ... Z

▶ Lowercase letters: a, b, c .. z

▶ Numbers: 0, 1, 2, ... 9

▶ Special characters: ~!@#$%^&*()-_=+[]{}|\;:'",.<>?/<space>

The following security attributes are used in this regard:

minloweralpha   Defines the minimum number of lower case alphabetic
                    characters that must be in a new password. The value is a
                    decimal integer string. The default is a value of 0, indicating
                    no minimum number. The allowed range is from 0 to
                    PW_PASSLEN.

minupperalpha   Defines the minimum number of upper case alphabetic
                    characters that must be in a new password. The value is a
                    decimal integer string. The default is a value of 0, indicating

no minimum number. The allowed range is from 0 to PW_PASSLEN.

mindigit   Defines the minimum number of digits that must be in a new password. The value is a decimal integer string. The default is a value of 0, indicating no minimum number. The allowed range is from 0 to PW_PASSLEN.

minspecialchar   Defines the minimum number of special characters that must be in a new password. The value is a decimal integer string. The default is a value of 0, indicating no minimum number. The allowed range is from 0 to PW_PASSLEN.

The following rules are applied to these attributes, while setting the password:

► Rule 1

- If minloweralpha > minalpha then minloweralpha=minalpha

- If minupperalpha > minalpha then minupperalpha=minalpha

- If minlowercase + minuppercase > minalpha then minuppercase=minalpha – minlowercase

Table 8-3 gives an example scenario for Rule 1.

Table 8-3   Example scenario for Rule 1

Value set for the attributes in the /etc/security/user file			Effective value while setting the password per Rule 1		
minupperalpha	minloweralpha	minalpha	minupperalpha	minloweralpha	minalpha
2	3	7	2	3	2
8	5	7	2	5	0
5	6	7	1	6	0

► Rule 2

- If mindigit > minother then mindigit=minother

- If minspecialchar > minother then minspecialchar=minother

- If minspecialchar + mindigit >minother then minspecialchar = minother – mindigit

Table 8-4 gives an example scenario for Rule 2.

Table 8-4   Example scenario for Rule 2

Value set for the attributes in the /etc/security/user file			Effective value while setting the password per Rule 2		
minspecialchar	mindigit	minother	minspecialchar	mindigit	minother
2	3	7	2	3	2

Value set for the attributes in the /etc/security/user file			Effective value while setting the password per Rule 2		
8	5	7	2	5	0
5	6	7	1	6	0

> **Note:** minother defines the minimum number of non-alphabetic characters in a password. The default is 0. The allowed range is from 0 to PW_PASSLEN.

Example 8-79 shows the usage of the `minloweralpha` security attribute.

Example 8-79   Usage of the minloweralpha security attribute

```
# chsec -f /etc/security/user -s default -a minloweralpha=5

# grep minloweralpha /etc/security/user
* minloweralpha Defines the minimum number of lower case alphabetic characters
*       Note: If the value of minloweralpha or minupperalpha attribute is
*             attribute. If 'minloweralpha + minupperalpha' is greater than
*             'minalpha - minloweralpha'.
        minloweralpha = 5
# chsec -f /etc/security/user -s default -a minalpha=8

# grep minalpha /etc/security/user
* minalpha      Defines the minimum number of alphabetic characters in a
*             greater than minalpha, then that attribute is reduce to minalpha
*             minalpha, then minupperalpha is reduce to
*             'minalpha - minloweralpha'.
*          'minalpha + minother', whichever is greater.  'minalpha + minother'
*          should never be greater than PW_PASSLEN.  If 'minalpha + minother'
*          'PW_PASSLEN - minalpha'.
        minalpha = 8
Changing password for "tester"
tester's Old password:
tester's New password: (the password entered is "comp")

3004-602 The required password characteristics are:
        a maximum of 8 repeated characters.
        a minimum of 8 alphabetic characters.
        a minimum of 5 lower case alphabetic characters.
        a minimum of 0 digits.


3004-603 Your password must have:
        a minimum of 8 alphabetic characters.
```

```
        a minimum of 5 lower case alphabetic characters.
tester's New password:
Enter the new password again:
$
```

8.5 Remote Statistic Interface (Rsi) client firewall support

In Rsi communication between xmservd/xmtopas and consumers, normally a random port was used by consumers. To force the consumers to open ports within the specified range, a new configuration line is introduced in AIX V7.1 and AIX 6.1 TL06. This new configuration enhancement is specified in the `Rsi.hosts` file. The Rsi agent first attempts to locate the `Rsi.hosts` file in the `$HOME` directory. If the file is not found, an attempt is made to locate the `Rsi.hosts` file in the `/etc/perf` directory, followed by a search in the `/usr/lpp/perfmgr` directory.

If an `Rsi.hosts` file is located, a specified range of ports is opened, including the starting and ending ports. If the `Rsi.hosts` file cannot be located in these directories or if the port range is specified incorrectly, the Rsi communication will make use of random ports.

You can specify the port range in the `Rsi.hosts` file as follows:

```
portrange <start_port> <end_port>
```

As an example:

```
    portrange 3001 3003
```

Once the Rsi agent is started, it makes use of the ports in the specified range. In the above example, the Rsi agent will use 3001 or 3002 or 3003. In this example, the Rsi agent can only listen on three ports (3001, 3002 and 3003). Subsequent Rsi communication will fail.

8.6 AIX LDAP authentication enhancements

AIX LDAP authentication has been enhanced with the following new features.

8.6.1 Case-sensitive LDAP user names

The LDAP uid and cn attributes are used to store user account name and group account name. Both uid and the cn attributes are defined as directory string and were case insensitive. Starting with AIX 6.1 TL06 and AIX 7.1, both uid and cn can be case sensitive by enabling the caseExactAccountName configuration parameter in the `/etc/security/ldap/ldap.cfg` file. Table 8-5 provides a list of the caseExactAccountName values.

Table 8-5   The caseExactAccountName values

Name	Value	Comments
caseExactAccountName	no (Default)	Case insensitive behavior
	yes	Exact case match

8.6.2 LDAP alias support

This feature allows AIX users to log in with an alias name defined in the LDAP directory entry, for example if an LDAP directory entry looks like the one shown in the following with an alias name `usr1`:

```
dn:uid=user1,ou=people,cn=aixdata
uid:user1
uid:usr1
objectclass:posixaccount
```

AIX LDAP authentication recognizes both uids `user1` and `usr1`. If a command **lsuser** is run for user name `user1` or `usr1` it displays the same information because they are aliases. Previously, LDAP authentication only recognized uid `user1`.

8.6.3 LDAP caching enhancement

The AIX LDAP secldapclntd client daemon caches user and group entries retrieved from the LDAP server. AIX 6.1 TL06 and AIX 7.1 offers the ability to control the caching mechanism through a new attribute called TO_BE_CACHED. This change translates into having an additional column in the existing mapping files located in the `/etc/security/ldap` directory. All attributes in the LDAP mapping files have a value of *yes* in the TO_BE_CACHED new field by default. Administrators can selectively set an attribute to *no* to disable the caching of that attribute.

Table 8-6 provides a list of TO_BE_CACHED attribute values.

Table 8-6   TO_BE_CACHED valid attribute values

Name	Value	Comments
TO_BE_CACHED	no	LDAP client sends query directly to the LDAP server.
	yes (Default)	LDAP client checks its cache before sending the query to the LDAP server.

8.6.4 Other LDAP enhancements

The following are additional LDAP enhancements:

► AIX LDAP supports Windows 2008 Active Directory (AD) and Active Directory application mode (ADAM).

► The `lsldap` command lists users, groups, NIS entities (hosts, networks, protocols, services, rpc, AND netgroup), automount maps, and RBAC entries (authorizations, roles, privileged commands, and devices). This command is extended to cover advance accounting.

► The AIX LDAP module is a full functional module covering both authentication and identification. It cannot be used as an authentication-only module as some customers have requested. This functionality is enhanced to have the same module support as a full functional module or an authentication-only module.

8.7 RealSecure Server Sensor

Multi-layered prevention technology in IBM RealSecure Server Sensor for AIX guards against threats from internal and external attacks.

Refer to the following website for further details about this product:

http://www.ibm.com/systems/power/software/aix/security/solutions/iss.html

9

Installation, backup, and recovery

The following AIX 7.1 topics are covered in this chapter:

9.1 AIX V7.1 minimum system requirements

This section discusses the minimum system requirements to install and run AIX V7.1.

9.1.1 Required hardware

Only 64-bit Common Hardware Reference Platform (CHRP) machines are supported with AIX V7.1. The following processors are supported:

► PowerPC® 970

► POWER4

► POWER5

► POWER6

► POWER7

To determine the processor type on an AIX system you can run the **prtconf** command, as shown in Example 9-1.

Example 9-1   Using prtconf to determine the processor type of a Power system

```
# prtconf | grep 'Processor Type'
Processor Type: PowerPC_POWER7
```

Note: The RS64, POWER3™, and 604 processors, 32-bit kernel, 32-bit kernel extensions and 32-bit device drivers are not supported.

Minimum firmware levels

Update your systems to the latest firmware level before migrating to AIX V7.1. Refer to the AIX V7.1 Release Notes for information relating to minimum system firmware levels required for AIX V7.1 at:

http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=/com.ibm.aix.ntl/releasenotes_kickoff.htm

For the latest Power system firmware updates, refer to the following website:

http://www14.software.ibm.com/webapp/set2/firmware/gjsn

Memory requirements

The minimum memory requirement for AIX V7.1 is 512 MB.

The current minimum memory requirements for AIX V7.1 vary based on the configuration of a system. It may be possible to configure a smaller amount of memory for a system with a very small number of devices or small maximum memory configuration.

The minimum memory requirement for AIX V7.1 may increase as the maximum memory configuration or the number of devices scales upward.

Paging space requirements

For all *new* and *complete overwrite* installations, AIX V7.1 creates a 512 MB paging space device named /dev/hd6.

Disk requirements

A minimum of 5 GB of physical disk space is required for a default installation of AIX V7.1. This includes all devices, the Graphics bundle, and the System Management Client bundle. Table 9-1 provides information relating to disk space usage with a default installation of AIX V7.1.

Table 9-1   Disk space requirements for AIX V7.1

Location	Allocated (Used)
/	196 MB (181 MB)
/usr	1936 MB (1751 MB)
/var	380 MB (264 MB)
/tmp	128 MB (2 MB)
/admin	128 MB (1 MB)
/opt	384 MB (176 MB)
/var/adm/ras/livedump	256 MB (1 MB)

> **Note:** If the `/tmp` file system has less than 64 MB, it is increased to 64 MB during a migration installation so that the AIX V7.1 boot image can be created successfully at the end of the migration.
>
> Starting with AIX V6.1 Technology Level 5, the boot logical volume is required to be 24 MB in size.
>
> The pre_migration script will check if the logical volume is the correct size. The script is located on your AIX V7.1 installation media or it can also be located in an AIX V7.1 NIM SPOT.
>
> If necessary, the boot logical volume, hd5, size will be increased. The logical partitions must be contiguous and within the first 4 GB of the disk. If the system does not have enough free space, a message is displayed stating there is insufficient space to extend the hd5 boot logical volume.

To install AIX V7.1, you must boot the system from the product media. The product media can be physical installation media such as DVD or it can be a NIM resource. For further information and instructions on installing AIX V7.1, refer to the *AIX Installation and Migration Guide*, SC23-6722, in the AIX Information Center at:

> http://publib.boulder.ibm.com/infocenter/aix/v7r1/topic/com.ibm.aix.
> install/doc/insgdrf/insgdrf_pdf.pdf

AIX edition selection

It is now possible to select the edition of the AIX operating system during the base operating system (BOS) installation.

AIX V7.1 is available in three different editions:

Express   This edition is the default selection. It is suitable for low-end Power systems for consolidating small workloads onto larger servers.

Standard   This edition is suitable for most workloads. It allows for vertical scalability up to 256 cores and 1024 threads.

Enterprise   This edition includes the same features as the Standard edition but with enhanced enterprise management capabilities. IBM Systems Directory Enterprise Edition and the Workload Partitions Manager™ for AIX are included. Systems Director Enterprise Edition also includes IBM Systems Director, Active Energy Manager, VMControl, IBM Tivoli® Monitoring and Tivoli Application Dependency Discovery Manager (TADDM).

Some of the differences between the AIX V7.1 editions are shown in Table 9-2.

Table 9-2   AIX edition and features

AIX V7.1 Feature	Express	Standard	Enterprise
Vertical Scalability	4 cores, 8 GB per core	256 cores, 1024 threads	256 cores, 1024 threads
Cluster Aware AIX	Only with PowerHA	Yes	Yes
AIX Profile Manager (requires IBM Systems Director)	Management target only	Yes	Yes
AIX 5.2 Versioned WPAR support (requires the AIX 5.2 WPAR for AIX 7 product)	Yes	Yes	Yes
Full exploitation of POWER7 features	Yes	Yes	Yes
Workload Partition support	Yes	Yes	Yes
WPAR Manager and Systems Director Enterprise Edition	No	No	Yes

As shown in Example 9-2, the administrator can change the AIX edition installed by selecting 5 `Select Edition` from the BOS installation menu.

Example 9-2   Selecting the AIX edition during a BOS installation

```
Installation and Settings

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  System Settings:
         Method of Installation.............New and Complete Overwrite
         Disk Where You Want to Install.....hdisk0

    2  Primary Language Environment Settings (AFTER Install):
         Cultural Convention................C (POSIX)
         Language...........................C (POSIX)
```

```
       Keyboard..........................C (POSIX)

    3  Security Model.......................Default
    4  More Options  (Software install options)
    5  Select Edition......................express
>>> 0  Install with the settings listed above.


                         +---------------------------------------------------
    88  Help ?           |  WARNING: Base Operating System Installation will
    99  Previous Menu    |       destroy or impair recovery of ALL data on the
                         |                destination disk hdisk0.
>>> Choice [0]:
```

Possible selections are `express`, `standard`, and `enterprise`. The default value is `express`. The edition value can also be set during non-prompted NIM installations by using the INSTALL_EDITION field in the control_flow stanza of the bosinst_data NIM resource. The AIX edition can be modified after BOS installation using the **chedition** command, as shown in Example 9-3.

Example 9-3   The chedition command flags and options

```
# chedition
Usage chedition: List current edition on the system
        chedition -l

Usage chedition: Change to express edition
        chedition -x [-d Device [-p]]

Usage chedition: Change to standard edition
        chedition -s [-d Device [-p]]

Usage chedition: Change to enterprise edition
        chedition -e [-d Device [-p]]
```

The edition selected defines the signature file that is copied to the `/usr/lpp/bos` directory. There are three signature files included in the bos.rte package. The files are located in `/usr/lpp/bos/editions`. These files are used by the IBM Tivoli License Manager (ITLM) to determine the current edition of an AIX system. When an edition is selected during installation (or modified post install), the corresponding signature file is copied to the `/usr/lpp/bos` directory.

For example, to change the edition from `express` to `enterprise` you would enter the command shown in Example 9-4 on page 369. You will notice that the corresponding signature file changes after the new selection.

Example 9-4   Modifying the AIX edition with the chedition command

```
# chedition -l
standard
# ls -ltr /usr/lpp/bos | grep AIX
-r--r--r--    1 root     system           50 May 25 15:25 AIXSTD0701.SYS2
# chedition -e
chedition: The edition of the system has been changed to enterprise.
# ls -ltr /usr/lpp/bos | grep AIX
-r--r--r--    1 root     system           50 May 25 15:25 AIXENT0701.SYS2
# chedition -l
enterprise
```

For further usage information relating to the `chedition` command, refer to the command reference section in the AIX Information Center at:

> http://publib.boulder.ibm.com/infocenter/aix/v7r1/topic/com.ibm.aix.
> cmds/doc/aixcmds1/chedition.htm

A SMIT interface to manage AIX editions is also available with the SMIT fastpath, `smit editions`.

For further information relating to managing AIX editions, refer to the AIX V7.1 Information Center at:

> http://publib.boulder.ibm.com/infocenter/aix/v7r1/topic/com.ibm.aix.
> install/doc/insgdrf/sw_aix_editions.htm

IBM Systems Director Command Agent

AIX V7.1 includes the IBM Systems Director Common Agent as part of the default install options. It is included in the System Management Client Software bundle.

When AIX is restarted, the Director agent and its prerequisite processes are automatically enabled and started. If these services are not required on a system, follow the instructions in the AIX V7.1 Release Notes to disable them.

Refer to the AIX V7.1 Release Notes in the AIX Information Center for additional information relating to minimum system requirements:

> http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=/c
> om.ibm.aix.ntl/releasenotes_kickoff.htm

9.2 Loopback device support in NIM

In addition to the Activation Engine, support for loopback devices will also be implemented in NIM. This support will allow a NIM administrator to use an ISO image, in place of the AIX installation media, as a source to create lpp_source and spot resources.

This functionality will rely on the underlying AIX loopback device feature introduced in AIX 6.1 via the `loopmount` command. Loopback device support was implemented in AIX 6.1, allowing system administrators to mount ISO images locally onto a system in order to read/write them.

This functionality limits the requirement of using the physical AIX installation media to create lpp_source and spot resources.

9.2.1 Support for loopback devices during the creation of lpp_source and spot resources

On the AIX Infocenter site at:

http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=/com.ibm.aix.kerneltechref/doc/ktechrf1/kgetsystemcfg.htm

it is specified that you can define an lpp_source in several ways. One is that an ISO image containing installation images can be used to create an lpp_source by specifying its absolute path name for the source attribute. For example:

```
nim -o define -t lpp_source -a server=master -a
location=/nim/lpp_source/lpp-71 -a source=/nim/dvd.71.v1.iso lpp-71
```

would define the lpp-71 lpp_source at /nim/lpp_source/lpp-71 on the master NIM server using the /nim/dvd.71.v1.iso ISO image.

If you wanted to define a spot labeled "spot-71" at /nim/spot/spot-71 on the master server using the /nim/dvd.71.v1.iso ISO image, then the following would be executed:

```
nim -o define -t spot -a server=master -a location=/nim/spot -a
source=/nim/dvd.71.v1.iso spot-71
```

9.2.2 Loopmount command

The `loopmount` command is the command used to associate an image file to a loopback device and optionally make an image file available as a file system via the loopback device.

It is described in the infocenter at:

http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=/c
om.ibm.aix.cmds/doc/aixcmds3/loopmount.htm

A loopback device is a device that can be used as a block device to access files.
It is described in the infocenter at:

http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=/c
om.ibm.aix.baseadmn/doc/baseadmndita/loopback_main.htm

The loopback file can contain an ISO image, a disk image, a file system, or a
logical volume image. For example, by attaching a CD-ROM ISO image to a
loopback device and mounting it, you can access the image the same way that
you can access the CD-ROM device.

Use the `loopmount` command to create a loopback device, to bind a specified file
to the loopback device, and to mount the loopback device. Use the `loopumount`
command to unmount a previously mounted image file on a loopback device, and
to remove the device. There is no limit on the number of loopback devices in AIX.
A loopback device is never created by default; you must explicitly create the
device. The block size of a loopback device is always 512 bytes.

The loopmount command restrictions

The following restrictions apply to a loopback device in AIX:

► The `varyonvg` command on a disk image is not supported.

► A CD ISO, DVD UDF+ISO, and other CD/DVD images are only supported in
read-only format.

► An image file can be associated with only one loopback device.

► Loopback devices are not supported in workload partitions.

Support of the loopmount command in NIM

In order to create an lpp_source or spot resource from an ISO image, NIM must
be able to mount ISO images using the `loopmount` executable.

NIM tries to mount the ISO image using:

/usr/sbin/loopmount -i image_pathname -m mount_point_pathname -o "-V
cdrfs -o ro

If the ISO image is already mounted, `loopmount` will return an error.

Since `umount` would unmount an ISO image, nothing has changed,

Add ISO image documentation to the Define a Resource smitty menu
(nim_mkres fastpath).

9.3 Bootlist command path enhancement

Configuration path commands such as `bootlist`, `lspath`, `chpath`, `rmpath`, and
`mkpath` have been enhanced with Multiple PATH I/O devices (MPIO) path
manipulation. It means that you can now include the pathid of a device.

9.3.1 Bootlist device pathid specification

The `bootlist` command includes the specification of the device pathid.

The AIX V7.1 man page for the `bootlist` command is shown in Example 9-5.

Example 9-5   Bootlist man page pathid concerns

```
Purpose
        Displays information about paths to a device that is capable of
        multiPath I/O.
Syntax
        bootlist [ { -m Mode } [ -r ] [ -o ] [ [ -i ] [ -V ] [ -F ]| [ [
-f File ] [ Device [ Attr=Value ... ] ... ] ] ] [ -v ]
Description
.......
When you specify a path ID, identify the path ID of the target disk by
using the pathid attribute. You can specify one or more path IDs with
the pathid attribute by entering a comma-separated list of the required
paths to be added to the boot list. When the bootlist command displays
information with the -o flag, the pathid attribute is included for each
disk that has an associated path ID.
Examples
11   To specify path ID 0 on disk hdisk0 for a normal boot operation,
type:
        bootlist -m normal hdisk0 pathid=0
12   To specify path ID 0 and path ID 2 on disk hdisk0 for a normal
boot operation, type one of the following commands:
        bootlist -m normal hdisk0 pathid=0,2
        bootlist -m normal hdisk0 pathid=0 hdisk0 pathid=2
```

> **Note:** Because the pathid argument can be repeated, both syntax pathid=0,2 and pathid=0 pathid=2 are equivalent.
>
> The order of the pathid arguments is how bootlist will process the paths. For example, pathid=2,0,1 will be different from patid=0,1,2.

The **bootlist** command display option specifies the pathid information; Example 9-6.

Example 9-6   bootlist -m normal -o command output

```
# bootlist -m normal -o
hdisk0 blv=hd5 pathid=0
```

9.3.2 Common new flag for pathid configuration commands

A new flag, **-i**, will print paths with the specified pathid specified as argument; Example 9-7.

Example 9-7   lspath, rmpath and mkpath command

```
lspath Command

Purpose
        Displays information about paths to an MultiPath I/O (MPIO)
capable device.
Syntax
        lspath [ -F Format | -t ] [ -H ] [ -l Name ] [ -p Parent ] [ -s
Status] [ -w Connection ] [ -i PathID ]
...
-i PathID
            Indicates the path ID associated with the path to be
displayed.
---------------------------------------------------------------------
rmpath Command

Purpose
        Removes from the system a path to an MPIO capable device.
Syntax
        rmpath [ -l Name ] [ -p Parent ] [ -w Connection ] [ -i PathID ]
...
-i PathID
            Indicates the path ID associated with the path to be
removed and is used to uniquely identify a path.
```

```
   -----------------------------------------------------------------
   mkpath Command

   Purpose
         Adds to the system another path to an MPIO capable device.
   Syntax
         mkpath [ -l Name ] [ -p Parent ] [ -w Connection ] [ -i PathID]
   ...
   -i PathID
               Indicates the path ID associated with the path to be added
   and is used to uniquely identify a path. This flag cannot be used with
   the -d flag.
```

> **Note:** The lspath command also gets a new flag, **-t**, which makes it possible
> to print information using the pathid field.
>
> **-t** displays the path ID in addition to the current default output. The **-t** flag
> cannot be used with the **-F** or the **-A** flags.
>
> ```
> # lspath -t
> Enabled hdisk0 vscsi0 0
> Enabled hdisk1 fscsi0 0
> Enabled hdisk2 fscsi0 0
> Enabled hdisk3 fscsi0 0
> Enabled hdisk4 fscsi0 0
> ```
>
> In case there is only one pathid, `lspath` and `lspath -i 0` get the same output.
>
> ```
> # lspath
> Enabled hdisk0 vscsi0
> Enabled hdisk1 fscsi0
> Enabled hdisk2 fscsi0
> Enabled hdisk3 fscsi0
> Enabled hdisk4 fscsi0
> ```

9.4 NIM thin server 2.0

With the AIX Network Installation Manager (NIM), you can manage the
installation of the Base Operating System (BOS) and any optional software on
one or more machines.

The NIM environment includes a server machine called master and clients that
receive resources from the server.

The Network Install component has provided several options for network security and firewall enhancements, but in AIX 6.1 it did not offer a method for encrypting or securing network data on resource servers in the NIM environment. In AIX 7.1 the NIM service handler (nimsh) provides NIM users with a client-configurable option for service authentication. Support of NFS V4 offers that capability.

NFS V4 support also permits support of the IPv6 network. The NIM server has been updated to support the IPv6 network.

An overview of the features and their implementation follows.

9.4.1 Functional enhancements

NFSv4 provides service authentication that provides information security in the following contexts:

► Identification - Creation and management of the identity of users, hosts, or services.
► Authentication - Validation of the identity of users, hosts or service.
► Authorization - Control of the information and data that a user or entity can access.

Some security attributes were then added to the NIM object database for the resource objects accessed through NFS V4.

You may specify the NFS export requirements for each NIM resource object when it is created or when changing options. The NFS protocol options available are summarized in the following table:

Table 9-3   NFS available options

option	values (default bolded)
version	**v3** or v4
security	**sys** or krb5

The Kerberos configuration specified with previous the krb5 flag must be created by you. Samples are available in `/usr/samples/nim/krb5`, and Kerberos credentials are viewable using query commands so clients can verify their credentials.

Note: In order to propagate the Kerberos configuration to NIM clients, the credentials must be valid for NFS access when strong security is enabled.

In the IPv6 network we can find two types of addresses:

► Link-local addresses prefixed by FE80::/16, which are used by hosts on the same physical network, that is, when there is only one hop of communication between nodes.

► Global addresses that uniquely identify a host on any network.

NIM supports installation of clients on IPv6 networks. Thin Server IPv6 network clients are also supported.

To support IPv6, NIM commands and SMIT menus have been preserved but new objects have been added; see Table 9-4.

Table 9-4   New or modified NIM objects

Object name	Meaning
ent6	Represents an Ethernet IPv6 network. IPv6 clients must be a member of this network.
if1 new semantic	The third field of if1 must contain the client's link-local address instead of the MAC address, such as If1="v6net myclient.company.com fe80:23d7::663:4"

Note: For IPv6 clients, BOOTP is not used but the boot image is downloaded directly through TFTP, which requires specification of a boot image file name. The convention being used is that the boot image file name is simply the hostname used by the client.

TFTP support is also available via new SMS menus for IPv6 added to the firmware. See an example in 9.4.5, "IPv6 boot firmware syntax" on page 378.

9.4.2 Considerations

Because the security options rely on exporting options for machine, network and group objects in the NIM environment, the mount options must be consistent across NFS client access:

► You cannot mix export options for an NFS mount specification.

► Only one single version support for a file system.

► You are limited to exporting NIM spot resources with an NFS security option of sys.

► You cannot define pseudo root mappings for NFS V4 exports. The NFS default of / will be used for accessing the NIM resources.

- The NFS options are only manageable from the NIM master. NIM clients can only do queries.
- The NFS attributes of the NFS protocols called nfs_vers and nfs_sec are what you get when mounting resources or restricting access.

> **Note:** The NFS server calls the `rpc.mountd` daemon to get the access rights of each client, so the daemon must be running on the server even if the server only exports file systems for NFS version 4 access.

- When master and client are on the same network, link-local addresses must be used.
- When master and client are on different networks, global addresses are used as normal.
- Gateway must *always* be link-local.
- NIM resources that are allocated to IPv6 clients must be exported using NFS4 with the option -a nfs_vers=4.
- Only AIX 6.1 TL1 and greater can be installed over IPv6.
- Only AIX 6.1 TL6 and greater thin servers can boot over IPv6.
- Only AIX 6.1 and greater can be installed at the same time as other IPv6 clients.
- Static IPv6 addresses are enforced so there is no DHCP support, no support for router discovery nor service discovery.

9.4.3 NIM commands option for NFS setting on NIM master

On the NIM master, if SMIT panels would drive you to specify the NFS options, the `nim` command is able to enable NFS client communication options:

- To enable the global use of NFS reserved ports type:

  `# nim -o change -a `**`nfs_reserved_port`**`=yes master`

- To disable global usage of NFS reserved ports type:

  `# nim -o change -a `**`nfs_reserved_port`**`=no master`

- To enable port checking on the NIM master NFS server type:

  `# nfso -o `**`portcheck`**`=1`

- To disable port checking on the NIM master NFS server.

  `# nfso -o portcheck=0`

9.4.4 Simple Kerberos server setting on NIM master NFS server

In order to use Kerberos security options for NFS you need to set a Kerberos server. A sample is provided in

```
/usr/samples/nim/krb5/config_rpcsec_server
```

To create a new system user-based on the principal name and password provided, just type:

```
/usr/samples/nim/krb5/config_rpcsec_server -p <password> -u <user
principal name>
```

If you want to delete the Kerberos V configuration information related to the Kerberos server and principals on the NIM master NFS server, just type the following command on the NIM master:

```
# /usr/sbin/unconfig.krb5
```

> **Note:** Because Kerberos is relying on time, a mechanism should be invoked to automatically synchronize time through the network. The NIM server must run the AIX timed daemon or an NTP daemon.

9.4.5 IPv6 boot firmware syntax

The **boot** command has changed to support IPv6 and the new format:

```
> boot
/lhea@23c00300/ethernet@23e00200:ipv6,ciaddr=FE80::214:5EFF:FE51:D5,
giaddr=FE80::20D:60FF:FE4D:C1CE,siaddr=FE80::214:5EFF:FE51:D51,
filename=mylparwar.domain.com
```

9.4.6 /etc/export file syntax

The syntax of a line in the /etc/exports file is:

```
directory  -option[,option]
```

directory is the full path name of the directory. Options can designate a simple flag such as ro or a list of host names. See the specific documentation of the /etc/exports file and the **exportfs** command for a complete list of options and their descriptions.

9.4.7 AIX problem determination tools

Numerous files and commands can be used to investigate problems.

syslogd NFS uses the syslog to write its error and debug information. Before carrying out any problem determination, the administrator should turn syslog logging on.

iptrace To examine network traffic, the developer should create an iptrace.

ipreport To decode an iptrace into a readable format, the developer should use ipreport and ensure that Kerberos packets are included in the log.

rpcinfo Used to check the status of remote procedural call servers.

fuser Used to determine mount problems. `fuser` lists the process numbers of local processes that use the local or remote files specified by the command's file parameter.

lsof Tool available at the following site for listing files opened by a process:

http://www.bullfreeware.com

nfs4cl Allows display of NFS v4 statistics. The command can also be used to modify current NFS v4 properties.

nfsstat Displays information about NFS and RPC calls.

errpt Can be used to determine why a daemon is not starting or core dumping during its execution.

9.5 Activation Engine for VDI customization

This feature first became available in AIX 6.1 TL 06. Documentation is available in the Information Center under the Activation Engine topic.

The main purpose of the Activation Engine (AE) is to provide a toolkit that allows one image of an AIX instance to be deployed onto many target machines, each with a different configuration.

The Activation Engine includes a script that runs at boot time and automatically configures the system with a set of defined system parameters. These parameters are specified by the system administrator in the virtual image template file on the optical media.

A generic system image, such as a Virtual Disk Image (VDI) or mksysb, can be used to boot multiple clients using different virtual image template files. Each of the target machines will then be deployed with a completely different configuration including network configuration, custom file systems, and user accounts.

9.5.1 Step-by-step usage

Activation Engine usage can be summarized in the following five steps:

1. Enable Activation Engine on the AIX system.
2. Capture a VDI using the current system as the source.
3. Create a virtual image template file for any systems you wish to deploy to.
4. Place virtual image templates on optical drives of the systems you are deploying to.
5. Boot the target systems using the VDI.

Enable Activation Engine on the AIX system

The Activation Engine needs to be enabled on the target system.

By running the `ae –o enable template_file` command we are telling AE to enable itself to run at the next boot-up through an inittab entry. It will execute the processing of the XML template called template_file.

> **Note:** We did not have to specify any scripts to run. The scripts are all defined and referenced in the XML template file itself.

The AIX Activation Engine is available in the bos.ae installp package. The contents of the package are listed below. It provides the `ae` command as well as some sample scripts.

Example 9-8   Content of the ae package

```
# lslpp -f bos.ae
  Fileset                File
  ----------------------------------------------------------------------
Path: /usr/lib/objrepos
  bos.ae 7.1.0.0         /usr/samples/ae/templates
                         /usr/samples/ae/scripts/ae_accounts
                         /opt/ibm/ae/dmtf_network_bringup
                         /opt/ibm/ae/ae
                         /usr/samples/ae
                         /opt/ibm
```

```
/opt/ibm/ae/ae_template.xsd
/usr/samples/ae/scripts
/usr/sbin/ae -> /opt/ibm/ae/ae
/usr/samples/ae/scripts/ae_filesystems
/opt/ibm/ae
/usr/samples/ae/templates/ae_template.xml
/usr/samples/ae/scripts/ae_network
/opt/ibm/ae/ae_template.dtd
```

The first step is to enable and configure AE on a target system. This is done by running the **ae -o enable** command as shown in Example 9-9, which creates an `aengine` entry in `/etc/inittab` that will be executed at boot time.

Example 9-9   .Enabling activation engine

```
# ae -o enable
Activation Engine was successfully enabled.
Using template 'ae_template.xml' from first available optical media.
# grep engine /etc/inittab
aengine:23456789:wait:/usr/sbin/ae -o run ae_template.xml
```

The argument `ae_template.xml` is the name of the XML template that will be read from the optical media at boot time. It is the default name. However, it can be specified as an argument to the **ae -o enable** command. See the command syntax in Example 9-10.

Example 9-10   The Activation Engine command syntax

```
# ae
USAGE: /usr/sbin/ae -o {enable | disable | status | check | run}

enable <template> - Enable the Activation Engine
disable - Disable the Activation Engine
status - Print current status of Activation Engine
check <template> - Validate a user created template against the
Activation Engine schema
run <template> - Execute the activation engine against a particular
template file
```

Capture a VDI using the current system as the source

The second step involves capturing an image of your current system. This is the image that you will use to deploy to other systems. The captured image must have the Activation Engine enabled so that AE can customize specific

parameters at boot time. This capture step is usually performed using VMControl, which is one of the main consumers of AE.

This step can also be done using the mksysb or NIM.

Note: Image creation must be performed after Activation Engine has been enabled.

Create a virtual image template

Since each deployed system gets configured with its own network address, custom users, and file system, you usually need to create separate template files for each system you plan to deploy to. These files must be stored in the root of the optical media, which must be mountable by the Activation Engine at boot time.

The configuration is organized using two types of files:

► The data contained in the XML template files.
► The scripts that perform actions using the data extracted from XML template files.

The template file example `/usr/samples/ae/templates/ae_template.xml` listed in Example 9-12 references the scripts associated with the network, user, and file systems sections as seen in the `grep` command output shown in Example 9-11.

Example 9-11   Grep of script in user created template file.

```
<!--<section name="network" script="ae_network">
<section name="accounts" script="ae_accounts">
<section name="filesystems" script="ae_filesystems">
```

These default scripts are available in `/usr/samples/ae/scripts`.

Example 9-12   Sample script /usr/samples/ae/templates/ae_template.xml

```
# cat /usr/samples/ae/templates/ae_template.xml
<?xml version="1.0" encoding="UTF-8"?>
<template name="Sample Activation Engine template">
        <settings>
                <!-- log directory is created automatically if it doesn't exist -->
                <logDirectory>/var/adm/ras/ae</logDirectory>
                <!-- / is assumed to be / dir of optical media -->
                <scriptsDirectory>/scripts</scriptsDirectory>
                <!-- Here we specify all user created templates that we want AE to
execute, in order. scripts are defined within -->
```

```
                    <extensions>

<!--<extendedTemplate>/user_template1.xml</extendedTemplate>-->
                    </extensions>
        </settings>
        <rules>
                <!-- the following section was commented to out prevent accidental
execution -->
                <!-- script paths are assumed to be relative to / directory of
optical media -->
                <!--<section name="network" script="ae_network">
                        <ruleSet>
                                <hostname>hostanme.domain</hostname>
                                <interface>en0</interface>
                                <address>XX.XX.XX.XX</address>
                                <mask>255.255.254.0</mask>
                                <gateway>XX.XX.XX.0</gateway>
                                <domain>hostname.domain</domain>
                                <nameserver>XX.XX.XX.XX</nameserver>
                                <start_daemons>yes</start_daemons>
                        </ruleSet>
                </section>
                <section name="accounts" script="ae_accounts">
                        <ruleSet>
                                <username>username</username>
                                <groups>sys</groups>
                                <admin>true</admin>
                                <home>/home/apuzic</home>
                        </ruleSet>
                </section>
                <section name="filesystems" script="ae_filesystems">
                        <ruleSet>
                                <mountpoint>/usr/testmount</mountpoint>
                                <type>jfs2</type>
                                <volume_group>rootvg</volume_group>
                                <size>16M</size>
                        </ruleSet>
                </section>-->
        </rules>
</template>
```

> **Note:** A template can reference as many scripts as it wants, as long as all those scripts are present on the optical media.

Creating AE scripts

Script creation must follow three distinct guidelines:

► The scripts must accept parameters defined in the <ruleSet> tags of the template file. (See Example 9-12 on page 382.)

► They must not pipe standard output or standard error to any external files because the Activation Engine pipes both of these to the specified log files. This makes debugging and status tracking easier.

► The script must return 0 after a successful execution. Any other return code is interpreted as a failure.

> **Note:** Each template can also link to other template files, which allows for further flexibility. For example, you can create one template to customize all network parameters on the system, another to create new file systems, and another to add new custom user accounts and groups. This allows for easier categorization of customized data. It also makes it easier to add new customized data to the image because you can create a new template and have one of the existing templates point to the newly created file.

Checking virtual image templates

Running `ae –o check template_name` against your own template checks your XML file against the schema and alerts you of any errors. It is a best practice that you do this before using your template files to make sure that you are not using the Activation Engine with an invalid template file in a production environment. A successful check is performed in Example 9-13.

Example 9-13   Successful Activation Engine template file structure check

```
# ae -o check ae_template.xml
Template 'ae_template.xml' is valid AE template
# cp /usr/samples/ae/scripts/* /
```

> **Note:** The `ae -o check` command only checks syntax of the XML file, not the content. It does not check the existence of the script files referenced in the XML file.

Place virtual image templates on the optical media

Once a valid XML template file and optional corresponding shell scripts have been created, burn the files to the optical media.

The template file has to be located in the root directory of the media in the optical device.

> **Note:** Activation Engine checks all bootable optical media for virtual image templates and uses the first one found. If you are booting a VDI on a system with two (or more) optical discs and all discs have virtual image templates, then AE will use the first template it finds on any of the mounted discs.

Boot the target systems using the VDI

Because the Activation Engine is executed at boot time through the inittab entry, the scripts will be executed and will only perform configurations limited to the boot phase. For example, you cannot expect to install new filesets using AE.

9.6 SUMA and Electronic Customer Care integration

In August 2004 AIX V5.3 introduced the Service Update Management Assistant (SUMA) tool, which allows system administrators to automate the download of maintenance updates such as Maintenance Levels (MLs), Technology Levels (TLs) and Service Packs (SPs). In the AIX V5.3 and AIX V6.1 releases SUMA uses the undocumented $fixget$ interface to initiate a standard multipart data HTTP POST transaction to the URL where the fix server's fixget script resides to retrieve AIX updates. The fix server's URL is configured through the FIXSERVER_URL parameter of the SUMA global configuration settings during the base configuration and can be viewed with the **suma -c** command. Example 9-14 shows the **suma -c** command output on an AIX V6.1 TL 6100-05 system after a SUMA base configuration has been performed.

Example 9-14   SUMA default base configuration on AIX V6.1

```
# suma -c
        FIXSERVER_PROTOCOL=http
        DOWNLOAD_PROTOCOL=ftp
        DL_TIMEOUT_SEC=180
        DL_RETRY=1
        MAX_CONCURRENT_DOWNLOADS=5
        HTTP_PROXY=
        HTTPS_PROXY=
        FTP_PROXY=
        SCREEN_VERBOSE=LVL_INFO
        NOTIFY_VERBOSE=LVL_INFO
        LOGFILE_VERBOSE=LVL_VERBOSE
        MAXLOGSIZE_MB=1
        REMOVE_CONFLICTING_UPDATES=yes
        REMOVE_DUP_BASE_LEVELS=yes
        REMOVE_SUPERSEDE=yes
```

```
TMPDIR=/var/suma/tmp
FIXSERVER_URL=www14.software.ibm.com/webapp/set2/fixget
```

A usage message for the fixget script is given at:

http://www14.software.ibm.com/webapp/set2/fixget

when entered in the address field of a web browser. Note that the fixget utility is not intended for direct customer use but is rather called internally by the SUMA tool.

Beginning with AIX V7.1, SUMA no longer uses fixget but instead utilizes the Electronic Customer Care (eCC) services to retrieve AIX updates.

IBM Electronic Customer Care services are strategically designed to offer a centralized access point to code updates for IBM systems. Independent of a given platform, similar terminology and application programming interfaces enable a standardized user interface with a consistent usage environment.

Currently eCC provides an update repository for instances such as Power Systems Firmware, Hardware Management Console (HMC), IBM BladeCenter, Linux, IBM i and now also for AIX 7. The eCC Common Client's Java API is used as a common interface by all supported platforms to download the updates. In AIX V7.1 the eCC Common Client functionality is available through the `bos.ecc_client.rte` fileset. The same fileset is also required to support the IBM Electronic Service Agent™ (ESA) and the Inventory Scout utility on AIX. This means that on AIX 7, SUMA, ESA, and the Inventory Scout are all consumers of the same eCC Common Client and share the eCC code, the libraries, and the connectivity settings. However, each of the named utilities will run individually in a separate Java Virtual Machine.

9.6.1 SUMA installation on AIX 7

As in previous AIX releases, the SUMA code is delivered in the `bos.suma` fileset. But on AIX 7 this fileset is not installed by default because it is no longer included in the `/usr/sys/inst.data/sys_bundles/BOS.autoi` file. In AIX 7 the `bos.suma` fileset is contained in the graphics software bundle (Graphics.bnd) and the system management software bundle (SystemMgmtClient.bnd). Both predefined system bundles are located in the `/usr/sys/inst.data/sys_bundles/` directory. The `bos.suma` fileset requires the installation of the `bos.ecc_client.rte` fileset, which in turn needs the support of Java 6 through the `Java6.sdk` fileset. Both SUMA and eCC rely on the support of the Perl programming language.

The `lslpp` command output in Example 9-15 shows the fileset dependencies of SUMA and eCC.

Example 9-15   The lslpp command output

```
75011p01:/> lslpp -p bos.suma bos.ecc_client.rte
  Fileset              Requisites
----------------------------------------------------------------------
Path: /usr/lib/objrepos
  bos.ecc_client.rte 7.1.0.0
                       *ifreq bos.rte 7.1.0.0
                       *prereq perl.rte 5.10.1.0
                       *prereq perl.libext 2.3.0.0
                       *prereq Java6.sdk 6.0.0.200
  bos.suma 7.1.0.0     *prereq bos.rte 7.1.0.0
                       *prereq bos.ecc_client.rte 7.1.0.0
                       *prereq perl.rte 5.8.2.0
                       *prereq perl.libext 2.1.0.0

Path: /etc/objrepos
  bos.ecc_client.rte 7.1.0.0
                       *ifreq bos.rte 7.1.0.0
                       *prereq perl.rte 5.10.1.0
                       *prereq perl.libext 2.3.0.0
                       *prereq Java6.sdk 6.0.0.200
  bos.suma 7.1.0.0     *prereq bos.rte 7.1.0.0
                       *prereq bos.ecc_client.rte 7.1.0.0
                       *prereq perl.rte 5.8.2.0
                       *prereq perl.libext 2.1.0.0
```

9.6.2 AIX 7 SUMA functional and configuration differences

The SUMA implementation in AIX V7.1 is governed by the following two guidelines:

▶ IBM AIX operating system release and service strategy

▶ Electronic Customer Care cross-platform service strategy for IBM Systems

The current AIX service strategy was introduced in 2007 and requires fixpacks such as Technology Levels (TL) or Service Packs (SP) to be downloaded in a single entity. The download of individual fixes or filesets is no longer supported. SUMA in AIX 7 adheres to this service strategy and supports the following request type (RqType) values for the `suma` command only:

ML   Request to download a specific maintenance or technology level.

TL	Request to download a specific technology level. The TL must be specified by the full name, for example 6100-03-00-0920 instead of 6100-03.
SP	Request to download a specific service pack. The SP must be specified by the full name, for example 6100-02-04-0920 instead of 6100-04-04.
PTF	Request to download a Program Temporary Fix (PTF). Only certain PTFs may be downloaded as an individual fileset. For example, PTFs containing bos.rte.install, bos.alt_disk_install.rte, or PTFs that come out in between service packs. Otherwise, the TL or SP must be downloaded.
Latest	Request to download the latest fixes. This RqType value returns the latest service pack of the TL specified in the FilterML field of the suma command. The FilterML field specifies a technology level to filter against; for example, 6100-03. If not specified, the value returned by `oslevel -r` on the local system will be used.

The following request type (RqType) values are obsolete and are no longer supported on AIX 7:

APAR	Request to download an APAR.
Critical	Request to download the latest critical fixes.
Security	Request to download the latest security fixes.
Fileset	Request to download a specific fileset.

Also, the field FilterSysFile that was once used to filter against the inventory of a running system is not supported on AIX 7.

The integration of SUMA and Electronic Customer Care has only been implemented on AIX 7 and not on any of the previous AIX releases. Nevertheless, SUMA on AIX 7 can be used to download AIX V5.3 TL 5300-06 and newer updates. AIX V5.3 TL 5300-06 was released in June 2007 and is the starting level of updates that are loaded into the eCC update repository.

The conversion of SUMA to use eCC instead of fixget has significant impact on the supported protocols utilized for fix server communication and to download updates. The following protocol-specific characteristics and changes are related to the relevant SUMA configuration parameters:

► FIXSERVER_PROTOCOL

   The FIXSERVER_PROTOCOL parameter specifies the protocol to be used for communication between the eCC Common Client and the eCC fix service provider as a part of the order request that SUMA will make to get the list of fixes. SUMA utilizes the Hypertext Transfer Protocol Secure (HTTPS) protocol since it is the only supported protocol for communication between the eCC Common Client and the IBM fix service provider. The only allowed value for

this configuration setting is https. The http setting of previous AIX releases is no longer supported.

► DOWNLOAD_PROTOCOL

The DOWNLOAD_PROTOCOL parameter specifies the protocol to be used for communication by the eCC Common Client for a download request from SUMA. SUMA takes advantage of the secure and multi-threaded Download Director Protocol (DDP) if the Hypertext Transfer Protocol (HTTP) has been configured. The HTTP protocol is specified by default and is recommended as eCC protocol for downloading updates. The related value for this configuration setting is http. The **suma** command can be used to modify the default configuration to use the HTTP Secure (HTTPS) protocol for downloads. But the related https setting restricts the secure downloads to single-threaded operations. The ftp setting of previous AIX releases is no longer supported.

Example 9-16 shows the **suma -c** command output on an AIX V7.1 TL 7100-00 system after a SUMA base configuration has been performed.

Example 9-16   SUMA default base configuration on AIX V7.1

```
7501lp01:/> suma -c
        FIXSERVER_PROTOCOL=https
        DOWNLOAD_PROTOCOL=http
        DL_TIMEOUT_SEC=180
        DL_RETRY=1
        HTTP_PROXY=
        HTTPS_PROXY=
        SCREEN_VERBOSE=LVL_INFO
        NOTIFY_VERBOSE=LVL_INFO
        LOGFILE_VERBOSE=LVL_VERBOSE
        MAXLOGSIZE_MB=1
        REMOVE_CONFLICTING_UPDATES=yes
        REMOVE_DUP_BASE_LEVELS=yes
        REMOVE_SUPERSEDE=yes
        TMPDIR=/var/suma/tmp
```

The SUMA-related eCC-specific base configuration properties are stored in the `eccBase.properties` file under the directory `/var/suma/data`. The initial version of the `eccBase.properties` file is installed as part of the `bos.suma` fileset. Example 9-17 shows the content of the `eccBase.properties` file after a SUMA default base configuration has been done on an AIX 7 system.

Example 9-17   eccBase.properties file after SUMA default base configuration

```
7501lp01:/> cat /var/suma/data/eccBase.properties
```

```
## ecc version: 1.0504
#Thu Apr 08 09:02:56 CDT 2010
DOWNLOAD_READ_TIMEOUT=180
INVENTORY_COLLECTION_CONFIG_DIR=/var/suma/data
DOWNLOAD_RETRY_WAIT_TIME=1
TRACE_LEVEL=SEVERE
DOWNLOAD_SET_NEW_DATE=TRUE
AUDITLOG_MAXSIZE_MB=2
CONNECTIVITY_CONFIG_DIR=/var/ecc/data
PLATFORM_EXTENSION_CLASS=com.ibm.esa.ea.tx.ecc.PlatformExtensions
TRACE_FILTER=com.ibm.ecc
WS_TRACE_LEVEL=OFF
AUDITLOG_COUNT=2
TRACELOG_MAXSIZE_MB=4
DOWNLOAD_MAX_RETRIES=3
LOG_DIR=/var/suma/log
RETRY_COUNT=1
DOWNLOAD_MONITOR_INTERVAL=10000
REQUEST_TIMEOUT=600
```

The CONNECTIVITY_CONFIG_DIR variable in the `eccBase.properties` file points to the directory where the connectivity configuration information is stored in the `eccConnect.properties` file. An initial version of this file is installed as part of the `bos.ecc_client.rte` fileset in the `/var/ecc/data` directory. The `eccConnect.properties` file connectivity configuration information is shared by SUMA, IBM Electronic Service Agent, and the Inventory Scout. This file holds the proxy server information if required for the service communication.

The proxy configuration task is supported by the SMIT panels that are dedicated to set up an AIX service configuration. System administrators can use the **smit srv_conn** fastpath to directly access the Create/Change Service Configuration menu. In this menu the Create/Change Primary Service Configuration selection will bring up the Create/Change Primary Service Configuration menu where the desired connection type can be configured.

The following three alternatives are available for the connection type: Not configured, Direct Internet, and HTTP_Proxy. For the connection type HTTP_Proxy selection you need to provide the IP address of the proxy server, the port number used, and an optional authentication user ID. Up to two additional service configurations (secondary, and tertiary) are supported to back up the primary connection in case of a failure. Note that the HTTP_PROXY selection in SMIT supports both HTTP_PROXY and HTTPS_PROXY if the customer proxy server is configured to support both http and https.

10

National language support

AIX Version 7.1 continues to extend the number of nations and regions supported by its national language support. In this chapter, details about the following features and facilities are provided:

► 10.1, "Unicode 5.2 support" on page 392

► 10.2, "Code set alias name support for iconv converters" on page 392

► 10.3, "NEC selected characters support in IBM-eucJP" on page 393

10.1 Unicode 5.2 support

As part of the continuous ongoing effort to adhere to the most recent industry standards, AIX V7.1 provides the necessary enhancements to the existing Unicode locales in order to bring them up to compliance with the latest version of the Unicode standard, which is Version 5.2, as published by the Unicode Consortium.

The Unicode is a standard character coding system for supporting the worldwide interchange, processing, and display of the written texts of the diverse languages used throughout the world. Since November 2007 AIX V6.1 supports Unicode 5.0, which defines standardized character positions for over 99,000 glyphs in total. More than 8,000 additional code points have been defined in Unicode 5.1 (1624 code points, April 2008) and Unicode 5.2 (6,648 code points, October 2009). AIX V7.1 provides the necessary infrastructure to handle, store and transfer all Unicode 5.2 characters.

For in-depth information about Unicode 5.2, visit the official Unicode home page at:

http://www.unicode.org

10.2 Code set alias name support for iconv converters

National Language Support (NLS) provides a base for internationalization in which data often can be changed from one code set to another. Support of several standard converters for this purpose is provided by AIX, and the following conversion interfaces are offered by any AIX system:

iconv command   Allows you to request a specific conversion by naming the FromCode and ToCode code sets.

libiconv functions   Allows applications to request converters by name.

AIX can transfer, store, and convert data in more than 130 different code sets. In order to meet market requirements and standards, the number of code sets has been increased dramatically by different venders, organizations, and standard groups in the past decade. However, many code sets are maintained and named in different ways. This may raise code set alias name issues. A code set with a specific encoding scheme can have two or more different code set names in different platforms or applications.

For instance, ISO-8859-13 is an Internet Assigned Numbers Authority (IANA) registered code set for Estonian, a Baltic Latin language. The code set

ISO-8859-13 is also named as IBM-921, CP921, ISO-IR-179, windows-28603, LATIN7, L7, 921, 8859_13 and 28603 in different platforms. For obvious interoperability reasons it is desirable to provide an alias name mapping function in the AIX `/usr/lib/libiconv.a` library to unambiguously identify code sets to the AIX converters.

AIX 7 introduces an AIX code set mapping mechanism in `libiconv.a` that holds more than 1300 code set alias names based on code sets and alias names of different vendors, applications, and open source groups. Major contributions are based on code sets related to the International Components for Unicode (ICU), Java, Linux, WebSphere®, and many others.

Using the new alias name mapping function, iconv can now easily map ISO-8859-13, CP921, ISO-IR-179, windows-28603, LATIN7, L7, 921, 8859_13 or 28603 to IBM-921 (AIX default) and convert the data properly, for example. The code set alias name support for iconv converters is entirely transparent to the system and no initialization or configuration is required on behalf of the system administrator.

10.3 NEC selected characters support in IBM-eucJP

There are 83 Japanese characters known as *NEC selected characters*. NEC selected characters refers to a proprietary encoding of Japanese characters historically established by the Nippon Electric Company (NEC) corporation. NEC selected characters have been supported by previous AIX releases through the IBM-943 and UTF-8 code sets.

For improved interoperability and configuration flexibility, AIX V7.1 and the related AIX V6.1 TL 6100-06 release extend the NEC selected characters support to the IBM-eucJP code set used for the AIX ja_JP local.

The corresponding AIX Japanese input method and the dictionary utilities were enhanced to accept NEC selected characters in the ja_JP local, and all IBM-eucJP code set related iconv converters were updated to handle the newly added characters.

Table 10-1 shows the local (language_territory designation) and code set combinations, all of which are now supporting NEC selected characters.

Table 10-1   Locales and code sets supporting NEC selected characters

Local	Local code set	Full local name	Category
JA_JP	UTF-8	JA_JP.UTF-8	Unicode

Local	Local code set	Full local name	Category
ja_JP	IBM-eucJP	ja_JP.IBM-eucJP	Extended UNIX Code (EUC)
Ja_JP	IBM-943	Ja_JP.IBM-943	PC

Requirements and specifications for Japanese character sets can be found at the official website of the Japanese Industrial Standards Committee:

http://www.jisc.go.jp/

11

Hardware and graphics support

This chapter discusses the new hardware support and graphic topics new in AIX Version 7.1, arranged as follows:

- ▶ 11.1, "X11 font updates" on page 396
- ▶ 11.2, "AIX V7.1 storage device support" on page 397
- ▶ 11.3, "Hardware support" on page 403

11.1 X11 font updates

AIX V7.1 contains font updates for X11 and the Common Desktop Environment (CDE) to properly exploit the latest TrueType fonts.

Existing fonts and their X Logical Font Description (XLFD) family names have changed to match the names provided. To preserve compatibility with prior releases of AIX, symbolic links have been provided to redirect the original file names to the new file names. Additionally, font aliases have been added to redirect the original XLFD names to the new names.

The Windows Glyph List (WGL) fonts have been removed in AIX V7.1. These fonts are already a subset of other fonts. It is not necessary to provide fonts that contain only the WGL. Table 11-1 lists the file names that have been removed.

Table 11-1   Removed WGL file names and fileset packages

File Name	Packaging Fileset
mtsans_w.ttf	X11.fnt.ucs.ttf
mtsansdw.ttf	X11.fnt.ucs.ttf
tnrwt_w.ttf	X11.fnt.ucs.ttf

A consideration with glyph subsets and the CDE: If one glyph in a font extends higher or lower than others, the font metrics will be affected such that a paragraph of text will appear to have excessive white space between each line.

To address this issue, the -dt interface user-* and -dt interface system-* font aliases used by CDE in many Unicode locales will, by default, point to fonts containing a reduced set of glyphs. This reduced set does not contain the large glyphs causing increased line height.

To override this default and force the use of fonts containing the complete set of glyphs, add `/usr/lib/X11/fonts/TrueType/complete` to the front of your font path, so that the -dt* font aliases in that directory are found before the ones in `/usr/lib/X11/fonts/TrueType`.

For example, if you select the EN_US locale at CDE login, but still need to be able to display Indic characters, you can run the following command:

```
# xset +fp /usr/lib/X11/fonts/TrueType/complete
```

Note that an alternative would be to have actually selected the EN_IN locale at CDE login instead of EN_US. Refer to the `/usr/lpp/X11/README` file for more information.

11.2 AIX V7.1 storage device support

AIX V7.1 expands the support for many IBM and vendor storage products.

The IBM System Storage® Interoperation Center (SSIC) provides a matrix for listing operating system support for the various IBM and vendor storage products.

The SSIC can be used to produce a matrix showing supported features and products by selecting search options, including:

► Operating system

► Operating system technology level

► Connection protocol

► Host platform

► Storage product family

The System Storage Interoperation Center can be found at:

http://www.ibm.com/systems/support/storage/config/ssic/displayesssearchwithoutjs.wss?start_over=yes

Note: At the time of publication, the SSIC was in the process of being updated to include support information for the AIX V7.1 release.

Figure 11-1 on page 398 shows the System Storage Interoperation Center.

Figure 11-1   The IBM System Storage Interoperation Center (SSIC)

By making selections from the drop-down boxes, the SSIC may be used to determine which features and products are available and supported for AIX V7.1.

In Figure 11-2 on page 400 multiple features and products are selected, which restricts the display results to combinations of these features and products.

> **Note:** The SSIC is updated regularly as feature and product offerings are added or removed. This search example was accurate at the time of publication but may change as features are added or removed.

Figure 11-2   The IBM SSIC - search example

The product version output from the System Storage Interoperation Center may be exported into a .xls format spreadsheet.

Figure 11-3 on page 402 shows an example search with the `Export Selected Product Version (xls)` selection option identified, and shown highlighted.

Figure 11-3   The IBM SSIC - the export to .xls option

Using the System Storage Interoperation Center can benefit system designers who are determining which features are available when designing new hardware and software architecture.

The SSIC can also be used as an entry reference point by storage and system administrators to determine prerequisite hardware or software dependencies when planning for upgrades to existing environments.

The SSIC is not intended to replace such tools as the IBM System Planning Tool (SPT) for POWER® processor-based systems or the IBM Fix Level Recommendation Tool (FLRT) for IBM POWER systems administrators. The SSIC should be used in conjunction with such tools as the SPT and FLRT, as well as any additional planning and architecture tools specific to your environment.

11.3 Hardware support

This section discusses the new hardware support and graphic topics new in AIX Version 7.1.

11.3.1 Hardware support

AIX V7.1 exclusively supports 64-bit Common Hardware Reference Platform (CHRP) machines with selected processors:

► PowerPC 970

► POWER4

► POWER5

► POWER6

► POWER7

The **prtconf** command can be used to determine the processor type of the managed system hardware platform.

Example 11-1 shows the root user running the **prtconf** command.

Example 11-1   The prtconf command to determine the processor type of the system

```
# whoami
root
# prtconf|grep 'Processor Type'
Processor Type: PowerPC_POWER7
```

The **prtconf** command run by LPAR shows that the processor type of the managed system hardware platform is `POWER7`.

To determine whether your managed system hardware platform may require firmware updates or additional prerequisites in order to run AIX V7.1, refer to the AIX V7.1 Release Notes, found at:

    `http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=/com.ibm.aix.ntl/releasenotes_kickoff.htm`

Abbreviations and acronyms

ABI	Application Binary Interface	**CD-ROM**	Compact Disk-Read Only Memory
AC	Alternating Current	**CDE**	Common Desktop Environment
ACL	Access Control List		
ACLs	Access Control Lists	**CEC**	Central Electronics Complex
AFPA	Adaptive Fast Path Architecture	**CHRP**	Common Hardware Reference Platform
AIO	Asynchronous I/O	**CID**	Configuration ID
AIX	Advanced Interactive Executive	**CLDR**	Common Locale Data Repository
APAR	Authorized Program Analysis Report	**CLI**	Command-Line Interface
API	Application Programming Interface	**CLVM**	Concurrent LVM
		CLiC	CryptoLight for C library
ARP	Address Resolution Protocol	**CMW**	Compartmented Mode Workstations
ASMI	Advanced System Management Interface	**CPU**	Central Processing Unit
AltGr	Alt-Graphic	**CRC**	Cyclic Redundancy Check
Azeri	Azerbaijan	**CSM**	Cluster Systems Management
BFF	Backup File Format		
BIND	Berkeley Internet Name Domain	**CT**	Component Trace
		CUoD	Capacity Upgrade on Demand
BIST	Built-In Self-Test		
BLV	Boot Logical Volume	**DAC**	Discretionary Access Controls
BOOTP	Boot Protocol		
BOS	Base Operating System	**DCEM**	Distributed Command Execution Manager
BSD	Berkeley Software Distribution	**DCM**	Dual Chip Module
CA	Certificate Authority	**DES**	Data Encryption Standard
CAA	Cluster Aware AIX	**DGD**	Dead Gateway Detection
CATE	Certified Advanced Technical Expert	**DHCP**	Dynamic Host Configuration Protocol
CD	Compact Disk	**DLPAR**	Dynamic LPAR
CD	Component Dump facility	**DMA**	Direct Memory Access
CD-R	CD Recordable	**DNS**	Domain Name Server

DR	Dynamic Reconfiguration		**HACMP™**	High Availability Cluster Multiprocessing
DRM	Dynamic Reconfiguration Manager		**HBA**	Host Bus Adapters
DST	Daylight Saving Time		**HMC**	Hardware Management Console
DVD	Digital Versatile Disk		**HPC**	High Performance Computing
DoD	Department of Defense		**HPM**	Hardware Performance Monitor
EC	EtherChannel		**HTML**	Hypertext Markup Language
ECC	Error Checking and Correcting		**HTTP**	Hypertext Transfer Protocol
eCC	Electronic Customer Care		**Hz**	Hertz
EGID	Effective Group ID		**I/O**	Input/Output
EOF	End of File		**IBM**	International Business Machines
EPOW	Environmental and Power Warning		**ICU**	International Components for Unicode
EPS	Effective Privilege Set		**ID**	Identification
eRAS	enterprise Reliability Availability Serviceability		**IDE**	Integrated Device Electronics
ERRM	Event Response Resource Manager		**IEEE**	Institute of Electrical and Electronics Engineers
ESA	Electronic Service Agent		**IETF**	Internet Engineering Task Force
ESS	Enterprise Storage Server®		**IGMP**	Internet Group Management Protocol
EUC	Extended UNIX Code		**IANA**	Internet Assigned Numbers Authority
EUID	Effective User ID			
F/C	Feature Code		**IP**	Internetwork Protocol
FC	Fibre Channel		**IPAT**	IP Address Takeover
FCAL	Fibre Channel Arbitrated Loop		**IPL**	Initial Program Load
FDX	Full Duplex		**IPMP**	IP Multipathing
FFDC	First Failure Data Capture		**IQN**	iSCSI Qualified Name
FLOP	Floating Point Operation		**ISC**	Integrated Solutions Console
FRU	Field Replaceable Unit		**ISSO**	Information System Security Officer
FTP	File Transfer Protocol		**ISV**	Independent Software Vendor
GDPS®	Geographically Dispersed Parallel Sysplex™		**ITSO**	International Technical Support Organization
GID	Group ID		**IVM**	Integrated Virtualization Manager
GPFS	General Parallel File System			
GSS	General Security Services			
GUI	Graphical User Interface			

iWARP	Internet Wide Area RDMA Protocol	**MIBs**	Management Information Bases
J2	JFS2	**ML**	Maintenance Level
JFS	Journaled File System	**MLS**	Multi Level Security
KAT	Kernel Authorization Table	**MP**	Multiprocessor
KCT	Kernel Command Table	**MPIO**	Multipath I/O
KDT	Kernel Device Table	**MPS**	Maximum Privilege Set
KRT	Kernel Role Table	**MTU**	Maximum Transmission Unit
KST	Kernel Security Table	**Mbps**	Megabits Per Second
L1	Level 1	**NDAF**	Network Data Administration Facility
L2	Level 2	**NEC**	Nippon Electric Company
L3	Level 3	**NFS**	Network File System
LA	Link Aggregation	**NIB**	Network Interface Backup
LACP	Link Aggregation Control Protocol	**NIH**	National Institute of Health
LAN	Local Area Network	**NIM**	Network Installation Management
LDAP	Light Weight Directory Access Protocol	**NIMOL**	NIM on Linux
LED	Light Emitting Diode	**NIS**	Network Information Server
LFS	Logical File System	**NLS**	National Language Support
LFT	Low Function Terminal	**NTP**	Network Time Protocol
LMB	Logical Memory Block	**NVRAM**	Non-Volatile Random Access Memory
LPA	Loadable Password Algorithm	**ODM**	Object Data Manager
LPAR	Logical Partition	**OFA**	OpenFabrics Alliance
LPP	Licensed Program Product	**OFED**	OpenFabrics Enterprise Distribution
LPS	Limiting Privilege Set	**OSGi**	Open Services Gateway Initiative
LRU	Least Recently Used page replacement demon	**OSPF**	Open Shortest Path First
LUN	Logical Unit Number	**PCI**	Peripheral Component Interconnect
LUNs	Logical Unit Numbers	**PIC**	Pool Idle Count
LV	Logical Volume	**PID**	Process ID
LVCB	Logical Volume Control Block	**PIT**	Point-in-time
LVM	Logical Volume Manager	**PKI**	Public Key Infrastructure
LWI	Light Weight Infrastructure	**PLM**	Partition Load Manager
MAC	Media Access Control		
MBps	Megabytes Per Second		
MCM	Multichip Module		

PM	Performance Monitor	**RNIC**	RDMA-capable Network Interface Controller
POSIX	Portable Operating System Interface	**RPC**	Remote Procedure Call
POST	Power-On Self-test	**RPL**	Remote Program Loader
POWER	Performance Optimization with Enhanced RISC (Architecture)	**RPM**	Red Hat Package Manager
		RSA	Rivet, Shamir, Adelman
PPC	Physical Processor Consumption	**RSCT**	Reliable Scalable Cluster Technology
PPFC	Physical Processor Fraction Consumed	**RSH**	Remote Shell
		RTE	Runtime Error
PSPA	Page Size Promotion Aggressiveness Factor	**RTEC**	Runtime Error Checking
		RUID	Real User ID
PTF	Program Temporary Fix	**S**	System Scope
PTX	Performance Toolbox	**SA**	System Administrator
PURR	Processor Utilization Resource Register	**SAN**	Storage Area Network
		SAS	Serial-Attached SCSI
PV	Physical Volume	**SCSI**	Small Computer System Interface
PVID	Physical Volume Identifier		
PVID	Port Virtual LAN Identifier	**SCTP**	Stream Control Transmission Protcol
QoS	Quality of Service		
RAID	Redundant Array of Independent Disks	**SDD**	Subsystem Device Driver
		SED	Stack Execution Disable
RAM	Random Access Memory	**SFDC**	Second Failure Data Capture
RAS	Reliability, Availability, and Serviceability	**SLs**	Sensitivity Labels
		SMI	Structure of Management Information
RBAC	Role Based Access Control		
RCP	Remote Copy	**SMIT**	Systems Management Interface Tool
RDAC	Redundant Disk Array Controller		
		SMP	Symmetric Multiprocessor
RDMA	Remote Direct Memory Access	**SMS**	System Management Services
RGID	Real Group ID	**SMT**	Simultaneous Muli-threading
RIO	Remote I/O	**SO**	System Operator
RIP	Routing Information Protocol	**SP**	Service Processor
RISC	Reduced Instruction-Set Computer	**SPOT**	Shared Product Object Tree
		SRC	System Resource Controller
RMC	Resource Monitoring and Control	**SRN**	Service Request Number

SSA	Serial Storage Architecture	**VPSS**	Variable Page Size Support
SSH	Secure Shell	**VRRP**	Virtual Router Redundancy Protocol
SSL	Secure Socket Layer	**VSD**	Virtual Shared Disk
SUID	Set User ID	**WED**	WebSphere Everyplace® Deployment V6.0
SUMA	Service Update Management Assistant	**WLM**	Workload Manager
SVC	SAN Virtualization Controller	**WPAR**	Workload Partitions
TCB	Trusted Computing Base	**WPS**	Workload Partition Privilege Set
TCP/IP	Transmission Control Protocol/Internet Protocol		
TE	Trusted Execution		
TEP	Trusted Execution Path		
TLP	Trusted Library Path		
TLS	Transport Layer Security		
TSA	Tivoli System Automation		
TSD	Trusted Signature Database		
TTL	Time-to-live		
UCS	Universal-Coded Character Set		
UDF	Universal Disk Format		
UDID	Universal Disk Identification		
UFST	Universal Font Scaling Technology		
UID	User ID		
ULM	User Loadable Module		
UPS	Used Privilege Set		
VG	Volume Group		
VGDA	Volume Group Descriptor Area		
VGSA	Volume Group Status Area		
VIPA	Virtual IP Address		
VLAN	Virtual Local Area Network		
VMM	Virtual Memory Manager		
VP	Virtual Processor		
VPA	Visual Performance Analyzer		
VPD	Vital Product Data		
VPN	Virtual Private Network		

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

For information about ordering these publications, see "How to get Redbooks" on page 415. Note that some of the documents referenced here may be available in softcopy only.

► *AIX Version 4.2 Differences Guide*, SG24-4807

► *AIX Version 4.3 Differences Guide*, SG24-2014

► *AIX 5L Differences Guide Version 5.2 Edition*, SG24-5765

► *AIX 5L Differences Guide Version 5.3 Edition*, SG24-7463

► *AIX 5L Differences Guide Version 5.3 Addendum*, SG24-7414

► *IBM AIX Version 6.1 Differences Guide*, SG24-7559

► *Sun Solaris to IBM AIX 5L Migration: A Guide for System Administrators*, SG24-7245

► *AIX Reference for Sun Solaris Administrators*, SG24-6584

► *IBM AIX 5L Reference for HP-UX System Administrators*, SG24-6767

► *AIX V6 Advanced Security Features Introduction and Configuration*, SG24-7430

► *Tivoli Management Services Warehouse and Reporting*, SG24-7290

► *AIX Logical Volume Manager from A to Z: Introduction and Concepts*, SG24-5432

► *IBM System p5 Approaches to 24x7 Availability Including AIX 5L*, SG24-7196

► *Introduction to Workload Partition Management in IBM AIX Version 6.1*, SG24-7431

► *IBM Power 710 and 730 Technical Overview and Introduction*, REDP-4636

► *IBM Power 720 and 740 Technical Overview and Introduction*, REDP-4637

► *IBM Power 750 and 755 Technical Overview and Introduction*, REDP-4638

► *IBM Power 770 and 780 Technical Overview and Introduction*, REDP-4639

- *IBM Power 795 Technical Overview and Introduction*, REDP-4640

Other publications

These publications are also relevant as further information sources:

- *Technical Reference: Kernel and Subsystems, Volume 1*, SC23-6612

Online resources

These Web sites are also relevant as further information sources:

- Software binary compatibility site:
  http://www.ibm.com/systems/power/software/aix/compatibility/

- *Technical Reference: Kernel and Subsystems, Volume 1*, SC23-6612 of the AIX product documentation:
  http://publib.boulder.ibm.com/infocenter/aix/v6r1/topic/com.ibm.aix.kerneltechref/doc/ktechrf1/ktechrf1.pdf

- Open Group Base Specifications, Issue 7
  http://www.unix.org/2008edition

- AIX V7.1 documentation
  http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=/com.ibm.aix.ntl/releasenotes_kickoff.htm

- SSD configuration information
  http://www.ibm.com/developerworks/wikis/display/WikiPtype/Solid+State+Drives
  http://www.ibm.com/developerworks/wikis/display/wikiptype/movies

- *Positioning Solid State Disk (SSD) in an AIX environment*
  http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101560

- *Writing AIX kernel extensions*
  http://www.ibm.com/developerworks/aix/library/au-kernelext.html

- *AIX Installation and Migration Guide*, SC23-6722
  http://publib.boulder.ibm.com/infocenter/aix/v7r1/topic/com.ibm.aix.install/doc/insgdrf/insgdrf_pdf.pdf

- AIX migration script
  http://publib.boulder.ibm.com/infocenter/aix/v7r1/topic/com.ibm.aix.install/doc/insgdrf/migration_scripts.htm

- ► AIX V7.1 technical references
  http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=/com.ibm.aix.doc/doc/base/technicalreferences.htm

- ► AIX man pages
  http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=/com.ibm.aix.baseadmn/doc/baseadmndita/aix_ev.htm

- ► *xCAT 2 Guide for the CSM System Administrator*, REDP-4437
  http://www.redbooks.ibm.com/redpapers/pdfs/redp4437.pdf

- ► IBM Systems Director publications
  http://www.ibm.com/systems/management/director/
  http://www.ibm.com/power/software/management/

- ► IBM Systems Director installation
  http://publib.boulder.ibm.com/infocenter/director/v6r2x/index.jsp?topic=/com.ibm.director.install.helps.doc/fqm0_t_preparing_to_install_ibm_director_on_aix.html
  http://publib.boulder.ibm.com/infocenter/director/v6r2x/index.jsp?topic=/com.ibm.director.cli.helps.doc/fqm0_r_cli_remote_access_cmds.html

- ► AIX Expansion Pack
  http://www.ibm.com/systems/power/software/aix/expansionpack/

- ► Detailed DWARF debugging information
  http://www.dwarfstd.org

- ► AIX Event Infrastructure
  http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=/com.ibm.aix.baseadmn/doc/baseadmndita/aix_ev.htm

- ► Active Memory Expansion
  http://www.ibm.com/systems/power/hardware/whitepapers/am_exp.html

- ► Internet System Consortium
  http://www.isc.org

- ► NTP protocol
  http://alumni.media.mit.edu/~nelson/research/ntp-survey99

- ► Network Time Protocol project
  http://www.ntp.org/
  http://www.isc.org/
  http://www.rfcs.org/

- ► *NTP Version 4 Release Notes*
  http://www.eecis.udel.edu/~mills/ntp/html/release.html

- ► *AIX V6 Advanced Security Features Introduction and Configuration*, SG24-7430
  http://www.redbooks.ibm.com/abstracts/sg247430.html?Open
- ► IBM RealSecure Server Sensor for AIX
  http://www.ibm.com/systems/power/software/aix/security/solutions/iss.html
- ► AIX V7.1 Release Notes
  http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=/com.ibm.aix.ntl/releasenotes_kickoff.htm
- ► IBM Power Systems firmware
  http://www14.software.ibm.com/webapp/set2/firmware/gjsn
- ► *AIX Installation and Migration Guide*, SC23-6722
  http://publib.boulder.ibm.com/infocenter/aix/v7r1/topic/com.ibm.aix.install/doc/insgdrf/insgdrf_pdf.pdf
- ► AIX chedition command reference
  http://publib.boulder.ibm.com/infocenter/aix/v7r1/topic/com.ibm.aix.cmds/doc/aixcmds1/chedition.htm
- ► Managing AIX Editions
  http://publib.boulder.ibm.com/infocenter/aix/v7r1/topic/com.ibm.aix.install/doc/insgdrf/sw_aix_editions.htm
- ► kgetsystemcfg Kernel Service
  http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=/com.ibm.aix.kerneltechref/doc/ktechrf1/kgetsystemcfg.htm
- ► loopmount command reference
  http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=/com.ibm.aix.cmds/doc/aixcmds3/loopmount.htm
- ► loopmount command guide
  http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=/com.ibm.aix.baseadmn/doc/baseadmndita/loopback_main.htm
- ► Bull freeware download
  http://www.bullfreeware.com
- ► fixget script download
  http://www14.software.ibm.com/webapp/set2/fixget
- ► Unicode home page
  http://www.unicode.org
- ► Japanese Industrial Standards Committee
  http://www.jisc.go.jp/
- ► System Storage Interoperation Center
  http://www.ibm.com/systems/support/storage/config/ssic/displayesssearchwithoutjs.wss?start_over=yes

- National Institute of Health
  ftp://elsie.nci.nih.gov/pub/

- My developerWorks Blogs, Chris's AIX blog:
  https://www.ibm.com/developerworks/mydeveloperworks/blogs/cgaix/?lang=en_us

- My developerWorks: Blogs, AIXpert blog:
  https://www.ibm.com/developerworks/mydeveloperworks/blogs/aixpert/?lang=en_us

- AIX 7.1 Information Center

  http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp

How to get Redbooks

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks publications, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

IBM

Redbooks

IBM AIX Version 7.1 Differences Guide

(1.0" spine)
0.875"<->1.498"
460 <-> 788 pages

IBM AIX Version 7.1 Differences Guide

IBM®

Redbooks®

AIX - The industrial strength UNIX operating system

AIX Version 7.1 Standard Edition enhancements

An expert's guide to the new release

This IBM Redbooks publication focuses on the enhancements to IBM AIX Version 7.1 Standard Edition. It is intended to help system administrators, developers, and users understand these enhancements and evaluate potential benefits in their own environments.

AIX Version 7.1 introduces many new features, including:

► Domain Role Based Access Control
► Workload Partition enhancements
► Topas performance tool enhancements
► Terabyte segment support
► Cluster Aware AIX functionality

AIX Version 7.1 offers many other new enhancements, and you can explore them all in this publication.

For clients who are not familiar with the enhancements of AIX through Version 5.3, a companion publication, *AIX Version 6.1 Differences Guide*, SG24-7559, is available.