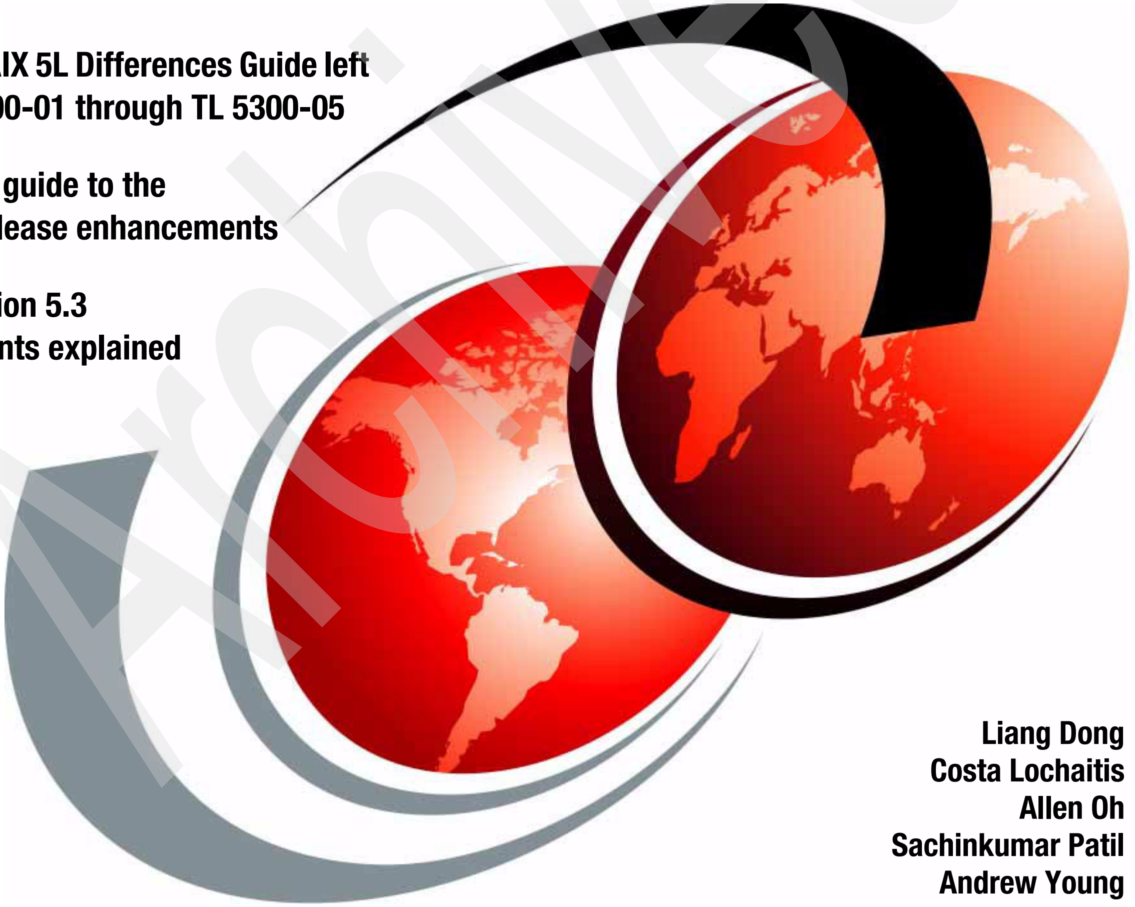
IBM

AIX 5L Differences Guide
Version 5.3 Addendum



Where the AIX 5L Differences Guide left off - ML 5300-01 through TL 5300-05

An expert's guide to the between-release enhancements

AIX 5L Version 5.3 enhancements explained

Liang Dong
Costa Lochaitis
Allen Oh
Sachinkumar Patil
Andrew Young

Redbooks

ibm.com/redbooks

International Technical Support Organization

AIX 5L Differences Guide Version 5.3 Addendum

April 2007

SG24-7414-00

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

First Edition (April 2007)

This edition applies to AIX 5L Version 5.3, program number 5765-G03 ML 5300-01 through TL 5300-05.

Contents

Figures

Tables

Examples

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

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

- Redbooks (logo) ®
- pSeries®
- AFS®
- AIX 5L™
- AIX®
- BladeCenter®
- DFS™
- Enterprise Storage Server®
- General Parallel File System™
- Geographically Dispersed Parallel Sysplex™
- GDPS®
- GPFS™
- HACMP™
- IBM®
- Parallel Sysplex®
- PowerPC®
- POWER™
- POWER Hypervisor™
- POWER3™
- POWER4™
- POWER5™
- POWER5+™
- PTX®
- Redbooks®
- System p™
- System p5™
- Tivoli®

The following terms are trademarks of other companies:

Oracle, JD Edwards, PeopleSoft, Siebel, and TopLink are registered trademarks of Oracle Corporation and/or its affiliates.

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

Active Directory, Microsoft, 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 differences introduced in AIX® 5L™ Version 5.3 since the initial AIX 5L Version 5.3 release. It is intended to help system administrators, developers, and users understand these enhancements and evaluate potential benefits in their own environments.

Since AIX 5L Version 5.3 was introduced, many new features (including JFS2, LDAP, trace and debug, installation and migration, NFSv4, and performance tools enhancements) were introduced. There are many other improvements offered through updates for AIX 5L Version 5.3, and you can explore them in this book.

For clients who are not familiar with the base enhancements of AIX 5L Version 5.3, a companion publication, *AIX 5L Differences Guide Version 5.3 Edition*, SG24-7463, is available.

The team that 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.

Liang Dong is an Advisory IT Specialist from China. He has five years of experience in AIX 5L country field support and more than four years of application development experience. He is also the Technical Leader of IBM Greater China Group in System p™ and AIX 5L, and he is certified as an Advanced Technical Expert in System p and AIX 5L, HACMP™ Systems Expert, and also holds several industry certificates. His areas of expertise include problem determination, performance tuning, networking, and system dump analyzing.

Costa Lochaitis is a Software Engineer in IBM South Africa. He has been with IBM for seven years within the IBM Global Services Division in customer support and services delivery. His areas of expertise include IBM System p hardware, AIX 5L, and Linux®. He is a Certified Advanced Technical Expert.

Allen Oh is a Senior System Engineer and Solutions Architect for MoreDirect Professional Services, an IBM Premier Business Partner authorized to sell and service IBM System p, x, and Storage throughout the United States. He has over ten years of experience in UNIX®, AIX, AIX 5L, and enterprise server and storage technology. Allen holds several senior level industry certifications, and is an IBM Certified Advanced Technical Expert in pSeries® and AIX 5L. He is a graduate of the University of Southern California.

Sachinkumar Patil is a Staff Software Engineer for IBM India Software Labs, Pune. He is the Technical Team Leader for the DFS™ L3 support team. He has more than seven years of experience in software development. He has a Bachelors degree in Computer Technology from the University of Mumbai and a Bachelor of Science in Electronics from North Maharashtra University. During the last four years, he has worked on the Distributed File System (IBM DFS). His areas of expertise are mainly in file system domain, AFS®, DFS, NFS, UNIX operating system internals, and software development in C and C++ on AIX 5L, Linux, and SUN Solaris™.

Andrew Young is a Senior AIX Support Specialist within the UK UNIX support center in Farnborough. He has seven years of experience providing AIX support to customers within his local geography and around the world. His main areas of expertise are performance analysis and memory management. He holds a Masters degree in Chemistry from the University of Southampton, specializing in superconductor research.

The project that produced this publication was managed by:

Scott Vetter, IBM Austin

Thanks to the following people for their contributions to this project:

Bob G Kovacs, Julie Craft, Eduardo L Reyes, Ann Wigginton, Grover Neuman, Grover Neuman, Octavian F Herescu, Eric P Fried, Shiv Dutta, Arthur Tysor, William Brown, Ravi A Shankar, Shawn Mullen, Augie Mena III, Bret Olszewski, David Sheffield, Michael S Williams, Michael Lyons

Become a published author

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

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

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

- Use the online **Contact us** review 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

1

Application development and system debug

In the area of application development, this chapter covers the following major topics:

- Editor enhancements (5300-05)
- System debugger enhancements (5300-05)
- Trace event timestamping (5300-05)
- xmalloc debug enhancement (5300-05)
- Stack execution disable protection (5300-03)
- Environment variable and library enhancements
- Vector instruction set support (5300-03)
- Raw socket support for non-root users (5300-05)
- IOCP support for AIO (5300-05)

1.1 Editor enhancements (5300-05)

Prior to AIX 5L 5300-05, editors used a static array of 8192 characters to hold the current line. The capabilities of editors such as vi, ex, and ed are enhanced as follows:

- vi, ex, and ed are now able to process files with a line limit greater than 8192. The line length is now limited to the available storage.
- vi now can open huge files (more than 600 MB).
- vi can readjust text properly on a window or xterm resize.

For more information see the MODS and **errctrl** command documentation.

1.2 System debugger enhancements (5300-05)

Beginning with AIX 5L Version 5.3 5300-05, the **dbx** command has been enhanced by introducing the following changes:

- Addition of the $stack_details variable
- Addition of the frame subcommand
- Addition of the addcmd subcommand
- Deferred event support (addition of $deferevents) variable
- Regular expression symbol search
- Thread level breakpoint and watchpoint support
- A dump subcommand enhancement

Example 1-1 shows a sample program that shows the **dbx** command enhancements used during this discussion.

Example 1-1 Sample program used for explaining enhanced features of dbx

```
#include<stdio.h>
int add(int x, int y)
{
        int z=0;
        for(;y>=1;y--)
        z=z+x;
        return z;
}
int mul(int x, int y)
{
        int z=0;
        z=add(x,y);
        return z;
}

int main(){

     int result;
     result=mul(100,5);
     printf(“Final result is %d\n", result);
}

#cc -g example1.c
#dbx a.out
Type 'help' for help.
reading symbolic information ...
(dbx) stop at 5
[1] stop at 5
(dbx) stop at 12
[2] stop at 12
(dbx) stop at 19
[3] stop at 19
(dbx) run
[3] stopped in main at line 19
   19        result=mul(100,5);
(dbx) where
main(), line 19 in “example1.c”
(dbx) cont
[2] stopped in mul at line 12
   12           z=add(x,y);
(dbx) cont
[1] stopped in add at line 5
    5           for(;y>=1;y--)
(dbx) where
add(x = 100, y = 5), line 5 in “example1.c”
mul(x = 100, y = 5), line 12 in “example1.c”
main(), line 19 in “example1.c”
(dbx)
```

1.2.1 The $stack_details variable

The $stack_details variable displays extended details of each stack frame. If the debug program $stack_details variable is set, it displays the frame number and the register set for each active function or procedure displayed by the where subcommand.

By default, the $stack_details variable is disabled in **dbx**. Example 1-2 shows the output of the where subcommand without setting the $stack_details variable.

Example 1-2 The where subcommand default output

```
(dbx) where
add(x = 100, y = 5), line 5 in “example1.c”
mul(x = 100, y = 5), line 12 in “example1.c”
main(), line 19 in “example1.c”
```

Example 1-3 shows the output of the where subcommand, once the $stack_details variable is set.

Example 1-3 The where subcommand output once $stack_details is set

```
(dbx) set $stack_details
(dbx) where
---------
  $r0:0x10000460  $stkp:0x2ff22b90   $toc:0x20000848    $r3:0x00000000
  $r4:0x00000005    $r5:0x2ff22d08    $r6:0x00000000    $r7:0x2ff22ff8
  $r8:0x00000000    $r9:0x05030050   $r10:0xf02d5318   $r11:0xdeadbeef
 $r12:0xdeadbeef   $r13:0xdeadbeef   $r14:0x00000001   $r15:0x2ff22d00
 $r16:0x2ff22d08   $r17:0x00000000   $r18:0xdeadbeef   $r19:0xdeadbeef
 $r20:0xdeadbeef   $r21:0xdeadbeef   $r22:0xdeadbeef   $r23:0xdeadbeef
 $r24:0xdeadbeef   $r25:0xdeadbeef   $r26:0xdeadbeef   $r27:0xdeadbeef
 $r28:0xdeadbeef   $r29:0xdeadbeef   $r30:0xdeadbeef   $r31:0x200005c0
 $iar:0x1000037c   $msr:0x0002d0b2    $cr:0x28222882  $link:0x10000408
 $ctr:0xdeadbeef   $xer:0x20000020
          Condition status = 0:e 1:l 2:e 3:e 4:e 5:l 6:l 7:e
        [unset $noflregs to view floating point registers]

0 add(x = 100, y = 5), line 5 in "example1.c"
---------
$stkp:0x2ff22be0   $r14:0x00000001   $r15:0x2ff22d00   $r16:0x2ff22d08
 $r17:0x00000000   $r18:0xdeadbeef   $r19:0xdeadbeef   $r20:0xdeadbeef
 $r21:0xdeadbeef   $r22:0xdeadbeef   $r23:0xdeadbeef   $r24:0xdeadbeef
 $r25:0xdeadbeef   $r26:0xdeadbeef   $r27:0xdeadbeef   $r28:0xdeadbeef
 $r29:0xdeadbeef   $r30:0xdeadbeef   $r31:0x200005c0
 $iar:0x10000404  $link:0x10000460
        [unset $noflregs to view floating point registers]

1 mul(x = 100, y = 5), line 12 in "example1.c"
---------
```

```
$stkp:0x2ff22c30   $r14:0x00000001   $r15:0x2ff22d00   $r16:0x2ff22d08
 $r17:0x00000000   $r18:0xdeadbeef   $r19:0xdeadbeef   $r20:0xdeadbeef
 $r21:0xdeadbeef   $r22:0xdeadbeef   $r23:0xdeadbeef   $r24:0xdeadbeef
 $r25:0xdeadbeef   $r26:0xdeadbeef   $r27:0xdeadbeef   $r28:0xdeadbeef
 $r29:0xdeadbeef   $r30:0xdeadbeef   $r31:0x200005c0
 $iar:0x1000045c  $link:0x100001ec
        [unset $noflregs to view floating point registers]

2 main(), line 19 in "example1.c"
(dbx)
```

1.2.2 The frame subcommand

The frame subcommand changes the current function to the function corresponding to the specified stack frame number *num*. The current function is used for resolving names. The numbering of the stack frame starts from the currently active function's stack frame (the function frame that is currently active is always numbered 0). If there are *n* frames, the frame of the main function will be numbered n-1. When no frame number is specified, information about the function associated with the current frame is displayed.

The use of the frame subcommand is as shown in the following:

```
(dbx) frame 2
main(), line 19 in “example1.c”
(dbx) frame
main(), line 19 in “example1.c”
(dbx) frame 1
mul(x = 100, y = 5), line 12 in “example1.c”
(dbx) frame
mul(x = 100, y = 5), line 12 in “example1.c”
(dbx) frame 0
add(x = 100, y = 5), line 5 in “example1.c”
(dbx) frame
add(x = 100, y = 5), line 5 in “example1.c”
```

1.2.3 The addcmd subcommand

By using the addcmd subcommand, you can associate any **dbx** subcommand to the specified events, which will be executed whenever the breakpoint, tracepoint, or watchpoint corresponding to the event is met.

Example 1-4 shows the where and registers subcommands with breakpoint events using the addcmd subcommand.

Example 1-4 The addcmd subcommand example

```
(dbx) stop at 5
[1] stop at 5
(dbx) addcmd 1 “where;registers”
(dbx) stop at 12
[2] stop at 12
(dbx) addcmd 2 “where”
(dbx) stop at 19
[3] stop at 19
(dbx) addcmd 3 “registers”
(dbx) run
[3] stopped in main at line 19
   19        result=mul(100,5);
  $r0:0x100001ec  $stkp:0x2ff22c30   $toc:0x20000848    $r3:0x00000001
  $r4:0x2ff22d00    $r5:0x2ff22d08    $r6:0x00000000    $r7:0x2ff22ff8
  $r8:0x00000000    $r9:0x05030050   $r10:0xf02d5318   $r11:0xdeadbeef
 $r12:0xdeadbeef   $r13:0xdeadbeef   $r14:0x00000001   $r15:0x2ff22d00
 $r16:0x2ff22d08   $r17:0x00000000   $r18:0xdeadbeef   $r19:0xdeadbeef
 $r20:0xdeadbeef   $r21:0xdeadbeef   $r22:0xdeadbeef   $r23:0xdeadbeef
 $r24:0xdeadbeef   $r25:0xdeadbeef   $r26:0xdeadbeef   $r27:0xdeadbeef
 $r28:0xdeadbeef   $r29:0xdeadbeef   $r30:0xdeadbeef   $r31:0x200005c0
 $iar:0x10000454   $msr:0x0002d0b2    $cr:0x28222882  $link:0x100001ec
 $ctr:0xdeadbeef   $xer:0x20000020    $mq:0xdeadbeef
          Condition status = 0:e 1:l 2:e 3:e 4:e 5:l 6:l 7:e
        [unset $noflregs to view floating point registers]
        [unset $novregs to view vector registers]
in main at line 19
0x10000454 (main+0x14) 38600064         li   r3,0x64
(dbx) cont
[2] stopped in mul at line 12
   12           z=add(x,y);
mul(x = 100, y = 5), line 12 in "example1.c"
main(), line 19 in "example1.c"
(dbx) cont
[1] stopped in add at line 5
    5           for(;y>=1;y--)
add(x = 100, y = 5), line 5 in "example1.c"
mul(x = 100, y = 5), line 12 in "example1.c"
main(), line 19 in "example1.c"
  $r0:0x10000460  $stkp:0x2ff22b90   $toc:0x20000848    $r3:0x00000000
  $r4:0x00000005    $r5:0x2ff22d08    $r6:0x00000000    $r7:0x2ff22ff8
  $r8:0x00000000    $r9:0x05030050   $r10:0xf02d5318   $r11:0xdeadbeef
 $r12:0xdeadbeef   $r13:0xdeadbeef   $r14:0x00000001   $r15:0x2ff22d00
 $r16:0x2ff22d08   $r17:0x00000000   $r18:0xdeadbeef   $r19:0xdeadbeef
 $r20:0xdeadbeef   $r21:0xdeadbeef   $r22:0xdeadbeef   $r23:0xdeadbeef
 $r24:0xdeadbeef   $r25:0xdeadbeef   $r26:0xdeadbeef   $r27:0xdeadbeef
 $r28:0xdeadbeef   $r29:0xdeadbeef   $r30:0xdeadbeef   $r31:0x200005c0
 $iar:0x1000037c   $msr:0x0002d0b2    $cr:0x28222882  $link:0x10000408
 $ctr:0xdeadbeef   $xer:0x20000020    $mq:0xdeadbeef
          Condition status = 0:e 1:l 2:e 3:e 4:e 5:l 6:l 7:e
        [unset $noflregs to view floating point registers]
        [unset $novregs to view vector registers]
in add at line 5
0x1000037c (add+0x14) 8061006c        lwz   r3,0x6c(r1)
```

1.2.4 Deferred event support ($deferevents variable)

AIX 5L has introduced a new variable, $deferevents, that allows events when symbols are not present. By default the $deferevents variable is turned off in **dbx**. The following example shows the usage of the deferevent variable. It shows how to set a break point at the sach function that is not yet loaded into the running program:

```
(dbx) stop in sach
“sach” is not defined
(dbx) set $deferevents
(dbx) stop in sach
“sach” is not loaded. Creating deferred event:
<5> stop in sach
```

1.2.5 Regular expression symbol search / and ? subcommands

AIX 5L has introduced the slash (/) and question mark (?) characters as new subcommands in **dbx**. By using these subcommands, you can search dumps using regular expressions in the current source, forward and backward, respectively.

The use of the / subcommand (searches forward) in **dbx** is as follows:

```
(dbx) / add
    2   int add(int x, int y)
(dbx) / a*d
    2   int add(int x, int y)
```

To repeat the previous search:

```
(dbx) /
    2   int add(int x, int y)
```

The use of the ? subcommand (searches backward) in dbx is as follows:

```
(dbx) ? mul
```

```
    9   int mul(int x, int y)
(dbx) ? m*l
    9   int mul(int x, int y)
```

To repeat the previous search, perform the following:

```
(dbx) ?
    9   int mul(int x, int y)
```

1.2.6 Thread level breakpoint and watchpoint support

When debugging a multi-threaded program, it is beneficial to work with individual threads instead of with processes. The **dbx** command only works with user threads. In the **dbx** command documentation, the word *thread* is usually used alone to mean user thread. The **dbx** command assigns a unique thread number to each thread in the process being debugged, and also supports the concept of a running and a current thread:

Running thread The user thread that was responsible for stopping the program by reaching a breakpoint. Subcommands that single-step through the program work with the running thread.

Current thread The user thread that you are examining. Subcommands that display information work in the context of the current thread.

The **dbx** command has added some new subcommands that enable you to work with individual attribute objects, condition variables, mutexes, and threads. They are provided in Table 1-1.

Table 1-1 dbx subcommands for thread level debugging

dbx subcommand	Description
attribute	Displays information about all attribute objects, or attribute objects specified by attribute number
condition	Displays information about all condition variables, condition variables that have waiting threads, condition variables that have no waiting threads, or condition variables specified by condition number
mutex	Displays information about all mutexes, locked or unlocked mutexes, or mutexes specified by mutex number
thread	Displays information about threads, selects the current thread, and holds and releases threads

dbx subcommand	Description
tstophwp	Sets a thread-level hardware watchpoint stop
ttracehwp	Sets a thread-level hardware watchpoint trace
tstop	Sets a source-level breakpoint stop for a thread
tstopi	Sets an instruction-level breakpoint stop for a thread
ttrace	Sets a source-level trace for a thread
ttracei	Sets an instruction-level trace for a thread
tnext	Runs a thread up to the next source line
tnexti	Runs a thread up to the next machine instruction
tstep	Runs a thread one source line
tstepi	Runs a thread one machine instruction
tskip	Skips breakpoints for a thread

A number of subcommands that do not work with threads directly are also affected when used to debug a multithreaded program.

For further details of the thread-level debugging with thread-level breakpoint and watchpoint, refer to the man page of the **dbx** command.

1.2.7 A dump subcommand enhancement

Beginning with AIX 5L Version 5.3 with TL 5300-05, the dump subcommand in **dbx** can recognize wildcards. The syntax is as follows:

```
dump [ procedure | "PATTERN" ] [ >File ]
```

The dump subcommand displays the names and values of all variables in the specified procedure or those that match with the specified pattern. If the procedure parameter is a period (.), then all active variables are displayed. If neither the procedure nor the PATTERN parameter is specified, the current procedure is used. The PATTERN parameter is a wildcard expression using the *, ?, and [] meta-characters. When PATTERN is used, it displays all the matching symbols in the global space (from all the procedures). If the >File flag is used, the output is redirected to the specified file.

The following are examples:

- To display names and values of variables in the current procedure, enter:

  ```
  dump
  ```

- To display names and values of variables in the add_count procedure, enter:

  `dump add_count`

- To display names and values of variables starting from the characters, enter:

  `dump "s*"`

- To redirect names and values of variables in the current procedure to the var.list file, enter:

  `dump > var.list`

Example 1-5 shows the output of the dump subcommand in **dbx** for a minimal C language program.

Example 1-5 A dump subcommand example

```
# dbx a.out
Type 'help' for help.
reading symbolic information ...
(dbx) step
stopped in main at line 19
   19        result=mul(100,5);
(dbx) dump
main(), line 19 in "example1.c"
result = 0
__func__ = "main"
(dbx) dump "z*"
example1.mul.z
example1.add.z
(dbx) dump "mu*"
mul
(dbx) dump "mai*"
main
(dbx)
```

1.3 Consistency checkers (5300-03)

The **kdb** kernel debugger command has been enhanced to include additional consistency checking for kernel structures. Consistency checkers provide automated data structure integrity checks for selected structures. This can include examining state typically checked in normal and debug asserts, supporting a component debug level that allows additional error-checking without a special compile or reboot, use of data structure eye-catchers, and, in general, improved data structure validation. The check subcommand has been added to

kdb to support consistency checkers that run in a debugger context. To display the list of known checkers, run the check subcommand without flags within **kdb**. Example 1-6 shows a sample output of the check subcommand.

Example 1-6 Available consistency checkers with the kdb check subcommand

```
(0)> check
Please specify a checker name:

Kernel Checkers         Description
--------------------------------------------------------------------------------
proc                    Validate proc and pvproc structures
thread                  Validate thread and pvthread structures

Kernext Checkers        Description
--------------------------------------------------------------------------------
```

1.4 Trace event timestamping (5300-05)

Beginning with AIX 5L Version 5.3 with 5300-05, the trace event subroutines (trchook, trchook64, utrchook, utrchook64, trcgen, trcgenk, and so on) are enhanced to always record a time stamp in the trace record. Now all the events are implicitly appended with a time stamp.

For more information see 3.2.4, “Trace event macro stamping (5300-05)” on page 50.

1.5 xmalloc debug enhancement (5300-05)

Beginning with AIX 5L Version 5.3 with 5300-05 Technology Level, random sampling of xmalloc allocations is enabled to catch memory leaks, buffer overruns, and accesses to freed data. The xmalloc debug function is similar to the previous memory overlay detection system (MODS). MODS is disabled by default on AIX 5L systems, which means that these types of problems can often not be resolved at first occurrence and require a second failure with the diagnostics enabled. The xmalloc enhancement makes a first-time detection of these issues more likely.

The MODS-related **bosdebug** commands such as -M still work. The **errctrl** command can also be used to alter the error checking level of the xmalloc component, proc.xmdbg. The syntax is:

```
errctrl errorchecklevel={0..9} -c proc.xmdbg[.heap0]
```

Controls can be applied to a specific heap. Unlike enabling MODS, the checking level can be raised without requiring a reboot.

The xm kdb subcommand is enhanced to support this new feature. **`xm -Q`** will show the current probability levels. **`xm -u`** will show the outstanding allocation records, although whether a given allocation had a record created is controlled by the alloc_record probabilty value.

To specifically disable the xmalloc debug Run-Time Error Checking (RTEC) feature, use the following command:

```
#errctrl errcheckoff -c alloc.xmdbg -r
```

To enable xmalloc debug, use the following command:

```
#errctrl errcheckon -c alloc.xmdbg -r
```

To persistently disable all AIX RTEC across reboots, use the following command:

```
#errctrl -P errcheckoff
```

1.6 Stack execution disable protection (5300-03)

On a computer system, security breaches can take many forms. One of the most common methods is by exploiting buffer overflows or overruns. Buffer overflows or overruns are common programming errors where a process attempts to store data beyond the boundaries of a fixed length buffer. The result is that the extra data overwrites adjacent memory locations. The overwritten data may include other buffers, variables, and program flow data. This can cause a program to crash or execute incorrect procedures. In such conditions, intruders can attack a system and insert code into a running process through the buffer overflow, changing the execution path of the process. The return address is overwritten and redirected to the inserted-code location. Common causes of breaches include improper or nonexistent bounds checking, or incorrect assumptions about the validity of data sources. For example, a buffer overflow can occur when a data object is large enough to hold 1 KB of data, but the program does not check the bounds of the input and hence can be made to copy more than 1 KB into that data object.

You can prevent these attacks by blocking execution of attack code entering through the buffer overflow. This can take the form of disabling execution on the memory areas of a process where it commonly does not take place (stack and heap memory areas).

AIX 5L has enabled the stack execution disable (SED) mechanism to disable the execution of code on the stack and select data areas of a process. By disabling the execution and then terminating an infringing program, the attacker is prevented from gaining root user privileges through a buffer overflow attack. While this feature does not stop buffer overflows, it provides protection by disabling the execution of attacks on buffers that have been overflowed.

Beginning with the POWER4™ family of processors, there was a page-level execution enable or disable feature for the memory. The AIX 5L SED mechanism uses this underlying hardware support to implement a no-execution feature on select memory areas. Once this feature is enabled, the operating system checks and flags various files during a program's execution. It then alerts the operating system memory manager and the process managers that the SED is enabled for the process being created. The select memory areas are then marked for no-execution. If any execution occurs on these marked areas, the hardware raises an exception flag and the operating system stops the corresponding process. The exception and application termination details are captured through AIX 5L error log events.

SED is implemented through the **sedmgr** command. The **sedmgr** command permits control of the system-wide SED mode of operation as well as setting the executable file-based SED flags. The SED facility is available only with the AIX 5L 64-bit kernel. The syntax is as follows:

```
sedmgr [-m {off | all | select | setidfiles}] [-o {on | off}]
[-c {system | request | exempt} {file_name | file_group}]
[-d {file_name | directory_name}] [-h]
```

You can use the command to enable and control the level of stack execution performed on the system. This command can also be used to set the various flags in an executable file, controlling the stack execution disable. Any changes to the system-wide mode setting will take effect only after a system reboot.

If invoked without any parameters, the **sedmgr** command will display the current setting in regards to the stack execution disable environment.

To change the system-wide SED mode flag to setidfiles and the SED control flag to on, enter:

```
sedmgr -m setidfiles -o on
```

With this command example, the setidfiles option sets the mode of operation so that the operating system performs stack execution disable for the files with the request SED flag set and enables SED for the executable files with the following characteristics:

- **setuid** files owned by root

- **setid** files with primary group as system or security

To change the SED checking flag to exempt for the plans file, enter:

```
sedmgr -c exempt plans
```

To change the SED checking flag to select for all the executable files marked as a TCB file, type use following command:

```
sedmgr -c request TCB_files
```

To display the SED checking flag of the plans file, enter:

```
sedmgr -d plans
```

1.7 Environment variable and library enhancements

This section presents the major changes or additions with respect to environment variables and library enhancements.

1.7.1 Environment variables

This section covers the following enhancements:

- DR_MEM_PERCENT (5300-03)
- AIXTHREAD_READ_GUARDPAGES (5300-03)

DR_MEM_PERCENT (5300-03)

Dynamic addition or removal of memory from an LPAR running multiple dynamic LPAR-aware programs can result in a conflict for resources. By default, each program is notified equally about the resource change. For example, if 1 GB of memory is removed from an LPAR running two dynamic-aware programs, then, by default, each program is notified that 1 GB of memory has been removed. Because the two programs are generally unaware of each other, both of them will scale down their memory use by 1 GB, leading to inefficiency. A similar efficiency problem can also occur when new memory is added.

To overcome this problem, AIX 5L now allows application scripts to be installed with a percentage factor that indicates the percentage of the actual memory resource change. The system then notifies the application in the event of a dynamic memory operation. While installing the application scripts using the **drmgr** command, you can specify this percentage factor using the DR_MEM_PERCENT name=value pair. The application script will need to output this name=value pair when it is invoked by the **drmgr** command with the scriptinfo subcommand. The value must be an integer between 1 and 100. Any value outside of this range is ignored, and the default value, which is 100, is used. Additionally, you can also set this name=value pair as an environment variable at the time of installation. During installation, the value from the environment variable, if set, will override the value provided by the application script. Similarly, in applications using the SIGRECONFIG signal handler and dr_reconfig() system call, you can control the memory dynamic LPAR notification by setting the DR_MEM_PERCENT name=value pair as an environment variable before the application begins running. This value, however, cannot be changed without restarting the application.

AIXTHREAD_READ_GUARDPAGES (5300-03)

Beginning with AIX 5L Version 5.3 release 5300-03, the AIXTHREAD_READ_GUARDPAGES environment variable is added into the AIX 5L system. The AIXTHREAD_READ_GUARDPAGES environment variable enables or disables read access to the guard pages that are added to the end of the pthread stack. It can be set as follows:

```
#AIXTHREAD_READ_GUARDPAGES={ON|OFF};
#export AIXTHREAD_READ_GUARDPAGES
```

The change takes effect immediately in the shell session and will be effective for its duration.

You can make the change permanent on a system by adding the AIXTHREAD_READ_GUARDPAGES={ON|OFF} command to the /etc/environment file.

1.7.2 LIBRARY variables

The following environment variables for library functions have been added to AIX 5L:

- LD_LIBRARY_PATH (5300-03)
- LDR_PRELOAD and LDR_PRELOAD64 (5300-05)

These are discussed in the following sections.

The LD_LIBRARY_PATH variable (5300-03)

Beginning with AIX 5L Version 5.3 with the 5300-03 Recommended Maintenance package, AIX 5L introduced the LD_LIBRARY_PATH loader environment variable in addition to the existing LIBPATH. The LIBPATH or LD_LIBRARY _PATH environment variable may be used to specify a list of directories in which shared libraries or modules can be searched.

The library search path for any running application or the dlopen or exec subroutines is now as follows:

1. The LIBPATH environment variable will be searched.
2. If the LIBPATH environment is set then LD_LIBRARY_PATH will be ignored. Otherwise, the LD_LIBRARY_PATH environment variable will be searched.
3. The library search paths included during linking in the running application will be searched.

LDR_PRELOAD and LDR_PRELOAD64 (5300-05)

The LDR_PREELOAD and LDR_PRELOAD64 environment variables request the preloading of shared libraries. The LDR_PRELOAD option is for 32-bit processes, and the LDR_PRELOAD64 option is for 64-bit processes.

During symbol resolution, the pre-loaded libraries listed in this variable are searched first for every imported symbol, and only when it is not found in those libraries will the other libraries be searched. Pre-emptying of symbols from preloaded libraries works for both AIX 5L default linking and run-time linking. Deferred symbol resolution is unchanged.

The following example shows the usage of these environment variables:

```
#LDR_PRELOAD=”libx.so:liby.so(shr.o)”
#LDR_PRELOAD64=”libx64.so:liby64.so(shr64.o)”
#export LDR_PRELOAD; export LDR_PRELOAD64
```

Once these environment variables are set, any symbol resolution will happen first in the libx.so shared object, then in the shr.o member of liby.a, and then finally within the process dependencies. All dynamically loaded modules (modules loaded with the dlopen() or load() calls) will also be resolved first from the preloaded libraries listed by these environment variables.

These environment variables are useful to correct faulty functions without relinking. These are also useful for running alternate versions of functions, without replacing the original library.

1.7.3 Named shared library areas (5300-03)

By default, AIX 5L shares libraries among processes using a global set of segments referred to as the global shared library area. For 32-bit processes, this area consists of one segment for shared library text (segment 0xD) and one segment for pre-relocated library data (segment 0xF). Sharing text and pre-relocating data improves performance on systems where a large number of processes use common shared libraries. Because the global shared library area is a single fixed-size resource, attempts to share a set of libraries that exceed the capacity of the area cannot succeed. In this situation, a portion of a process libraries are loaded privately. Loading libraries privately, as opposed to shared, consumes private address space in the process and places greater demands on memory, leading to a degradation in overall system performance.

AIX 5L now allows the designation of named shared library areas that can replace the global shared library area for a group of processes. A named shared library area enables a group of processes to have the full shared library capacity available to them at the same location in the effective address space as the global shared library area (segments 0xD and 0xF). The named shared library area feature is enabled using the NAMEDSHLIB option to the LDR_CTRL environment variable as follows:

```
LDR_CNTRL=NAMEDSHLIB=shared1 dbstartup.sh
```

If the shared1 library area does not exist then the system will dynamically create it and the dbstartup.sh process will load its libraries there. Additional processes will be able to attach to a segment once it has been created. The system will dynamically purge libraries from the area as it fills up. However, slibclean can be run manually if required:

```
LDR_CNTRL=NAMEDSHLIB=shared1 slibclean
```

When the last process attached to the segment exits, the area will be dynamically removed. Multiple named shared library areas can be active on the system simultaneously, provided that they have a unique name. Named shared library areas can only be used by 32-bit processes.

By default, the named shared library area works in the same way as the global area, designating one segment for shared library data and one for text. However, it is possible to use an alternate memory model that dedicates both segments to shared library text. To do this, you can specify the doubletext32 option for the named shared library area:

```
LDR_CNTRL=NAMEDSHLIB=shared1,doubletext32 dbstartup.sh
```

This is useful for process groups that need to use more than 256 MB for shared library text. However, it does mean that library data will not be preloaded that may have additional performance implications. This option should therefore be considered on a case-by-case basis.

1.7.4 Modular I/O library (5300-05)

The Modular I/O (MIO) library allows you to analyze and tune I/O at the application level for optimal performance. The MIO library addresses the need for an application-level method for optimizing I/O. Using the MIO library, users can tune different applications that have conflicting needs for better I/O performance.

MIO architecture

The Modular I/O library consists of five I/O modules that may be invoked at runtime on a per-file basis. The modules currently available are:

mio module	The interface to the user program
pf module	A data prefetching module
trace module	A statistics-gathering module
recov module	A module to analyze failed I/O accesses and retry in case of failure
aix module	The MIO interface to the operating system

The default modules are mio and aix. The other modules are optional.

Examples of using MIO

There are many scenarios that are relevant to the MIO library:

- MIO can be implemented by linking to libtkio to redirect I/O calls to the MIO library.
- MIO can be implemented by adding the libmio.h header file to an application's source file in order to redirect I/O calls to the MIO library.
- MIO library diagnostic data is written to a stats file when the MIO_close subroutine is called.
- You can configure MIO at the application level.

For more detailed information about the modular I/O library and its references, refer to AIX 5L publications, located at:

`http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=/com.ibm.aix.prftungd/doc/prftungd/mio.htm`

1.7.5 POSIX prioritized I/O support (5300-03)

POSIX prioritized I/O is a real-time option for its asynchronous input and output operations (AIO). Beginning with AIX 5L Version 5.3 Release 5300-03, POSIX AIO interfaces are updated to allow prioritization. It affects the following interfaces:

- aio_write: asynchronous write to a file
- aio_read: asynchronous read from a file
- lio_listio: initiates a list of I/O requests with a single function call

This is done through the new support of the aio_reqprio field.

The current AIX POSIX AIO behavior involves up to three processes:

1. The user process that requests the asynchronous I/O
2. The kernel process posix_aioserver that serves the asynchronous I/O, but does not necessarily start the disk I/O
3. The syncd daemon that may start the disk I/O

The effective I/O may be started in the user process itself (if fastpath is set), in the posix_aio server process if the request applies on a file opened in synchronous mode or in direct mode, or in the syncd process in the standard case.

The priority assigned to each PIO request is set in the aio_reqprio field and is an indication of the desired order of execution of the request relative to other PIO requests for this file. The standard states that EINVAL must be returned for an invalid value of aio_reqprio independently to the process scheduling policy.

The aiocb structure is used for all POSIX operations. This structure is defined in the /usr/include/aio.h file and contains the following members:

```
int                 aio_fildes
off_t               aio_offset
char               *aio_buf
size_t              aio_nbytes
int                 aio_reqprio
struct sigevent     aio_sigevent
int                 aio_lio_opcode
```

In prioritized I/O supported operations, the asynchronous operation is submitted at a priority equal to the scheduling priority of the process minus aiocbp->aio_reqprio.

A PIO request will be queued in priority order. The queue will be chosen as AIO does it now, but only PIO requests will be linked to it. This means that the queues for AIO and PIO will coexist.

PIO requests are kept in order within priority queues. Each priority queue is executed in order, while AIO requests can be executed as resources are available.

It is important to note that the priority of the request can be lowered with respect to the priority of the process or thread that scheduled it.

1.8 Vector instruction set support (5300-03)

The AltiVec instruction set was developed between 1996 and 1998 by Keith Diefendorff, the distinguished scientist and director of microprocessor architecture at Apple Computer. Motorola trademarked the term AltiVec, so Apple uses the name Velocity Engine. With respect to IBM implementation, it is known as Single Instruction, Multiple Data (SIMD) or vector extension. In some cases the term VMX is used.

1.8.1 What is SIMD

Normally, a single instruction to a computer does a single thing. A single SIMD instruction will also generally do a single thing, but it will do it to multiple pieces of data at once. Thus, the vector processors might store fifty or more pieces of data in a single vector register.

Selected PowerPC® processors implement a SIMD-style vector extension. Often referred to as AltiVec or VMX, the vector extension to the PowerPC architecture provides an additional instruction set for performing vector and matrix mathematical functions.

The Vector Arithmetic Logic Unit is an SIMD-style arithmetic unit in which a single instruction performs the same operation on all the data elements of each vector. AIX 5L Version 5.3 with ML 5300-03 is the first AIX 5L release to enable vector programming. The IBM PowerPC 970 processor is the first processor supported by AIX 5L that implements the vector extension. These processors are currently found in the JS20/JS21 blade servers offered with the BladeCenter®.

1.8.2 Technical details

The vector extension consists of an additional set of 32 128-bit registers that can contain a variety of vectors including signed or unsigned 8-bit, 16-bit, or 32-bit integers, or 32-bit IEEE single-precision floating point numbers. There is a vector status and control register that contains a sticky status bit indicating saturation, as well as a control bit for enabling Java™ or non-5L Java mode for floating-point operations.

The default mode initialized by AIX 5L for every new process is Java-mode enabled, which provides IEEE-compliant floating-point operations. The alternate non-Java mode results in a less precise mode for floating point computations, which might be significantly faster on some implementations and for specific operations. For example, on the PowerPC 970 processor running in Java mode, some vector floating-point instructions will encounter an exception if the input operands or result are denormal, resulting in costly emulation by the operating system. For this reason, you are encouraged to consider explicitly enabling the non-Java mode if the rounding is acceptable, or to carefully attempt to avoid vector computations on denormal values.

The vector extension also includes more than 160 instructions providing load and store access between vector registers and memory, in register manipulation, floating point arithmetic, integer arithmetic and logical operations, and vector comparison operations. The floating point arithmetic instructions use the IEEE 754-1985 single precision format, but do not report IEEE exceptions. Default results are produced for all exception conditions as specified by IEEE for untrapped exceptions. Only IEEE default round-to-nearest rounding mode is provided. No floating-point division or square-root instructions are provided, but instead a reciprocal estimate instruction is provided for division, and a reciprocal square root estimate instruction is provided for square root.

There is also a 32-bit special purpose register that is managed by software to represent a bitmask of vector registers in use. This allows the operating system to optimize vector save and restore algorithms as part of context switch management.

1.8.3 Compiler support

There are a few compilers who support vector programming. In this publication, the IBM XL C/C++ V8.0 will be discussed as an example. In V8.0 of XL C/C++, SIMD-related compiler options and directives are added to support vector programming:

The -qvecnvol option

The -qvecnvol option specifies whether to use volatile or non-volatile vector registers, where volatile vector registers are those registers whose value is not preserved across function calls or across save context, jump or switch context system library functions. The -qvecnvol option instructs the compiler to use both volatile and non-volatile vector registers while -qnovecnvol instructs the compiler to use only volatile vector registers. This option is required for programs where there is risk of interaction between modules built with libraries prior to AIX 5L Version 5.3 with 5300-03 and vector register use. Restricting the compiler to use only volatile registers will protect your vector programs, but it potentially forces the compiler to store vector data to memory more often and therefore results in additional processing.

Note:

- In order the generate vector-enabled code, you should explicitly specify the -qenablevmx option.
- In order to use the -qvecnvol option, you need bos.adt.include Version 5.3.0.30 or later to be installed on your system .
- When the -qnoenablevmx compiler option is in effect, the -qnovecnvol option is ignored.
- The -qnovecnvol option performs independently of -qhot=simd | nosimd, -qaltivec | -qnoaltivec, and also vector directive NOSIMD.
- On AIX 5.3 with 5300-03, by default, 20 volatile registers (vr0–vr19) are used, and 12 non-volatile vector registers (vr20–vr31) are not used. You can use these registers only when -qvecnvol is in effect.
- The -qvecnvol option should be enabled only when no older code that saves and restores non-volatile registers is involved. Using -qvecnvol and linking with older code may result in runtime failure.

The -qenablevmx option

The -qenablevmx option enables generation of vector instructions. -qnoenablevmx is set by default. However, you can take advantage of the vector extensions by explicitly specifying the -qenablevmx compiler option.

Note:

- Some processors are able to support vector instructions. These instructions can offer higher performance when used with algorithmic-intensive tasks such as multimedia applications.
- The -qenablevmx compiler option enables generation of vector instructions, but you should not use this option unless your operating system version supports vector instructions.
- If -qnoenablevmx is in effect, -qaltivec, -qvecnvol, and -qhot=simd cannot be used.

The -qaltivec option

The -qaltivec option enables compiler support for vector data types. The AltiVec Programming Interface specification describes a set of vector data types and operators. This option instructs the compiler to support vector data types and operators and has effect only when -qarch is set or implied to be a target architecture that supports vector instructions and the -qenablevmx compiler option is in effect. Otherwise, the compiler will ignore -qaltivec and issue a warning message.

Also, if the -qnoenablevmx option is in effect. The compiler will ignore -qaltivec and issue a warning message.

When -qaltivec is in effect, the following macros are defined:

- __ALTIVEC__ is defined to 1.
- __VEC__ is defined to 10205.

The -qhot option

The -qhot option instructs the compiler to perform high-order loop analysis and transformations during optimization. -qhot adds the following new suboptions:

```
simd | nosimd
```

The compiler converts certain operations that are performed in a loop on successive elements of an array into a call to a vector instruction. This call calculates several results at one time, which is faster than calculating each result sequentially.

If you specify -qhot=nosimd, the compiler performs optimizations on loops and arrays, but avoids replacing certain code with calls to vector instructions.

This suboption has effect only when the effective architecture supports vector instructions and you specify -qenablevmx.

Note: If you specify an architecture that supports vector instructions and -qhot and -qenablevmx, -qhot=simd will be set as the default.

The simd suboption optimizes array data to run mathematical operations in parallel where the target architecture allows such operations. Parallel operations occur in 16-byte vector registers. The compiler divides vectors that exceed the register length into 16-byte units to facilitate optimization. A 16-byte unit can contain one of the following types of data:

- 4 integers
- 8 two-byte units
- 16 one-byte units

1.9 Raw socket support for non-root users (5300-05)

Sockets were developed in response to the need for sophisticated interprocess facilities to meet the following goals:

- Provide access to communications networks such as the Internet.
- Enable communication between unrelated processes residing locally on a single host computer and residing remotely on multiple host machines.

Sockets provide a sufficiently general interface to allow network-based applications to be constructed independently of the underlying communication facilities. They also support the construction of distributed programs built on top of communication primitives.

The socket subroutines serve as the application program interface for Transmission Control Protocol/Internet Protocol (TCP/IP).

Each socket has an associated type, which describes the semantics of communications using that socket. The socket type determines the socket communication properties such as reliability, ordering, and prevention of duplication of messages. The basic set of socket types on AIX is defined in the sys/socket.h file:

```
/*Standard socket types */
#define  SOCK_STREAM           1 /*virtual circuit*/
#define  SOCK_DGRAM            2 /*datagram*/
#define  SOCK_RAW              3 /*raw socket*/
#define  SOCK_RDM              4 /*reliably-delivered message*/
#define  SOCK_CONN_DGRAM       5 /*connection datagram*/
```

Other socket types can be defined.

The SOCK_RAW type is a raw socket that provides access to internal network protocols and interfaces. Raw sockets allow an application to have direct access to lower-level communication protocols. Raw sockets are intended for advanced users who want to take advantage of a protocol feature that is not directly accessible through a normal interface, or who want to build new protocols on top of existing low-level protocols.

Raw sockets are normally datagram-oriented, though their exact characteristics are dependent on the interface provided by the protocol.

Prior to AIX 5L Version 5.3 with TL 5300-05, raw sockets are available only to processes with root-user authority. If the application is run without root privilege, the following error returned is:

```
socket: Permission denied
```

After AIX 5L Version 5.3 with TL 5300-05, raw sockets can be opened by non-root users who have the CAP_NUMA_ATTACH capability. For non-root raw socket access, the **chuser** command assigns the CAP_NUMA_ATTACH capability, along with CAP_PROPAGATE.

For the user who is to be permitted raw socket use, the sysadmin should set the CAP_NUMA_ATTACH bit. While opening a raw socket, if the user is non-root, it is checked if this bit is on. If yes, raw socket access is permitted. If no, it is prohibited. The capabilities are assigned to a user using the syntax:

```
# chuser "capabilities=CAP_NUMA_ATTACH,CAP_PROPAGATE" <user>
```

This command adds the given capabilities to the user in /etc/security/user file.

1.10 IOCP support for AIO (5300-05)

The Asynchronous I/O (AIO) subsystem has been enhanced by the interaction of I/O completion ports (IOCP). Previously, the AIO interface that was used in a threaded environment was limited in that aio_nwait() collects completed I/O requests for all threads in the same process. In other words, one thread collects completed I/O requests that are submitted by another thread. Another limit was that multiple threads cannot invoke the collection routines (such as aio_nwait()) at the same time. If one thread issues aio_nwait() while another thread is calling it, the second aio_nwait() returns EBUSY. This limitation can affect I/O performance when many I/Os must run at the same time and a single thread cannot run fast enough to collect all the completed I/Os.

Using I/O completion ports with AIO requests provides the capability for an application to capture results of various AIO operations on a per-thread basis in a multithreaded environment. This design provides threads with a method of receiving completion status for only the AIO requests initiated by the thread.

The IOCP subsystem only provides completion status by generating completion packets for AIO requests. The I/O cannot be submitted for regular files through IOCP.

The behavior of AIO remains unchanged. An application is free to use any existing AIO interfaces in combination with I/O completion ports. The application is responsible for harvesting completion packets for any noncanceled AIO requests that it has associated with a completion port.

The application must associate a file with a completion port using the CreateIoCompletionPort() IOCP routine. The file can be associated with multiple completion ports, and a completion port can have multiple files associated with it. When making the association, the application must use an application-defined CompletionKey to differentiate between AIO completion packets and socket completion packets. The application can use different CompletionKeys to differentiate among individual files (or in any other manner) as necessary.

Important: This functionality enhancement may change performance in certain Oracle® database environments. If your system runs Oracle, consult with your IBM service representative prior to upgrading to AIX 5L 5300-05.

2

File systems and storage

In this chapter the following topics relating to storage management are discussed:

- JFS2 file system enhancements
- The mirscan command (5300-03)
- AIO fast path for concurrent I/O (5300-05)
- FAStT boot support enhancements (5300-03)
- Tivoli Access Manager pre-install (5300-05)
- Geographic Logical Volume Manager (5300-03)

2.1 JFS2 file system enhancements

In addition to the existing JFS2 features, the following enhancements have been added since the base AIX 5L Version 5.3 Release 5300:

- JFS2 file system freeze and thaw feature (5300-01)
- JFS2 file system rollback (5300-03)
- Enhancement for backup of files on a DMAPI-managed JFS2 file system (5300-03)
- JFS2 inode creation enhancement (5300-03)

These enhancements are discussed in the sections that follow.

2.1.1 JFS2 file system freeze and thaw (5300-01)

The JFS2 file system freeze and thaw feature was added to AIX 5L Version 5.3 with the 5300-01 Recommended Maintenance package. This feature provides an external interface whereby an application can request that a JFS2 file system freeze, or stay quiescent. After the freeze operation, the file system must remain quiescent until it is thawed or until a specified time-out has passed.

This means that the act of freezing a file system produces a nearly consistent on-disk image of the file system, and writes all dirty file system metadata and user data to the disk. In its frozen state, the file system is read-only, and anything that attempts to modify the file system or its contents must wait for the freeze to end. Modifications to the file system are still allowed after it is thawed, and the file system image might no longer be consistent after the thaw occurs.

Requests for freeze or thaw can be performed by using the **chfs** command or from the fscntl() API. To perform these operations, root authority is required.

Usage of fscntl for freeze and thaw file system is as follows:

```
fscntl(vfs, FSCNTL_FREEZE, (caddr_t)timeout, 0)
fscntl(vfs, FSCNTL_REFREEZE, (caddr_t)timeout, 0)
fscntl(vfs, FSCNTL_THAW, NULL, 0)
```

The parameters are described in Table 2-1.

Table 2-1 API fcntl parameter details for file system freeze and thaw features

Parameters	Details
FSCNTL_FREEZE	The file system specified by vfs_id is frozen for a specified amount of time. The argument is treated as an integral time-out value in seconds (instead of a pointer). The file system is thawed by FSCNTL_THAW or when the timeout expires. The timeout, which must be a positive value, can be renewed using FSCNTL_REFREEZE. The argument size must be 0.
FSCNTL_REFREEZE	The file system specified by vfs_id, which already must be frozen, has its timeout value reset. If the command is used on a file system that is not frozen, an error is returned. The argument is treated as an integral timeout value in seconds (instead of a pointer). The file system is thawed by FSCNTL_THAW or when the new timeout expires. The timeout must be a positive value. The argument size must be 0.
FSCNTL_THAW	The file system specified by vfs_id is thawed. If the file system is not frozen at the time of the call, an error is returned. The argument and argument size must both be 0.

Note: For all applications using this interface, use FSCNTL_THAW to thaw the file system rather than waiting for the timeout to expire. If the timeout expires, an error log entry is generated as an advisory.

The following show the usage of the **chfs** command to freeze and thaw a file system.

To freeze and thaw a file system, use the following command:

```
chfs -a freeze=<timeout | 0 | "off"> <file system name>
```

Example 2-1 shows the file system’s read-only behavior during its freeze timeout period.

Example 2-1 Freeze a file system using the chfs command

```
# chfs -a freeze=60 /tmp; date; echo "TEST FREEZE TIME OUT" >
/tmp/sachin.txt; cat /tmp/sachin.txt;date
Mon Dec 11 16:42:00 CST 2006
TEST FREEZE TIME OUT
Mon Dec 11 16:43:00 CST 2006
```

Similarly, the following command can be used to refreeze a file system:

```
chfs -a refreeze=<timeout> <file system name>
```

2.1.2 JFS2 file system rollback (5300-03)

File system rollback restores an entire file system to a valid point-in-time snapshot (target snapshot). It applies to JFS2 only and is implemented as a low-level block copy from snapshot storage to file system storage. A file system must be unmounted and remains inaccessible for the duration of the rollback. Rollback requires up to several minutes to complete. If the rollback is interrupted for any reason, the file system remains inaccessible until the rollback is restarted and completes.

The rollback restart procedure is simply to retry the failed **`rollback`** command. During a retry, the same snapshot must be targeted again.

To roll back a JFS2 file system to a point-in-time snapshot, you can use **`smit rollbacksnap`** or the **`rollback`** command, that has the following syntax:

```
rollback [-v ] [ -s ] [-c] snappedFS snapshotObject
```

Table 2-2 The rollback command parameter details

Parameters	Details
-v	This causes a count of blocks restored to be written to stdout. Useful in monitoring the progress of the rollback.
-s	This causes rollback not to delete the logical volumes associated with lost snapshots.
-c	This causes rollback to continue even when read/write errors are encountered when doing the block level copy.
snappedFS	The JFS2 system to roll back.
snapshotObject	The logical volume of the snapshot to revert to.

Note: The –c option should be used with care.

2.1.3 Backup of files on a DMAPI-managed JFS2 file system (5300-03)

Beginning with AIX 5L Version 5.3 with the 5300-03 Recommended Maintenance package, there are options in the **`tar`** and **`backbyinode`** commands that allow you to back up the extended attributes (EAs).

With the **backbyinode** command on a DMAPI file system, only the data resident in the file system at the time the command is issued is backed up. The **backbyinode** command examines the current state of metadata to do its work. This can be advantageous with DMAPI, because it backs up the state of the managed file system. However, any offline data will not be backed up.

To back up all of the data in a DMAPI file system, use a command that reads entire files, such as the **tar** command. This can cause a DMAPI-enabled application to restore data for every file accessed by the **tar** command, moving data back and forth between secondary and tertiary storage, so there can be performance implications.

2.1.4 JFS2 inode creation enhancement (5300-03)

JFS2 inode creation is enhanced by using variable inode extent sizes of less than 16 KB. Inodes are allocated dynamically by allocating inode extents that are contiguous chunks of inodes on disk. This helps the cases where there may be plenty of space available in the file system, but it is too fragmented to allow one 16 KB allocation. The new inode creation enhancement allows users to continue to create files in such circumstances.

2.1.5 JFS2 CIO and AIO fast path setup

Although AIO fast path for Concurrent I/O (CIO) is supported in AIX 5L Version 5.3 TL 5, it is not enabled by default. To turn on AIO fast path for CIO, use the command:

```
aioo –o fsfastpath=1
```

The **aioo** command change is dynamic and must be re-run after every system reboot.

The default setting for fsfastpath is 0. This should not be confused with the fastpath setting seen in the following command:

```
lsattr –El aio0
```

It should also not be confused with the setting in the following:

```
smitty aio
```

This setting is for raw logical volumes and should always be set to enable.

When the AIO fast path for CIO is enabled (fsfastpath =1), it is optional to reset maxservers and maxreqs to the system default of 10 or leave it as is. There is no performance difference in either case.

2.2 The mirscan command (5300-03)

The **mirscan** command provides additional resilience to LVM mirroring by allowing administrators to search for and correct physical partitions that are stale or unable to perform I/O operations. This can serve two purposes:

- Detection of partitions on a disk that have failed but have not recently been accessed.
- If a disk is to be replaced, the command can be used to ensure that the last good copy of a logical partition is not removed.

The command can be run against a logical volume, either against the entire LV or a specific copy, a physical volume, or a volume group. It will generate a report to standard out on the status of the partitions scanned. In addition, the command can be requested to attempt corrective actions to recover data.

2.2.1 The mirscan command syntax

The **mirscan** command has the following syntax:

```
mirscan -v vgname | -l lvname | -p pvname | -r reverse_pvname [ -a ]
[ -o ] [ -q nblks ] [ -c lvcopy ] [ -s strictness ]
[ -u upperbound ]
```

Common flags are described in Table 2-3 and additional information is available in the command man page.

Table 2-3 Commonly used flags for mirscan command

Flag	Description
-v vgname	Specifies the volume group to be scanned.
-l lvname	Specifies the logical volume to be scanned.
-p pvname	Specifies the physical volume to be scanned.
-r reverse_pvname	Specifies that any partitions in the volume group should be scanned if they do not reside on pvname but they do have a mirror copy on pvname.
-c lvcopy	Identifies a particular copy of the logical volume. The -c flag can only be specified in conjunction with the -l flag.
-a	Specifies that corrective action should be taken.

The -r reverse_pname flag takes a disk device as its argument and checks all partitions that do not reside on that device but that have a mirrored copy located there. This is useful for ensuring the integrity of a logical volume, prior to removing a failing disk.

2.2.2 Report format

The **mirscan** command generates a report to standard out describing the operations performed and the results. Example 2-2 shows a sample output.

Example 2-2 Report output from mirscan command

```
START TIME: Wed Nov 29 09:41:20 CST:2006
OP STATUS  PVNAME       PP  SYNC    IOFAIL  LVNAME       LP  CP  TARGETPV
TARGETPP
s  SUCCESS hdisk0       116 synced  no      fslv00        1   1
s  SUCCESS hdisk0       117 synced  no      fslv00        2   1
s  SUCCESS hdisk0       118 synced  no      fslv00        3   1
s  SUCCESS hdisk0       119 synced  no      fslv00        4   1
s  SUCCESS hdisk1       115 synced  no      fslv00        1   2
s  SUCCESS hdisk1       116 synced  no      fslv00        2   2
s  SUCCESS hdisk1       117 synced  no      fslv00        3   2
s  SUCCESS hdisk1       118 synced  no      fslv00        4   2
END TIME: Wed Nov 29 09:48:43 CST:2006
```

The report has 11 columns that are described in Table 2-4.

Table 2-4 Output columns from mirscan command

Field	Description
OP	Indicates the operation performed, s sync, r resync, f force resync, and m migration.
STATUS	Shows whether the operation was a success or a failure.
PVNAME	Identifies the name of the physical volume where the partition being operated on resides.
PP	Identifies the physical partition number of the partition being operated on.
SYNC	Shows whether the partition is synced or stale.
IOFAIL	The valid values for this field are yes or no. The value indicated refers to the state of the partition after the operation has been completed.
LVNAME	Identifies the name of the logical volume where the partition being operated on resides.

Field	Description
LP	Identifies the logical partition number of the partition being operated on.
CP	Identifies the logical copy number of the partition being operated on.
TARGETPV	Identifies the name of the physical volume that was used as the target for a migration operation.
TARGETPP	Identifies the physical partition number of the partition that was used as the target for a migration operation.

2.2.3 Corrective actions

If the -a flag is specified, the **mirscan** command will attempt corrective actions to resolve any issues. There are three possible actions, as follows:

Resync Attempts to resync a stale partition. This is the same process as performed by the **syncvg** command.

Forced Resync Re-reads a partition that is incapable of I/O. This is intended to trigger bad block relocation or hardware relocation in order to recover the partition.

Migration If the partition is still unreadable, the command attempts to migrate that partition to a new location. By default, the new location that is selected adheres to the strictness and upperbound policies for the logical volume that contains the partition.

Partitions on non-mirrored logical volumes are scanned and included in all reports, but no sync or migration operation is possible for such partitions. Partitions on striped logical volumes can be synced but cannot be migrated. Partitions on paging devices cannot be migrated, because this would result in a system hang if the **mirscan** process were to be paged out. Partitions on the boot logical volume cannot be migrated. An informative error message is generated in the corrective action report for each of the preceding cases.

2.3 AIO fast path for concurrent I/O (5300-05)

With previous versions of AIX and AIX 5L, disk drives accessed asynchronously using either the Journaled File System (JFS) or the Enhanced Journaled File System (JFS2) had all I/O routed through the Asynchronous I/O kprocs (kernel processes). Disk drives accessed asynchronously that were using a form of raw logical volume management did not have disk I/O routed through the Asynchronous I/Os kprocs.

AIX 5L Version 5.3 5300-05 has implemented Asynchronous I/O fast path for concurrent I/O. This is similar to the Logical Volume Manager fast path and is meant to be used with JFS2 concurrent I/O. This results in less context switching and immediate start of the I/O leading to a performance improvement.

The Asynchronous I/O fast path for concurrent I/O is yet another I/O optimization. It allows I/O requests to be submitted directly to the disk driver strategy routine through the Logical Volume Manager (LVM). The fast path for concurrent I/O is supported exclusively with the JFS2 file system. Without the fast path, I/O must be queued to Asynchronous I/O kernel processes (kprocs) to handle the requests. In addition, the number of Asynchronous I/O kprocs must be tuned carefully to handle the I/O load generated by the application.

> **Note:** The kproc path can result in slower performance than the fast path due to additional CPU or memory usage and inadequate Asynchronous I/O kproc tuning.

2.4 FAStT boot support enhancements (5300-03)

In previous versions of AIX 5L, there was a requirement for an AIX 5L logical partition to be configured with a path to each working controller in order to allow that partition to boot off a FAStT.

Due to the fact that IBM System p machines can have upto 254 partitions and that more customers are configuring their partitions to boot from external storage devices such as the FAStT, AIX 5L 5300-03 has made enhancements to allow a partition to only have one path configured if the partition is required to boot off the FAStT.

2.4.1 SAN boot procedures

This procedure assumes that the following steps have already been done:

- Hardware installation is complete.
- HBAs are mapped to the correct storage subsystem.
- The size of the boot device that you plan to use is at least 2.2 GB.
- You have ensured that you have the AIX 5L operating system installation CD.

> **Note:** This procedure is an outline of the common installation steps required to install AIX 5L. For step-by-step information refer to your system's installation guide.

1. Ensure that all external devices (for example, storage controllers) are powered on.
2. Power on the server and insert the AIX 5L Volume 1 CD into the optical device.
3. When the system beeps, press 1 or F5 (the function key is system dependent). This will launch the System Management Services menu (SMS) .
4. From the SMS menu, select your installation source (ROM) and boot from the AIX Product CD. (See your server's installation manual for detailed instructions.)
5. When the Welcome to Base Operating System Installation and Maintenance window displays, type `2` in the Choice field to select Change/Show Installation Settings and Install, and press Enter.
6. When the Change Method of Installation window displays, select 1 for New and Complete Overwrite, and press Enter.
7. When the Change Disk(s) window displays, you can change/select the destination disk for the installation. At this point, you can select the appropriate SAN hdisks and deselect any other drives (for example, SCSI).
8. When you have finished selecting the disks and verified that your choices are correct, type `0` in the Choice field, and press Enter. The Installation and Settings window displays with the selected disks listed under System Settings.
9. Make any other changes your OS installation requires and proceed with the installation.

2.5 Tivoli Access Manager pre-install (5300-05)

A simple-to-use, policy-based security system, Tivoli® Access Manager for System p is available as a preinstalled option on selected System p servers. This security system can securely lock down business-critical applications, files, and operating platforms to help prevent unauthorized access. These security capabilities help block both insiders and outsiders from unauthorized access to and use of valuable client, employee, and IBM Business Partner data.

Highlights of Tivoli Access Manager are that it:

- Defends against the top security threat that enterprises face such as malicious or fraudulent behavior by internal users and employees.
- Combines full-fledged intrusion prevention-host-based firewall, application and platform protection, user tracking and controls with robust auditing and compliance checking.
- Provides Persistent Universal Auditing to document compliance with government regulations, corporate policy, and other security mandates.
- Provides best-practice security policy templates to minimize implementation effort and time.
- Delivers mainframe-class security and auditing in a lightweight, easy-to-use product.

Tivoli Access Manager is available at no additional charge on all System p servers sold in the United States. The Tivoli Access Manager server component will be pre-installed by default on all IBM System p5™ 9117-570, 9119-590, and 9119-595 servers. Pre-install of the Tivoli Access Manager server on all other System p servers is available by request (by default it will *not* be installed).

2.6 Geographic Logical Volume Manager (5300-03)

The Geographic Logical Volume Manager (GLVM) is a new AIX 5L software-based technology for real-time geographic data mirroring over standard TCP/IP networks. GLVM can help protect your business from a disaster by mirroring your mission-critical data to a remote disaster recovery site. If a disaster, such as a fire or flood, were to destroy the data at your production site, you would already have an up-to-date copy of the data at your disaster recovery site.

GLVM builds upon the AIX 5L Logical Volume Manager (LVM) to allow you to create a mirror copy of data at a geographically distant location. Because of its tight integration with LVM, users who are already familiar with LVM should find GLVM easy to learn. You configure geographically distant disks as remote physical volumes and then combine those remote physical volumes with local physical volumes to form geographically mirrored volume groups. These are managed by LVM very much like ordinary volume groups.

GLVM was originally made available as part of the Extended Distance (XD) feature of HACMP for AIX 5L Version 5.2. The HACMP documentation refers to this technology as HACMP/XD for GLVM. The AIX 5L GLVM technology provides the same geographic data mirroring functionality as HACMP/XD for GLVM, only without the automated monitoring and recovery that is provided by

HACMP. This technology is intended for users who need real-time geographic data mirroring but do not require HACMP to automatically detect a disaster and move mission-critical applications to the disaster recovery site.

The document *Using the Geographic LVM in AIX* located at the following Web site contains the additional information you need to manage a standalone GLVM installation in AIX 5L without HACMP:

`http://www.ibm.com/servers/aix/whitepapers/aix_glvm.html`

Formal user documentation for GLVM is provided with the HACMP product. The *HACMP/XD for Geographic LVM: Planning and Administration Guide* is available online at the following HACMP documentation page:

`http://www.ibm.com/servers/eserver/pseries/library/hacmp_docs.html`

Reliability, availability, and serviceability

Reliability, Availability, Serviceability (RAS) is a collective term for those characteristics that enable a system to do the following:

- Perform its intended function during a certain period under given conditions.
- Perform its function whenever it is needed.
- Quickly determine the cause and the solution to a problem or error that affects system operation.

In the area of RAS, this chapter covers the following topics:

- Trace enhancements
- Run-Time Error Checking
- Dump enhancements
- Redundant service processors (5300-02)
- Additional RAS capabilities

3.1 Advanced First Failure Data Capture features

AIX 5L Version 5.3 with the TL 5300-03 Technology Level package introduces new First Failure Data Capture (FFDC) capabilities. The set of FFDC features is further expanded in the TL 5300-05 Technology level. These features are described in the sections that follow, and include:

- Lightweight Memory Trace (LMT)
- Run-Time Error Checking (RTEC)
- Component Trace

These features are enabled by default at levels that provide valuable FFDC information with minimal performance impacts. The advanced FFDC features can be individually manipulated, as will be explained below in their individual descriptions. Additionally, a SMIT dialog has been provided as a convenient way to persistently (across reboots) disable or enable the features through a single command. To enable or disable all three advanced FFDC features, enter the following command:

```
smit ffdc
```

> **Note:** You can then choose to enable or disable FFDC features. Note that a **`bosboot`** and reboot are required to fully enable or disable all FFDC features. Any change will not take effect until the next boot.

3.2 Trace enhancements

The following sections discuss the enhancements made to trace.

3.2.1 System Trace enhancements

In prior versions of AIX 5L, system trace traced the entire system. The system trace facility has been enhanced by new flags, which enables the trace to run only for specified processes, threads, or programs.

The system trace can be used to trace processor utilization register (PURR) to provide more accurate event timings in a shared processor partition environment. In previous versions of AIX 5L and AIX, the trace buffer size for a regular user is restricted to a maximum of 1 MB. Version 5.3 allows the system group users to set the trace buffer size either through a new command, **`trcctl`**, or using a new SMIT menu called Manage Trace.

3.2.2 Lightweight memory trace (5300-03)

The Lightweight Memory Trace (also known as LMT) is an efficient, default-on, per CPU, in-memory kernel trace. It is built upon the trace function that already exists in kernel subsystems, and is of most use for those who have AIX 5L source-code access or a deep understanding of AIX 5L internals. The LMT is intended for use by IBM service personnel. Therefore not all of its commands have been documented externally.

Overview

LMT provides system trace information for First Failure Data Capture (FFDC). It is a constant kernel trace mechanism that records software events occurring during system operation. The system activates LMT at initialization, then tracing runs continuously. Recorded events are saved into per processor memory trace buffers. There are two memory trace buffers for each processor—one to record common events, and one to record rare events. The memory trace buffers can be extracted from system dumps accessed on a live system by service personnel. The trace records look like traditional AIX 5L system trace records. The extracted memory trace buffers can be viewed with the **`trcrpt`** command, with formatting as defined in the /etc/trcfmt file.

LMT differs from the traditional AIX 5L system trace in several ways:

- LMT is more efficient.
- LMT is enabled by default, and has been explicitly tuned as an FFDC mechanism. However, a traditional AIX 5L trace will not be collected until requested.

Unlike traditional AIX 5L system trace, you cannot selectively record only certain AIX 5L trace hook IDs with LMT. With LMT, you either record all LMT-enabled hooks or you record none. This means that traditional AIX 5L system trace is the preferred Second Failure Data Capture (SFDC) tool, as you can more precisely specify the exact trace hooks of interest given knowledge gained from the initial failure. All trace hooks can be recorded using traditional AIX 5L system trace, but it may produce a large amount of data this way. Traditional system trace also provides options that allow you to automatically write the trace information to a disk-based file (such as /var/adm/ras/trcfile). LMT provides no such option to automatically write the trace entries to disk when the memory trace buffer fills. When an LMT memory trace buffer fills, it wraps, meaning that the oldest trace record is overwritten, similar to circular mode in traditional AIX 5L trace.

LMT allows you to view some history of what the system was doing prior to reaching the point where a failure is detected. As previously mentioned, each CPU has a memory trace buffer for common events, and a smaller memory trace buffer for rare events. The intent is for the common buffer to have a 1 to 2 second retention (in other words, have enough space to record events occurring during the last 1 to 2 seconds without wrapping). The rare buffer is designed for around one hour's retention. This depends on workload, on where developers place trace hook calls in the AIX 5L kernel source, and on what parameters they trace. AIX 5L Version 5.3 ML3 is tuned such that overly expensive, frequent, or redundant trace hooks are not recorded using LMT. Note that all of the kernel trace hooks are still included in traditional system trace (when it is enabled), so a given trace hook entry may be recorded in LMT, system trace, or both. By default, the LMT-aware trace macros in the source code write into the LMT common buffer, so there is currently little rare buffer content in ML3.

LMT has proven to be a very useful tool during the development of AIX 5L Version 5.3 with the ML 5300-03 and in servicing problems in the field.

Enabling and disabling LMT

LMT by default is turned on but LMT can be disabled (or can be re-enabled) by changing the mtrc_enabled tunable using the **/usr/sbin/raso** command. The **raso** command is documented in the *AIX 5L Version 5.3 Commands Reference, Volume 4* or **man** manual. To turn off (disable) LMT, enter the following:

```
raso -r -o mtrc_enabled=0
```

To turn on (enable) LMT, type the following:

```
raso -r -o mtrc_enabled=1
```

Note: In either case, the boot image must be rebuilt (**bosboot** needs to be run), and the change does not take effect until the next reboot.

When LMT is disabled, the trace memory buffers are not allocated, and most of the LMT-related instruction path-length is also avoided.

LMT performance impact and memory consumption

LMT has been implemented such that it has negligible performance impacts. The impact on the throughput of a kernel-intensive benchmark is just one percent, and is much less for typical user workloads. LMT requires the consumption of a small amount of pinned kernel memory. The default amount of memory required for the memory trace buffers is automatically calculated based on factors that influence software trace record retention, with the target being sufficiently large buffers to meet the retention goals previously described. There are several factors that may reduce the amount of memory automatically used. The behavior differs slightly between the 32-bit (unix_mp) and 64-bit (unix_64) kernels. For the 64-bit kernel, the default calculation is limited such that no more than 1/128th of system memory can be used by LMT, and no more than 256 MB by a single processor. The 32-bit kernel uses the same default memory buffer size calculations, but further restricts the total memory allocated for LMT (all processors combined) to 16 MB. Table 3-1 shows some examples of default LMT memory consumption.

Table 3-1 LMT memory consumption

Machine	Number of CPUs	System memory	Total LMT memory: 64-bit Kernel	Total LMT memory: 32-bit Kernel
POWER3™ (375 MHz CPU)	1	1 GB	8 MB	8 MB
POWER3 (375 MHz CPU)	2	4 GB	16 MB	16 MB
POWER5™ (1656 MHz CPU, shared processor LPAR, 60% ent cap, simultaneous muilti-threading)	8 logical	16 GB	120 MB	16 MB
POWER5 (1656 MHz CPU)	16	64 GB	512 MB	16 MB

To determine the amount of memory being used by LMT, enter the following shell command:

```
echo mtrc | kdb | grep mt_total_memory
```

The following example output is from a IBM System p5 machine with four logical CPUs, 1 GB memory, and 64-bit kernel (the result may vary on your system):

```
# echo mtrc | kdb | grep mt_total_memory
mt_total_memory... 00000000007F8000
```

The preceding output shows that the LMT is using 8160 KB (that is in hex 0x7F8000 bytes) memory.

The 64-bit kernel resizes the LMT trace buffers in response to dynamic reconfiguration events (for both POWER4 and POWER5 systems). The 32-bit kernel does not. It will continue to use the buffer sizes calculated during system initialization. Note that for either kernel, in the rare case that there is insufficient pinned memory to allocate an LMT buffer when a CPU is being added, the CPU allocation will fail. This can be identified by a CPU_ALLOC_ABORTED entry in the error log, with detailed data showing an Abort Cause of 0000 0008 (LMT) and Abort Data of 0000 0000 0000 000C (ENOMEM).

For the 64-bit kernel, the **/usr/sbin/raso** command can also be used to increase or decrease the memory trace buffer sizes. This is done by changing the mtrc_commonbufsize and mtrc_rarebufsize tunable variables. These two variables are dynamic parameters, which means that they can be changed without requiring a reboot. For example, to change the per CPU rare buffer size to sixteen 4 KB pages, for this boot as well as future boots, you would enter:

```
raso -p -o mtrc_rarebufsize=16
```

For more information about the memory trace buffer size tunables, see the **raso** command documentation.

Internally, LMT tracing is temporarily suspended during any 64-bit kernel buffer resize operation.

For the 32-bit kernel, the options are limited to accepting the default (automatically calculated) buffer sizes, or disabling LMT (to completely avoid buffer allocation).

Using LMT

This section describes various commands available to make use of the information captured by LMT. LMT is designed to be used by IBM service personnel, so these commands (or their new LMT-related parameters) may not be documented in the external documentation in InfoCenter. Each command can display a usage string if you enter **command -?**.

To use LMT, use the following steps:

1. Extract the LMT data from a system dump or a running system.
2. Format the contents of the extracted files to readable files.
3. Analyze the output files and find the problem.

Get LMT records from a system dump

The LMT memory trace buffers are included in a system dump. You manipulate them similarly to how traditional system trace buffers are used. The most basic method is to use the **trcdead** command to extract the LMT buffers from the dump. The **trcdead** command can be used to extract the eight active system trace channels, all component trace buffers, and the LMT buffers from the system dump. System trace channel 0 is extracted when no flags are provided. A system trace channel other than channel 0 is identified through a -channelnum flag. LMT buffers are identified through the -M flag. Only one type of trace buffer, or one specific system trace channel, can be extracted at one time. If you use the

-M parameter on the **trcdead** command, it will extract the buffers into files in the LMT log directory. By default this is /var/adm/ras/mtrcdir. For example, to extract LMT buffers from a dump image called dumpfile, you would enter:

```
trcdead -M dumpfile
```

Each buffer is extracted into a unique file, with a control file for each buffer type. It is similar to the per CPU trace option of traditional system trace. As an example, executing the previous command on a dump of a two-processor system would result in the creation of the following files:

```
# ls /var/adm/ras/mtrcdir
mtrccommon      mtrccommon-1   mtrcrare-0
mtrccommon-0    mtrcrare       mtrcrare-1
```

To extract Lightweight Memory Trace information from dump image vmcore.0 and put it into the /tmp directory, enter:

```
trcdead -o /tmp -M vmcore.0
```

The new -M parameter of the **trcrpt** command can then be used to format the contents of the extracted files. The **trcrpt** command allows you to look at the common files together, or the rare files together, but will not display a totally merged view of both sets. All LMT trace record entries are time-stamped, so it is straight forward to merge files when desired.

As described in 3.2.4, “Trace event macro stamping (5300-05)” on page 50, the **trcrpt** command is enhanced in TL5 to support merging of the various trace event sources. This includes supporting a -M all option. Also remember that in the initial version of AIX 5L Version 5.3 ML3, rare buffer entries are truly rare, and most often the interesting data will be in the common buffers. Continuing the previous example, to view the LMT files that were extracted from the dumpfile, you could enter:

```
trcrpt -M common
```

and

```
trcrpt -M rare
```

Other **trcrpt** command parameters can be used in conjunction with the -M flag to qualify the displayed contents. As one example, you could use the following command to display only VMM trace event group hookids that occurred on CPU 1:

```
trcrpt -D vmm -C 1 -M common
```

The **trcrpt** command is the most flexible way to view LMT trace records. However, it is also possible to use the **kdb** dump reader and KDB debugger to view LMT trace records. This is done via the new mtrace subcommand. Without any parameters, the subcommand displays global information relating to LMT. The -c parameter is used to show LMT information for a given CPU, and can be combined with the common or rare keyword to display the common or rare buffer contents for a given CPU. The -d flag is the other flag supported by the mtrace subcommand. This option takes additional subparameters that define a memory region using its address and length. The -d option formats this memory region as a sequence of LMT trace record entries. One potential use of this option is to view the LMT entries described in the dmp_minimal cdt of a system dump.

> **Note:** Any LMT buffer displayed from kdb/KDB contains only generic formatting, unlike the output provided by the **trcrpt** command. The kdb/KDB subcommand is a more primitive debug aid. It is documented in the external KDB documentation for those wishing additional details. As a final comment regarding kdb and LMT, the mtrace subcommand is not fully supported when the kdb command is used to examine a running system. In particular, buffer contents will not be displayed when the **kdb** command is used in this live kernel mode.

Get LMT records from a running system

The final option for accessing LMT trace records is to extract them on a running system. The new **mtrcsave** command is used to extract the trace memory buffers into disk files in the LMT log directory. Recording of new LMT entries is temporarily suspended while the trace buffers are being extracted. The extracted files look identical to the files created when LMT buffers are extracted from a system dump with the **trcdead** command. And as with LMT files created by the **trcdead** command, the **trcrpt** command is used to view them.

Without any parameters, the **mtrcsave** command will extract both common and rare buffers for every CPU to the LMT log directory. The -M flag can be used to specify a specific buffer type, common or rare. The -C flag can be used to specify a specific CPU or a list of CPUs. CPUs in a CPU list are separated by commas, or the list can be enclosed in double quotation marks and separated by commas or blanks. The following example shows the syntax for extracting the common buffer only, for only the first two CPUs of a system. CPU numbering starts with zero. By default, the extracted files are placed in /var/adm/ras/mtrcdir:

```
mtrcsave -M common -C 0,1
ls /var/adm/ras/mtrcdir
mtrccommon    mtrccommon-0   mtrccommon-1
```

The **snap** command can be used to collect any LMT trace files created by the **mtrcsave** command. This is done using the **gettrc** snap script, which supports collecting LMT trace files from either the default LMT log directory or from an explicitly named directory. The files are stored into the

/tmp/ibmsupt/gettrc/<logdirname> subdirectory. Using the **snap** command to collect LMT trace files is only necessary when someone has explicitly created LMT trace files and wants to send them to service. If the machine has crashed, the LMT trace information is still embedded in the dump image, and all that is needed is for snap to collect the dump file. You can see the options supported by the **gettrc** snapscript by executing:

```
/usr/lib/ras/snapscripts/gettrc -h
```

As an example, to collect general system information, as well as any LMT trace files in the default LMT log directory, you would enter:

```
snap -g "gettrc -m"
```

The preceding discussions of the **trcdead**, **trcrpt**, **mtrcsave**, and **snap** commands mention the LMT log directory. The **trcdead** and **mtrcsave** commands create files in the LMT log directory, the **trcrpt** command looks in the LMT log directory for LMT trace files to format, and the **gettrc** snap script may look in the LMT log directory for LMT trace files to collect. By default, the LMT log directory is /var/adm/ras/mtrcdir. This can be changed to a different directory using the **trcctl** command. For example, to set the LMT log directory to a directory associated with a dump being analyzed, you might enter:

```
trcctl -M /mypath_to_dump/my_lmt_logdir
```

This sets the system-wide default LMT log directory to /mypath_to_dump/my_lmt_logdir, and subsequent invocations of **trcdead**, **trcrpt**, **mtrcsave**, and the **gettrc** snapscript will access the my_lmt_logdir directory. This single system-wide log directory may cause issues on multi-user machines where simultaneous analysis of different dumps is occurring. So beginning with AIX 5L Version 5.3 TL5, the **mtrcsave** command also supports a -d <*directory*> flag to override the global default directory when extracting LMT buffers.

LMT support introduced with AIX 5L Version 5.3 with ML 5300-03 represents a significant advance in AIX first failure data capture capabilities, and provides service personnel with a powerful and valuable tool for diagnosing problems.

3.2.3 Component trace facility (5300-05)

The component trace facility allows the capture of information about a specific kernel component, kernel extension, or device driver. It is new in AIX 5L Version 5.3 with TL 5300-05.

Component trace facility overview

Component trace (CT) provides two high-level capabilities:

- It can be used as a filter for existing system trace, as a component hierarchy can now be associated with trace.
- It provides a uniform framework to manage and retrieve First Failure Data Capture (FFDC) and Second Failure Data Capture (SFDC) information that may currently be traced according to component-specific methods.

Component trace is an important FFDC and SFDC tool available to the kernel, kernel extensions, and device drivers. CT allows a component to capture trace events to aid in both debugging and system analysis and provide focused trace data on larger server systems.

The component trace facility provides system trace information for specific system components. This information allows service personnel to access component state information through either in-memory trace buffers or through traditional AIX 5L system trace. CT is enabled by default.

Component trace uses mechanisms similar to system trace. Existing TRCHKxx and TRCGEN macros can be replaced with CT macros to trace into system trace buffers or memory trace mode private buffers. Once recorded, CT events can be retrieved using the `ctctrl` command. Extraction using the `ctctrl` command is relevant only to in-memory tracing. CT events can also be present in a system trace. The `trcrpt` command is used in both cases to process the events.

Component trace modes

Component trace has two modes that can be used simultaneously:

- The system trace mode sends trace entries to the existing system trace.

  There are two properties associated with this mode:

  on/off By default, this mode is on.

  level of trace The default level of system trace is CT_LVL_NORMAL, for example 3.

- The memory trace mode stores the trace entries in a memory buffer, either private to the component or to a per-CPU memory buffer dedicated to the kernel's lightweight memory tracing.

  The following settings may be changed:

  on/off By default, this mode is off.

  level of trace The default level is CT_LVL_MINIMAL for example 1.

  private buffer By default, the size is 0.

Component trace entries may be traced to a private component buffer, the lightweight memory trace, or the system trace. The destination is governed by flags specified to the CT_HOOK and CT_GEN macros. The MT_COMMON flag causes the entry to be traced into the common, lightweight memory trace buffer, and MT_RARE causes it to go to the rare, lightweight memory trace buffer. You should not specify both MT_COMMON and MT_RARE. MT_PRIV traces the entry into the component's private buffer. MT_SYSTEM puts the entry into the system trace if system trace is active. Thus, an entry may be traced into the lightweight memory trace, the component's private buffer, and the system trace, or any combination of these destinations. Generic trace entries, traced with the CT_GEN macro, cannot be traced into the lightweight memory trace.

In the memory trace mode, you have the choice for each component, at initialization, to store their trace entries either in a component private buffer or in one of the memory buffers managed by the lightweight memory trace. In the second case, the memory type (common or rare) is chosen for each trace entry. The component private buffer is a pinned memory buffer that can be allocated by the framework at component registration or at a later time and only attached to this component. Its size can be dynamically changed by the developer (through the CT API) or the administrator (with the **ctctrl** command).

Private buffers and lightweight memory buffers will be used in circular mode, meaning that once the buffer is full, the last trace entries overwrite the first one.

Moreover, for each component, the serialization of the buffers can be managed either by the component (for example, managed by the component owner) or by the component trace framework. This serialization policy is chosen at registration and may not be changed during the life of the component.

The system trace mode is an additional functionality provided by component trace. When a component is traced using system trace, each trace entry is sent to the current system trace. In this mode, component trace will act as a front-end filter for the existing system trace. By setting the system trace level, a component can control which trace hooks enter the system trace buffer. Tracing into the system trace buffer, if it is active, is on at the CT_LVL_NORMAL tracing level by default.

Using the component trace facility

CT commands can be found in the **ctctrl** command documentation.

The following are examples of using the **ctctrl** command to manipulate the CT:

- To dump the contents of all CT buffers to the default directory (/var/adm/ras/trc_ct), enter:

```
ctctrl -D -c all
```

- To dump the contents of the mbuf CT buffer to /tmp, enter:

```
ctctrl -D -d /tmp -c mbuf
```

After that, a file named mbuf is created in /tmp with the content of the mbuf component trace. To view the trace record, enter:

```
trcrpt /tmp/mbuf
```

- To query the state of all CT aware components, enter:

```
ctctrl -q
```

- To query the state of only the netinet components, enter:

```
ctctrl -c netinet -q -r
```

- To disable the use of in-memory CT buffers persistently across reboots by using:

```
ctctrl -P memtraceoff
```

- CT can be persistently enabled by running:

```
ctctrl -P memtraceon
```

> **Note:** The **bosboot** command is required to make the memory trace enablement of disablement persistent on the next boot.

3.2.4 Trace event macro stamping (5300-05)

The trace facility is useful for observing a running device driver and system. The trace facility captures a sequential flow of time-stamped system events, providing a fine level of detail on system activity. Events are shown in time sequence and in the context of other events. The trace facility is useful in expanding the trace event information to understand who, when, how, and even why the event happened. For AIX 5L Version 5.3, tracing can be limited to a specified set of processes or threads. This can greatly reduce the amount of data generated and allow you to target the trace to report on specific tasks of interest.

The operating system is shipped with predefined trace hooks (events). You need only to activate trace to capture the flow of events from the operating system. You can also define trace events in your code during development for tuning purposes. This provides insight into how the program is interacting with the system.

A trace event can take several forms. An event consists of the following:

- Hookword
- Data words (optional)

- A TID, or thread identifier
- Timestamp

Macros to record each possible type of event record are defined in the /usr/include/sys/trcmacros.h file. The following macros should always be used to generate trace data. Do not call the tracing functions directly. There is a macro to record each possible type of event record. The macros are defined in the sys/trcmacros.h header file. Most event IDs are defined in the sys/trchkid.h header file. Include these two header files in any program that is recording trace events.

The macros to record system (channel 0) events with a time stamp are:

```
TRCHKL0T (hw)
TRCHKL1T (hw,D1)
TRCHKL2T (hw,D1,D2)
TRCHKL3T (hw,D1,D2,D3)
TRCHKL4T (hw,D1,D2,D3,D4)
TRCHKL5T (hw,D1,D2,D3,D4,D5)
```

Similarly, to record non-time stamped system events (channel 0) on versions prior to AIX 5L Version 5.3 with the 5300-05 Technology Level, use the following macros:

```
TRCHKL0 (hw)
TRCHKL1 (hw,D1)
TRCHKL2 (hw,D1,D2)
TRCHKL3 (hw,D1,D2,D3)
TRCHKL4 (hw,D1,D2,D3,D4)
TRCHKL5 (hw,D1,D2,D3,D4,D5)
```

In AIX 5L Version 5.3 with the 5300-05 Technology Level and later, a time stamp is recorded with each event regardless of the type of macro used. The time stamp is useful in merging system trace events with events in LMT and component trace buffers.

There are only two macros to record events to one of the generic channels (channels 1–7). They are:

```
TRCGEN (ch,hw,d1,len,buf)
TRCGENT (ch,hw,d1,len,buf)
```

These macros record a hookword (hw), a data word (d1), and a length of data (len) specified in bytes from the user's data segment at the location specified (buf) to the event stream specified by the channel (ch). In AIX 5L Version 5.3 with the 5300-05 Technology Level and later, the time stamp is recorded with both macros.

3.3 Run-Time Error Checking

The Run-Time Error Checking (RTEC) facility provides service personnel with a method to manipulate debug capabilities that are already built into product binaries. RTEC provides service personnel with powerful first failure data capture and second failure data capture error detection features. The basic RTEC framework is introduced in TL 5300-03, and extended with additional features in TL 5300-05.

RTEC features include the Consistency Checker and Xmalloc Debug features already described in 1.3, “Consistency checkers (5300-03)” on page 10, and 1.5, “xmalloc debug enhancement (5300-05)” on page 11. Features are generally tunable with the **errctrl** command. Some features also have attributes or commands specific to a given subsystem, such as the **sodebug** command associated with new socket debugging capabilities. The enhanced socket debugging facilities are described in the AIX publications. Some of the other RTEC features are expected to hold the most value for internal service personnel, and therefore have received less external documentation. The sections that follow describe some of the capabilities.

3.3.1 Detection of Excessive Interrupt Disablement

AIX 5L Version 5.3 ML3 introduces a new feature that can detect a period of excessive interrupt disablement on a CPU, and create an error log record to report it. This allows you to know if privileged code running on a system is unduly (and silently) impacting performance. It also helps to identify and improve such offending code paths before the problems manifest in ways that have proven very difficult to diagnose in the past.

Functional description

Use a kernel profiling approach to detect disabled code that runs for too long. The basic idea is to take advantage of the regularly scheduled clock ticks that generally occur every 10 milliseconds, using them to approximately measure continuously disabled stretches of CPU time individually on each logical processor in the configuration.

> **Note:** This is a statistical sampling approach, so resolution is limited to avoid excessive false positives.

This approach will alert you to partially disabled code sequences by logging one or more hits within the offending code. It will alert you to fully disabled code sequences by logging the i_enable that terminates them. In the special case of timer request block (trb) callouts, the possible detection is triggered by controlling the disablement state within the clock routine, which invokes registered trb handlers in succession.

Note: See the tstart kernel service for more information.

The primary detail data logged is a stack trace for the interrupted context. This will reveal one point from the offending code path and the call sequence that got you there. For example, a heavy user of bzero will be easily identified even though bzero may have received the interrupt. Due to the sampling implementation, it is likely that the same excessively and partially disabled code will log a different IAR and traceback each time it is detected. Judgement is required to determine whether two such detection's are representative of the same underlying problem.

To get function names displayed in the traceback, it is necessary that the subject code be built with the proper traceback tables. AIX kernel code and extensions are compiled with -q tbtable=full to ensure this.

The most recent Lightweight Memory Trace (LMT) entries have been copied to the error log record.

Example error log

To see how this actually works and what is reported, a deliberate disablement was coded as a kernel extension, dtest_ext. Its busy loop looks like this:

```
int dtest_ext()
{
...
    /* partially disabled loop */
    ipri = i_disable(INTOFFL0);
    looper(looptime);
    i_enable(ipri);
...
}

void looper(int uSec)
...
      while( 1 ) {
       curtime(&ctime);
          if( ntimercmp(ctime, etime, >) )
                break;
      }
}
```

This loops until the current time, returned by curtime, passes a predetermined end time, set for this example to cause the loop to run for 550 ms. In particular, the culprit code is looper, which calls curtime repeatedly in its busy loop.

Here is what you will see in the error log summary after running this program:

```
# errpt
IDENTIFIER TIMESTAMP  T C RESOURCE_NAME   DESCRIPTION
A2205861   0920142405 P S SYSPROC         Excessive interrupt
disablement time
The detail looks like this:

LABEL:             DELAYED_INTS
IDENTIFIER:        A2205861

Date/Time:          Tue Sep 20 14:24:22 2005
Sequence Number:    102
Machine Id:         0001288F4C00
Node Id:            victim64
Class:              S
Type:               PERF
Resource Name:      SYSPROC

Description
Excessive interrupt disablement time

Probable Causes
SOFTWARE PROGRAM

Failure Causes
SOFTWARE PROGRAM

        Recommended Actions
        REPORT DETAILED DATA

Detail Data

TICKS 51 (3) decr, IAR CD204 convert_tb+170, LR CD270 curtime+24
convert_tb+170
curtime+20
looper+E8
dtest_ext+A8
[37F4]
--unknown--
```

```
LMT
0000 0000 0000 0001 0000 0000 0000 B017 0000 004B 7535 DE29 8000
0028 10C0 0000
<snip>
0000 0000 0000 0000 0000 0000 0008 B02D 0000 0085 1DA1 F27E 0000
0028 4AF0 0000
```

When dtest_ext was run and called looper, requesting a 55 tick busy loop, it became eligible for detection after running disabled for more than 500 ms. The detector consequently reacted when the busy loop exceeded the default 50 tick threshold and created an error log record. There are three significant parts to this record:

TICKS 51 (3) This tells you that excessive disablement was detected on the 51st timer tick, and that the minimum count of total tick interrupts of 3 was reached, revealing that the culprit was only partially disabled.

convert_tb+170 This is the start of the traceback, and tells you that the convert_tb function was called by curtime, which was called by looper, which was called by dtest_ext. In this case, it is obvious that looper was the culprit, and that curtime and its subroutine convert_tb were innocent. In general, this may not always be so obvious.

LMT recent entries These can be analyzed to gain more information about the path leading up to the detection. The most recent traced event is last in this area, and will always be the 4AF hookid, which is that of disablement detection itself.

In the case where the detected code has been running fully disabled, the ticks information will be slightly different:

- The TICKS value will report the (tick-accurate) actual length of the disabled interval, since it could not be detected until it was over.
- The (count) value will be 1.
- The top of the traceback chain will likely be i_enable.

When the excessive disablement is detected in a trb handler, an extra line of detail data precedes the traceback to identify the responsible routine. When a trb routine has run fully disabled, this is the only way to identify it. For example:

```
Detail Data

TICKS 55 (1) decr, IAR 9474 i_enable+174, LR 73A88 clock+1D0
trb_called.func 3DA65DC trb_looper
```

```
i_enable+174
clock+1CC
i_softmod+2BC
[DF64C]
```

In this case, the busy loop was planted in a trb callout function, trb_looper. Because the excession could not be detected until after the trb routine returned to clock, and clock enabled, there is no direct evidence of it in the traceback, which merely shows clock enabling for interrupts.

Controlling disablement detection

The only externally documented interface to this function is through the standard RAS framework. You can turn error checking persistently off at the system level with:

```
errctrl -P errcheckoff
```

Or persistently on with:

```
errctrl -P errcheckon
```

The disablement detector registers as a child of the proc component, with the name proc.disa, which can be specified as the component name in place of all. To affect just the disablement detector, for example:

```
errctrl errcheckoff -c proc.disa
```

This level of control is not fully supported, and is not persistent over system reboot. More granular internal-use-only controls are also present, as described below.

Detection threshold

You can use the error-checking level of the RAS framework to imply the threshold on the number of ticks that must be accumulated to trigger an error report. These values are:

```
disabled n/a
minimal 50 ticks or 1/2 second
normal 10 ticks or .1 second
detail 4 ticks or .04 second
```

Since the default error-checking level of a debug-off kernel is minimal, the default disablement detection threshold is 50 ticks. One way to override this would be errctrl errchecknormal -c proc.disa, which would change the threshold to 10 ticks. Any of the errchecknormal, errcheckminimal, or errcheckdetail subcommands can be specified to change the threshold, as indicated in the table above.

Note: A debug-on kernel has a default error-checking level of detail, and consequently a default disablement detection threshold of only 4.

A private errctrl subcommand can also be used to set the threshold as an alternative to using the values implied by the error-checking level.

```
errctrl threshold=n -c proc.disa
```

Valid thresholds are in the range of 3–62 clock ticks (63 is used to indicate that detection is suspended, as described below). Results may not be useful if the threshold is set too low, so values less than 3 are rejected. Setting the threshold to 0 will disable checking. The actual threshold will be determined by whichever of the checking level or the explicit threshold was last set. Higher threshold values will detect disablement excession more accurately, while lower threshold values will detect more potential offenders.

Note: Ticks are not an absolute measure of CPU time or CPU cycles. They are a measure of elapsed time. For this reason, detection is disabled by default in shared LPAR partitions.

Error disposition

The framework can also communicate an error disposition for each error severity. In this case, there is only one error being detected, whose severity will be considered MEDIUM_SEVERITY. Only the following error dispositions are supported, when set for medium severity with the medsevdisposition subcommand of errctrl:

- ERROR_IGNORE - do not log the error, medsevdisposition=48
- ERROR_LOG - log the error, medsevdisposition=64
- ERROR_SYSTEM_DUMP - abend, medsevdisposition=112

Other dispositions are not applicable here.

If an event is ignored, either because of the error disposition (here), or the maxlog limit (see below), the event will still be recorded using LMT. Only the IAR, LR, and current trb callout address will be traced.

Limiting error logging

Another private subcommand errctrl -c proc.disa maxlog=n will set a strict limit on the number of errors you will log, PER BOOT. The default will be 1, meaning that, over the life of a system, only one excessive disablement entry will be logged. If more are desired, this private command must be used after each system boot to allow them.

All events will be logged in the normal LMT buffer. This implies that all events are also logged to the system trace, when enabled. Only the binary IAR, LR, and trb callout addresses will be traced.

Exemption

In special cases it may be impractical to immediately rewrite an algorithm that contains excessive disablement. For this reason, two kernel services are exported to kernel extensions:

```
long disablement_checking_suspend(void)
```

A call to this service will temporarily disable the detection of excessive disablement, just for the duration of the current critical section. This call should be inserted at the beginning of the exempt critical section, immediately after it disables if this is base-level code, or as soon as possible within interrupt handling code. The temporary exemption automatically lapses either when the program re-enables to INTBASE, or when the interrupt handling completes.

To cancel the exemption explicitly, perhaps because the exempt code is one of potentially many interrupt level callout routines, you will also have:

```
void disablement_checking_resume(long)
```

This resume function is called after re-enabling at the end of the critical section, not within the critical section. This is necessary because, in the case of an INTMAX critical section, the tick counting will have been deferred by the disablement until the moment of enablement. You want to still be in suspend mode at this instant.

The suspend function returns the previous suspension state to the caller. This is later passed to the paired resume function, which will restore that state. This enables nesting of exempted critical sections. As an example, to specifically exempt just the looper function of the previous example:

```
int ipri, dc;

ipri = i_disable(INTOFFL0);
dc = disablement_checking_suspend();
looper(looptime);
i_enable(ipri);
disablement_checking_resume(dc);
```

3.3.2 Kernel Stack Overflow Detection (5300-05)

Beginning with the AIX 5L 5300-05 technology level package, the kernel provides enhanced logic to detect stack overflows. All running AIX 5L code maintains an area of memory called a stack, which is used to store data necessary for the execution of the code. As the code runs, this stack grows and shrinks. It is possible for a stack to grow beyond its maximum size and overwrite other data. These problems can be difficult to service. AIX 5L TL5 introduces an asynchronous run-time error checking capability to examine if certain kernel stacks have over-flowed. The default action upon overflow detection is to log an entry in the AIX 5L error log. The stack overflow run-time error checking feature is controlled by the ml.stack_overflow component in the RAS component hierarchy. This will typically only need to be manipulated based on the advice of IBM service personnel.

3.3.3 Kernel No-Execute Protection (5300-05)

Beginning with the AIX 5L 5300-05 technology level package, no-execute protection is set for various kernel data areas that should never be treated as executable code. This exploits the same page-level execution enable/disable hardware feature described previously in 1.6, “Stack execution disable protection (5300-03)” on page 12. The benefit is immediate detection if erroneous device driver or kernel code inadvertently make a stray branch onto one of these pages. Previously the behavior would likely lead to a crash, but was undefined. This enhancement improves kernel reliability and serviceability by catching attempts to execute invalid addresses immediately, before they have a chance to cause further damage or create a difficult-to-debug secondary failure. This feature is largely transparent to the user, since most of the data areas being protected should clearly be non-executable. The two general kernel-mode heaps are the exception. Two new boolean tunables for the raso command, kern_heap_noexec and mbuf_heap_noexec, will enable no-execute protection for the kernel heap and netmalloc heaps, respectively. They will default to 0, leaving both heaps unprotected, to preserve binary compatibility in case there are kernel extensions currently in the field that are making use of the ability to execute code in these heaps. This is not expected to be common, but the conservative approach was taken as the default.

3.4 Dump enhancements

A system generates a system dump when a severe error occurs. System dumps can also be initiated by users with root user authority. A system dump creates a picture of your system's memory contents. System administrators and programmers can generate a dump and analyze its contents when debugging new applications. System dumps are an important component part of the AIX service strategy, and AIX 5L Version 5.3 contains a number of enhancements to this subsystem.

3.4.1 Minidump facility (5300-03)

A system dump is not always completed successfully for various reasons. If a dump is not collected at crash time, it is often difficult to determine the cause of the crash. To combat this, the minidump facility has been introduced in AIX 5L Version 5.3 with ML5300-03.

The minidump is a small, compressed dump. The minidump stores level 0 crash information into NVRAM, and places it in the error log on reboot with a label of MINIDUMP_LOG and a description of COMPRESSED MINIMAL DUMP. The minidump is taken in addition to any full system dump when a system crashes. The minidump is visible in the error log after operating system reboot, and is included in the failure information sent to IBM service for diagnosis. It is targeted at service personnel.

The benefits of the minidump include:

- First Failure Data Capture (FFDC) even when there is a system dump failure. The minidump records key information at the time point the system crashes, such as stack trace.
- A history of the past dump (including content) in the error log, in case the dump device gets overwritten.
- Aiding in identifying duplications of problems.
- Easier to manage than a full system dump. It is stored in the errlog and has a very small size. It can also be transferred much more easily than a full system dump.

Users can help to provide minidump information to IBM service personnel by:

```
snap -gc
```

Or the raw error log file that is usually located at /var/adm/ras/errlog.

3.4.2 Parallel dump (5300-05)

An AIX 5L system generates a system dump when a severe error occurs. System dumps can also be user-initiated by users with root user authority. A system dump creates a picture of your system's memory contents.

However, systems have an increasingly large amount of memory and CPUs. Larger systems experience longer dump times. Thus a new feature parallel dump is introduced in AIX 5L Version 5.3 with TL 5300-05. This is a dump performance enhancement, and the speed of generating a dump is improved on systems with multiple processors.

As a by-product of parallel dump, a new compressed dump format is introduced. A new -S flag is introduced to the **sysdumpdev** command that allows you to determine whether a given dump device contains a valid compressed dump:

```
sysdumpdev -L -S Device
```

The dump must be from an AIX 5L release with parallel dump support. This flag can be used only with the -L flag.

3.4.3 The dmpuncompress command (5300-05)

You can specify that all future dumps will be compressed before they are written to the dump device by using:

```
sysdumpdev -C
```

A main change for AIX 5L Version 5.3 with TL 5300-05 of dump format is that the compressed dump will not use the **compress** command format, so it cannot be extracted by the **uncompress** command. A new dump compression method is introduced, and the copied dump has a name with a suffix of .BZ instead of a .Z. So a new **dmpuncompress** command is added to extract the new-format compressed dump file. The syntax is as follows:

```
/usr/bin/dmpuncompress [ -f ] [ File ]
```

The **dmpuncompress** command restores original dump files.

Each compressed file specified by the file parameter is removed and replaced by an expanded copy. The expanded file has the same name as the compressed version, but without the .BZ extension. If the user has root authority, the expanded file retains the same owner, group, modes, and modification time as the original file. If the user does not have root authority, the file retains the same modes and modification time, but acquires a new owner and group.

The -f File flag forces expansion. It will overwrite the file if it already exists. The system does not prompt the user that an existing file will be overwritten. File size might not actually shrink in cases where data compression is already high.

The following is an example to uncompress the dump.BZ file:

```
/usr/lib/ras/dmpuncompress dump.BZ
```

The dump.BZ file is uncompressed and renamed dump.

3.4.4 Other system dump enhancements

In AIX 5L Version 5.3, system dump compression is turned on by default. For information about dump compression, see the **sysdumpdev** command documentation.

System dump is enhanced to support DVD-RAM as the dump media. A DVD-RAM can be used as the primary or secondary dump device. The **snap** command can use a DVD-RAM as a source as well as an output device.

Extended system failure status information is captured as part of the dump, detailing dump success or failure. Display the extended information by using the **sysdumpdev** command.

Following a system crash, there exist scenarios where a system dump might crash or fail without one byte of data written out to the dump device. For cases where a failed dump does not include the dump minimal table, the failures cannot be easily diagnosed. As an enhancement to the dump procedure, a small minidump is now taken when the system crashes. The minidump stores level 0 crash information into NVRAM, and places it in the error log on reboot. The **sysdumpdev -vL** command can then be used to discover the reason for the failure. This information is also included in the failure information sent to IBM service for diagnosis.

Dump information is displayed on the TTY during the creation of the system dump.

The **dump** command can now take a wildcard (5300-05).

A new option -c has been added to the **dmpfmt** command to verify the integrity of a dump.

3.5 Redundant service processors (5300-02)

The service processor enables POWER™ Hypervisor and Hardware Management Console surveillance, selected remote power control, environmental monitoring, reset and boot features, and remote maintenance and diagnostic activities, including console mirroring. On systems without an HMC, the service processor can place calls to report surveillance failures with the POWER Hypervisor™, critical environmental faults, and critical processing faults.

AIX 5L has been enhanced to support redundant service processors.

The service processor provides the following services:

- Environmental monitoring
- Mutual surveillance with the POWER Hypervisor
- Self protection for the system to restart from an unrecoverable firmware error, firmware hang, hardware failure, or environmentally induced failure
- Fault monitoring and operating system notification at system boot

All IBM System p5 9119-590 and p5 9119-595 servers are shipped with dual service processors, and it is an option on a System p5 9117-570 and 9116-561. The purpose of dual flexible service processors is to provide automatic failover support in the event of a failure.

3.5.1 Redundant service processor requirements

The following are the major requirements to support redundant service processors:

- Firmware levels
  - 9117/9406 570: 01SF235_160_160
  - 9117/9406 570 with FC 8338 (2.2 GHz P5+ processors) or FC 7782 (1.9 GHz POWER5+™ processors): 01SF240_201_201
  - 9116 561: 01SF240_201_201
- One HMC at the following levels:
  - 9117/9406 570: Version 5, Release 1 with PTF MH000607
  - 9116 561: Version 5, Release 2 with PTF MH000610
- Two Physical Processor Drawer (CEC) Enclosures

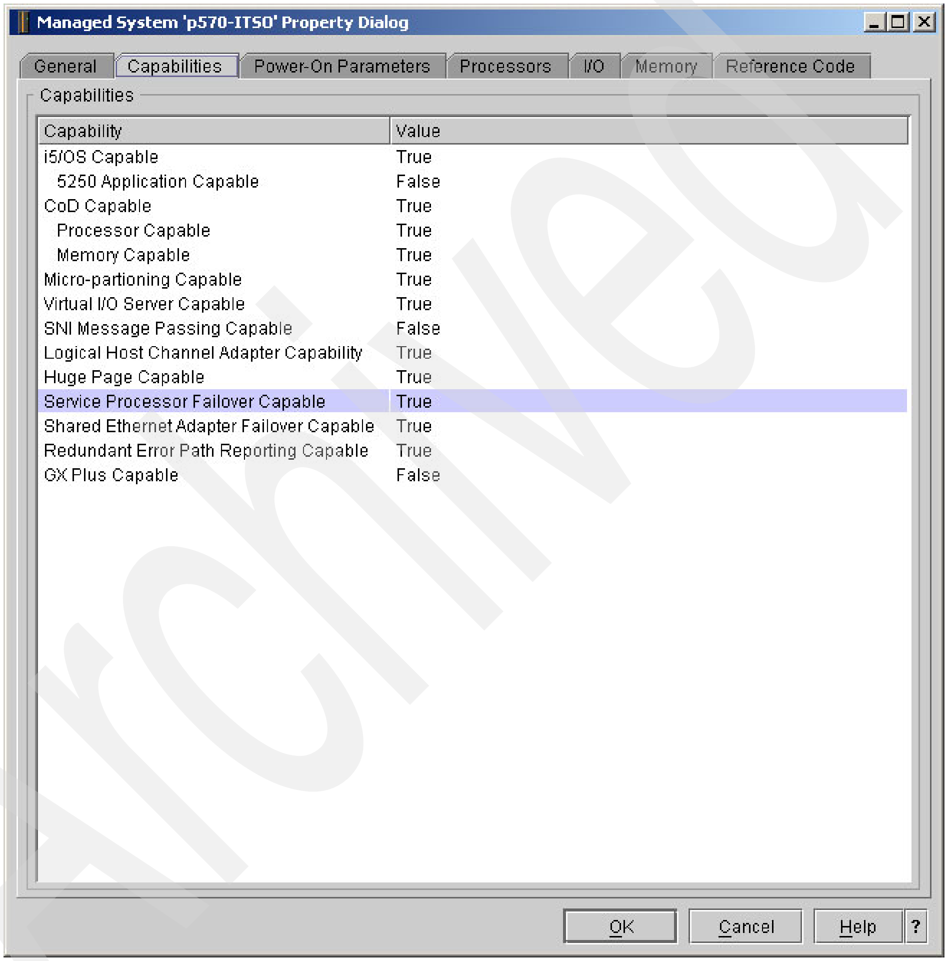
> **Note:** Firmware and HMC levels listed are strictly minimums. We recommend installation of the latest available service packs for your code stream, as they contain fixes and enhancements that will provide more robust performance and improve reliability.

3.5.2 Service processor failover capable

If you have two service processors installed in your system and you want to confirm that service processor failover is enabled, follow these steps:

1. In the content area right-click the managed system.

2. Select **Properties**.
3. Select **Capabilities**.
4. Service Processor Failover Capable should be set to True. (When True, the managed system can switch from using a primary service processor to using a secondary service processor automatically if the primary service processor fails). See Figure 3-1 for a view of an enabled system.



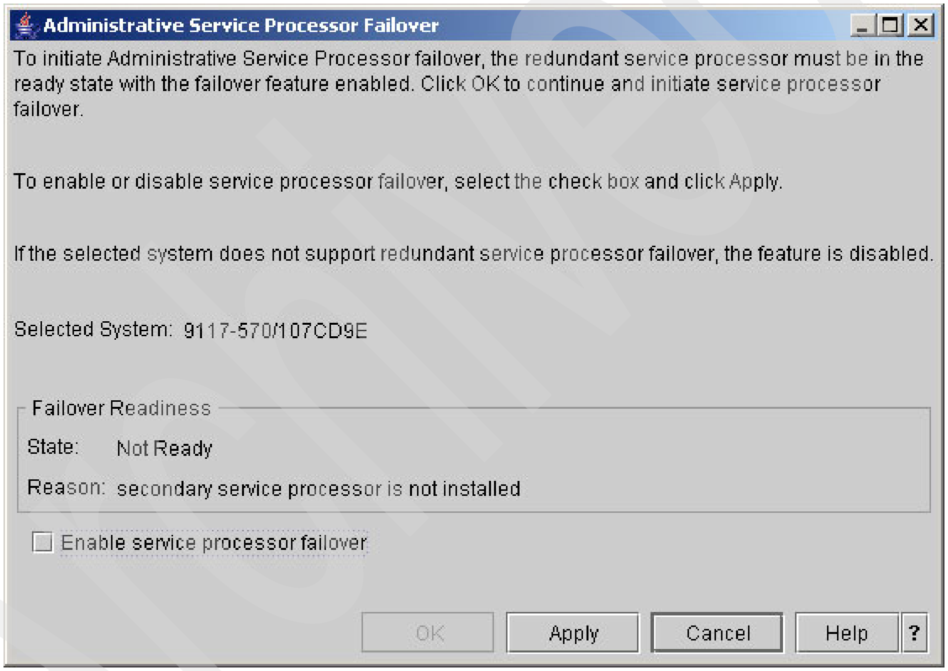
Capability	Value
i5/OS Capable	True
5250 Application Capable	False
CoD Capable	True
Processor Capable	True
Memory Capable	True
Micro-partioning Capable	True
Virtual I/O Server Capable	True
SNI Message Passing Capable	False
Logical Host Channel Adapter Capability	True
Huge Page Capable	True
Service Processor Failover Capable	True
Shared Ethernet Adapter Failover Capable	True
Redundant Error Path Reporting Capable	True
GX Plus Capable	False

Figure 3-1 Managed System Properties

3.5.3 Performing a service processor failover

Regarding Figure 3-2 on page 65, you can both enable or disable failover and execute an administrative failover. The Apply button is used to save changes when a failover is enabled or disabled. If a failover is enabled, setting OK will initiate a failover. As shown in Figure 3-2, the secondary service processor is not installed, nor has it been deconfigured. Therefore, performing a failover would not be possible. If a second service processor was installed, the failover readiness state needs to be set to Ready in order for the failover to occur.

1. In the navigation area expand **Service Applications**.
2. In the contents area, select **Service Utilities**.
3. On the next window, select the managed system you want to work with.
4. From the menu, select **Selected**.
5. Select **Service Processor Failover**.



To initiate Administrative Service Processor failover, the redundant service processor must be in the ready state with the failover feature enabled. Click OK to continue and initiate service processor failover.

To enable or disable service processor failover, select the check box and click Apply.

If the selected system does not support redundant service processor failover, the feature is disabled.

Selected System: 9117-570/107CD9E

Failover Readiness

State: Not Ready

Reason: secondary service processor is not installed

☐ Enable service processor failover

Figure 3-2 Administrator Service Processor Failover

Information regarding configuration of a redundant service processor can be found at:

http://www.redbooks.ibm.com/abstracts/sg247196.html?Open

3.6 Additional RAS capabilities

The following sections highlight recent changes to some of the additional AIX 5L RAS utilities and infrastructure.

3.6.1 Error log hardening

An error log may become corrupted or incomplete when a system is terminated without stopping error logging. The current recovery strategy is to make a copy of the log and then reset the log as though it were a new log, rather than attempt to recover the existing log entries. AIX 5L Version 5.3 introduces a recovery method wherein the log is recovered when the errdemon is started. It checks for the error log consistency. If the errdemon detects a corrupted error log, it makes a backup copy of the existing error log file to /tmp/errlog.save and then repairs the existing log.

AIX 5L error logging now supports up to 4096 bytes of event data (see the /usr/include/sys/err_rec.h file). However, this size of error log entry is intended only for restricted system use, and general error log entries should continue to contain 2048 bytes or less of event data. While up to 4096 bytes of detail data is allowed, this size entry may be truncated across a reboot in certain circumstances. The largest detail data size guaranteed not to be truncated is 2048 bytes. A large error log entry reduces the non-volatile storage available to the system dump facility in the event of a system crash.

3.6.2 The snap command enhancements

The function of the **snap** command has been enhanced so that it can now split the snap output file into user-specified sizes (smaller). To do this, the **snap** command invokes a the **snapsplit** command.

The **snap** command is enhanced to support the following:

- Independent service vendors (ISVs) can use custom scripts to collect their custom problem data as part of the snap process. For programming and process details, see “Copying a System Dump” in *AIX 5L Version 5.3 Kernel Extensions and Device Support Programming Concepts.*
- Large outputs can be split into smaller files for ease of transport.

Output can be written to DVD-RAM media.

3.6.3 Configuring a large number of devices

For each device configured in the system, an entry is made in the /dev directory. On systems with many devices, it is possible for the system to run out of space in the root file system or to run out of inodes. Prior versions of AIX 5L did not report the cause of errors. In AIX 5L Version 5.3, the **cfgmgr** command reports the cause.

3.6.4 Core file creation and compression

AIX 5L Version 5.3 allows the users to compress the core file and specify its name and destination directory. Two new commands, **lscore** and **chcore**, have been introduced to check the settings for the corefile creation and change them, respectively.

4

System administration

In this chapter the following major topics are discussed:

- AIX 5L release support strategy (5300-04)
- Command enhancements
- Multiple page size support (5300-04)
- Advanced Accounting
- National language support
- LDAP enhancements (5300-03)

4.1 AIX 5L release support strategy (5300-04)

AIX 5L has changed some of its current service strategy directions and instituted new release rules. One of the reasons for this is the amount of change present in maintenance levels and the frequency with which they are released. Technology levels, service packs, and concluding service packs are the new concepts that have been introduced.

4.1.1 Technology level (TL)

A technology level will contain new hardware and software features in addition to service updates. The first technology level will be restricted to hardware features and enablement, as well as software service. The second technology level will include new hardware features and enablement, software service, and new software features, making it the larger of the two yearly releases. A technology level will have all of its requisites added so that the whole technology level is installed, and will not allow for it to be partially installed.

```
oslevel -r
5300-03
```

4.1.2 Service pack (SP)

The Service Pack concept will allow service-only updates (known as PTFs) that are released between technology levels to be grouped together for easier identification. These fixes will be highly pervasive, critical, or security-related issues. Service packs will be provided for the N and N-1 releases (for example, Version 5.3 and 5.2). The **oslevel** command has a new -s flag, which applies all flags to service packs.

```
oslevel -s
5300-03-01
```

4.1.3 Concluding service pack (CSP)

Concluding service pack is the term that will identify the last service pack on a technology level. The concluding service pack will contain fixes for highly pervasive, critical, or security-related issues, just like a service pack, but it may also contain fixes from the newly released technology level that fall into these categories. Therefore, a concluding service pack will contain a very small subset of service that was just released as a part of a new technology level.

```
oslevel -s
5300-03-CSP
```

4.1.4 Interim fix (IF)

The term *interim fix* or i-fix is used in AIX 5L as a replacement for emergency fix or interim fix in order to simplify terminology across IBM and not cause confusion when dealing with other products. While the term *emergency fix* is still applicable in some situations (a fix given under extreme conditions), the term interim fix is more descriptive in that it implies a temporary state until an update can be applied that has been through the normal distribution process.

4.1.5 Maintenance strategy models

The current strategy is for two AIX 5L updates per year. In the first half of the year a technology level will be released, and in the second half of the year the next technology level will be released. Roughly 4–8 weeks after a technology level has been released, a concluding service pack will be released for the previous technology level. For example:

- 1H / 2006 - TL4 5300-04 Roughly 4-8 weeks Concluding Service Pack 4
- 2H / 2006 - TL5 5300-05 Roughly 4-8 weeks Concluding Service Pack 5

Between technology levels, one or more service packs and PTFs will be released in support of the current technology level.

4.2 Command enhancements

The following topics are covered in this section:

- The id command enhancement (5300-03)
- cron and at command mail enhancement (5300-03)
- The more command search highlighting (5300-03)
- The ps command enhancement (5300-05)
- Commands to show locked users (5300-03)
- The -l flag for the cp and mv commands (5300-01)
- For information about performance command enhancements see 4.3.1, “Performance command enhancements” on page 83

4.2.1 The id command enhancement (5300-03)

The **id** command has been enhanced by the addition of the -l flag. This flag specifies that the **id** command write the login ID instead of the real or effective ID. Previously, this information could only be retrieved by examining kernel structures using the **kdb** command. It can be invoked with either the -u flag to write the login UID or the -g flag to write the primary group ID for the login user. When username is passed with the -l option, the **id** command displays the ID details of the user name instead of the login ID details.

For example, if the user andyy logged into the system and then used the **su** command to switch to the root user, **id -un** would report the following:

```
# id -un
root
```

With the -l flag you would see the following output:

```
# id -unl
andyy
```

If a username is specified, the output is reported for that user rather than the login user:

```
# id -unl root
root
```

4.2.2 cron and at command mail enhancement (5300-03)

Prior to AIX 5L Version 5.3 with ML 5300-03 release, the cron daemon sends a mail to the user who submitted the cron job, once it is complete. This mail is intended to update the users on the status of the job execution. The cron daemon also sends a mail if the output of the **cron** job is not redirected to stdout or stderr. This mail would contain the output of the executed cron job or error messages in case the job failed. For sending mail to the users, the cron daemon uses the **mail** command. But the mail sent by the cron daemon does not contain a subject line.

From AIX 5L Version 5.3 with ML 5300-03, mail from the cron daemon will have a subject line, and two different cron mail formats are introduced. One is for internal cron errors and the other is for the completion of cron jobs.

Mail format for mails resulting due to cron internal errors (such as errors encountered while processing crontab file entries) is shown in Example 4-1.

Example 4-1 Mail format for cron internal errors

```
Subject:
Cron Job Failure

Content:
Cron Environment:
 SHELL= < Shell Name>
 PATH= <PATH string>
 CRONDIR = <cron directory name>
 ATDIR = < atjobs directory name>

Output from cron as follows:
Brief description on the error encountered
```

Mail format for mail on cron jobs is shown in Example 4-2.

Example 4-2 Mail format for cron jobs completion

```
Subject:
Output from “<at | cron>” job <jobname>, username@hostname, exit status
<Exit Code>

Content:
Cron Environment:
 SHELL= < Shell Name>
 PATH= <PATH string>
 CRONDIR = <cron directory name>
 ATDIR = < at jobs directory name>

Your “<cron | at>” job executed on <machine name> on <scheduled time>
[“cron” | “at” job name]
produced the following output:
<Output from the Job or any error messages reported>
```

Example 4-3 shows an example of mail sent by cron on cron job completion.

Example 4-3 A cron job completion message

```
Message  1:
From daemon Mon Dec  4 14:26:00 2006
Date: Mon, 4 Dec 2006 14:26:00 -0600
From: daemon
To: root
Subject: Output from cron job /usr/bin/sleep,
root@lpar01.itsc.austin.ibm.com, exit status 2

Cron Environment:
 SHELL =

PATH=/usr/bin:/etc:/usr/sbin:/usr/ucb:/usr/bin/X11:/sbin:/usr/java14/jr
e/bin:/usr/java14/bin:/usr/local/bin
 CRONDIR=/var/spool/cron/crontabs
 ATDIR=/var/spool/cron/atjobs
 LOGNAME=root
 HOME=/root

Your "cron" job executed on lpar01.itsc.austin.ibm.com on Mon Dec  4
14:26:00 CST 2006
/usr/bin/sleep

produced the following output:

Usage: sleep Seconds

*****************************************************************
        cron: The previous message is the standard output
        and standard error of one of the cron commands.
```

This example shows the execution of the **cron** command with an exit status 2 (return code), and the standard output contains the information of:

```
Usage: sleep Seconds
```

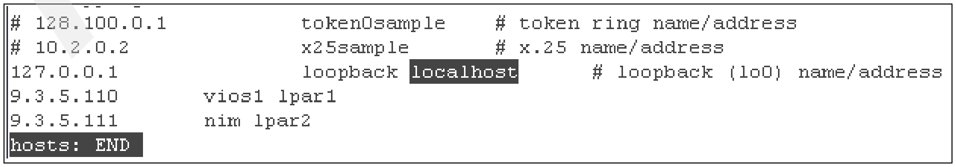
The improved mail facility contains more useful information than the older version does. A similar enhancement has been made to the **at** command.

4.2.3 The more command search highlighting (5300-03)

The **more** command now supports search highlighting. This is the default behavior. When matching a search pattern, all matches of the search pattern are highlighted. There are two ways to disable it:

- The **more** command -H option to disable the default search highlighting feature from the command line.
- H can also be used as a subcommand in an active session to toggle highlighting on or off.

For example, use the **more** command to open the /etc/hosts, and use a slash (/) to search the pattern *localhost*, and you will get all the words *localhost* highlighted. Figure 4-1 provides an example of this.



```
# 128.100.0.1           token0sample    # token ring name/address
# 10.2.0.2              x25sample       # x.25 name/address
127.0.0.1               loopback localhost      # loopback (lo0) name/address
9.3.5.110       vios1 lpar1
9.3.5.111       nim lpar2
hosts: END
```

Figure 4-1 Search highlighting sample of the more command

In the session shown, you can just type H (case-sensitive) to disable highlighting.

4.2.4 The ps command enhancement (5300-05)

The **ps** command has been enhanced in AIX 5L Version 5.3 5300-05. The command is used to show the current status of processes. Now it also provides process hierarchy information and a listing of descendant processes for given PIDs. AIX 5L introduces three new options for the **ps** command, provided in Table 4-1.

Table 4-1 Flags of ps command

Flag	Purpose
-Z	Displays the page size settings of processes using three columns: DPGSZ indicates the data page size of a process. SPGSZ indicates the stack page size of a process. TPGSZ indicates the text page size of a process.
-L pidlist	Generates a list of descendants of each and every PID that has been passed to it in the pidlist variable. The pidlist variable is a list of comma-separated process IDs. The list of descendants from all of the given PID is printed in the order in which they appear in the process table.
-T pid	Displays the process hierarchy rooted at a given PID in a tree format using ASCII. This flag can be used in combination with the -f, -F, -o, and -l flags.

Example 4-4 shows the **ps** command flags to find all the descendants of the inetd process.

Example 4-4 The ps command new flags

```
# ps -ef|grep inetd
    root 180324 188532   0 08:37:15      -  0:00 /usr/sbin/inetd
# ps -L 180324
    PID    TTY  TIME CMD
 135340      -  0:00 telnetd
 180324      -  0:00 inetd
 209020      -  0:00 telnetd
 254084  pts/2  0:00 ksh
 286890      -  0:00 bootpd
 307372  pts/0  0:00 ksh
 311474      -  0:00 telnetd
 319694  pts/0  0:00 topas
 323780      -  0:02 xmtopas
 364790  pts/2  0:00 ps
```

You can find all of he processes including their hierarchy in an ASCII tree format by entering:

```
ps -T 0
```

The -T option is used to find all of the processes and sub-processes under a specific user by providing the user’s Telnet IP address.

1. Find out the pts number of the user by giving the host name or the IP address:

```
# who
root        pts/0       Nov 30 08:41     (kcyk04t.itsc.austin.ibm.com)
root        pts/1       Nov 30 08:44     (proxy)
root        pts/2       Nov 30 08:50     (kcbd0na.itsc.austin.ibm.com)
```

2. Find the shell for this user:

```
# ps -ef|grep "pts/2"
    root 254084 250022   0 08:50:49  pts/2  0:00 -ksh
```

3. Use ps -T options:

```
# ps -fT 254084
```

The -Z option of the **ps** command is added to support different page sizes. For more information about Huge Page support and Large Page support in AIX 5L Version 5.3, refer to 4.3, “Multiple page size support (5300-04)” on page 79.

4.2.5 Commands to show locked users (5300-03)

In previous versions of AIX 5L and AIX there was no unique command available to show a list of locked users. If a system administrator needed to collect a list of locked users, they would have to write a shell script using various commands such as **usrck**, **pwdck**, **lssec**, and so on, then parse the output and access the relevant security files needed to obtain the information.

AIX 5L Version 5.3 ML 5300-03 has new command-line options for the **usrck** command to display locked users.

The usrck command syntax

The following example shows the command syntax:

```
usrck [ -l [-b] | -p | -y | -n | -t ] [ user ... | ALL ]
```

The two most common new flags are described in table Table 4-2. Additional information is available in the command man page.

Table 4-2 Commonly used usrck command flags and their descriptions

Flag	Description
-b	Reports users who are not able to access the system and the reasons, with the reasons displayed in a bit-mask format. The -l flag must be specified if the -b flag is specified. Note: The bit mask does not report criteria 10 (user denied login to terminal), since this cannot be considered a complete scenario when determining whether a system is inaccessible to a user. Likewise, the bit mask does not report criteria 9 (user denied access by applications) if at least one but not all of the attributes values deny authentication. This criteria is only reported when all four attribute values deny authentication.
-l	Scans all users or the users specified by the user parameter to determine whether the users can access the system.

Command examples

Example 4-5 and Example 4-6 on page 78 would be used by an administrator to scan all users or the users specified by the user parameter to determine whether a user can access the system, and also give a description as to why the user account has no access.

Example 4-5 The usrck -l command

```
# usrck -l ALL
The system is inaccessible to daemon, due to the following:
        User account is expired.
        User has no password.
The system is inaccessible to bin, due to the following:
        User account is expired.
        User has no password.
The system is inaccessible to sys, due to the following:
        User account is expired.
        User has no password.
The system is inaccessible to adm, due to the following:
        User has no password.
The system is inaccessible to uucp, due to the following:
        User has no password.
        User denied access by login, rlogin applications.
The system is inaccessible to guest, due to the following:
        User has no password.
The system is inaccessible to nobody, due to the following:
        User account is expired.
        User has no password.
```

Example 4-6 The usrck -l user command

```
# usrck -l test
The system is inaccessible to test, due to the following:
        User account is locked.
```

The next example uses both the -b and -l options. The output consists of two fields, the user name and a 16-digit bit mask, separated by a tab. This output is a summary of the previous command and lists all of the user accounts that have been locked out.

Example 4-7 The usrck -l -b command

```
# usrck -l -b ALL
daemon                  0000000000001010
bin                     0000000000001010
sys                     0000000000001010
adm                     0000000000001000
uucp                    0000000000001000
guest                   0000000000001000
nobody                  0000000000001010
lpd                     0000000000001010
lp                      0000000001001000
invscout                0000000001001000
snapp                   0000000001001000
ipsec                   0000000001001000
nuucp                   0000000001001000
sshd                    0000000001001001
test                    0000000000000001
```

4.2.6 The -I flag for the cp and mv commands (5300-01)

A new **-I** flag is available for the **`cp`** and **`mv`** commands. The **-I** flag suppresses the warning message during ACL conversion. When ACL conversion succeeds with the **`cp -p`** or **`mv`** commands, a warning message is printed out to stderr. You can use the **-I** flag to suppress this warning message.

The cp command syntax

The **cp** command syntax is as follows:

- To copy a file to another file, type the following:

```
cp [ -E{force|ignore|warn} ] [ -f ] [ -h ] [ -i ] [ -p ] [ -I ]
[ -U ] [ - ] SourceFile TargetFile
```

- To copy a file to a directory, type the following:

```
cp [ -E{force|ignore|warn} ] [ -f ] [ -h ] [ -i ] [ -p ]
[[ -r | -R ] [ -H | -L | -P ]] [ -I ] [ -U ]
[ - ] SourceFile ... TargetDirectory
```

- To copy a directory to a directory, type the following:

```
cp [ -E{force|ignore|warn} ] [ -f ] [ -h ] [ -i ] [ -p ]
{ -r | -R } [ -H | -L | -P ] [ -I ] [ -U ]
[ - ] SourceDirectory ... TargetDirectory
```

The mv command syntax

The **mv** command syntax is as follows:

- To move files to a directory, maintaining the original file names, type the following:

```
mv [ -E{force|ignore|warn} ] [ -i | -f ] [ -I ] SourceFile
TargetFile
```

- To move and rename a file or directory, type the following:

```
mv -E{force|ignore|warn} ] [ -i | -f ] [ -I ] SourceFile ...
TargetDirectory
```

4.3 Multiple page size support (5300-04)

AIX 5L 5.3 Version 5300-04 includes 64-bit kernel support for virtual memory page sizes of 64 KB and 16 GB that are supported by POWER5+ processors. These virtual memory page sizes are supported in addition to the previously supported virtual memory page sizes of 4 KB and 16 MB. While 16 GB pages are intended to only be used in very high-performance environments, 64 KB pages are general-purpose, and most workloads are likely to see a benefit by using 64 KB pages rather than 4 KB pages.

Using a larger virtual memory page size for an application's memory can significantly improve an application's performance and throughput due to hardware efficiencies associated with larger page sizes. Using a larger page size can decrease the hardware latency of translating a virtual page address to a physical page address. This decrease in latency is due to improving the efficiency of hardware translation caches such as a processor's translation lookaside buffer (TLB). Because a hardware translation cache only has a limited number of entries, using larger page sizes increases the amount of virtual memory that can be translated by each entry in the cache. This increases the amount of memory that can be accessed by an application without incurring hardware translation delays.

The specific page sizes supported on a system depend on the processor type. Table 4-3 shows the page sizes supported on different hardware platforms.

Table 4-3 Page size support by platform

Page size	Required hardware	Requires user configuration	Restricted	Kernel
4 KB	All	No	No	32 and 64
64 KB	IBM POWER5+ or later	No	No	64 only
16 MB	IBM POWER4 or later	Yes	Yes	64 only
16 GB	IBM POWER5+ or later	Yes	Yes	64 only

You can use the **pagesize -af** command to display all of the virtual memory page sizes supported by AIX 5L on a system. Example 4-8 provides a sample output.

Example 4-8 Output of the pagesize -af command

```
# pagesize -af
4K
64K
16M
16G
```

You can specify the page sizes to use for three regions of a process's address space using an environment variable or settings in an application's XCOFF binary with the **ldedit** or **ld** commands, as shown in Table 4-4.

Table 4-4 The ld and ldedit command arguments for page size specification

Region	ld/ledit option	LDR_CNTRL environment variable	Description
Data	-bdatapsize	DATAPSIZE	Initialized data, bss, heap
Stack	-bstackpsize	STACKPSIZE	Initial thread stack

Region	ld/ledit option	LDR_CNTRL environment variable	Description
Text	-btextpsize	TEXTPSIZE	Main executable text

For example, the following command causes a.out to use 64 KB pages for its data, 4 KB pages for its text, and 64 KB pages for its stack on supported hardware:

```
LDR_CNTRL=DATAPSIZE=64K@TEXTPSIZE=4K@STACKPSIZE=64K a.out
```

Unless page sizes are selected using one of the previous mechanisms, a process will continue to use 4 KB pages for all three process memory regions by default.

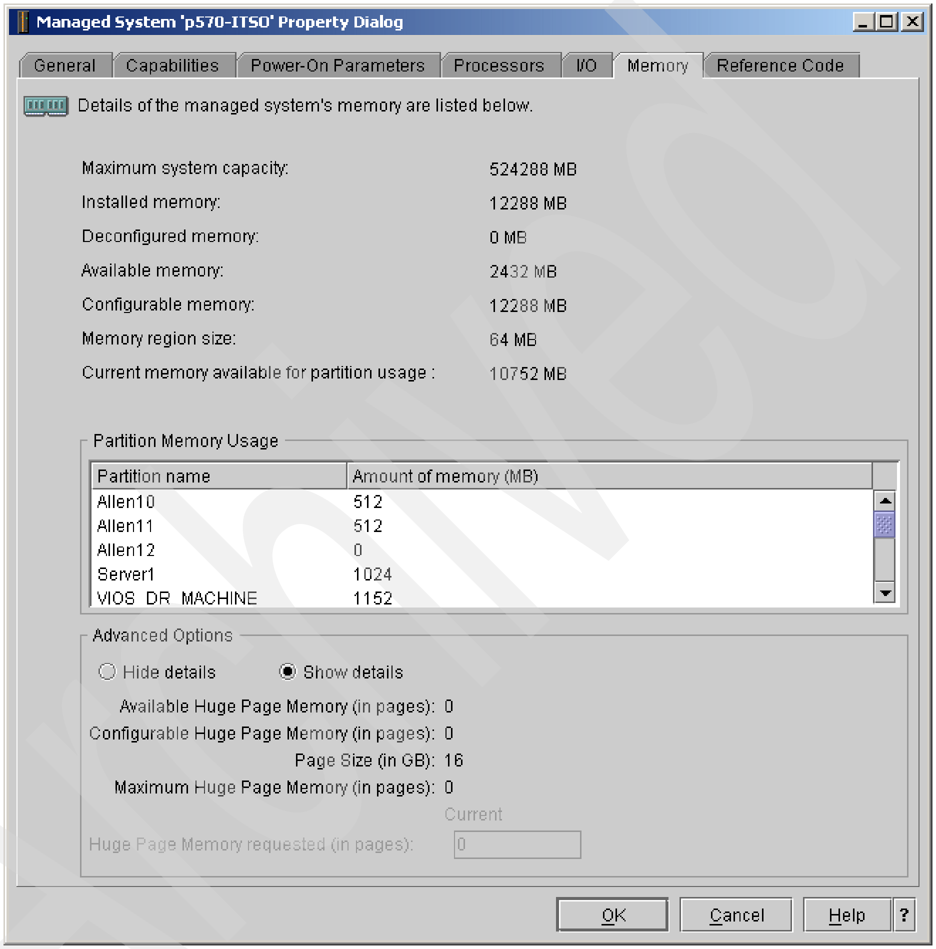
The 4 KB and 64 KB page sizes are intended to be general purpose, and no configuration changes are necessary to enable a system to use these page sizes. The 16 MB large page size and 16 GB huge page size are intended only to be used in very high-performance environments, and an administrator must configure a system to use these page sizes. Furthermore, the support for 16 MB large pages and 16 GB huge pages is limited. 16 MB large pages are only supported for process data and shared memory, and 16 GB huge pages are only supported for shared memory.

To enable 16 MB page support, the **vmo** command can be used as follows:

```
# vmo -p -o lgpg_regions=32 -o lgpg_size=16777216
```

This will configure thirty-two 16 MB pages, giving 512 MB in total. The operation can be performed dynamically provided that the system is capable of performing dynamic LPAR operations. Operations to change the number of large pages on the system may succeed partially. If a request to increase or decrease the size of the pool cannot fully succeed (for example, if lgpg_regions is tuned to 0 but there are large pages in use by applications), the **vmo** command will add or remove pages to get as close as possible to the requested number of pages.

16 GB huge pages must be configured using a system's Hardware Management Console (HMC). Under the Managed System's Properties menu, a system administrator can configure the number of 16 GB huge pages on a system by selecting **Show Details** in the Advanced Options field of the Memory tab. Figure 4-2 shows this option.



Details of the managed system's memory are listed below.

Maximum system capacity:	524288 MB
Installed memory:	12288 MB
Deconfigured memory:	0 MB
Available memory:	2432 MB
Configurable memory:	12288 MB
Memory region size:	64 MB
Current memory available for partition usage :	10752 MB

Partition Memory Usage

Partition name	Amount of memory (MB)
Allen10	512
Allen11	512
Allen12	0
Server1	1024
VIOS_DR_MACHINE	1152

Advanced Options

○ Hide details ● Show details

Available Huge Page Memory (in pages): 0
Configurable Huge Page Memory (in pages): 0
Page Size (in GB): 16
Maximum Huge Page Memory (in pages): 0
Current
Huge Page Memory requested (in pages): 0

Figure 4-2 Configuring huge pages on a managed system using the HMC

Changing the number of 16 GB huge pages on a system requires the entire managed system to be powered off. Once a managed system has been configured with 16 GB huge pages, they can be assigned to partitions by changing a partition's profile.

The **vmo** command can be used to globally disable 64 KB and 16 GB pages using the vmm_mpsize_support option:

```
vmo -r -o vmm_mpsize_support=0
```

This option will take effect after a reboot, and once it is set, only 4 KB and 16 MB pages are available, regardless of the **ld** command options used.

4.3.1 Performance command enhancements

The **ps**, **svmon**, and **vmstat** commands have been enhanced to provide information regarding multiple page size usage.

The ps command

The **ps** command now has an additional flag, -Z, which displays the page sizes being used for the data, stack, and text memory regions of a running process:

```
# ps -Z
    PID    TTY  TIME DPGSZ SPGSZ TPGSZ CMD
  233636  pts/0  0:00    4K    4K    4K ksh
  262346  pts/0  0:00   64K   64K    4K sleep
  278670  pts/0  0:00    4K    4K    4K ps
```

The svmon command

Most **svmon** command options have been enhanced to provide information regarding pagesize. Example 4-9 shows the output from **svmon -G**, while Example 4-10 on page 84 shows the **svmon -P** command.

Example 4-9 The svmon -G command showing multiple page size information

```
# svmon -G
               size       inuse        free         pin     virtual
memory       262144       91197      170947       45147       77716
pg space     131072         669

               work        pers        clnt
pin           45147           0           0
in use        77716           0       13481

PageSize   PoolSize       inuse        pgsp         pin     virtual
s   4 KB          -       72109         669       36235       58628
m  64 KB          -        1193           0         557        1193
```

Example 4-10 The svmon -P command showing multiple page size information

```
Pid Command          Inuse      Pin     Pgsp  Virtual 64-bit Mthrd  16MB
262346 sleep            15963     7556        0    15962      N     N     N

     PageSize      Inuse        Pin       Pgsp    Virtual
     s   4 KB      11307       7540          0      11306
     m  64 KB        291          1          0        291

    Vsid      Esid Type Description             PSize  Inuse   Pin Pgsp
Virtual
       0         0 work kernel segment              s  11306  7540    0 11306
    a0ad         d work shared library text         m    278     0    0   278
   1a32a         f work shared library data         m      8     0    0     8
    1331         2 work process private             m      5     1    0     5
   10320         1 clnt code,/dev/hd2:279           s      1     0    -     -
```

The vmstat command

Two new flags (-p and -P) have been added to the **vmstat** command to provide information about multiple page sizes. Table 4-5 describes these flags, and Example 4-11 shows the outputs.

Table 4-5 The vmstat command new flags and descriptions

Flag	Description
-p	Displays global vmstat information along with a breakdown of statistics per page size
-P	Displays per page size statistics

Example 4-11 The vmstat command output using the -p and -P flags

```
# vmstat -p ALL

System configuration: lcpu=2 mem=1024MB ent=0.10

kthr    memory              page              faults              cpu
----- ----------- ------------------------ ------------ -----------------------
 r  b   avm   fre  re  pi  po  fr   sr  cy  in   sy  cs us sy id wa    pc    ec
 1  1 77757 170904   0   0   0   0    0   0  33   41 101  0  0 99  0  0.00
0.0

  psz   avm   fre  re  pi  po  fr   sr  cy     siz
   4K 58670 67639   0   0   0   0    0   0  139792
  64K  1193  6454   0   0   0   0    0   0    7647

# vmstat -P ALL
```

```
System configuration: mem=1024MB

pgsz            memory                           page
----- -------------------------- -------------------------------------
            siz      avm      fre    re    pi    po    fr     sr    cy
   4K   139792    58671    67637     0     0     0     0      0     0
  64K     7647     1193     6454     0     0     0     0      0     0
```

4.4 Advanced Accounting

There have been three major changes to the Advanced Accounting functionality:

- A new reporting tool has been introduced to enable the processing of accounting information.
- IBM Tivoli Usage and Accounting Manager has been integrated with AIX 5L Advanced Accounting.
- Advanced Accounting has been integrated with LDAP.

These topics are discussed in this section.

4.4.1 Advanced Accounting reporting (5300-03)

The Advanced Accounting subsystem was introduced in AIX 5L Version 5.3. Initially, there were no reporting functions available and it was the responsibility of the user to take the raw data and process it. Now, with the release of AIX 5L 5300-03, new facilities are provided to assist with this. The **acctrpt** command is able to generate reports from the raw accounting data. There are also changes to the libaacct.a library and the SMIT interfaces for Advanced Accounting data analysis statistics reporting.

The acctrpt command

The **acctrpt** command is designed to take data from the accounting file and generate a report. The Advanced Accounting subsystem captures three levels of accounting data: system, process, and transaction. System level accounting data provides global information about configuration, paging activity, remote and local file systems, networks, and disks resource usage. Process-level accounting data provides job-related information such as CPU and virtual memory usage, paging activity, remote and local file system use, and network activity. Transaction-based accounting data provides information about client requests in server processes. Each of these report types are described in the following sections.

Process accounting

For process accounting, users can generate accounting reports by projects, by groups, by users, by commands, or by a combination of these four identifiers. The syntax for this aspect is shown in the following:

```
acctrpt [ -f filename ] [ -F ] [ -U uid ] [ -G gid ] [ -P projID ]
[ -C command ] [ -b begin_time ] [ -e end_time ] [ -p projfile ]
[ -n ]
```

The commonly used flags are described in Table 4-6.

Table 4-6 The acctrpt command flags for process accounting

Flag	Description
-f filename	Specifies the path name of the accounting data file to be used. More than one file can be specified using a comma-separated list. If the -f flag is not specified, the /var/aacct/aacctdata file is used by default.
-U uid	Displays process accounting statistics for the specified UIDs. More than one UID can be specified using a comma-separated list. To display all UIDs, specify -U ALL.
-G gid	Displays process accounting statistics for the specified GIDs. More than one GID can be specified using a comma-separated list. To display all GIDs, specify -G ALL.
-P projID	Displays process accounting statistics for the specified project ID. More than one project ID can be specified using a comma-separated list. To display all projects, specify -P ALL.
-C command	Displays process accounting statistics for the specified command. More than one command name can be specified using a comma-separated list. Only the first 12 characters of the base command name are considered. To display all commands, specify -C ALL.
-b begin_time	Specifies the begin time of an interval. The begin_time parameter is a10-character string in the *MMDDhhmmyy* format, where *MM* is month, *DD* is day, *hh* is hour, *mm* is minute, and *yy* is the last two digits of the year. All characters are numeric. If begin_time is not specified, all encountered records that were written before end_time are considered. If neither end_time nor begin_time is specified, all records are considered.

Flag	Description
-e end_time	Specifies the end time of an interval. The end_time parameter is a 10-character string in the *MMDDhhmmyy* format, where *MM* is month, *DD* is day, *hh* is hour, *mm* is minute, and *yy* is the last two digits of the year. All characters are numeric. If end_time is not specified, all encountered records that were written after begin_time are considered. If neither end_time nor begin_time is specified, all records are considered.
-p projfile	Specifies the project definition file to be used to resolve the projects associated with the transaction records. If -p is not specified, the projects are resolved using the currently loaded projects.

Example 4-12 gives a sample process output, while Example 4-13 shows a report filtered to give information about the **`projctl`** command as run by root.

Example 4-12 The acctrpt command output

```
#acctrpt

Process Accounting Report
-------------------------
                                                                            (C) PELAPSE  TELAPSE  CPU   (sec)
                                                                            (M) VMEM     DMEM     PMEM  (pg)
                                                                            (F) LFILE    DFILE          (MB)
TIMESTAMP  PROJID   UID      GID      PID      CMD          STARTED  EXITED (S) LSOCKET  DSOCKET        (MB)
---------  ------   ---      ---      ---      ---          -------  ------ --- -------  -------  ------
12011107   System   root     system   274574   acctctl      12011107 E       C: 1.0611   1.0611   0.0045
                                                                             M: 76       82       0
                                                                             F: 0.00     0.00
                                                                             S: 0.00     0.00
12011107   System   root     system   209144   ksh          12011107 E       C: 1.1301   1.1301   0.0058
                                                                             M: 183      189      0
                                                                             F: 0.00     0.00
                                                                             S: 0.00     0.00
12011111   System   root     system   274650   smitty       12011111 N       C: 74.4906  74.4906  0.1425
                                                                             M: 61281    62249    0
                                                                             F: 0.02     0.02
                                                                             S: 0.00     0.00
```

Example 4-13 The acctrpt command output filtered for command projctl and user root

```
# acctrpt -U 0 -C projctl

Process Accounting Report
-------------------------
                                               (C) PELAPSE  TELAPSE  CPU      (sec)
                                               (M) VMEM     DMEM     PMEM     (pg)
                                               (F) LFILE    DFILE             (MB)
PROJID   UID      GID      CMD          CNT    (S) LSOCKET  RSOCKET           (MB)
         ---               ---          ---    --- -------- -------- --------
-        root     -        projctl      7      C:  0.0      0.0      0.0
                                               M:  0        0        0
                                               F:  1.9      0.0
                                               S:  0.0      0.0
```

The fields are described in Table 4-7.

Table 4-7 The acctprt command fields for process output

Field	Description
PROJID	Project name (Project ID)
UID	User name (User ID)
GID	Group name (Group ID)
CMD	Base name of the executed command
CNT	Count of transaction records aggregated per row of accounting report
PELAPSE	Total process elapsed time
TELAPSE	Total threads elapsed time
CPU	CPU time (in seconds)
LFILE	Local file I/O (in MBs)
DFILE	Other file I/O (in MBs)
LSOCKET	Local socket I/O (in MBs)
RSOCKET	Other socket I/O (in MBs)
DMEM	Page seconds of disk pages
PMEM	Page seconds of real pages
VMEM	Page seconds of virtual memory

System accounting

For system accounting, reports can be generated that describe the system-level use of resources, such as processors, memory, file systems, disks, and network interfaces. The system accounting interval must be enabled to collect accounting statistics for system resources. This function is often refereed to as LPAR accounting. However, it is not restricted to partitioned environments and can be run on any system running AIX 5l Version 5.3. The syntax for these reports is shown here:

```
acctrpt [ -f filename ] [ -F ] -L resource [ -b begin_time ]
[ -e end_time ]
```

The main flag not described in the process section is shown in Table 4-8.

Table 4-8 The acctrpt command flags for system reporting

Flag	Description	
-L resource	Displays system accounting statistics for the specified resource. The resource parameter must be one of the following values:	
	cpumem	CPU and memory statistics
	filesys	File system statistics
	netif	Network interface statistics
	disk	Disk statistics
	vtarget	VSCSI target statistics
	vclient	VSCSI client statistics
	ALL	All system resource statistic

Example 4-14 shows a sample system report.

Example 4-14 The acctrpt command system output

```
acctrpt  -L  ALL

CPU and Memory Accounting Report
--------------------------------

           (C) IDLE     IOWAIT   SPROC    UPROC    INTR    (sec)
CNT        (U) IO       PGSPIN   PGSPOUT  LGPGUTIL PGRATE
---        --- ----     ------   -------  -------- ------
3          C:  320.8    1.7      0.2      4.9      0.5
           U:  30157    0        0        0.0      0.0


File Systems Accounting Report
------------------------------

CNT    DEVNAME            MOUNTPT    FSTYPE RDWR    OPEN    CREATE  LOCKS   XFERS(MBs)
---    -------            -------    ------ ----    ----    ------  -----   ----------
1      specfs             specfs     16     1339    1194    0       2       26.9
1      pipefs             pipefs     16     3167    0       0       0       0.1
1      /dev/hd10opt       /opt       0      0       0       0       0       0.0
1      /proc              /proc      6      0       0       0       0       0.0
1      /dev/hd1           /home      0      0       0       0       0       0.0
1      /dev/hd3           /tmp       0      414     143     138     0       0.6
1      /dev/hd9var        /var       0      29959   109     4       62      0.4
1      /dev/hd2           /usr       0      12955   9672    0       0       25.9
1      /dev/hd4           /          0      2301    749     45      87      1.4


Network Interfaces Accounting Report
------------------------------------

CNT        NETIFNAME       NUMIO      XFERS(MBs)
---        ---------       -----      ----------
```

```
2          lo0             36         0.001459
2          en0             1344       0.178871

Disks Accounting Report
-----------------------

CNT        DISKNAME        BLKSZ XFERS(MBs) READ       WRITE
---        --------        ----- ---------- ----       -----
2          hdisk0:0        512   2136       47982      254392
2          hdisk0          512   2136       47982      254392
2          vscsi1          0     154815488  446        1690

VSCSI Clients Accounting Report
-------------------------------

CNT        SERVER#   SERVERID UNITID    BYTESIN(MBs) BYTESOUT
---        -------   -------- ------    ------------ --------
2          4         805306408 9367487224930631680 23.428711    124.214844
```

The fields for each section are described in Table 4-9.

Table 4-9 The acctprt command fields for system output

Field	Description
CPU and Memory accounting report	
CNT	Count of transaction records aggregated per row of accounting report
IDLE	CPU idle time (in seconds)
IOWAIT	CPU I/O wait time (in seconds)
SPROC	System process time (in seconds)
UPROC	User process time (in seconds)
INTR	Interrupt time (in seconds)
IO	Number of I/Os
PGSPIN	Number of page swap-ins
PGSPOUT	Number of page swap-outs
LGPGUTIL	Average utilization of large page pool
PGRATE	Average page rate (per second)
File system accounting report	
CNT	Count of transaction records aggregated per row of accounting report
DEVNAME	Device name
MOUNTPT	Mount point name
FSTYPE	File system type

Field	Description
RDWR	Number of reads and writes.
OPEN	Number of file opens
CREATE	Number of file creates
LOCKS	Number of file locks
XFERS	Data transferred (in MBs)
Network interfaces accounting report	
CNT	Count of transaction records aggregated per row of accounting report
NETIFNAME	Network interface name
NUMIO	Number of I/Os
XFERS	Data transferred (in MBs)
Disk accounting report	
CNT	Count of transaction records aggregated per row of accounting report
DISKNAME	Disk name
BLKSZ	Disk block size (in bytes)
XFERS	Number of disk transfers
READ	Number of reads from the disk
WRITE	Number of writes to the disk
VSCSI clients accounting report	
CNT	Count of transaction records aggregated per row of accounting report
SERVER#	Server partition number
SERVERID	Server Unit ID
UNITID	Device logical unit ID
BYTESIN	Data in (in MBs)
BYTESOUT	Data out (in MBs)

Transaction accounting

For transaction accounting, users can generate accounting reports describing application transactions. Transaction reports provide scheduling and accounting information, such as transaction resource usage requirements. These reports use data that is produced by applications that are instrumented with the Application Resource Management (ARM) interfaces to describe the transactional nature of their workloads. Advanced Accounting supports ARM interfaces by recording information that is presented through these interfaces in the accounting data file. The **acctrpt** command can then process these files and report information. The transaction report syntax is as follows:

```
acctrpt [ -f filename ] [ -F ] -T [ -b begin_time ] [ -e end_time ]
```

The -T flag specifies that a transaction report is required. Example 4-15 gives a sample output.

Example 4-15 The acctrpt command transaction report

```
/usr/bin/acctrpt –T –f /var/aacct/acctdata

                     (A)  CLASS           NAME            USER     GROUP    TRANSACTION
PROJID   CNT         (T)  RESPONSE        QUEUED          CPU                             (sec)
------   ---         ---  --------        -------        ----    -------   ------------
System   144         A:  -                WebSphere       -       server1   URI
                     T:  0.00             0.00           0.00
System   32          A:  -                IBM Webserving  -       IBM_SERV  Apache/1.3.28(Unix)
                     T:  0.00             0.00           67.01
```

The fields for the transaction report are shown in Table 4-10.

Table 4-10 The acctprt command fields for transaction output

Field	**Description**
PROJID	Project name (Project ID)
CNT	Count of transaction records aggregated per row of accounting report
CLASS	Account class
GROUP	Application group name
NAME	Application name
TRANSACTION	Transaction name
USER	User name
RESPONSE	Response time (in milliseconds)
QUEUED	Queued time (in milliseconds)
USER	CPU time (in milliseconds)

4.4.2 IBM Tivoli Usage and Accounting Manager integration (5300-05)

IBM Tivoli Usage and Accounting Manager (ITUAM) is an enterprise-wide accounting application that is capable of collecting, analyzing, reporting, and billing based on usage and costs of shared computing resources. With Version 6.1 of ITUAM and AIX 5L 5300-03, Advanced Accounting can be used as a data collector to provide accounting records. For more information about configuring and using these features, see *IBM Tivoli Usage and Accounting Manager Data Collectors for UNIX and Linux User's Guide*.

Additional Advanced Accounting information can be found in 4.6.1, “Advanced Accounting LDAP integration (5300-03)” on page 97.

4.5 National language support

National language support enhancements include the following topics:

- Multi-language software %L in bundles (5300-03)
- geninstall and gencopy enhancements (5300-03)
- Additional national languages support

4.5.1 Multi-language software %L in bundles (5300-03)

Previously, in AIX 5L and AIX, it has been necessary to package separate bundle files for each different locale and then have the system administrator choose between the multiple bundles during installation, such as in the following:

```
dfs.html.en_US
dfs.msg.en_US
dfs.pdf.en_US
dfs.msg.jp_JP
dfs.pdf.jp_JP
dfs.html.jp_JP
dfs.msg.es_ES
dfs.pdf.es_ES
dfs.html.es_ES
.....
..... similar packages for other languages.
.....
dfs.client.rte
dfs.server.rte
```

Now, with AIX 5L Version 5.3 Release 5300-03, support for a %L wildcard in software bundles has been added. This allows the creation of a bundle with file set entries that contain %L, which will resolve at runtime to the applicable locale. This will be replaced with an appropriate language specifier (for example, 'en_US') when installing, copying, or listing software. The **geninstall** command has been modified to handle variable languages in bundle files. The following example shows how the DocServices.bnd is shortened:

```
dfs.msg.%L
dfs.pdf.%L
dfs.html.%L
dfs.client.rte
dfs.server.rte
```

Instead of having to create 14 separate bundle files for 14 separate locales (one for each locale supported), only one set of %L bundles will be used, and it will be correctly resolved at runtime to the applicable locale.

4.5.2 geninstall and gencopy enhancements (5300-03)

Enhancements have been made in AIX 5L 5300-03 to include the %L wildcard option for the **geninstall** and **gencopy** commands.

The **geninstall** and **gencopy** commands are able to process the %L wildcard in a bundle file. This wildcard is replaced at runtime with the value of the appropriate locale environment variable. LC_ALL is checked first, then LC_MESSAGES, and then LANG. The advantage is that you can now create a single bundle file corresponding to multiple installation configurations.

As an example, assume that you provide the ABC product, which requires the abc.rte and abc.com filesets, as well as a message catalog fileset and a documentation fileset. You then provide the message and documentation filesets in English, French, and German, as follows:

```
abc.cat.en_US
abc.cat.fr_FR
abc.cat.de_DE

abc.doc.en_US
abc.doc.fr_FR
abc.doc.de_DE
```

The following bundle file would cause the appropriate combination of filesets to be installed according to the locale variables on the target system:

```
I:abc.rte
I:abc.com
```

```
I:abc.cat.%L
I:abc.doc.%L
```

> Note: If expanding the %L wildcard does not yield a fileset name corresponding to a fileset available on the installation media, then the UTF-8 version of the current locale will be tried, then en_US, and then EN_US.

The following command lists the contents of the media:

```
geninstall -L -d Media [-e LogFile] [ -D ]
```

The output format is the same as the **installp -Lc** command format, with additional fields at the end for ISMP and RPM formatted products.

In Example 4-16, the **geninstall** command is run with the -L option.

Example 4-16 The geninstall -L command output

```
# geninstall -L -d /dev/cd0
Tivoli_Management_Agent.client:Tivoli_Management_Agent.client.rte:3.7.1
.0::I:C::
:::N:Management Framework Endpoint Runtime"::::
bos:bos.rte:5.3.0.50::S:C:::::N:Base Operating System Runtime::::
bos:bos.rte.ILS:5.3.0.50::S:C:::::N:International Language Support::::
bos:bos.rte.SRC:5.3.0.50::S:C:::::N:System Resource Controller::::
bos:bos.rte.aio:5.3.0.50::S:C:::::N:Asynchronous I/O Extension::::
bos:bos.rte.archive:5.3.0.50::S:C:::::b:Archive Commands::::
bos:bos.rte.bind_cmds:5.3.0.50::S:C:::::N:Binder and Loader
Commands::::
bos:bos.rte.boot:5.3.0.50::S:C:::::b:Boot Commands::::
bos:bos.rte.bosinst:5.3.0.50::S:C:::::N:Base OS Install Commands::::
bos:bos.rte.commands:5.3.0.50::S:C:::::b:Commands::::
bos:bos.rte.compare:5.3.0.50::S:C:::::N:File Compare Commands::::
bos:bos.rte.console:5.3.0.50::S:C:::::N:Console::::
bos:bos.rte.control:5.3.0.50::S:C:::::N:System Control Commands::::
bos:bos.rte.cron:5.3.0.50::S:C:::::N:Batch Operations::::
bos:bos.rte.date:5.3.0.50::S:C:::::N:Date Control Commands::::
bos:bos.rte.devices:5.3.0.50::S:C:::::b:Base Device Drivers::::
bos:bos.rte.devices_msg:5.3.0.50::S:C:::::N:Device Driver Messages::::
bos:bos.rte.diag:5.3.0.50::S:C:::::N:Diagnostics::::
```

The following command lists the install packages on the media:

```
gencopy -L -d Media [ -D ]
```

This listing is colon separated and contains the following information:

```
file_name:package_name:fileset:V.R.M.F:type:platform:Description
bos.sysmgt:bos.sysmgt:bos.sysmgt.nim.client:4.3.4.0:I:R:Network
Install Manager - Client Tools
bos.sysmgt:bos.sysmgt:bos.sysmgt.smit:4.3.4.0:I:R:System Management
Interface Tool (SMIT)
```

In Example 4-17, the **gencopy** command is used with the -L option.

Example 4-17 The gencopy -L -d command output

```
# gencopy -L -d /dev/cd0
Tivoli_Management_Agent.client:Tivoli_Management_Agent.client:Tivoli_Management_
Agent.client.rte:3.7.1.0:I:R:Management Framework Endpoint Runtime"
bos:bos:bos.rte:5.3.0.50:O:R:Base Operating System Runtime
bos.rte.edit_5.3.0.50.bff:bos:bos.rte.edit:5.3.0.50:S:R:Editors
bos.rte.diag_5.3.0.50.bff:bos:bos.rte.diag:5.3.0.50:S:R:Diagnostics
bos.rte.devices_msg_5.3.0.50.bff:bos:bos.rte.devices_msg:5.3.0.50:S:R:Device Dri
ver Messages
bos.rte.devices_5.3.0.50.bff:bos:bos.rte.devices:5.3.0.50:S:R:Base Device Driver
s
bos.rte.date_5.3.0.50.bff:bos:bos.rte.date:5.3.0.50:S:R:Date Control Commands
bos.rte.ILS_5.3.0.50.bff:bos:bos.rte.ILS:5.3.0.50:S:R:International Language Sup
port
bos.rte.SRC_5.3.0.50.bff:bos:bos.rte.SRC:5.3.0.50:S:R:System Resource Controller
bos.rte.aio_5.3.0.50.bff:bos:bos.rte.aio:5.3.0.50:S:R:Asynchronous I/O Extension
bos.rte.archive_5.3.0.50.bff:bos:bos.rte.archive:5.3.0.50:S:R:Archive Commands
bos.rte.bind_cmds_5.3.0.50.bff:bos:bos.rte.bind_cmds:5.3.0.50:S:R:Binder and Loa
```

4.5.3 Additional national languages support

In addition to preexisting NLS support in the base AIX 5L Version 5.3, the following national language support is added since that introduction:

Malayalam	UTF-8 codeset (ML_IN) NLS enablement (5300-03)
Kannada	UTF-8 codeset (KN_IN) NLS enablement (5300-03)
Bengali	UTF-8 codeset (BN_IN) NLS enablement (5300-05)
Assamese	UTF-8 codeset (AS_IN) NLS enablement (5300-05)
Punjabi	UTF-8 codeset (PA_IN) NLS enablement (5300-05)
Oriya	UTF-8 codeset (OR_IN) NLS enablement (5300-05)
Estonian	ISO8859-4 codeset (et_EE) NLS enablement
Latvian	ISO8859-4 codeset (lv_LV) NLS enablement

4.6 LDAP enhancements (5300-03)

The following sections describe the LDAP enhancements available on AIX 5L 5300-03.

4.6.1 Advanced Accounting LDAP integration (5300-03)

On prior levels of AIX 5L, the system administrator had to maintain the Advanced Accounting configuration for each system separately. Through this feature, Advanced Accounting is enhanced to support remote management of Advanced Accounting configurations using LDAP. The system administrator can load/upload/download the policies and projects from the LDAP server to other client machines.

The **mkprojldap** command can be used to convert the LDAP client and server to support advanced accounting subsystem data. To support Advanced Accounting on the LDAP server, the LDAP schema for Advanced Accounting must be uploaded to the server. This allows consolidation of accounting from multiple servers to a central common server. AIX 5L LDAP Advanced Accounting is functionally dependent on the secldapclntd daemon.

4.6.2 LDAP client support for Active Directory (5300-03)

Customers now use Microsoft® Active Directory® as an LDAP directory and authentication server for AIX 5L systems.

4.6.3 LDAP ldap.cfg password encryption (5300-03)

AIX 5L Version 5.3 ML3 introduces a password encryption feature. The binddn password can now be stored encrypted in the /etc/security/ldap/ldap.cfg file. This is a security enhancement to protect the password in the client's LDAP configuration file. The LDAP client daemon can read either clear text or the encrypted password to provide backward compatibility.

4.6.4 lsldap: List LDAP records command (5300-03)

A new **lsldap** command has been added to query LDAP entries. The supported queries include users, groups, hosts, automount maps, hosts, protocols, networks, netgroup, RPC, services, and aliases. The **lsldap** command uses the secldapclntd daemon, which must be running for the command to function. By default, the **lsldap** command displays on the distinguished name (DN) of the returned object. By adding the -a flag, all of the associated attributes can be viewed (Example 4-18).

The syntax of the command is shown in the following:

```
   lsldap [-a] [ entity [entry_name | filter] ]
            entity: aliases, automount, bootparams, ethers, group, hosts
                    netgroup, networks, passwd, protocols, rpc, services
```

If no entity names of flags are supplied from the command line, then **lsldap** displays container entries of the entities, as shown in Example 4-18.

Example 4-18   Default lsldap output with no flags

```
# lsldap
dn: cn=Directory Administrators, dc=example,dc=com
dn: ou=Groups, dc=example,dc=com
dn: ou=Special Users,dc=example,dc=com
dn: ou=rpc,dc=example,dc=com
dn: ou=protocols,dc=example,dc=com
dn: ou=networks,dc=example,dc=com
dn: ou=netgroup,dc=example,dc=com
dn: ou=aliases,dc=example,dc=com
dn: ou=hosts,dc=example,dc=com
dn: ou=services,dc=example,dc=com
dn: ou=ethers,dc=example,dc=com
dn: ou=profile,dc=example,dc=com
dn: nismapname=auto_home,dc=example,dc=com
dn: nismapname=auto_appl,dc=example,dc=com
dn: nismapname=auto_master,dc=example,dc=com
dn: nismapname=auto_apps,dc=example,dc=com
dn: nismapname=auto_next_apps,dc=example,dc=com
dn: nismapname=auto_opt,dc=example,dc=com
dn: ou=group,dc=example,dc=com
dn: ou=people,dc=example,dc=com
#
```

Example 4-19 shows how to use the **lsldap** command with the passwd option to list the distinguished names of all users.

Example 4-19   Using lsldap to show user entries

```
# lsldap passwd
dn: uid=user1,ou=people,dc=example,dc=com
dn: uid=user2,ou=people,dc=example,dc=com
dn: uid=user3,ou=people,dc=example,dc=com
#
```

To retrieve all of the information for the user named user3 run the **lsldap** command with the -a passwd option and the user3 name, as shown in Example 4-20.

Example 4-20 Using lsldap by root to show entry for user3

```
# lsldap -a passwd user3
dn: uid=user3,ou=people,dc=example,dc=com
uidNumber: 20003
uid: user3
gidNumber: 20000
gecos: ITSO  user3
homeDirectory: /home/user3
loginShell: /bin/ksh
cn: ITSO user3
shadowLastChange: 12996
shadowInactive: -1
shadowMax: -1
shadowFlag: 0
shadowWarning: -1
shadowMin: 0
objectClass: posixAccount
objectClass: shadowAccount
objectClass: account
objectClass: top
userPassword: {crypt}hTQBG25y7pz6Q
```

All users can run the **lsldap** command, but when normal users run this command they will only be able to see public information. For example, using the same command as a normal user you see the information shown in Example 4-21.

Example 4-21 Normal user using lsldap to view user3

```
$ lsldap -a passwd user3
dn: uid=user3,ou=people,dc=example,dc=com
uidNumber: 20003
uid: user3
gidNumber: 20000
gecos: ITSO  user3
homeDirectory: /home/user3
loginShell: /bin/ksh
cn: ITSO user3
objectClass: posixAccount
objectClass: shadowAccount
objectClass: account
objectClass: top
```

For more information reference the **`lsldap`** command man page in the AIX 5L Version 5.3 InfoCenter commands and *Integrating AIX into Heterogeneous LDAP Environments*, SG24-7165.

`http://www.redbooks.ibm.com/abstracts/SG247165.html?Open`

5

Performance monitoring

In this chapter the following major topics are discussed:

- Performance tools enhancements (5300-05)
- The gprof command enhancement (5300-03)
- The topas command enhancements
- The iostat command enhancement (5300-02)
- PMAPI user tools (5300-02)
- Memory affinity enhancements (5300-01)
- The fcstat command (5300-05)

5.1 Performance tools enhancements (5300-05)

The performance monitoring tools **svmon**, **vmstat**, **curt**, **netpmon**, and **tprof** have been enhanced in AIX 5L Version 5.3 with TL 5300-05. Milicode and hypervisor support, automatic performance metric recording, and VIOS performance monitoring is provided and improved in 5300-05.

5.1.1 The svmon command enhancement

The **svmon** command is a tool developed to capture and analyze a snapshot of virtual memory. It is enhanced to support multiple page size in AIX 5.3 TL 5300-05.

For more information about multiple page sizes see 4.3, “Multiple page size support (5300-04)” on page 79.

The **svmon** command has been enhanced to provide a per-page size break-down of statistics. For example, to display global statistics about each page size, the -G option can be used with the **svmon** command, as shown in Example 5-1.

Example 5-1 Example output of svmon -G

```
# svmon -G
               size      inuse       free        pin    virtual
memory      8208384    5714226    2494158     453170    5674818
pg space     262144      20653

               work       pers       clnt
pin          453170          0          0
in use      5674818        110      39298

PageSize   PoolSize      inuse       pgsp        pin    virtual
s    4 KB          -    5379122      20653     380338    5339714
m  64 KB          -      20944          0       4552      20944
```

5.1.2 The vmstat command enhancement

The **vmstat** command is enhanced to display information about multiple page sizes. The -p and -P options of the **vmstat** command display VMM statistics for each supported page size.

The following displays global **vmstat** information along with a breakdown of statistics per page size:

```
vmstat -p
```

The following displays per page size statistics.

```
vmstat -P
```

Example 5-2 shows the output from -p.

Example 5-2 Example output of vmstat -p

```
# vmstat -p ALL

System configuration: lcpu=2 mem=6144MB

kthr    memory              page              faults        cpu
----- ----------- ------------------------ ------------ -----------
 r  b   avm   fre  re  pi  po  fr   sr  cy  in   sy  cs us sy id wa
 1  1 380755 207510   0   0   0   2    8   0  12 1518 172  0  0 99  0

  psz   avm    fre  re  pi  po  fr   sr  cy     siz
   4K 380755 207510   0   0   0   2    8   0 1528754
  16M      0     10   0   0   0   0    0   0      10
```

Both options take a comma-separated list of specific page sizes or the keyword *all* to indicate that information should be displayed for all supported page sizes that have one or more page frames. Example 5-3 displays per-page size information for all of the page sizes with page frames on a system.

Example 5-3 Example output of svmon -P

```
# vmstat -P all

System configuration: mem=1024MB

pgsz            memory                           page
----- -------------------------- ------------------------------------
           siz      avm      fre    re    pi    po    fr     sr    cy
   4K   262144   116202   116313     0     0     0     3      8     0
  64K    31379      961     3048     0     0     0     0      0     0
```

5.1.3 The curt command enhancement

The CPU Utilization Reporting Tool (**curt**) command converts a trace file into a number of statistics related to CPU utilization and process, thread, or pthread activity. These statistics ease the tracking of specific application activity. The **curt** command works with both uniprocessor and multiprocessor AIX Version 4 and AIX 5L Version 5 traces. However, it is shipped with AIX 5L Version 5.2. In AIX 5L Version 5.3 with TL 5300-05, the command is enhanced to support NFS v4.

The **curt** command report will be divided into many sections. The enhanced part of it will be:

- System NFS Calls Summary
- Pending NFS System Calls Summary

The following is an example about how the command supports NFS v4:

1. Get a system trace and **gensyms** command output. (This section does not cover how to trace.)
2. Use the **curt** command on the trace to get the command report:

```
curt -i trace.raw -n gensyms.out -o curt.out
```

3. Open the curt.out file with a text editor, and then you see that the report covers many sections:

```
General Information
System Summary
System Application Summary
Processor Summary
Processor Application Summary
Application Summary by TID
Application Summary by PID
Application Summary by Process Type
Kproc Summary
Application Pthread Summary by PID
System Calls Summary
Pending System Calls Summary
Hypervisor Calls Summary
Pending Hypervisor Calls Summary
System NFS Calls Summary
Pending NFS System Calls Summary
Pthread Calls Summary
Pending Pthread Calls Summary
FLIH Summary
SLIH Summary
```

- Information on completed NFS operations (System NFS Calls Summary) that includes the name of the NFS operation, the number of times the NFS operation was executed, and the total CPU time, expressed in milliseconds, and as a percentage with average, minimum, and maximum time the NFS operation call was running.
- Information on pending NFS operations (Pending NFS System Calls Summary), where the NFS operations did not complete before the end of the trace. The information includes the sequence number for NFS V2/V3, or opcode for NFS V4, the thread or process which made the NFS operation, and the accumulated CPU time that the NFS operation was running, expressed in milliseconds.

Example 5-4 shows sample output from the **curt** command.

Example 5-4 Sample output of curt report (partial)

```
...(lines omitted)...
                       System NFS Calls Summary
                     ------------------------
   Count   Total Time  Avg Time  Min Time  Max Time  % Tot  % Tot  Opcode
               (msec)    (msec)    (msec)    (msec)   Time  Count
========  ===========  ========  ========  ========  =====  =====  =============
       4       2.9509    0.7377    0.0331    2.4040  80.05   4.44  OPEN
      27       0.1935    0.0072    0.0036    0.0159   5.25  30.00  PUTFH
       4       0.1202    0.0300    0.0177    0.0397   3.26   4.44  READDIR
...(lines omitted)...
      4       0.0047    0.0012    0.0011    0.0013   0.13   4.44  SAVEPH
--------  -----------  --------  --------  --------  -----  -----  -------------
      90       3.6862    0.0410                                    NFS V4 SERVER
TOTAL

      15       0.8462    0.0564    0.0286    0.0797  40.33   1.96  NFS4_RFS4CALL
       2       0.2126    0.1063    0.0965    0.1161  10.13   0.26  NFS4_CREATE_ATTR
......
      1       0.0005    0.0005    0.0005    0.0005   0.02   0.13  NFS4_ADD_SETATTR
--------  -----------  --------  --------  --------  -----  -----  -------------
     765       2.0984    0.0027                                    NFS V4 CLIENT
TOTAL

                     Pending NFS Calls Summary
                     -------------------------
Accumulated   Sequence Number  Procname (Pid  Tid)
Time (msec)   Opcode
============  ===============  ==========================
      0.0275  NFS4_NFS4WRITE   kbiod(147528 483553)
      0.0191  NFS4_LOOKUP1     ls(438466 1155087)
      0.0201  NFS4_LOOKUP1     ls(438448 1167581)
      0.0091  NFS4_SETATTR     touch(438450 1167583)
...(lines omitted)...
```

The System NFS Calls Summary has the fields provided in Table 5-1.

Table 5-1 System NFS calls summary

Fields	Meaning
Opcode	The name of the system NFS call
Count	The number of times that a certain type of system NFS call (see Opcode) has been called during the monitoring period
Total Time (msec)	The total CPU time that the system spent processing system NFS calls of this type, expressed in milliseconds
Avg Time (msec)	The average CPU time that the system spent processing one system NFS call of this type, expressed in milliseconds
Min Time (msec)	The minimum CPU time that the system needed to process one system NFS call of this type, expressed in milliseconds
Max Time (msec)	The maximum CPU time that the system needed to process one system NFS call of this type, expressed in milliseconds
% Tot Time	The total CPU time that the system spent processing the system NFS calls of this type, expressed as a percentage of the total processing time
% Tot Count	The number of times that a system NFS call of a certain type was made, expressed as a percentage of the total count

The Pending System NFS Calls Summary has the fields provided in Table 5-2.

Table 5-2 Pending NFS calls summary

Fields	Meaning
Procname (Pid Tid)	The name of the process associated with the thread that made the system NFS call, its process ID, and the thread ID.

Fields	Meaning
Accumulated Time (msec)	The accumulated CPU time that the system spent processing the pending system NFS call, expressed in milliseconds.
Sequence Number	The sequence number represents the transaction identifier (XID) of an NFS operation. It is used to uniquely identify an operation and is used in the RPC call/reply messages. This number is provided instead of the operation name because the name of the operation is unknown until it completes.
Opcode	The name of pending operation NFS V4.

5.1.4 The netpmon command enhancement

The **netpmon** command uses the trace facility to obtain a detailed picture of network activity during a time interval. Because it uses the trace facility, the **netpmon** command can be run only by a root user or by a member of the system group.

The **netpmon** command cannot run together with any of the other trace-based performance commands such as **tprof** or **filemon**. In its usual mode, the **netpmon** command runs in the background while one or more application programs or system commands are being executed and monitored. The **netpmon** command is enhanced to support NFS v4, and it can report NFS server or client statistics now.

The following example shows how to use the **netpmon** command online to get the NFS performance report:

1. Collect trace data for 10 minutes using the **netpmon** command:

   ```
   netpmon -o netpmon.out -O all; sleep 600; trcstop
   ```

2. Open the netpmon.out file by using a text editor, and you will see the report. The output of the **netpmon** command is composed of two different types of reports: global and detailed. Example 5-5 shows an example of the **netpmon** command output.

Example 5-5 Sample output of netpmon (partial)

```
...(lines omitted)...
NFSv4 Client RPC Statistics (by Server):
----------------------------------------
Server                      Calls/s
----------------------------------
```

```
nim                           0.04
----------------------------------------------------------------------
Total (all servers)           0.04

======================================================================

NFSv4 Server Statistics (by Client):
------------------------------------
                    Read             Write            Other
Client              Ops/s            Ops/s            Ops/s
----------------------------------------------------------------------
nim                 0.00             0.00             0.14
----------------------------------------------------------------------
Total (all clients) 0.00             0.00             0.14
======================================================================
...(lines omitted)...
Detailed NFSv4 Client RPC Statistics (by Server):
-------------------------------------------------
SERVER: nim
calls:                  27
  call times (msec):    avg 1.569   min 0.083   max 11.697  sdev 3.322
COMBINED (All Servers)
calls:                  27
  call times (msec):    avg 1.569   min 0.083   max 11.697  sdev 3.322
======================================================================
Detailed NFSv4 Server Statistics (by Client):
---------------------------------------------
CLIENT: nim
writes:                 1
  write times (msec):   avg 0.062   min 0.062   max 0.062   sdev 0.000
other ops:              82
  other times (msec):   avg 0.448   min 0.001   max 11.543  sdev 2.012
COMBINED (All Clients)
writes:                 1
  write times (msec):   avg 0.062   min 0.062   max 0.062   sdev 0.000
other calls:            82
  other times (msec):   avg 0.448   min 0.001   max 11.543  sdev 2.012
...(lines omitted)...
```

5.1.5 The tprof command enhancement

The **tprof** command reports CPU usage for individual programs and the system as a whole. It uses trace facility. It is enhanced in AIX 5.3 TL05 for supporting named shared library areas.

AIX 5L now allows the designation of named shared library areas that can replace the global shared library area for a group of processes. A named shared library area enables a group of processes to have the full shared library capacity available to them at the same location in the effective address space as the global shared library area (segments 0xD and 0xF). The **tprof** command is named shared library ready now.

For more information about the named shared library area, see 1.7.3, “Named shared library areas (5300-03)” on page 16.

5.2 The gprof command enhancement (5300-03)

The **gprof** command produces an execution profile of C, Pascal, FORTRAN, or COBOL programs. The effect of called routines is incorporated into the profile of each caller. The **gprof** command is useful in identifying how a program consumes CPU resource. This can be used to determine a profile of which functions (routines) in the program are using the CPU.

The profile data is taken from the call graph profile file (gmon.out by default) created by programs compiled with the **cc** command using the -pg option. The -pg option also links in versions of library routines compiled for profiling, and reads the symbol table in the named object file (a.out by default), correlating it with the call graph profile file. If more than one profile file is specified, the **gprof** command output shows the sum of the profile information in the given profile files.

In the release of AIX 5L Version 5.3 with ML 5300-03, the **gprof** is enhanced to support applications containing more than 32 million symbols.

5.3 The topas command enhancements

There have been two major changes to the functionality of the **topas** command:

- The **topas** command can now monitor information from multiple LPARs within the same managed system.
- It is now possible to record information from **topas** and generate historical reports based on the data, both for an individual system and across multiple LPARs.

This section describes the use of these enhancements.

5.3.1 The topas command cross partition monitoring (5300-03)

The **topas** command is now able to collect statistics from multiple partitions on the same hardware platform. This can be invoked using the new -C flag or by using the C subcommand from any other panel. Example 5-6 shows the cross-partition monitor panel.

Example 5-6   The topas command cross-partition monitor panel

```
Topas CEC Monitor             Interval:  10             Thu Nov 30 10:48:23
2006
Partitions     Memory (GB)               Processors
Shr:  2        Mon: 2.2  InUse: 1.0      Shr:  1  PSz:  1   Shr_PhysB:  0.01
Ded:  1        Avl: 4.0                  Ded:  1  APP:  1.0 Ded_PhysB:  0.00

Host         OS  M Mem InU Lp  Us Sy Wa Id  PhysB  Ent  %EntC Vcsw PhI
-------------------------------------shared------------------------------------
-
lpar01       A53 U 1.0 0.4  4   0  0  0 99   0.01  0.50   1.5  412   1
VIO_Server1  A53 U 0.5 0.3  4   0  0  0 99   0.01  0.50   1.4  409   0
------------------------------------dedicated----------------------------------
-
lpar04       A53 S 0.8 0.3  2   0  0  0 99   0.00
```

The display is split into two sections, the global region and the partition region.

Global region

This region represents aggregated data from the partition set and is split into three sections: partitions, memory, and processors. The fields have the descriptions provided in Table 5-3.

Table 5-3   Global region fields

Field	Description
Partitions	
Shr	The number of shared partitions that are available for monitor.
Ded	Indicates the number of dedicated partitions.
Memory	
Mon	The monitored partitions' total memory.
Avl	Total memory available to the partition set. This field requires the partition to query information from the HMC. More details on this functionality are provided later.

Field	Description
InUse	Total memory in use by the monitored partitions.
	Processors
Shr	Number of shared CPUs.
Ded	Number of dedicated CPUs.
Psz	Active physical CPUs in the shared processor pool being used by this LPAR.
App	Available physical processors in the shared pool.
Shr_PhysB	Shared Physical Busy.
Ded_PhysB	Dedicated Physical Busy.

Partition region

This region displays information about individual partitions that are split depending on whether they use shared or dedicated CPUs. The fields in this region are provided in Table 5-4.

Table 5-4 Partition region fields

Field	Description	
Host	Host name of partition.	
OS	Operating system level.	
Mode	This shows the operating mode of the partition. Possible values are:	
	Shared partitions	
	C	Simultaneous multithreading enabled and capped
	c	Simultaneous multithreading disabled and capped
	U	Simultaneous multithreading enabled and uncapped
	u	Simultaneous multithreading disabled and uncapped
	Dedicated partitions	
	S	Simultaneous multithreading enabled

Field	Description	
	' ' (blank)	Simultaneous multithreading disabled
Mem	Total memory in GB.	
InU	Memory in use in GB.	
Lp	Number of logical processors.	
Us	Percentage of CPU used by programs executing in user mode.	
Sy	Percentage of CPU used by programs executing in kernel mode.	
Wa	Percentage of time spent waiting for I/O.	
Id	Percentage of time the CPUs are idle.	
Ded_PhysB	Dedicated Physical Busy.	
Ent	Entitlement granted (shared-only).	
%Entc	Percent Entitlement consumed (shared-only).	
Vcsw	Virtual context switches average per second (shared only).	
Phi	Phantom interrupts average per second (shared only).	

Within the cross-partition view there are additional subcommands available. Pressing s and d toggle on and off the shared and dedicated partition views, respectively, while pressing r will force the **topas** command to search for HMC configuration changes if a connection is available. This includes the discovery of new partitions, processors, or memory allocations. Finally, the g subcommand toggles the detail level of the global region. Example 5-7 shows the detailed view.

Example 5-7 The topas command detailed partition view without HMC data

```
Topas CEC Monitor             Interval:  10             Thu Nov 30 11:35:51 2006
Partition Info    Memory (GB)        Processor
Monitored  :  3   Monitored  : 2.2  Monitored  :2.0   Shr Physical Busy:  0.01
UnMonitored:  -   UnMonitored:   -  UnMonitored:  -   Ded Physical Busy:  0.00
Shared     :  2   Available  :   -  Available  :  -
Dedicated  :  1   UnAllocated:   -  UnAllocated:  -   Hypervisor
Capped     :  1   Consumed   : 1.0  Shared     :  1   Virt. Context Switch: 565
Uncapped   :  2                     Dedicated  :  1   Phantom Interrupts  :   0
                                     Pool Size  :  1
                                     Avail Pool :  1.0

Host         OS  M Mem InU Lp  Us Sy Wa Id  PhysB  Ent  %EntC Vcsw PhI
-------------------------------------shared-------------------------------------
VIO_Server1  A53 U 0.5 0.3  4   0  0  0 99   0.01 0.50   1.3  392   0
lpar01       A53 u 1.0 0.4  2   0  0  0 99   0.00 0.50   0.9  173   0
```

```
-----------------------------------dedicated----------------------------------
lpar04       A53   0.8 0.3  1   0  0  0 99   0.00
```

Implementation

In this example there are various fields that are incomplete. As of AIX 5L Version 5.3 Technology Level 5, a partition is able to query the HMC to retrieve further information regarding the CEC as a whole. This is enabled using the following process:

1. Install OpenSSH at the partition. This is available on the AIX 5L expansion pack or for download from the Web.
2. Enable remote command support on the HMC for user hscroot to allow SSH connections to be opened from the partition.
3. Configure SSH on the HMC to not require a password for the HMC user hscroot when queried from the selected partition. This requires the .ssh/authorized_keys2 on the HMC for user login hscroot.
4. Run the **ssh -l hscroot *<hmc address>* date** command from the partition to check whether the date is displayed without entering a password.
5. Utilize the following options to specify the managed system and HMC names when executing the **topas** command:

   ```
   -o managedsys=<managed system name under which this partition is
   configured>
   -o hmc=<HMC name under which this partition is configured>
   ```

Once this is configured, you should get an output similar to Example 5-8.

Example 5-8 The topas command detailed partition view with HMC data

```
Topas CEC Monitor             Interval:  10           Thu Nov 30 11:54:27
2006
Partition Info    Memory (GB)         Processor
Monitored   :  3   Monitored   : 2.2   Monitored  :2.0   Shr Physical Busy:  0.01
UnMonitored:  0   UnMonitored: 1.8   UnMonitored:0.0   Ded Physical Busy:  0.00
Shared      :  2   Available   : 4.0   Available  :  2
Dedicated   :  1   UnAllocated: 1.4   UnAllocated:0.0   Hypervisor
Capped      :  1   Consumed    : 1.0   Shared      :  1   Virt. Context Switch:
606
Uncapped    :  2                       Dedicated   :  1   Phantom Interrupts   :
0
                                       Pool Size   :  1
                                       Avail Pool :  1.0

Host         OS  M Mem InU Lp  Us Sy Wa Id  PhysB  Ent  %EntC Vcsw PhI
-------------------------------------shared------------------------------------
VIO_Server1  A53 U 0.5 0.3  4   0  0  0 99   0.01  0.50   1.2  379   0
```

```
lpar01       A53 u 1.0 0.4  2   0  0  0 99   0.00  0.50   1.0  227   0
-----------------------------------dedicated---------------------------------
lpar04       A53   0.8 0.3  1   0  0  0 99   0.00
```

If you are unable to obtain this information dynamically from the HMC, you can provide it as a command-line option, as shown in Table 5-5.

Table 5-5 Command options and their descriptions

Option	Description
-o availmem	Total memory size allocated to all partitions in MB.
-o unavailmem	Total memory size unallocated from the HMC in MB.
-o availcpu	Total CPUs allocated for all partitions on the HMC.
-o unavailcpu	Total CPUs unallocated on the HMC.
-o partitions	Number of partitions defined on the HMC.
-o reconfig	Number of seconds between checking for HMC configuration changes. Allowed values are 30, 60, 90, 120, 180, 240, and 300 seconds. The default is 60 seconds.
-o poolsize	Defined pool size, required if HMC Processor Utilization Authority restricts access.

5.3.2 Performance statistics recording (5300-05)

AIX 5L is now able to record performance statistics over time, which can be used to generate reports. There are two ways of performing this, depending on whether you are recording data for a local system or across multiple partitions. Local data can be recorded using the xmwlm agent, while statistics across partitions can be collected using the **topas -R** command. Data from both sources can be analyzed by the **topasout** command to generate reports.

The xmwlm agent

The xmwlm agent provides recording capability for a set of local system performance metrics when started with the -L flag. These include common CPU, memory, network, disk, and partition metrics typically displayed by the topas command. By default, daily recordings are stored in the /etc/perf/daily directory with files formatted as xmwlm.YYMMDD. An alternative directory can be specified with the -d flag. All recordings cover 24-hour periods and are retained for two days. The xmwlm agent can be started from the command line. Alternatively, the /usr/lpp/perfagent/config_aixwle.sh configuration script allows this function to be configured as an inittab process.

The topas command

When invoked with the -R flag, the **topas** command operates as a background process and display functions are disabled. Data available from the **topas** command cross-partition LPAR monitoring capability (-C flag) is then recorded to a 24-hour log file. The command can either be invoked manually or the /usr/lpp/perfagent/config_topas.sh configuration script allows this function to be configured as an inittab process. Data is placed in the /etc/perf/ directory and files are formatted as topas_cec.YYMMDD. The files are retained for seven days.

The topasout command

Once the recording files have been generated, the **topasout** command can be used to format the data. The command syntax is as follows:

```
topasout [-c|-s|-R daily|-R weekly] [-R detailed|-R summary|-R disk]
[ -i MM -b HHMM -e HHMM]][ -m type ]
[xmwlm_recording|topas_recording]
```

The flags relevant to **xmwlm** and **topas -R** outputs are provided in Table 5-6.

Table 5-6 The xmwlm and topas command flags

Flag	Description	
-c	Specifies that topasout should format the output files as comma-separated ASCII. Each line in the output files contains one time stamp and one observation. Both fields are preceded by a label that describes the fields.	
-s	Specifies that topasout should format the output files in a format suitable for input to spreadsheet programs.	
-m type	By default, the post-processor only outputs mean values. Other recorded values are available via the following type options:	
	min	Minimum value
	max	Maximum value
	mean	Mean value
	stdev	Standerd deviation
	set	The full set of outputs as a coma-separated list
	exp	The full set of outputs on separate lines
-R	Generates additional report outputs of the following type:	
	summary	Prints a single-line formatted output report of the system

Flag	Description	
	detailed -	Prints a formatted output report similar to the topas main display
	disk	Prints a single-line formatted output report of disk statistics

The options in Table 5-7 apply only to the reports generated by **topas -R**. If the -b and -e options are not used, the daily recording will be processed from beginning to end.

Table 5-7 The topas specific command options

Flag	Description
-i MM	Splits the recording reports into equal size time periods. This must be a multiple of the recording interval, which by default is 5 minutes. Allowed values are 5, 10, 15, 30, and 60.
-b HHMM	Begin time in hours (HH) and minutes (MM). Range is between 0000 and 2400.
-e HHMM	End time in hours (HH) and minutes (MM). Range is between 0000 and 2400 and must be greater than the begin time.

The following examples show a selection of data outputs generated by the topasout function. Example 5-9 gives an ASCII formatted output from **xmwlm**, while Example 5-10 on page 117 gives a summary output from **topas -R**, split using the -i flag.

Example 5-9 ASCII formatted output generated by topas out from xmwlm data file

```
Monitor: xmtrend recording--- hostname: nim ValueType: mean
Time="2006/11/29 00:00:02", CPU/gluser=0.32
Time="2006/11/29 00:00:02", CPU/glkern=0.97
Time="2006/11/29 00:00:02", CPU/glwait=0.01
Time="2006/11/29 00:00:02", CPU/glidle=98.70
Time="2006/11/29 00:00:02", CPU/numcpu=2.00
Time="2006/11/29 00:00:02", CPU/osver=5.30
Time="2006/11/29 00:00:02", NFS/Server/v3calls=0.00
Time="2006/11/29 00:00:02", NFS/Server/v2calls=0.00
Time="2006/11/29 00:00:02", NFS/Server/calls=0.00
Time="2006/11/29 00:00:02", NFS/Client/v3calls=0.00
Time="2006/11/29 00:00:02", NFS/Client/v2calls=0.00
Time="2006/11/29 00:00:02", NFS/Client/calls=0.00
```

Example 5-10   Summary output generated by topasout from topas -R data file

```
Report: CEC Summary  --- hostname: nim                        version:1.1
Start:11/28/06 00:00:21   Stop:11/28/06 14:24:21  Int:60 Min  Range: 864 Min
Partition Mon:  2  UnM:  0  Shr:  2  Ded:  0  Cap:  1  UnC:  1
       -CEC------  -Processors------------------------- -Memory (GB)------------
Time   ShrB  DedB  Mon  UnM  Avl UnA  Shr Ded  PSz  APP  Mon  UnM  Avl  UnA  InU
01:00  0.01  0.00  0.7  0.0  0.0   0  0.7   0  2.0  2.0  1.0  0.0  0.0  0.0  0.6
02:01  0.01  0.00  0.7  0.0  0.0   0  0.7   0  2.0  2.0  1.0  0.0  0.0  0.0  0.6
03:01  0.01  0.00  0.7  0.0  0.0   0  0.7   0  2.0  2.0  1.0  0.0  0.0  0.0  0.6
04:01  0.01  0.00  0.7  0.0  0.0   0  0.7   0  2.0  2.0  1.0  0.0  0.0  0.0  0.6
05:01  0.01  0.00  0.7  0.0  0.0   0  0.7   0  2.0  2.0  1.0  0.0  0.0  0.0  0.6
06:01  0.01  0.00  0.7  0.0  0.0   0  0.7   0  2.0  2.0  1.0  0.0  0.0  0.0  0.6
07:01  0.01  0.00  0.7  0.0  0.0   0  0.7   0  2.0  2.0  1.0  0.0  0.0  0.0  0.6
```

It is also possible to generate a **topas** command output from **topas -R**. Example 5-11 provides a sample output.

Example 5-11   Output generated by topasout from topas -R data files

```
Report: CEC Detailed --- hostname: nim                        version:1.1
Start:11/30/06 08:38:33   Stop:11/30/06 09:39:33  Int:60 Min  Range:  61 Min

Time: 09:38:32
-----------------------------------------------------------------
Partition Info   Memory (GB)        Processors
Monitored   :  3  Monitored   :  2.2  Monitored   :  2.0  Shr Physcl Busy:  0.05
UnMonitored:  0  UnMonitored:  0.0  UnMonitored:  0.0  Ded Physcl Busy:  0.00
Shared      :  2  Available   :  0.0  Available   :  0.0
Dedicated   :  1  UnAllocated:  0.0  Unallocated:  0.0  Hypervisor
Capped      :  1  Consumed    :  1.4  Shared      :  1.0  Virt Cntxt Swtch:  1642
UnCapped    :  2                      Dedicated   :  1.0  Phantom Intrpt  :    34
                                      Pool Size   :  1.0
                                      Avail Pool  :  1.0
Host         OS  M  Mem  InU Lp Us Sy Wa Id PhysB  Ent  %EntC  Vcsw PhI
-------------------------------------shared-------------------------------------
nim          A53 U  1.0  0.7  4  0  1  0 95  0.04  0.5   7.07   926  33
VIO_Server1  A53 U  0.5  0.3  4  0  0  0 99  0.01  0.5   2.30   716   2
------------------------------------dedicated-----------------------------------
appserver    A53 S  0.8  0.3  2  0  0  0 99  0.00
```

5.4 The iostat command enhancement (5300-02)

The **iostat** command has been enhanced since the first offering of AIX 5L Version 5.3. The major improvement is the addition of the -D flag, which allows the reporting of extended disk service time metrics and VSCSI statistics.

5.4.1 Extended disk service time metrics

When invoked with the -D flag, the **iostat** command now produces extended information covering disk service times. This allows for more detailed analysis of read and write performance along with additional statistics that were not previously available. The command can report on all disks or be limited to specific devices. Example 5-12 gives the new fields. Here they are presented in the default 80 column format. However, the information can be recorded on a single line using the -l flag.

Example 5-12 The iostat -D command output

```
iostat -D

System configuration: lcpu=2 drives=1 paths=1 vdisks=1

hdisk0        xfer:  %tm_act      bps      tps      bread      bwrtn
                        0.1      2.0K     0.3      221.2       1.8K
              read:       rps  avgserv  minserv  maxserv   timeouts      fails
                        0.0      4.7      0.2     25.2          0          0
             write:       wps  avgserv  minserv  maxserv   timeouts      fails
                        0.3      7.5      0.8     51.0          0          0
             queue:  avgtime  mintime  maxtime  avgwqsz    avgsqsz     sqfull
                        7.8      0.0      1.4S     0.0        0.0      31816
```

The output is split into four sections (xfer, read, write, and queue) discussed in the following sections.

The xfer column

The xfer column provides an overview of the devices' performance, similar to the standard **iostat** output. The fields are as follows.

% tm_act Indicates the percentage of time the physical disk was active (bandwidth utilization for the drive).

bps Indicates the amount of data transferred (read or written) per second to the drive. Different suffixes are used to represent the unit of transfer. Default is in bytes per second.

tps Indicates the number of transfers per second that were issued to the physical disk. A transfer is an I/O request to the physical disk. Multiple logical requests can be combined into a single I/O request to the disk. A transfer is of indeterminate size.

bread Indicates the amount of data read per second, from the drive. Different suffixes are used to represent the unit of transfer. Default is in bytes per second.

bwrtn Indicates the amount of data written per second, to the drive. Different suffixes are used to represent the unit of transfer. Default is in bytes per second.

The read column

This read column provides a more detailed breakdown to read statistics for the device. The following details are provided:

rps Indicates the number of read transfers per second.

avgserv Indicates the average service time per read transfer. Different suffixes are used to represent the unit of time. Default is in milliseconds.

minserv Indicates the minimum read service time. Different suffixes are used to represent the unit of time. Default is in milliseconds.

maxserv Indicates the maximum read service time. Different suffixes are used to represent the unit of time. Default is in milliseconds.

timeouts Indicates the number of read timeouts per second.

fails Indicates the number of failed read requests per second.

The write column

This write column provides a more detailed breakdown to write statistics for the device. The following details are provided:

wps Indicates the number of write transfers per second.

avgserv Indicates the average service time per write transfer. Different suffixes are used to represent the unit of time. Default is in milliseconds.

minserv Indicates the minimum write service time. Different suffixes are used to represent the unit of time. Default is in milliseconds.

maxserv Indicates the maximum write service time. Different suffixes are used to represent the unit of time. Default is in milliseconds.

timeouts Indicates the number of write timeouts per second.

fails Indicates the number of failed write requests per second.

The queue column

This queue column provides the following information regarding wait queues at the device:

avgtime Indicates the average time spent by a transfer request in the wait queue. Different suffixes are used to represent the unit of time. Default is in milliseconds.

mintime Indicates the minimum time spent by a transfer request in the wait queue. Different suffixes are used to represent the unit of time. Default is in milliseconds.

maxtime Indicates the maximum time spent by a transfer request in the wait queue. Different suffixes are used to represent the unit of time. Default is in milliseconds.

avgwqsz Indicates the average wait queue size.

avgsqsz Indicates the average service queue size.

sqfull Indicates the number of times the service queue becomes full (that is, the disk is not accepting any more service requests) per second.

For all the fields, the outputs have a default unit. However, as values change order of magnitude, different suffixes are provided to improve clarity. The possible values are shown in Table 5-8.

Table 5-8 Possible suffixes for iostat -D command fields

Suffix	Description
K	1000 bytes.
M	1 000 000 bytes if displayed in xfer metrics. Minutes, if displayed in read/write/wait service metrics.
G	1 000 000 000 bytes.
T	1 000 000 000 000 bytes.
S	Seconds.
H	Hours.

> **Note:** The M suffix is used both for data and time metrics. Therefore a value of 1.0 M could indicate 1 MB of data or 1 minute, depending on the context.

As with the standard **iostat** command, two report types can be generated. If the command is run without an interval, the system generates a summary report of the statistics since boot. If an interval is specified, then data is collected over the given time period. For interval data, the values for the max and min fields reported give the respective values for the whole data collection period rather than that specific interval. In Example 5-13, the value of 77.4 ms for the queue maxtime occurred during the second interval and was not surpassed in the third, so it is reported again. If the -R flag is specified, the command will report the maximum value just for that interval.

Example 5-13 The iostat -D command interval output

```
-------------------------------------------------------------------------------
hdisk0         xfer:  %tm_act      bps      tps      bread      bwrtn
                         0.0      0.0      0.0        0.0        0.0
               read:      rps  avgserv  minserv  maxserv   timeouts      fails
                         0.0      0.0      0.5      0.5           0          0
              write:      wps  avgserv  minserv  maxserv   timeouts      fails
                         0.0      0.0      0.0      0.0           0          0
              queue:  avgtime  mintime  maxtime  avgwqsz    avgsqsz     sqfull
                         0.0      0.0      0.0      0.0      0.0             0
-------------------------------------------------------------------------------
hdisk0         xfer:  %tm_act      bps      tps      bread      bwrtn
                        20.0    208.9K    45.0        0.0      208.9K
               read:      rps  avgserv  minserv  maxserv   timeouts      fails
                         0.0      0.0      0.5      0.5           0          0
              write:      wps  avgserv  minserv  maxserv   timeouts      fails
                        45.0      8.2      1.7     21.5           0          0
              queue:  avgtime  mintime  maxtime  avgwqsz    avgsqsz     sqfull
                        20.3      0.0     77.4      1.4      0.2            23
-------------------------------------------------------------------------------
hdisk0         xfer:  %tm_act      bps      tps      bread      bwrtn
                         0.0      0.0      0.0        0.0        0.0
               read:      rps  avgserv  minserv  maxserv   timeouts      fails
                         0.0      0.0      0.5      0.5           0          0
              write:      wps  avgserv  minserv  maxserv   timeouts      fails
                         0.0      0.0      1.7     21.5           0          0
              queue:  avgtime  mintime  maxtime  avgwqsz    avgsqsz     sqfull
                         0.0      0.0     77.4      0.0      0.0             0
-------------------------------------------------------------------------------
```

5.4.2 Extended virtual SCSI statistics

In addition to the extended disk metrics, the **iostat** command can now provide further details on performance of virtual adaptors. This can be done by combining the -a flag to generate adapter data with the -D flag. Example 5-14 provides a sample output.

Example 5-14 The iostat -aD command sample output

```
# iostat -aD

System configuration: lcpu=2 drives=1 paths=1 vdisks=1

Vadapter:
vscsi1               xfer:     Kbps      tps   bkread     bkwrtn partition-id
                              11.5      1.7      1.0         0.7            0
                     read:      rps  avgserv  minserv  maxserv
                               0.0     32.0S     0.2     25.3S
                    write:      wps  avgserv  minserv  maxserv
                             11791.1      0.0      1.2     25.3S
                    queue:  avgtime  mintime  maxtime  avgwqsz    avgsqsz
sqfull
                             0.0      0.0      0.0     0.0        0.0
0


Paths/Disks:
hdisk0               xfer:  %tm_act      bps      tps      bread      bwrtn
                               0.5     11.8K     1.7        8.3K       3.5K
                     read:      rps  avgserv  minserv  maxserv   timeouts
fails
                             1.0      4.2      0.2    22.9           0
0
                    write:      wps  avgserv  minserv  maxserv   timeouts
fails
                             0.7      7.8      1.2    35.2           0
0
                    queue:  avgtime  mintime  maxtime  avgwqsz    avgsqsz
sqfull
                              5.8      0.0    191.6      0.0       0.0
605
-------------------------------------------------------------------------------
```

The output reports data for the adapter and then all of the devices attached to it. For non-VSCSI adapters, the adapter statistics will not report the full detail of output, but the disk devices will. The fields in the output are split into the same sections as for the extended disk output. However, there are some extra fields and slight alterations to the outputs. As with the disk statistics, different suffixes are used to represent data of different orders of magnitude.

The xfer column

The xfer column provides the following information:

Kbps Indicates the amount of data transferred (read or written) in the adapter in KB per second

tps Indicates the number of transfers per second issued to the adapter

bkread Number of blocks received per second from the hosting server to this adapter

bkwrtn Number of blocks per second sent from this adapter to the hosting server

partition-id The partition ID of the hosting server, which serves the requests sent by this adapter

The read column

The read column provides the following information:

rps Indicates the number of read requests per second

avgserv Indicates the average time to receive a response from the hosting server for the read request sent

minserv Indicates the minimum time to receive a response from the hosting server for the read request sent.

maxserv Indicates the maximum time to receive a response from the hosting server for the read request sent

The write column

The write column provides the following information:

wps Indicates the number of write requests per second

avgserv Indicates the average time to receive a response from the hosting server for the write request sent

minserv Indicates the minimum time to receive a response from the hosting server for the write request sent

maxserv Indicates the maximum time to receive a response from the hosting server for the write request sent

The queue column

The queue column provides the following information:

avgtime Indicates the average time spent by a transfer request in the wait queue. Different suffixes are used to represent the unit of time. Default is in milliseconds.

mintime Indicates the minimum time spent by a transfer request in the wait queue.

maxtime Indicates the maximum time spent by a transfer request in the wait queue.

avgwqsz Indicates the average wait queue size.

avgsqsz Indicates the average service queue size.

sqfull Indicates the number of times the service queue becomes full (that is, the hosting server is not accepting any more service requests) per second.

5.5 PMAPI user tools (5300-02)

PMAPI is a library with APIs that provide access to the hardware performance counters. Beginning with AIX 5L Version 5.3 with 5300-02 Technology Level, PMAPI has been enhanced and modifications made to the HPMtoolkit library (HPMlib). Two new commands have also been added, **hpmstat** and **hpmcount**.

5.5.1 The hpmcount command

The purpose of the **hpmcount** command is to measure application performance. The **hpmcount** command provides the execution wall clock time, hardware performance counters information, derived hardware metrics, and resource utilization statistics (obtained from the getrusage() system call) for a specified command. The syntax is as follows. Parameters are described in Table 5-9.

```
hpmcount [-a] [-d] [-H] [-k] [-o file] [-s set] command
hpmcount [-h]
```

Table 5-9 The hpmcount command parameters details

Parameter	Description
a	Aggregates the counters on POE runs.
d	Adds detailed set counts for counter multiplexing mode.
h	Displays help message.

Parameter	Description
H	Adds hypervisor activity on behalf of the process.
k	Adds system activity on behalf of the process.
o	File output file name.
s	Set lists a predefined set of events or a comma-separated list of sets. (1 to N, or 0 to select all. See the **pmlist** command.) When a comma-separated list of sets is used, the counter multiplexing mode is selected.

Example 5-15 shows the usage of the **hpmcount** command to check the activities when issuing the **ls** command.

Example 5-15 The hpmcount command example

```
#/usr/pmapi/tools/hpmcount -s 1 ls
Sachin_xlc      lost+found      paging          sach_test.conf  spots
dump            lpp_source      resources       sach_test.txt   tmp
home            mksysbs         root            scripts
 Execution time (wall clock time): 0.003478 seconds

 ########  Resource Usage Statistics  ########

 Total amount of time in user mode            : 0.000794 seconds
 Total amount of time in system mode          : 0.001816 seconds
 Maximum resident set size                    : 284 Kbytes
 Average shared memory use in text segment    : 0 Kbytes*sec
 Average unshared memory use in data segment  : 0 Kbytes*sec
 Number of page faults without I/O activity   : 78
 Number of page faults with I/O activity      : 0
 Number of times process was swapped out      : 0
 Number of times file system performed INPUT  : 0
 Number of times file system performed OUTPUT : 0
 Number of IPC messages sent                  : 0
 Number of IPC messages received              : 0
 Number of signals delivered                  : 0
 Number of voluntary context switches         : 0
 Number of involuntary context switches       : 0

 #######  End of Resource Statistics  ########

 Set: 1
 Counting duration: 0.003147321 seconds
  PM_FPU_1FLOP (FPU executed one flop instruction)            :               0
  PM_CYC (Processor cycles)                                   :          561038
  PM_MRK_FPU_FIN (Marked instruction FPU processing finished) :               0
  PM_FPU_FIN (FPU produced a result)                             :            45
  PM_INST_CMPL (Instructions completed)                          :        264157
  PM_RUN_CYC (Run cycles)                                        :        561038


  Utilization rate                                     :           9.751 %
  MIPS                                                 :          75.951 MIPS
  Instructions per cycle                               :           0.471
  HW floating point instructions per Cycle             :           0.000
  HW floating point instructions / user time           :           0.133 M
HWflops/s
  HW floating point rate (HW Flops / WCT)              :           0.013 M
HWflops/s
```

5.5.2 The hpmstat command

The **hpmstat** command provides system-wide hardware performance counter information. The **hpmstat** command provides the execution wall clock time, hardware performance counters information, and derived hardware metrics. It can only be used by a user with root privilege.

When specified without command-line options, the **hpmstat** command counts the default 1 iteration of user, kernel, and hypervisor (for processors supporting hypervisor mode) activity for 1 second for the default set of events. It then writes the raw counter values and derived metrics to standard output. By default, runlatch is disabled so that counts can be performed while executing in idle cycle.

```
hpmstat [-d] [-H] [-k] [-o file] [-r] [-s set] [-T] [-U] [-u]
interval count
or
hpmstat [-h]
```

Table 5-10 provides the **hpmstat** command parameters.

Table 5-10 The hpmstat command parameter details

Parameter	Description
d	Adds detailed set counts for counter multiplexing mode.
H	Adds detailed set counts for counter multiplexing mode.
h	Displays help message.
k	Counts system activity only.
o	Output file name.

Parameter	Description
r	Enables runlatch and disables counts while executing in idle cycle.
s	Lists a predefined set of events or a comma-separated list of sets. (1 to N, or 0 to select all. See the **pmlist** command.) When a comma-separated list of sets is used, the counter multiplexing mode is selected.
T	Writes time stamps instead of time in seconds.
U	Puts counting time interval in microseconds. This option is ignored if the counter multiplexing mode is specified.
u	Counts user activity only.

Example 5-16 shows the usage of the **hpmstat** command to check the activities of system.

Example 5-16 The hpmstat command example

```
#/usr/pmapi/tools/hpmstat -s 7
 Execution time (wall clock time): 1.000307 seconds

 Set: 7
 Counting duration: 2.000129112 seconds
  PM_TLB_MISS (TLB misses)                             :          37114
  PM_CYC (Processor cycles)                            :       17995529
  PM_ST_REF_L1 (L1 D cache store references)           :         874385
  PM_LD_REF_L1 (L1 D cache load references)            :        1835177
  PM_INST_CMPL (Instructions completed)                :        8820856
  PM_RUN_CYC (Run cycles)                              :       14684180


  Utilization rate                                   :          1.087 %
  MIPS                                               :          8.818 MIPS
  Instructions per cycle                             :          0.490
  Total load and store operations                    :          2.710 M
  Instructions per load/store                        :          3.255
  Number of loads per TLB miss                       :         49.447
  Number of load/store per TLB miss                  :         73.006
```

For further information about PMAPI, see the following:

http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=/com.ibm.aix.prftools/doc/prftools/perfmon_api.htm

5.6 Memory affinity enhancements (5300-01)

IBM POWER-based hardware models contain modules that are capable of supporting single, dual, or multiple core chips depending on the particular configuration. The system memory is then attached to these modules. AIX 5L has a feature called memory affinity that allows the operating system to consider this hardware structure when making memory allocations. This means that, when a process generates a page fault, memory can be allocated from the module containing the processor that it was running on. This can reduce latency by utilizing memory local to the processor, and in turn can give significant performance benefits in certain environments.

When memory affinity is enabled, each module has its own vmpool, which contains one or more memory pools. During system boot, AIX 5L uses hardware topology information to discover the memory attached to the various modules and define the memory pools. The amount of memory in each pool is based on how much memory is available in the module or allocated to the VMM by the hypervisor layer. This means that different memory pools can have different sizes, depending on the exact way in which the hypervisor allocates resources when the partition is activated. Each memory pool has its own instance of the page replacement daemon, lrud, and page faults within individual memory pools are dealt with independently as far as possible.

Starting with AIX 5L 5300-01, the **vmo** command has been enhanced to provide more control over user memory placement across these pools. Memory can be allocated in one of two ways, first-touch and round robin. With the first-touch scheduling policy, memory is allocated from the chip module that the thread was running on when it first touched that memory segment, which is the first page fault. With the round-robin scheduling policy, memory allocation is striped across each of the vmpools. The following parameters have been added to the **vmo** command to control how different memory types are allocated and can either have a value of 1, signifying the first touch scheduling policy, or 2, signifying the round-robin scheduling policy.

memplace_data This parameter specifies the memory placement for data of the main executable that is either initialized or uninitialized, heap segment data, shared library data, and data of object modules that are loaded at run time. The default value of this parameter is 2.

memplace_mapped_file This parameter specifies the memory placement for files that are mapped into the address space of a process, such as the shmat() function and the mmap() function. The default value for this parameter is 2.

memplace_shm_anonymous This parameter specifies the memory placement for anonymous shared memory that acts as working storage memory that is created by a call to the shmget() function or the mmap() function. The memory can only be accessed by the creating process or its descendants, and it is not associated with a name or a key. The default value for this parameter is 2.

memplace_shm_named This parameter specifies the memory placement for named shared memory that acts as working storage memory that is created by a call to the shmget() function or the shm_open() function. It is associated with a name or a key that allows more than one process to access it simultaneously. The default value for this parameter is 2.

memplace_stack This parameter specifies the memory placement for the program stack. The default value for this parameter is 2.

memplace_text This parameter specifies the memory placement for the application text of the main executable, but not for its dependencies. The default value for this parameter is 2.

memplace_unmapped_file This parameter specifies the memory placement for unmapped file access, such as with the read() or write() functions. The default value for this parameter is 2.

By default, memory affinity is enabled on all AIX 5L systems. However, it can be disabled using the memory_affinity vmo parameter. In AIX 5L release 5300-01, this parameter was deprecated and at the 5300-01 and 5300-02 levels. Memory affinity is always on. With APAR IY73792, which is part of release 5300-03, the parameter was re-enabled, and in subsequent releases it can be turned on and off. Disabling memory affinity may provide beneficial results if the hypervisor allocates memory to a partition such that the memory pools become very unbalanced in size. This can lead to unexpected page replacement behavior, as small pools can be subject to memory usage fluctuations. The number and size of the memory pools on a system can be determined using the **kdb** command's mempool subcommand, as shown in Example 5-17.

Example 5-17 Using the mempool subcommand to show a systems memory pools

```
(0)> mempool *
                 VMP MEMP  NB_PAGES  FRAMESETS        NUMFRB
```

```
memp_frs+010000  00  000   0001672C  000 001         0000DEE2
memp_frs+010280  00  001   000174A1  002 003         0000E241
(0)>
```

A reboot is required to enable or disable memory affinity as memory pool information is defined at boot time.

5.7 The fcstat command (5300-05)

AIX 5L Version 5.3 technology level 5 introduces the **fcstat** command. The purpose of this command is to display statistics gathered by the specified Fibre Channel device driver. It collects the statistics using the following procedure:

- Opens the message catalog of fcstat and checks the parameter list.
- Accesses the ODM database for information relating to the selected adapter.
- Accesses the ODM database for information relating to ports of the selected adapter.
- Opens and access adapter statistics.
- Reports statistics and exits.

The **fcstat** command used in Example 5-18 displays statistics for the Fibre Channel device driver fcs0. At this moment their are no flags associated with this command.

Example 5-18 The fcstat command

```
#fcstat fcs0
FIBRE CHANNEL STATISTICS REPORT: fcs0

Device Type: FC Adapter (df1000f9)
Serial Number: 1E313BB001
Option ROM Version: 02C82115
Firmware Version: B1F2.10A5
Node WWN: 20000000C9487B04
Port WWN: 10000000C9416DA4

FC4 Types
Supported:
0x0000010000000000000000000000000000000000000000000000000000000000
Active:
0x0000010000000000000000000000000000000000000000000000000000000000
Class of Service: 4
Port FC ID: 011400
Port Speed (supported): 2 GBIT
Port Speed (running): 1 GBIT
Port Type: Fabric

Seconds Since Last Reset: 345422

Transmit Statistics Receive Statistics
------------------- ------------------
Frames: 1 Frames: 1
Words: 1 Words: 1

LIP Count: 1
NOS Count: 1
Error Frames: 1
Dumped Frames: 1
Link Failure Count: 1
Loss of Sync Count: 1
Loss of Signal: 1
Primitive Seq Protocol Err Count: 1
Invalid Tx Word Count: 1
Invalid CRC Count: 1

IP over FC Adapter Driver Information
No DMA Resource Count: 0
No Adapter Elements Count: 0

FC SCSI Adapter Driver Information
No DMA Resource Count: 0
No Adapter Elements Count: 0
No Command Resource Count: 0

IP over FC Traffic Statistics
Input Requests: 0
Output Requests: 0
Control Requests: 0
Input Bytes: 0
Output Bytes: 0

FC SCSI Traffic Statistics
Input Requests: 16289
Output Requests: 48930
Control Requests: 11791
Input Bytes: 128349517
Output Bytes: 209883136
```

Table 5-11 provides descriptions of the statistics fields.

Table 5-11 Statistics fields and their descriptions

Statistics	Description
Device Type	Displays the description of the adapter
Serial Number	Displays the serial number from the adapter
Option ROM Version	Displays the version of the Options ROM on the adapter
Firmware Version	Displays the version of the firmware on the adapter
Node WWN	Displays the worldwide name of the adapter
Port FC ID	Displays the SCSI ID of the adapter
Port Type	Displays the adapter's connection type
Port Speed	Displays the speed of the adapter
Port WWN	Displays the worldwide name of the port
Seconds Since Last Reset	Displays the seconds since last reset of the statistics on the adapter
Frames	Displays the number of frames transmitted and received
Words	Displays the number of words transmitted and received
LIP Count	Displays the LIP count
NOS Count	Displays the NOS count
Error Frames	Displays the number of frames that were in error
Dumped Frames	Displays the frames that were dumped
Link Failure Count	Displays the Link Failure Count
Loss of Sync Count	Displays the number of times Sync was lost
Loss of Signal	Displays the number of times signal was lost
Primitive Seq Protocol Err Count	Displays the number of times a primitive sequence was in error
Invalid Tx Word Count	Displays the number of invalid transfers that occurred
Invalid CRC Count	Displays the number of CRC errors that occurred
IP over FC Adapter Driver Information: No Adapter Elements Count	Displays the number of times there were no adapter elements available

Statistics	Description
FC SCSI Adapter Driver Information: No DMA Resource Count	Displays the number of times DMA resources were not available
FC SCSI Adapter Driver Information: No Adapter Elements Count	Displays the number of times there were no adapter elements available
FC SCSI Adapter Driver Information: No Command Resource Count	Displays the number of times there were no command resources available
IP over FC Traffic Statistics: Input Requests	Displays the number of input requests
IP over FC Traffic Statistics: Output Requests	Displays the number of output requests
IP over FC Traffic Statistics: Control Requests	Displays the number of control requests
IP over FC Traffic Statistics: Input Bytes	Displays the number of input bytes
IP over FC Traffic Statistics: Output Bytes	Displays the number of output bytes
FC SCSI Traffic Statistics: Input Requests	Displays the number of input requests
FC SCSI Traffic Statistics: Output Requests	Displays the number of output requests
FC SCSI Traffic Statistics: Control Requests	Displays the number of control requests
FC SCSI Traffic Statistics: Input Bytes	Displays the number of input bytes
FC SCSI Traffic Statistics: Output Bytes	Displays the number of output bytes

Note : Some adapters might not support a specific statistic. The value of non-supported statistic fields is always 0.

5.8 Virtualization performance enhancements

This section contains enhancements related to virtual I/O and micropartitioned processors.

5.8.1 SCSI queue depth

The maximum transfer size for virtual SCSI client adapters is set by the Virtual I/O Server, which determines that value based on the resources available on that server and the maximum transfer size for the physical storage devices on that server. Other factors include the queue depth and maximum transfer size of other devices involved in mirrored volume group or MPIO configurations. Increasing the queue depth for some devices might reduce the resources available for other devices on that same parent adapter and decrease throughput for those devices.

Increasing the value of queue_depth above the default value of three might improve the throughput of the disk in some configurations. However, there are several other factors that must be taken into consideration. These factors include the value of the queue_depth attribute for all of the physical storage devices on the Virtual I/O Server being used as a virtual target device by the disk instance on the client partition. It also includes the maximum transfer size for the virtual SCSI client adapter instance that is the parent device for the disk instance.

The most straightforward configuration is when there is a physical LUN used as the virtual target device. In order for the virtual SCSI client device queue depth to be used effectively, it should not be any larger than the queue depth on the physical LUN. A larger value wastes resources without additional performance. If the virtual target device is a logical volume, the queue depth on all disks included in that logical volume must be considered. If the logical volume is being mirrored, the virtual SCSI client queue depth should not be larger than the smallest queue depth of any physical device being used in a mirror. When mirroring, the LVM writes the data to all devices in the mirror, and does not report a write as completed until all writes have completed. Therefore, throughput is effectively throttled to the device with the smallest queue depth. This applies to mirroring on the Virtual I/O Server and the client.

We recommend that you have the same queue depth on the virtual disk as the physical disk. If you have a volume group on the client that spans virtual disks, keep the same queue depth on all of the virtual disks in that volume group. This is most important if you have mirrored logical volumes in that volume group, because the write will not complete before the data is written to the last disk.

In MPIO configurations on the client, if the primary path has a much greater queue depth than the secondary, there might be a sudden loss of performance as the result of a failover.

The virtual SCSI client driver allocates 512 command elements for each virtual I/O client adapter instance. Two command elements are reserved for the adapter to use during error recovery, and three command elements are reserved for each device that is open to be used in error recovery. The rest are left in a common pool for use in I/O requests. As new devices are opened, command elements are removed from the common pool. Each I/O request requires one command element for the time that it is active on the Virtual I/O Server.

Increasing the queue depth for one virtual device reduces the number of devices that can be open at one time on that adapter. It also reduces the number of I/O requests that other devices can have active on the Virtual I/O Server.

As an example, consider the case shown in Figure 5-1. In this scenario, we map a physical disk to a virtual disk.

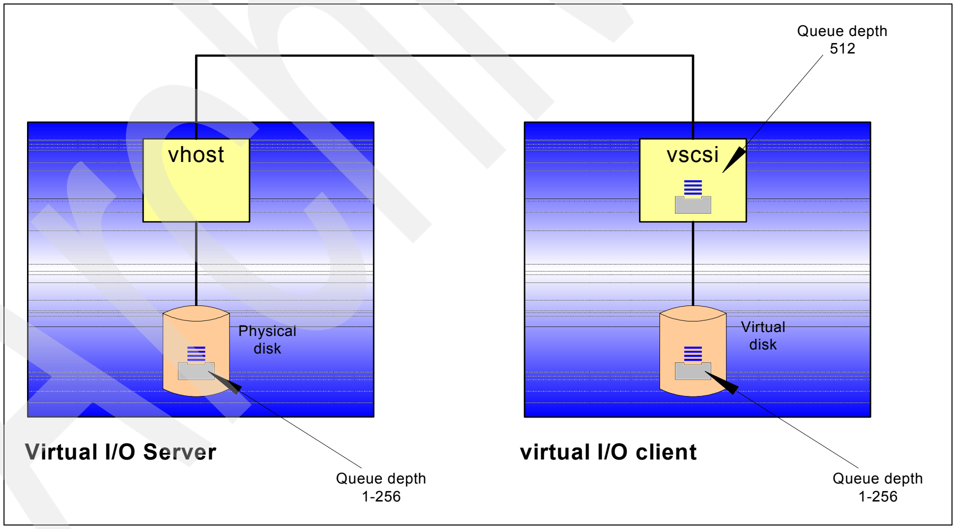


Figure 5-1 Physical disk mapped to a virtual disk

On the Virtual I/O Server, you can run the **lsdev -dev hdisk*N* -attr** command specifying the physical disk that you virtualized to the virtual I/O client. Record the queue depth, as shown in bold in Example 5-19.

Example 5-19 Using the lsdev command on the Virtual I/O Server

```
$ lsdev -dev hdisk2 -attr
attribute       value                            description
user_settable

PCM             PCM/friend/scsiscsd              Path Control Module
False
algorithm       fail_over                        Algorithm
True
dist_err_pcnt   0                                Distributed Error Percentage
True
dist_tw_width   50                               Distributed Error Sample Time
True
hcheck_interval 0                                Health Check Interval
True
hcheck_mode     nonactive                        Health Check Mode
True
max_transfer    0x40000                          Maximum TRANSFER Size
True
pvid            00cddeec68220f190000000000000000 Physical volume identifier
False
queue_depth     3                                Queue DEPTH
False
reserve_policy  single_path                      Reserve Policy
True
size_in_mb      36400                            Size in Megabytes
False
$
```

On the virtual I/O client, run the **chdev -l hdiskN -a queue_depth=*x*** command, where *x* is the number that you recorded previously on the Virtual I/O Server. Using this approach, you will have balanced the physical and virtual disk queue depth. Remember that the size of the queue depth will limit the number of devices that you can have on the virtual I/O client SCSI adapter because it partitions the available storage to the assigned resources.

Figure 5-2 on page 137 shows another case. A logical volume on the Virtual I/O Server is mapped to a virtual disk on the virtual I/O client. Note that every logical volume on the Virtual I/O Server shares the same disk queue. The goal in fine-grained performance tuning is to avoid flooding the disk queue. Consider the example where you have a physical disk with a default queue depth of three. You also have three logical volumes virtualized on that disk, and all the virtual disks on the virtual I/O clients have a default queue depth of three. In this case, you might end up with nine pending I/Os on a queue depth of three. One of the solutions is to increase the queue depth of the physical disk to match the total of the virtual disk queue depth.

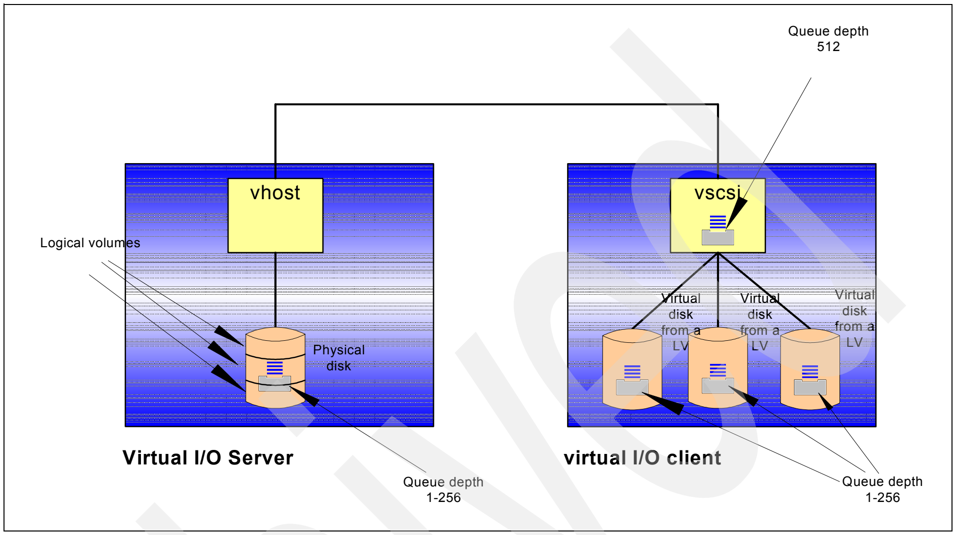


Figure 5-2 LV mapped to virtual disk

Increasing the VSCSI client queuing can be a useful optimization when:

- The storage is Fibre Channel attached.

  SCSI queue depth is already a limiting factor using the default setting of three.

- The VSCSI device is attached using LUNs:
  - The Virtual I/O Server does not allow striping LVs across multiple disks/LUNs.
  - LVs could benefit when on a LUN with multiple disks and there is not great contention on the LUN from requests from multiple clients.

- The LUN contains multiple physical disks.

  The more disks in the LUN, the better the possibility of more I/Os in flight at the same time.

- The workload has enough processes or asynchronous I/O to drive a lot of outstanding I/O requests.

5.8.2 Ethernet largesend option

IBM System p Gigabit or higher Ethernet adapters support TCP segmentation offload (also called largesend). This feature extends the TCP largesend feature to virtual Ethernet adapters and Shared Ethernet Adapters (SEA). In largesend environments, the TCP will send a big *chunk* of data to the adapter when TCP knows that the adapter supports largesend. The adapter will break this big TCP packet into multiple smaller TCP packets that will fit the outgoing MTU of the adapter, saving system CPU load and increasing network throughput.

The TCP largesend feature is extended from LPAR all the way up to the real adapter of VIOS. The TCP stack on the LPAR will determine whether the VIOS supports largesend. If VIOS supports TCP largesend, the LPAR sends a big TCP packet directly to VIOS.

If virtual Ethernet adapters are used in a LPAR-LPAR environment, however, the large TCP packet does not need to be broken into multiple smaller packets. This is because the underlying hypervisor will take care of sending the big chunk of data from one LPAR to another LPAR.

This feature allows the use of a large MTU for LPAR-LPAR communication, resulting in very significant CPU savings and increasing network throughput.

The largesend option using Virtual I/O Server Version 1.3 is available on the SEA when it is used as a bridge device. The largesend option for packets originating from the SEA interface is not available using Virtual I/O Server Version 1.3 (packets coming from the Virtual I/O Server itself).

To use the SEA largesend feature, ensure that the largesend attribute has a value of 1 for all of the virtual adapters of the LPARs.

You can check the largesend option on the SEA using Virtual I/O Server Version 1.3, as shown in Example 5-20. In this example, it is set to off.

Example 5-20 Largesend option for SEA

```
$ lsdev -dev ent6 -attr
attribute   value   description
user_settable

ctl_chan         Control Channel adapter for SEA failover
True
ha_mode     disabled High Availability Mode
True
largesend   0     Enable Hardware Transmit TCP Resegmentation
True
```

```
netaddr     0      Address to ping
True
pvid       1      PVID to use for the SEA device
True
pvid_adapter  ent5     Default virtual adapter to use for
non-VLAN-tagged packets         True
real_adapter ent0   Physical adapter associated with the SEA
True
thread     1     Thread mode enabled (1) or disabled (0)
True
virt_adapters ent5     List of virtual adapters associated with the SEA
(comma separated) True
```

5.8.3 Processor folding (5300-03)

In a micropartitioned environment, the configuration of virtual processors for any given partition can be controlled by the system administrator using the HMC. As such, deciding on the appropriate number of virtual processors to assign to a partition requires some planning and testing because this can affect the performance for both capped and uncapped partitions.

There is processing in the hypervisor associated with the maintenance of online virtual processors, so consider their capacity requirements before choosing values for these attributes. However, AIX 5L Version 5.3 ML3 introduces the processor folding feature to help manage idle virtual processors. The kernel scheduler has been enhanced to dynamically increase and decrease the use of virtual processors in conjunction with the instantaneous load of the partition, as measured by the physical utilization of the partition.

Essentially, the kernel measures the load on the system every second and uses that value to determine whether virtual processors need to be enabled or disabled. The number of enabled virtual processors, and the number that can be enabled in a second, can be tuned using the **schedo** command setting, vpm_xvcpus.

A more detailed explanation is that once every second, AIX 5L calculates the CPU utilization consumed in the last one-second interval. This metric is called p_util. If p_util is greater than 80% of the capacity provided by the number of virtual CPUs currently enabled in the partition, then the value 1 + vpm_xvcpus additional virtual CPUs are enabled. If ceiling value (p_util + vpm_xvcpus) is less than the number of currently enabled virtual CPUs in the partition, then one virtual CPU is disabled. Therefore, each second the kernel can disable, at most, one virtual CPU, and it can enable at most the value 1 + vpm_xvcpus virtual CPUs.

When virtual processors are deactivated, they are not dynamically removed from the partition as with dynamic LPAR. The virtual processor is no longer a candidate to run or receive unbound work. However, it can still run bound jobs. The number of online logical processors and online virtual processors that are visible to the user or applications does not change. There is no impact to the middleware or the applications running on the system because the active and inactive virtual processors are internal to the system.

The default value of the vpm_xvcpus tunable is 0, which signifies that folding is enabled. This means that the virtual processors are being managed. You can use the **schedo** command to modify the vpm_xvcpus tunable.

The following example disables the virtual processor management feature:

```
# schedo -o vpm_xvcpus=-1
Setting vpm_xvcpus to -1
```

To determine whether the virtual processor management feature is enabled, use the following command:

```
# schedo -a | grep vpm_xvcpus
            vpm_xvcpus = -1
```

To increase the number of virtual processors in use by 1, use the following command:

```
# schedo -o vpm_xvcpus=1
Setting vpm_xvcpus to 1
```

Each virtual processor can consume a maximum of one physical processor. The p_util + vpm_xvcpus value is, therefore, rounded up to the next integer.

The following example describes how to calculate the number of virtual processors to use.

Over the last interval, partition A uses two and a half processors. The vpm_xvcpus tunable is set to 1. Using the previous equation:

- Physical CPU utilization = 2.5
- Number of additional virtual processors to enable (vpm_xvcpus) = 1
- Number of virtual processors needed = 2.5 + 1 = 3.5

Rounding up the value that was calculated to the next integer equals 4. Therefore, the number of virtual processors needed on the system is four. So if partition A was running with eight virtual processors and the load remains constant, four virtual processors are disabled (over the next four seconds) and four virtual processors remain enabled. If simultaneous multithreading is enabled, each virtual processor yields two logical processors. So eight logical processors are disabled and eight logical processors are enabled.

In the following example, a modest workload that is running without the folding feature enabled consumes a minimal amount of each virtual processor that is allocated to the partition. The following output using the **mpstat -s** command on a system with four virtual processors indicates the utilization for the virtual processor and the two logical processors that are associated with it:

```
# mpstat -s 1 1

System configuration: lcpu=8 ent=0.5

     Proc0           Proc2           Proc4           Proc6
     19.15%          18.94%          18.87%          19.09%
 cpu0    cpu1    cpu2    cpu3    cpu4    cpu5    cpu6    cpu7
 11.09%   0.07% 10.97%   7.98% 10.93%   7.93% 11.08%   8.00%
```

When the folding feature is enabled, the system calculates the number of virtual processors needed with the preceding equation. The calculated value is then used to decrease the number of virtual processors to what is needed to run the modest workload without degrading performance.

The following output using the **mpstat -s** command on a system with four virtual processors indicates the utilization for the virtual processor and the two logical processors that are associated with it:

```
# mpstat -s 1 1

System configuration: lcpu=8 ent=0.5

     Proc0           Proc2           Proc4           Proc6
     54.63%           0.01%           0.00%           0.08%
 cpu0    cpu1    cpu2    cpu3    cpu4    cpu5    cpu6    cpu7
 38.89%  15.75%   0.00%   0.00%   0.00%   0.00%   0.03%   0.05%
```

As you can determine from this data, the workload benefits from a decrease in utilization and maintenance of ancillary processors, and increased affinity when the work is concentrated on one virtual processor. When the workload is heavy, however, the folding feature does not interfere with the ability to use all of the virtual processors, if needed.

Enabling this feature can be very useful for uncapped partitions because it allows the partition to be configured with a larger number of virtual processors without significant performance issues. As a general rule, we recommend configuring a number of virtual processors that is reasonable to accommodate immediate spikes or rapid short-term growth requirements.

6

Networking and security

This chapter covers the following major topics:

- TCP retransmission granularity change (5300-05)
- Enhanced dead interface detection (5300-05)
- NFS enhancements
- Network intrusion detection (5300-02)
- Radius Server Support (5300-02)
- Web-based System Manager and PAM (5300-05)
- IPFilters open source ported (5300-05)
- Network Data Administration Facility (5330-05)
- AIX Security Expert (5300-05)

6.1 TCP retransmission granularity change (5300-05)

Prior to AIX 5L Version 5.3 with TL 5300-05, the TCP retransmission timer is handled by the tcp_slowtimo() function, which runs every 500 ms. Also, TCP retransmission time-out takes on a minimum value of 3 seconds. With high-speed networks such as gigabit Ethernet and 10 gigabit Ethernet coming, a lower TCP retransmission timer (faster timer) is required to archive a high-speed, low-latency network.

The timer-wheel algorithm has been updated to achieve lower granularity for the retransmission timer in AIX 5L Version 5.3 with TL 5300-05.

6.1.1 Overview of timer-wheel algorithm

Figure 6-1 shows a simplified model of a timer-wheel algorithm.

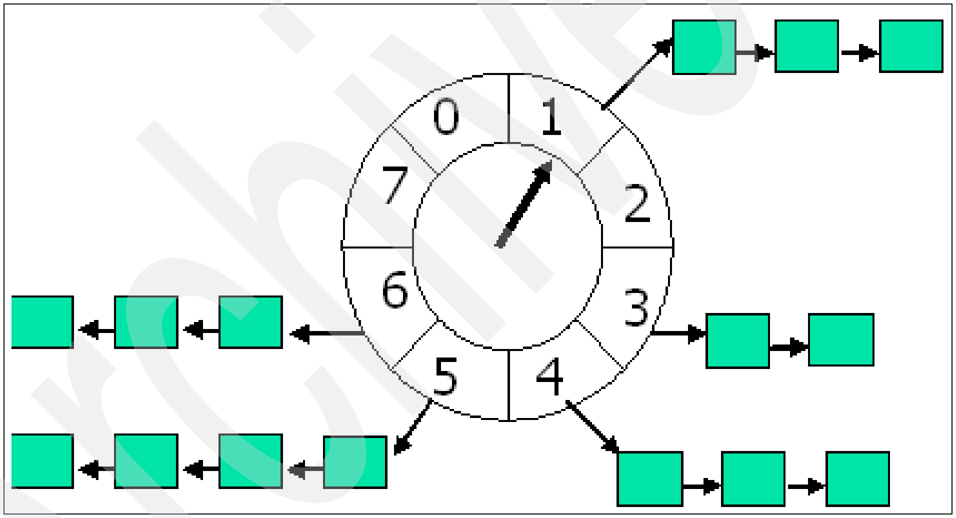


Figure 6-1 Timer-wheel algorithm simplified model

A timer wheel has N number of slots. A slot represents a time unit, named si (slot interval). A curser in the timing wheel moves one location every time unit (just like the seconds hand on a clock). Whenever a curser moves to a slot, named cs (current slot), it implies that the list of timers in that slot, if any, expire at that instant or when the curser reaches the same slot in the subsequent cycles.

When a new timer with timer interval named ti (time interval) is to be added to this wheel, the slot for the new timer named ts (timer slot) is calculated as :

```
ts = ( cs + (ti / si)) % N
```

Assume that the maximum timer interval value that any timer takes on does not exceed an upper limit, named tmax. If N is sufficiently large to accommodate tmax within a rotation from the current curser position, then it would mean that when the curser moves to a specific slot, all timers in that slot expire at the same instant or that there should be no subsequent cycles. This prevents traversing the list to check which of the timers expire now and which of them expire in the subsequent cycles.

The timer wheel has the following attributes:

- The maximum value for RTO is 64 (tmax) seconds.
- The least granularity for the timer is 10 ms.

6.1.2 Options to enable or disable the use of low RTO feature

To support timer-wheel algorithm for TCP's retransmission timer, options are introduced to both the **no** command and the **ifconfig** command.

The no command timer_wheel_tick option

The timer_wheel_tick option specifies the slot interval of the timer wheel in ticks, where a tick=10 ms. By default, timer_wheel_tick is set to 0 (the timer wheel clock does not run). The range of value for timer_wheel_tick is 0–100.

Since the granularity of the system clock is 10 ms, the value specified for timer_wheel_tick must be multiplied by 10 ms to equal the actual slot interval.

> **Note:** If this option is modified, a system reboot is required for the change to take place.

The no command tcp_low_rto option

The tcp_low_rto option, if set, specifies the RTO for all TCP connections that begin to experience packet drops. The range of value set is 0–3000, specified in milliseconds. This range for tcp_low_rto allows the user to configure retransmission time out to a value less than the current least retransmission time out of 3 seconds.

> **Note:** The value set for tcp_low_rto should be equal to or a multiple of (10*timer_wheel_tick). Also, timer_wheel_tick should be set to a non-zero value before setting the tcp_low_rto option.

The ifconfig command tcp_low_rto option

It is interface specific to use the **ifconfig** command to enable low RTO timer:

```
   ifconfig Interface [ tcp_low_rto rto | -tcp_low_rto ]
```

This enables the use of lower retransmission time outs (RTOs) for TCP connections on a low latency, fast network, such as Gigabit Ethernet and 10 Gigabit Ethernet). If the networks experience packet drops, the respective TCP connections use the rto value for RTO. The rto values range from 0–3000 ms. This runtime option must be set in the if_isno flags field. The use_isno option of the **no** command must also be set for this flag to be effective.

6.2 Enhanced dead interface detection (5300-05)

AIX supports multipath routing that enables users to have multiple routes to the same destination (including multiple default routes). The cost (hopcount) is used determine the route to use when there are multiple routes to the same destination. For default routes, AIX 5L provides a facility called Active Dead Gateway Detection, and it will detect if a first-hop gateway is down and modify the routing table to use routes through alternate gateways if they exist.

When there are multiple routes with the same destination and cost, then the route selection depends on the multipath routing policy set by the user defaulting to round-robin. One of the main reasons for having multipath routing was to provide some degree of fault tolerance by having alternate routes to reach a given destination in addition to providing a load balancing feature.

Prior to AIX 5L Version 5.3 with TL 5300-05, the Dead Gateway Detection will not detect if a local network adapter is down (or not working), nor will the multipath routing function. Even when the interface is down it will be trying to send packets through the dead interface. With multipath routing, if you have two interfaces to reach a certain destination network and one interface goes down, this could result in 50% packet loss because the round-robining policy for multipath routing is being used.

In AIX 5L Version 5.3 with TL 5300-05, you can choose to monitor and update the interface status (UP/DOWN) based on the current link status of the underlying network device. Also, route selection will be based on the current interface status in addition to the route status. This dead interface detection is provided by adding options to the **ifconfig** command:

```
   ifconfig Interface monitor
   ifconfig Interface -monitor
```

The monitor flag enables the underlying adapter to notify the interface layer of link status changes. The adapter must support link status callback notification. If multipath routing is used, alternate routes are selected when a link goes down. The -monitor flag disables adapter link status monitoring.

Once monitoring is turned on it will be updated by the interface status based on the underlying adapter link status. All routing entries going through an interface will be marked up/down based on the interface status. If an interface status is changed by using **`ifconfig en0 down`**, the routing entries are updated as well.

6.3 NFS enhancements

In addition to the existing NFS enhancements in AIX 5L Version 5.3, the following features have been added to further extend NFS capabilities

- NFS DIO and CIO support (5300-03)
- NFSv4 replication and global name space (5300-03)
- NFSv4 delegation (5300-03)
- Release-behind-on-read support for NFS mounts (5300-03)
- I/O pacing support for NFS (5300-03)

For detailed descriptions and implementation considerations about the NFS features introduced in the AIX 5L Version 5.3 with 5300-03 release, one can see the IBM Redbooks publication *Implementing NFSv4 in the Enterprise: Planning and Migration Strategies,* SG24-6657, available at:

`http://www.redbooks.ibm.com/redbooks/pdfs/sg246657.pdf`

- NFS server grace period (5300-05)
- NFS server proxy serving (5300-05)

6.3.1 NFS DIO and CIO support

AIX 5L Version 5.3 with the 5300-03 Recommended Maintenance package supports direct I/O and concurrent I/O on the NFS client for both the Version 3 and 4 protocols. The implementation of DIO and CIO only involves the client. Using DIO and CIO, client-side cache is bypassed, reducing processing overhead and memory usage. This feature is useful for applications that do not benefit from client caching such as databases and HPC environments.

Direct I/O for NFS

DIO allows applications to perform reads and writes directly to the NFS server without going through the NFS client caching layer (Virtual Memory Manager) or incurring the associated overhead of caching data.

Under DIO, application I/O requests are serviced using direct Remote Procedure Calls (RPC) to the NFS server. Using the **`mount`** command with the -o dio option sets the DIO option. Without the **`mount`** option, you can also enable DIO per-file by using the AIX O_DIRECT flag in open() call. For more information about DIO, see the -o option for the **`mount`** command.

Concurrent I/O for NFS

With CIO, application reads and writes that are issued concurrently run concurrently without reads blocking for the duration of writes, or the reverse. Multiple writes also run concurrently.

When CIO is in effect, direct I/O is implied. CIO is set either via the AIX **`mount`** command with the **`-o cio`** option, or the O_CIO flag for the open() call is used. For more information about CIO see the -o option for the **`mount`** command.

6.3.2 NFSv4 global name space and replication

AIX 5L Version 5.3 with the 5300-03 Recommended Maintenance package supports NFSv4 Replication and Global Name Space functionality, which is compliant with NFSv4 protocol (RFC 3530).

NFSv4 global namespace

The global namespace feature is also known as *referral*. Referral uses a special object created in the namespace of a server to which location information is attached. The server uses this to redirect clients to the server specified in the location information. The referral forms a building block for integrating data on multiple NFS servers into a single file namespace tree, which referral-aware NFSv4 Clients can navigate. A referral should point to the root of an exported file system for AIX, which relates to an NFSv4 FSID. Referrals can list multiple locations if the data is copied on several servers.

A referral can be created using the -refer option of the **`exportfs -o`** command. A referral export can only be made if replication is enabled on the server. Use the **`chnfs -R on`** command to enable replication.

For more information about referral, see the -o option of the **`exportfs`** command.

NFSV4 replication

Replication allows the hosting of data on multiple NFSv4 servers. Multiple copies of the same data placed on different servers are known as replicas, while servers holding copies of data are known as replica servers. The unit of replication used in NFSv4 replication is a mounted file system. The NFSv4 server communicates a location data list to all NFSv4 clients. Thus, the AIX 5L NFSv4 client becomes replica aware. In the event of primary data server failure, the client switches to an alternate location. The location list from the server is assumed to be ordered with the first location the most preferred. The client may override this with the **nfs4cl prefer** command. The default client fail-over behavior is governed by the timeo NFS mount option. If the client cannot contact the server in two NFS time-out periods, it will initiate failover processing to find another server. Failover processing can be influenced with the nfso replica_failover_time option.

A replica export can only be made if replication is enabled on the server. By default, replication is not enabled. The **chnfs** command can be used to enable or disable replication.

```
#chnfs -R {on|off|host[+hosts]}
```

Changing the replication mode can only be done if no NFSv4 exports are active. Replicas can be exported using the -replica option to the **exportfs -o** command.

6.3.3 NFSv4 delegation

Delegation is the ability of a server to delegate certain responsibilities to the client. When the server grants a delegation for a file to a client, the client is guaranteed certain semantics with respect to sharing that file with other clients. Delegation allows a client to open and lock a file without making server calls. The client obtains delegation during initial access (open), then caches the data. Then the client can locally service operations such as OPEN, CLOSE, LOCK, LOCKU, READ, and WRITE without immediate interaction with the server. Delegation is enabled by default for both the client and the server.

Server delegation

The AIX server supports read delegation and is available with the 64-bit AIX kernel. At the server, delegation can be controlled globally with the **nfso** command (server_delegation), or per export with the new deleg option of the **exportfs** command.

Server delegation can be disabled with the **nfso -o server_delegation=0** command. Administrators can use the **exportfs deleg=yes | no** option to disable or enable the granting of delegations on a per- file system basis, which will override the **nfso** command setting.

Client delegation

The AIX 5L client supports both read and write delegation on both the 32-bit and 64-bit kernel. Client delegation can be disabled with the **nfso -o client_delegation=0** command. If client delegation is to be disabled, the client_delegation **nfso** option should be set to 0 before any mounts take place on the client.

All delegation statistics can be extracted with the **nfsstat -d** command.

6.3.4 Release-behind-on-read support for NFS mounts

Data caching for sequential reads of large files might result in heavy page replacement activity as memory is filled with the NFS data cache. Performance can be improved by avoiding the page replacement activity by using the release-behind-on-read option during **mount** for NFSv2, NFSv3, or NFSv4 or **nfs4cl setfsoptions** for NFSv4. The rbr option is used for this as follows.

```
#mount -r rbr
or
#nfs4cl setoptions rbr
```

For sequential reads of large files, the real memory for the previous reads is then freed as the sequential reads continue.

> **Note:** To avoid the release of memory that is going to be needed again, the nfs_auto_rbr_trigger tunable of the **nfso** command can be used.

6.3.5 I/O pacing support for NFS

I/O pacing support for NFS is introduced in AIX 5L Version 5.3 with the 5300-03 maintenance package. I/O pacing is used to prevent a large number of I/O page outputs for one file from monopolizing the I/O resources. I/O pacing can be tuned on a per-file system basis. This tuning is set using the **mount** command:

```
#mount -o minpout=40, maxpout=60 /nfs
or
#nfs4cl setfsoptions /nfs minpout=40, maxpout=60
```

6.3.6 NFS server grace period

The grace period is disabled by default. To enable the grace period on the server, the **chnfs** command line interface can be used. See Table 6-1.

Table 6-1 NFS commands to change the server grace period

Command	Description
chnfs -g on | off	Controls the NFSv4 Grace Period enablement. The possible values are on or off. When the no command -g option is specified, the grace period is disabled by default.
chnfs -G	Controls the NFSv4 Grace Period bypass. When this option is specified, the grace period will be bypassed regardless of how the -g option is specified.
chnfs -x extend_cnt	Controls the NFSv4 Grace Period automatic extension. The extend_cnt parameter specifies the total number of automatic extensions allowed for the grace period. If no -x option is specified, the number of allowed automatic extensions defaults to 1. A single extension cannot extend the grace period for more than the length of the NFSv4 lease period.

The NFSv4 subsystem uses runtime metrics (such as the time of the last successful NFSv4 reclaim operation) to detect reclamation of the state in progress, and extends the grace period for a length of time up to the duration of the given number of iterations.

6.3.7 NFS server proxy serving

An AIX NFS server can be configured to export a view of another NFS server's data (Figure 6-2).

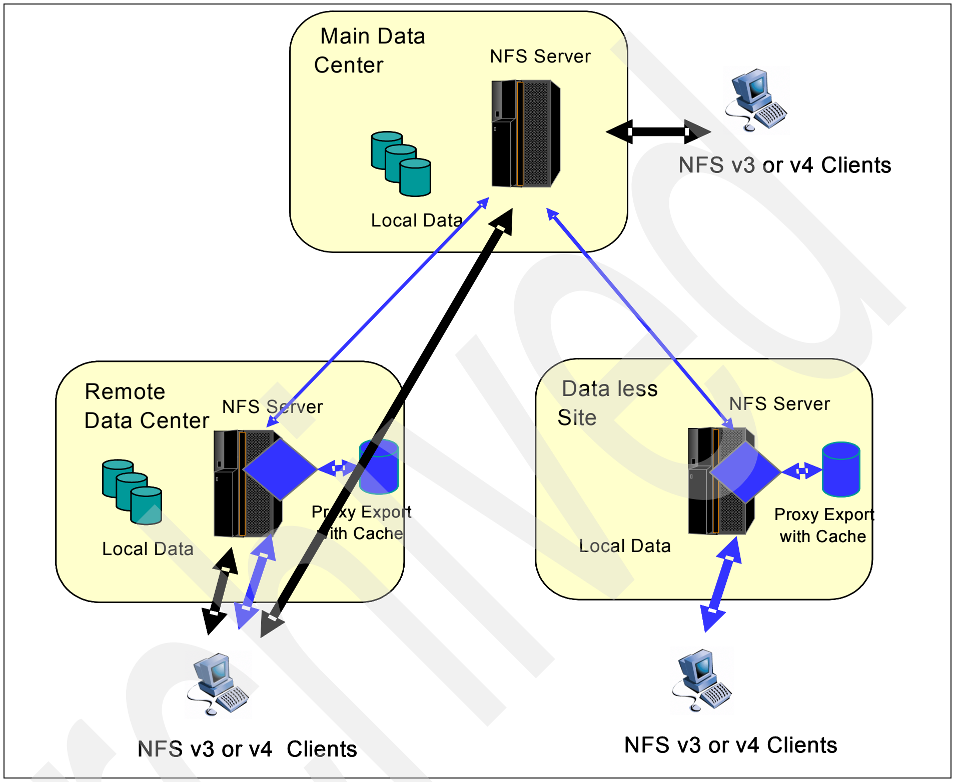


Figure 6-2 NFS server proxy serving

NFS proxy serving uses disk caching of accessed data to serve similar subsequent requests locally with reduced network traffic to the back-end server. Proxy serving can potentially extend NFS data access over slower or less reliable networks with improved performance and reduced network traffic to the primary server where the data resides. Depending on availability and content management requirements, proxy serving can provide a solution for extending NFS access to network edges without the need for copying data. You can configure NFS proxy serving using the **mknfsproxy** command.

Proxy caching can be used with both the NFS version 3 and 4 protocols. The protocol between the proxy and the connected clients must match the protocol used between the proxy and the back-end server for each exported proxy instance. Both reads and writes of data are supported in addition to byte range advisory locks. Proxy serving requires the 64-bit kernel environment. Enhanced JFS file system (JFS2) is used as the cached file system with proxy serving.

6.4 Network intrusion detection (5300-02)

AIX 5L provides powerful tools to detect and prevent network intrusions. Intrusion detection is the action of monitoring and analyzing system events in order to intercept and reject any attempted unauthorized system access. In AIX 5L, this detection of unauthorized access or attempted unauthorized access is done by observing certain actions, and then applying filter rules to these actions. The available filtering options have been expanded to include stateful filters.

6.4.1 Stateful filters

Stateful filters are designed to make filtering decisions based on an entire session rather than just an individual packet and its header information. They examine information within headers such as source and destination addresses, port numbers, and status. Then, by applying IF, ELSE, or ENDIF filter rules to these header flags, stateful systems can make decisions in the context of the session as a whole rather than on a packet-by-packet basis. Stateful inspection can examine both incoming and outgoing communication packets. When stateful filter rules are activated with the **mkfilt -u** command, the rules in the ELSE block are always examined until the IF rule is satisfied. After the IF rule or condition is satisfied, the rules in the IF block are used until the filter rules are reactivated with the **mkfilt -u** command. As well as the introduction of stateful filters, a new command has been added, the **ckfilt** command, to check the filters for consistency.

The ckfilt command

The stateful filter rules allow for extended actions such as IF, ELSE, and ENDIF. Thus, it is possible to have syntax errors in the rules set, such as an IF without an ENDIF, or an ELSE or ENDIF without a preceding IF. The **ckfilt** command checks for such errors. Nesting of IF rules is permitted. The **ckfilt** command displays the filter rules, indenting the rules within IF statements in a scoping fashion. If the -O flag is used, filter rules and all of their attributes are displayed in a scoped fashion. IPsec filter rules for this command can be configured using the **genfilt** command, **smit**, or Web-based System Manager in the Virtual Private Network submenu. The syntax of the command is as follows:

```
ckfilt [ -O ] [ -v 4 | 6 ]
```

The -v flag specifies whether IPv4 or IPv6 are to be used. When run, the command will produce output similar to that shown in Example 6-1. Here dummy rules have been used in order to demonstrate the format.

Example 6-1 The ckfilt command output

```
%ckfilt -v4
Beginning of IPv4 filter rules.
Rule 2
IF Rule 3
    IF Rule 4
        Rule 5
    ELSE Rule 6
        Rule 7
    ENDIF Rule 8
ELSE Rule 9
    Rule 10
ENDIF Rule 11
Rule 0
```

6.5 Radius Server Support (5300-02)

The Remote Authentication Dial-In User Service (RADIUS) server implements a client-server protocol, based on the Internet Engineering Task Force (IETF) Request for Comments (RFCs) 2865 and 2866, that enables remote access clients to communicate with a central server to gain access to a network. The RADIUS server authenticates users, authorizes their requests to services, and writes accounting data. The initial release for the RADIUS server is AIX 5L Version 5.3.0.10.

Typical clients in a RADIUS environment are a terminal server, authenticating LAN device, or wireless access point.

The RADIUS server consists of three services, managed by a monitor process:

- Authentication
- Authorization
- Accounting

Primary means that these daemons are always loaded when the RADIUS server is functioning properly. The processes are running at an effective user ID of radiusd and do not have permanent root authority. You can stop and start the processes using the SRC master commands:

```
startsrc -s radiusd
stopsrc -s radiusd
```

To view whether the processes are loaded, you can run the **`ps -ef | grep radiusd`** command.

6.5.1 Monitor process

The monitor process is a management process that starts or re-starts all other processes associated with the RADIUS server. At startup, the monitor process reads and processes all of the configuration files (for example, radiusd.conf, clients, proxy, dictionary, default.policy). The process also starts the communications to the syslog daemon for logging purposes. Services to the RADIUS server are processes that provide network services. The RADIUS server can start and manage two services:

- RADIUS authentication
- RADIUS accounting

The monitor process can start 1 to *N* instances of each service. Each instance of a service listens on a unique network port. The default configured port for authentication is the standard defined port 1812. The port for accounting is 1813. You can define additional ports in the main RADIUS configuration file, radiusd.conf. Each port listed in the radiusd.conf file starts a service listening on that port number.

Service instances might be of the same service type or of different types. For example, the monitor process might start three instances of the authentication service and two instances of the accounting service.

Remember, each instance of a service must run on a unique port.

After all services are available and functioning, the monitor process watches the services processes. If a service process should fail or abort, the monitor process logs the event and restarts the service.

> **Note:** If you refresh the RADIUS server frequently, you can cause a diminishment of services.

6.5.2 Authentication process

The general role of the authentication process is to respond to access requests from terminal servers, authenticating LAN devices, or wireless access points in one of the following ways:

- Granting access
- Denying access
- Challenging access and requiring additional information
- Validating authorization policies
- Replying with authorization information

Terminal servers manage and proxy remote access devices (for example, modems) for the network. If the authentication process grants access, the remote device attaches and becomes a part of the existing network. If access is denied, then the remote device can try to authenticate again or disconnect.

6.5.3 Accounting process

The purpose of the accounting process is to write accounting data about usage. The terminal server can be configured to generate RADIUS accounting information once a client has been authenticated. The RADIUS accounting consists of two message types generated by the terminal server, as follows:

- Start accounting message
- Stop accounting message

6.5.4 Software requirements for Radius Support

Installation of the AIX Remote Authentication Dial-In User Service (RADIUS) server requires the following software:

- AIX 5L Version 5.3.0.10 or later
- radius.base
- bos.msg.LANG.rte
- bos.help.msg.en_US.smit

To find the radius.base fileset you can apply APAR IY65978.

6.5.5 RADIUS IP address pooling (5300-03)

Previously for authorization, the server could send back attributes on what services the user could use once he was authenticated. The attributes are usually attribute=value pairs. One of these attributes is usually an IP address. This IP address was a static address, but the RADIUS Server has changed to access a DHCP server to obtain a dynamic IP address to return to the client.

6.5.6 RADIUS enhanced shared secret (5300-03)

The underlying security in the RADIUS client/server architecture is based on a shared secret that is never passed on the wire. In previous versions there was only support for a 64-byte ASCII shared secret. Pprintable characters, now the length of this data, have been increased to allow a maximum of 256 bytes. Also, it allows hex data as a shared secret.

For more information see *RADIUS Server in AIX 5L Versions 5.3 Security* on the following Web site:

http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=/com.ibm.aix.doc/aixbman/security-kickoff.htm

6.6 Web-based System Manager and PAM (5300-05)

Starting with AIX 5L 5300-05, if Pluggable authentication module (PAM) is configured on the system, then Web-based System Manager will use PAM services for authentication. Otherwise, a traditional authentication mechanism will be used.

You can enable PAM support on systems by setting auth_type to PAM_AUTH in the /etc/security/login.cfg file. When a user tries to log on through the Web-based System Manager's login panel, the login authorization is validated using PAM libraries. This authentication mechanism is transparent to the user. Setting the auth_type to STD_AUTH enables standard AIX authentication for Web-based System Manager logins. The example has the relevant text of the login.cfg file that shows the auth_type field set to PAM_AUTH, as shown in Example 6-2.

Example 6-2 The login.cfg file

```
maxlogins = 32767
logintimeout = 60
auth_type = PAM_AUTH
```

6.7 IPFilters open source ported (5300-05)

IPFilter is a software package that can be used to provide network address translation (NAT) or firewall services. IPFilter Version 4.1.13 open source software, has been ported to AIX 5L, consistent with the licensing presented on the IP Filter Web site, whose URL is:

`http://coombs.anu.edu.au/~avalon/`

6.8 Network Data Administration Facility (5330-05)

AIX 5L Network Data Administration Facility (NDAF) is a new component of AIX 5L that provides secure centralized management of NFS V4 distributed file systems including data placement, replication, and data and namespace administration. Its primary purpose is to facilitate the provisioning of data on NFSv4 servers and the creation of a single NFSv4 exported file namespace that spans multiple server systems. NDAF resides on the AIX 5L 5.3 Expansion Pack media.

NFS clients that support NFS V4 protocol features for replication and data referrals can perform a single mount to a server in the NDAF managed environment in order to access the data available at all the participating servers. Additionally, these clients can use read-only replicas to provide higher availability and load distribution.

6.8.1 NDAF concepts

NDAF consists of several components. They include a domain where the network servers reside, a centralized server that controls the collection of data servers, and a directory tree of file system objects for management purposes.

Physical file system

A physical file system (PFS) is a file system that can access attached disk storage on a system and can be exported by an NFS server.

NDAF domain

An NDAF domain consists of one or more administration clients (systems from which an administrator can control the NDAF environment through the **`dmf`** command), one or more NDAF-enabled NFS servers, and potentially, one or more non-NDAF enabled NFS servers grouped around an NDAF administration server.

All systems in the NDAF domain share the same user and group definitions. For example, if NDAF is deployed using Kerberos security, all systems in the domain are members of the same Kerberos realm. The NDAF domain and the NFSv4 domain must be the same domain.

In an NDAF domain, the NDAF administration server receives its process information from commands run by one or more system administrators over a command-line interface (CLI). The NDAF administration server initiates all NDAF actions at the NFS data server systems that are part of the domain.

Data set

The basic unit of NDAF management is a data set. A data set is a directory tree. NDAF creates data sets and manages their attributes, mounting, and contained data.

Data sets, also called dsets, are manageable units of data that can be shared across a network. They provide basic management for replicating network data and inserting it into a namespace. They are linked together using data set references to form the file namespace. NDAF supports thousands of data sets or replicas across all managed servers. Read-only replicas of a data set can be created, distributed, and maintained across multiple data servers. When the master source for a collection of replicas is modified, the **`dmf update replica`** command is used to propagate the changes to all replica locations. A given data set can have as many as 24 replicas.

Data that is copied into data set directories is NFS exported, but the data is not visible until the data set is mounted with the **`dmf mount dset`** command. File system objects in a data set include files, directories, access control lists (ACLs), links, and more.

Unless specified when they are created, data sets are created in the directory specified when the dms daemon is started by the -ndaf_dataset_default parameter or, if unspecified, the -ndaf_dir parameter.

Cell

Data sets can be grouped with other data sets and organized into a single file namespace. This grouping is called a cell.

A cell is a unit of management and namespace that is hosted by an administration server. After a cell is defined on an administration server, more data sets can be created on that server using that cell. Each cell in an administration server is independent of all other cells hosted by that administration server. A cell contains its own namespace, consisting of data sets, and its own role-based security objects. Roles are privileges attached to a set of users that manage the resources within a cell. As many as eight distinct roles can be defined for each cell.

After a cell is created using the **dmf create cell *name*** command, it is automatically placed on the administration server. You cannot use the **dmf place cell** name to place a cell on the administration server. Placing a cell results in the transfer of the cell's root directory information from the administration server to the targeted data server. A cell can place its data sets on any server defined for the administration server on which the cell is hosted. NFSv4 clients mount the root directory of the cell to access the cell's full namespace.

All NFSv4 clients can view the objects within a cell by mounting the root of the cell from any NDAF server on which the cell has been placed.

NDAF supports up to 64 cells for every deployed NDAF instance (domain) that has cells residing on one or more data servers. When a cell is destroyed, all of its data sets and replicas are also destroyed.

Replicas

These read-only data sets are called replicas. A replica is placed in the global namespace in the same way as a data set. Multiple replicas of the same data set can be placed on different servers so that if the primary server of a replica becomes unavailable to the client, the client can automatically access the same files from a different server. Replicas will not reflect updates that are made to the data set unless the replica is updated using the **dmf update replica** command. A given data set can have as many as 24 replicas.

Unless specified when they are created, replicas are created in the directory specified when the dms daemon is started by the -ndaf_replica_default parameter or, if unspecified, the -ndaf_dir parameter.

Master replica location

The master replication location is the place where the replica was first created, and it is the first location updated on any update action request. The other replica locations are updated afterwards asynchronously.

You can change the master location to another replica location using the master action request.

A master replica can never be unplaced before another master replica location is defined as a replacement for the first location.

Replica clones

For replicas, the **dmf place replica** command creates a clone of the replica at a specified location on the server.

If the replica is mounted in the cell, this clone location is added to the NFS replica list that is returned to the NFS clients that are accessing the replica. The order of the referrals in this list depends on the network configuration. Every clone location of a replica is updated asynchronously upon **`dmf`** update commands. The **`dmf place replica`** command takes as parameters the server and, optionally, the local path on the server.

A clone location of a replica can be removed from a server, as in the following example:

```
dmf unplace replica my_server local_path -a my_admin -c my_cell
-o my_replica
```

In this example, *my_server* is the name of the server on which the clone resides and *my_replica* is the name of the replica. The clone location is unexported, and its content is destroyed. This location is also removed from the file systems locations data list returned by NFSv4 for this replica in the cell. The other locations of the replica remain the same. The **`dmf update replica`** command updates clones along with their original replicas to be refreshed with the content of the original source data set.

Replication updates

The master replica is a read-only copy of the source data set, and the clones are copies of the master replica. If the source data set is updated, the replicas are not updated until explicitly done so using the dmf update replica command.

There are two methods of data transfer:

copy Performs data transfer using full file tree copy. The copy method implements the data transfer method plugin interface and performs a data transfer operation by doing a complete walk of the directory tree for the data set and transmitting all objects and data to the target.

rsync Performs data transfer using rsync-like algorithm. The rsync method performs a data transfer operation by doing a complete walk of the directory tree for the data set and transmitting only deltas for directories and data to the target. It is beneficial when updating replicas because it only sends changed blocks of information, so it reduces network bandwidth considerably.

Administration client

An administration client is any system in the network that has the ndaf.base.client file set installed on it from which the **`dmf`** command can be run.

The NDAF administration server receives its process information from commands run by system administrators over a command-line interface. The program name for this administration client is the **dmf** command.

Administration server

The NDAF administration server is a data server that runs daemon processes and acts as the central point of control for the collection of NDAF data servers.

It receives commands from the system administrators who use the administration client (the **dmf** command), the **dms** command, and the **dmadm** command. The NDAF administration server maintains a master database of configuration information and sends commands to the data servers. When a loss of communication occurs between the administration server and the data servers, the NDAF-managed data already present on the data server can still be accessed. After network connectivity is restored, the transactions between the systems are eventually completed.

The administration server is configured before all other NDAF components. This server requires 64-bit systems running the AIX 5L 64-bit kernel.

Administration server databases are created in an admin subdirectory in the directory specified by the -ndaf_dir parameter when the dmadm daemon is started.

Data server

A data server is the server that runs the daemon process (the dms daemon) controlling an NFS file server and its associated physical file system (PFS). The data provisioned with NDAF resides at the data server.

The dms process runs at each data server and carries out actions in underlying file systems. It sets default directories, timeout values, level of logging, security method used, Kerberos keytab path, Kerberos principal, and communication ports on each data server. Only data servers within the NDAF domain can be replicated.

Data server databases are created in a server subdirectory in the directory specified by the -ndaf_dir parameter when the **dms** daemon is started. These servers require 64-bit systems running the AIX 5L 64-bit kernel. The administration server also serves as a data server.

Principal

A principal is an authorized NDAF user that Kerberos and other security methods panel for during security checks.

Principals control how objects can be manipulated and by which operations.

Only the first user to run the **dmf create admin** command, called the DmPrincipal, can create cells, servers, and roles. Additional NDAF principals can be added to manage an object with the **dmf add_to object DmPrincipal=login** command. All members of the DmPrincipal list are considered to be owners of the object and can control it.

NDAF principals can also be removed using the **dmf remove_from** action.

6.8.2 Graphical representation of an NDAF domain

The basic concepts of the functioning objects of an NDAF domain can be depicted graphically. Figure 6-3 shows the organization of various objects within NDAF.

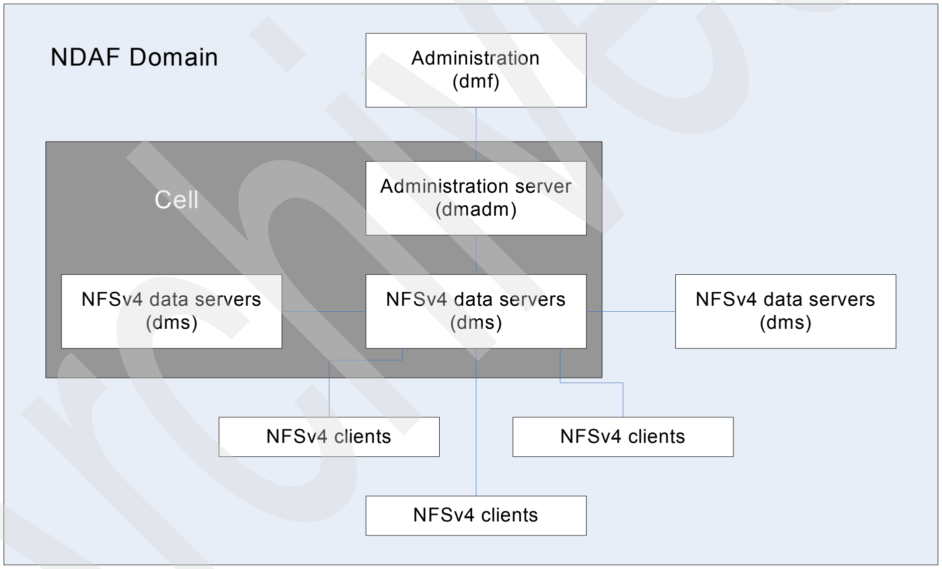


Figure 6-3 The NDAF domain

6.8.3 NDAF commands

The four primary commands that NDAF uses to perform its operations are **dmf**, **dms**, **dmadm**, and **dms_enable_fs**.

The dmf command

In NDAF command strings, **dmf** is the prefix for all command-line interface commands. The **dmf** command implements the AIX NDAF administration client executable.

The **dmf** command (/usr/sbin/dmf) has the following syntax:

```
dmf repair server  [-a admin_server] [-c server] [params]
```

The **dmf** command takes one or more of the following optional parameter values:

-V Verify. Results in a report on all data set or replica state data found with problems.

-U Unexport. Results in a report. Also, all bad data sets or replicas are unexported through NFS.

-R Repair. Results in a report. Also, all bad data sets or replicas are unexported and then recovery is invoked to try to restore the state data to working order. If successful, the location is then re-exported.

The dmadm command

The **dmadm** command operates NDAF on the administration server. Both the **dmadm** and **dms** commands are required services on the administration server's Kerberos keytab. The dms processes must be launched along with the dmadm daemons for the administration machine to function correctly.

With corresponding parameters, the **dmadm** command sets default directories, timeout values, level of logging, security method used, Kerberos keytab path, Kerberos principal, and communication ports on the admin server within an NDAF domain.

The **dmadm** command takes one of the following optional parameter values:

[-rpc_timeout=val] Sets the timeout for an RPC connect or call. Default is 300 seconds.

[-log_level=val] Sets the level of logging for the log files. Default is 0. Possible values include the following:

0	Critical errors
1	Errors
2	Warning
3	Notice
4	Information

[-security=val] Sets the type of security method used. The default is krb5. Values include:

- **auth_sys** For uid/gid authentication
- **krb5** For Kerberos authentication
- **krb5i** F or Kerberos integrity authentication
- **krb5p** For Kerberos privacy authentication

[-krb5_principal=val] Sets the Kerberos principal used for the kinit.

[-admin_port=val] Sets the dmadm port waiting for RPC of the dmf client. Default value is 28000.

[-serv_port=val] Sets the dms port waiting for the dmadm RPC. Default value is 28001.

[-ndaf_dir=val] Sets the base directory for NDAF. It contains default databases, logs, and directories for cells, dsets, and replicas. The default for the base directory is /var/dmf. Other defaults include the following directories:

${ndaf_dir}/var/dmf/log for logs

${ndaf_dir}/var/dmf/admin for admin databases

[-krb5_keytab=val] Indicates the Kerberos keytab path. If not specified without SRC, uses the KRB5_KTNAME variable if positioned. Otherwise, uses the default keytab file specified in the /etc/krb5/krb5.conf file. If not specified with SRC, uses the default keytab file specified in the /etc/krb5/krb5.conf file.

[-admin_cb_port=val] Sets the dmadm port waiting for the dms RPC callbacks. The default is 28002.

To start dmadm using SRC on the admin server, enter:

```
startsrc -s dmadm
```

To start dmadm using SRC and specifying auth_sys security, enter:

```
startsrc -a "-security=auth_sys" -s dmadm
```

The dms command

The **dms** command operates NDAF on a data server.

With corresponding parameters, the **dms** command sets default directories, timeout values, level of logging, security method used, Kerberos keytab path, Kerberos principal, and communication ports on a data server within an NDAF domain.

The **dms** command takes one of the following optional parameter values:

[**-rpc_timeout=val]** Sets the timeout for an RPC connect or call. Default is 300 seconds.

[-log_level=val] Sets the level of logging for the log files. Default is 0. Possible values include the following:

- **0** Critical errors
- **1** Errors
- **2** Warning
- **3** Notice
- **4** Information

[-security=val] Sets the type of security method used. The default is krb5. Values include:

- **auth_sys** For uid/gid authentication
- **krb5** For Kerberos authentication
- **krb5i** For Kerberos integrity authentication
- **krb5p** For Kerberos privacy authentication

[-krb5_principal=val] Sets the Kerberos principal used for the kinit.

[-serv_port=val] Sets the dms port waiting for the dmadm RPC. Default value is 28001.

[-serv_serv_port=val] Sets the dms port waiting for the other dms RPC. Default value is 28003.

[-ndaf_dir=val] Sets the base directory for NDAF. It contains default databases, logs, and directories for cells, dsets, and replicas. The default for the base directory is /var/dmf. Other defaults include the following directories:

${ndaf_dir}/log for logs

${ndaf_dir}/serverfor data server databases

${ndaf_dir}/server/dsetsfor dsets, if the -ndaf_dataset_default parameter is not set

${ndaf_dir}/server/replicasfor replicas, if the -ndaf_replica_default parameter is not set

Note: At least either the **-ndaf_dataset**_default and **-ndaf_replica_default** parameters, or the **-ndaf_dir** parameter, have to be specified. The creation of cells, data sets, and replicas must have been enabled using the **dms_enable_fs** command on the file systems containing the specified directories to store the datasets and replicas.

[-ndaf_dataset_default=val] Sets the default directory for dsets.

[-ndaf_replica_default=val] Sets the default directory for replicas.

[-krb5_keytab=val] Indicates the Kerberos keytab path. If not specified without SRC, uses the KRB5_KTNAME variable if positioned. Otherwise, uses the default keytab file specified in the /etc/krb5/krb5.conf file. If not specified with SRC, uses the default keytab file specified in the /etc/krb5/krb5.conf file.

[-admin_cb_port=val] Sets the dmadm port waiting for the dms RPC callbacks. The default is 28002.

To start dms using SRC on a dataset server, enter:

```
startsrc -s dms
```

To start dms using SRC and specifying auth_sys security, enter:

```
startsrc -a "-security=auth_sys" -s dms
```

The dms_enable_fs command

The **dms_enable_fs** command enables, disables, or queries the capability to create cells, data sets, and replicas on a file system.

The **dms_enable_fs** command (/usr/sbin/dms_enable_fs) has the following syntax:

```
dms_enable_fs [-sqh] pathname
```

The **dms_enable_fs** command enables, disables, or queries the capability to create cells, data sets, and replicas on a file system. It generates the .DSETINFO directory in the root of the file system. This directory must not be deleted.

The command has the following flags:

-h Displays usage of the **dms_enable_fs** command.

-q Checks to see whether VFS (path name within VFS) is enabled. If it is, 0 is returned. Otherwise, a nonzero value is returned.

-s Enables a VFS (path name of VFS) for filesets.

To enable the /ndafexp file system for datasets, enter:

```
dms_enable_fs -s /ndafexp
```

Additional NDAF commands

NDAF uses three configuration commands to prepare systems for running its processes. The **mkndaf** command configures the system to run NDAF. The **chndaf** command changes various parameter settings used by the **dms** command and **dmadm** command. The **rmndaf** command configures the system to stop running NDAF daemons. In addition to these three configuration commands, an additional **dmf** command verb for recovering data is defined in this section.

mkndaf The **mkndaf** command configures the system to run AIX NDAF.

rmndaf The **rmndaf** command changes the configuration of the system to stop running the AIX NDAF daemons.

chndaf The **chndaf** command changes the configuration of the AIX Network Data Administration Facility (NDAF).

dmf The **dmf** command implements the AIX NDAF administration client executable.

6.8.4 NDAF security

NDAF systems, including the point of administration, can use strong security based on Kerberos and open network computing (ONC™) remote procedure call (RPC) with RPCSEC-GSS for authentication when communicating with each other.

RPCSEC_GSS is a security method that can optionally be applied to ONC RPC. RPCSEC-GSS is a protocol that applies Generic Security Services (GSS) to RPC.

Security for NFS file access

To manage access to NFS clients, the NFSv4 access control list (ACL) method can be used for directories containing data sets, at file system level, and on the mounting point (nfsroot) of each server.

Exporting with Kerberos

By default, the data servers only export file systems for NFSv4 access with all security types allowed.

Roles

Roles are privileges attached to a set of NDAF principals for managing the resources within a cell. NDAF roles are a distinct function separate from AIX administrative roles.

> **Note:** For more information and examples about NDAF security, refer to the *IBM AIX Information Center* publication on NDAF at:
>
> `http://publib.boulder.ibm.com/infocenter/pseries/v5r3/topic/com.ibm.aix.ndaf/doc/ndaf/NDAF_security.htm`

6.8.5 NDAF installation and configuration

You can install NDAF by using SMIT (fastpath: **`smitty install_all`**) or the **`installp`** command to install the ndaf.base file set. If the recommended Kerberos 5 security is required, install the krb5.client file set.

Installing NDAF

To install NDAF, the system must have IBM AIX 5L Version 5.3 with the 5300-05 Technology Level or later installed. The system must be using the 64-bit kernel.

NDAF servers must be configured as NFSv4 servers.

The NDAF administration server and all NDAF data servers and administration clients must be configured as Kerberos clients.

In order to communicate correctly, each server must be aware of the ports the other servers are listening on and emitting to.

A given system can assume one of three roles in an NDAF domain. Different pieces of the ndaf.base file set must be installed depending on the roles. The roles are:

Administration server For this system, ndaf.base.admin and ndaf.base.server must be installed. There is only one administration server for a federation of servers.

Data servers For these systems, ndaf.base.server must be installed.

Administration clients For these systems, only ndaf.base.client must be installed.

For information about the command and flags, see the **installp** command in the *AIX 5L Version 5.3 Commands Reference*.

> **Note:** By default, the NDAF daemons are not started after installing the package. They are also not configured to be started automatically on the next system boot.

Configuring an NDAF data server

The primary configuration for an NDAF data server involves starting the dms daemon.

Configuring the NDAF administration server

When configuring NDAF, you must first configure the administration server. There is one NDAF administration server for a given federation of NDAF data servers.

Configuring an NDAF administration client

An NDAF administration client is any system that is used to run data-management commands that are handled by the NDAF administration server.

> **Note:** For more information and examples about NDAF installation and configuration, refer to the *IBM AIX Information Center* publication on NDAF at:
>
> http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=/com.ibm.aix.ndaf/doc/ndaf/NDAF_installation_and_configuration.htm

6.8.6 Managing NDAF

You must add the NDAF server and the data servers before you can perform other NDAF management tasks. You can then create and manage cells, roles, data sets, and replicas; populate data sets; construct cell namespaces; and federate servers without the NDAF into an NDAF environment.

Adding servers for NDAF

You must add the NDAF administration server and all of the data servers to the system before you can perform any other NDAF management tasks.

You can use the **dmf** command to add the NDAF administration server:

```
dmf create admin name [-r] [-a admin_server]
```

The flags have the following definitions:

-a admin_server Specifies the Domain Name System (DNS) name or IP address of the administration server. The port can be added using a colon separator.

name Specifies the name for the administration server to be created.

-r Prints the universally unique identifier (uuid) assigned to the request.

> **Note:** Entering `dmf create admin my_admin` also creates the my_admin server object.

You can use the **`dmf`** command to add a data server:

```
dmf create server name dns_target [-e] [-r] [-a admin_server]
```

The flags have the following definitions:

-a admin_server Specifies the DNS name or IP address of the administration server. The port can be added using a colon separator.

dns_target Specifies the DNS name or IP address of the server. The port can be added using a colon separator.

-e Specifies that the object is external to NDAF.

name Specifies the name for the data server to be created.

-r Prints the uuid assigned to the request.

To add the NDAF administration server using SMIT, perform the following steps:

1. From the NDAF menu, select **NDAF Management** → **Administration Server Management** → **Create Admin Server**.

   > **Note:** You can also use the ndafadmincreate fastpath.

2. Specify the DNS name or IP address of the administration server in the Admin Server DNS name field and press Enter.
3. Specify a name for the new administration server in the Admin Server name field and press Enter.

To add an NDAF data server using SMIT, perform the following steps:

1. From the NDAF menu, select **NDAF Management** → **Data Server Management** → **Create Data Server**.

Note: You can also use the ndafdscreate fastpath.

2. Specify the DNS name or IP address of the administration server in the Admin Server DNS name field.
3. Specify a name for the new data server in the Data Server name field and press Enter.
4. Specify the DNS name or IP address of the new data server in the Data Server DNS name field and press Enter.

Creating and managing cells

A cell is a collection of data sets organized into a single file namespace for use with NFSv4 servers. NFSv4 clients are expected to mount the root directory of the cell to access the cell's full namespace.

Creating a cell

You can create a cell for use with NFSv4 servers. You must create an NDAF administration server before you can create a cell.

You can use the **dmf** command to create a cell:

```
dmf create cell name  [-w timeout] [-r] [-a admin_server]
```

The flags have the following definition:

-a admin_server Specifies the DNS name or IP address of the administration server. The port can be added using a colon separator.

name Specifies the name for the cell to be created.

-r Prints the uuid assigned to the request.

-w timeout Specifies how long the command can wait before completing.

Perform the following steps to create a cell using SMIT:

1. From the NDAF menu, select **NDAF Management** → **Namespace (cell) Management** → **Create cell namespace**. You can also use the ndafcellcreate fastpath.
2. Specify the DNS name or IP address of the administration server that manages the NDAF domain in the Admin name field.
3. Enter a name for the new cell in the Cell name field and press Enter.

For example, to create a cell named cell1 in the NDAF domain that is managed by the NDAFServer1 administration server, perform the following steps:

1. From the NDAF menu, select **NDAF Management** → **Namespace (cell) Management** → **Create cell namespace**.
2. Enter `NDAFServer1` in the Admin name field.
3. Enter `cell1` in the Cell name field and press Enter.

Listing cell namespaces

You can list the cells that have been defined for a specified administration server.

You can use the **`dmf`** command to list cells:

```
dmf enumerate admin cell [pattern] [-r] [-a admin_server]
```

The flags have the following definitions:

-a admin_server Specifies the DNS name or IP address of the administration server. The port can be added using a colon separator.

pattern Optional matching text pattern. Valid values include ? and *.

-r Prints the uuid assigned to the request.

Perform the following steps to lists cells using SMIT:

1. Select **NDAF** → **NDAF Management** → **Namespace (cell) Management** → **List Cell Namespaces**.
2. Specify the name of the administration server that manages the NDAF domain in the Admin name field and press Enter.

Showing and changing cell attributes

You can show the attributes of a specified cell. You can modify some of these attributes if you have the necessary authorization.

You can use the **`dmf`** command to show the attributes for a cell:

```
dmf show cell [-r] [-a admin_server] [-c container]
```

The flags have the following definitions:

-a admin_server Specifies the DNS name or IP address of the administration server. The port can be added using a colon separator.

-c container Specifies the cell name this command is addressed to.

-r Prints the uuid assigned to the request.

You can use the **dmf** command to change the attributes for a cell:

```
dmf set cell key=value [-r] [-a admin_server] [-c container]
```

The flags have the following definitions:

-a admin_server Specifies the DNS name or IP address of the administration server. The port can be added using a colon separator.

-c container Specifies the cell name this command is addressed to.

key=value Specifies an attribute and the value to assign to it. Valid keys are DmLogLevel and DmLocsMax.

-r Prints the uuid assigned to the request.

Perform the following steps to show and change the attributes for a specific cell using SMIT:

1. Select **NDAF** → **NDAF Management** → **Namespace (cell) Management** → **Change/show cell attributes**.
2. Specify the name of the administration server that manages the NDAF domain in the Admin name field.
3. Enter the name of the cell in the Cell name field (or choose one from the list by pressing F4). The following attributes are displayed:
   - Admin server DNS name (or IP address)

     Specifies the DNS name or IP address of the administration server that manages the NDAF domain.
   - Admin server name

     Specifies the name of the administration server that manages the NDAF domain.
   - Cell name

     Specifies the name of the cell.
   - Cell UUID

     Specifies the uuid for the cell.
   - Maximum number of reported locations

     Specifies the maximum number of NFS location referrals that can be returned to an NFS client for an object.
   - NDAF principals

     Enter the list of users, separated by commas, directly in the input field. Users from this list are owners of this cell and can manipulate the cell.

Removing a cell namespace

You can remove a cell object and clean the databases of all of the objects that have been defined within the cell.

You can use the **dmf** command to remove a cell namespace:

```
dmf destroy cell [-r] [-f] [-a admin_server] [-c container]
```

The flags have the following definitions:

-a admin_server Specifies the DNS name or IP address of the administration server. The port can be added using a colon separator.

-c container Specifies the container (that is, the cell name).

-f Forces the action without confirmation.

-r Prints the uuid assigned to the request.

Perform the following steps to remove a cell namespace using SMIT:

1. From the NDAF menu, select **NDAF Management** → **Namespace (cell) Management** → **Remove cell namespace**.
2. Specify the name of the administration server that manages the NDAF domain in the Admin name field and press Enter.
3. Enter the name of the cell to be removed in the Cell name field and press Enter. A standard SMIT dialog box displays to confirm the destruction of the cell.

Adding a server to a cell namespace

A cell can use a data server to host the cell's data set.

You can use the **dmf** command to enable a cell to use a data server to host the cell's data set:

```
dmf place cell server_name [-r] [-a admin_server] [-c container]
```

The flags have the following definitions:

server_name Specifies the server on which the cell should be made available for mounting by NFS.

-r Prints the uuid assigned to the request.

-a admin_server Specifies the DNS name or IP address of the administration server. The port can be added using a colon separator.

-c container Specifies the cell name.

Perform the following steps to enable a cell to use a data server to host the cell's data set using SMIT:

1. From the NDAF menu, select **NDAF Management → Namespace (cell) Management → Add Server to a Cell Namespace**.
2. Specify the name of the administration server that manages the NDAF domain in the Admin name field and press Enter.
3. Enter the name of the cell in the Cell Name field (or choose one from the list by pressing F4) and press Enter.
4. Enter the name of the data server in the Data server name field (or choose one from the list by pressing F4) and press Enter.

Removing a server from a cell namespace

You can prevent a cell from using a data server to host the cell's data sets.

You can use the **dmf** command to prevent a cell from using a data server to host the cell's data sets:

```
dmf unplace cell server_name [-r] [-f] [-a admin_server] [-c
container]
```

The flags have the following definitions:

server_name Specifies the server on which the cell should become unavailable for mounting by NFS.

-r Prints the uuid assigned to the request.

-f Forces the action without confirmation.

-a admin_server Specifies the DNS name or IP address of the administration server. The port can be added using a colon separator.

-c container Specifies the cell name.

Perform the following steps to prevent a cell from using a data server to host the cell's data sets using SMIT:

1. From the NDAF menu, select **NDAF Management → Namespace (cell) Management → Remove Server from a Cell Namespace**.
2. Specify the name of the administration server that manages the NDAF domain in the Admin name field and press Enter.
3. Enter the name of the cell in the Cell Name field (or choose one from the list by pressing F4) and press Enter.
4. Enter the name of the data server in the Data server name field (or choose one from the list by pressing F4) and press Enter.

Creating and managing roles

A role is a set of privileges associated with a set of users. Roles are used to manage resources within a cell. Administrators can create, list, remove, validate, and change the options of roles.

Creating and managing data sets

A data set (dset) is a directory tree of file system objects (files, directories, ACLs, links, and so on).

Creating and managing replicas

A replica is a read-only copy of a data set. You can distribute a replica across multiple data servers. You can create, remove, validate, update, list, mount, unmount, place, and unplace replicas and change their options.

Populating data sets

There are two basic approaches to populating newly created data sets with data: by local file system access on the server, or remotely by NFSv4. With the exception of replicas under NDAF control and the creation of data set references for data set mounts, the creation and manipulation of data inside file systems with data sets uses normal file system access mechanism such as libc calls and NFS. The primary form of access to NDAF-managed file systems is expected to be through NFS.

Constructing a cell namespace from data sets

You can construct a cell namespace from data sets.

Federating servers without NDAF into an NDAF environment

You can create an external non-NDAF managed server so that existing local file system data can be connected to an NDAF cell namespace. External servers are used to tie data from non-NDAF enabled servers or preexisting data on NDAF-enabled servers into the NDAF cell namespace.

> **Note:** For more information and examples about NDAF management, refer to the *IBM AIX Information Center* publication at:
>
> `http://publib.boulder.ibm.com/infocenter/pseries/v5r3/topic/com.ibm.aix.ndaf/doc/ndaf/NDAF_management.htm`

6.8.7 Troubleshooting NDAF

Table 6-2 lists common problems and actions to troubleshoot NDAF.

Table 6-2 Common problems and actions to troubleshoot NDAF

Problem	Action
The dms or dmadm daemons will not start.	You must specify either ndaf_dir or both ndaf_dataset_default and ndaf_replica default when starting dms. You must specify ndaf_dir when starting dmadm. The specified directory for data sets and replicas must belong to a data-set enabled file system. Use dms_enable_fs to enable the file system. If using Kerberos, check the Kerberos keytab file.
Can read files, but cannot create or modify data set files from NFS client.	The DmMode for the data set might not permit writes. To fix this, use: `dmf set dset DmMode=<required mode>`
Cannot navigate to the cell or data set directory or access data set data from NFS client.	nfsd, nfsrgyd, and dms must be running on the NDAF data server.
The **dmf** command fails with `Cannot contact remote host` message.	Make sure that the host name of the administration server is specified correctly in the **dmf** command and that the dmadm daemon is running on the administration server.
Cannot specify directory when creating a data set.	The specified directory must belong to a file system that has been enabled for data sets on the server with the **dms_enable_fs** command. The specified directory must not exist, and will be subsequently created by the **dmf** command.
Source data set is down and a replica exists, but the client will not fail over to the replica.	This feature is not supported. Clients will fail over from one replica to another, but not from the source data set to a replica.
Failover from one replica to another takes too long or is too quick.	The timeout by default is approximately the timeo value multiplied by the retrans value (from the **mount** command or the **nfs4cl** command). This can be overridden with the nfs_replica_failover_timeout option of the **nfso** command.
Cell or data set is not NFS-exported.	Make sure that nfsroot is set. The command to set it is: `chnfs -r <root_dir>` To make the exports happen after setting the root, you must restart dms.

Problem	Action
Cannot create a cell or data set.	The file system where the data set will be created must be enabled for data sets with the **dms_enable_fs** command.
Cannot see data in the data set when mounted with NFSv4. Might be able to see the data when mounted from some servers, but not from others.	NFSv4 must be configured with replicas enabled on all NDAF servers. The command to enable NDAF servers is: `chnfs -R on`

NDAF checker

To help diagnose problems, NDAF provides commands to check the consistency of the databases used to manage the NDAF objects on administration and data servers.

You can use the following **dmf** command code to check the validity and consistency of the administration server database:

```
dmf check_adm admin [-r] [-a admin_server]
```

The flags have the following definitions:

-r Prints the uuid assigned to the request.

-a admin_server Specifies the DNS name or IP address of the admin server. The port can be added using a colon separator.

You can use the following **dmf** command to check the validity and consistency of the data server database on the specified data server or on every managed data server if none is specified:

```
dmf check_serv server [-r] [-a admin_server] [-c container]
```

The flags have the following definitions:

-r Prints the uuid assigned to the request.

-a admin_server Specifies the DNS name or IP address of the admin server. The port can be added using a colon separator.

-c container Specifies the name of the server to check.

You can use the following **dmf** command to check the consistency of the database on the administration server with the database on the specified data server or with the databases on every managed data server if none is specified:

```
dmf check_adm_serv {admin|server} [-r] [-a admin_server] [-c
container]
```

The flags have the following definitions:

-r Prints the uuid assigned to the request.

-a admin_server Specifies the DNS name or IP address of the admin server. The port can be added using a colon separator.

-c container Specifies the name of the server to check.

If the data from this command indicates inconsistencies between the databases, it might be necessary to recover from a backup to restore correct behavior. See NDAF data backup and NDAF data recovery for more information.

Example 6-3 shows sample output from running the **dmf** command.

Example 6-3 The dmf command output

```
# dmf check_adm admin -a ndaf10
---------- STARTING REPORT FROM SERVER my_admin ----------
1        ERROR(S)
---------- DATABASE CHECK ----------
ERROR:  1
DATABASE:       gldb:///admin/dmf.ldb
DESCRIPTION:
Error DmGroup root doesn't exist for DmUuid
8c9d4200-e973-11da-b214-de83e0005002 with path \
/var/dmf/server/dsets/8c9d416a-e973-11da-b214-de83e0005002
---------- END OF REPORT FROM SERVER my_admin ----------
```

The sample output shows that an NDAF object has a DmGroup parameter that is not valid. In this situation, use the **dmf show** command to find the corresponding data set, which is identified by the DmUuid parameter. In this case, the group specified for the data set is not valid because it does not exist in the /etc/group file. The solution might be to use the dmf set **dset** command to change this data set parameter.

NDAF data backup

An important consideration when backing up NDAF data is that the NDAF state data needs to be backed up along with the data set data.

NDAF data recovery

Assuming tha tthe NDAF data set data and state data is backed up as described previously, recovery depends on what is lost.

Note: For more information and examples about NDAF troubleshooting procedures and facilities, refer to the *IBM AIX Information Center* publication on NDAF at:

http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=/com.ibm.aix.ndaf/doc/ndaf/troubleshooting_ndaf_failures.htm

6.9 AIX Security Expert (5300-05)

AIX Security Expert (aixpert) is a network and system security hardening tool that incorporates the information in advanced UNIX security white papers and the expertise of leading security experts into one tool. The AIX Security Expert can be run using a GUI from Web-based System Manager, which allows you to control hundreds of complex security configuration settings with the click of a mouse.

AIX Security Expert can be used to implement the appropriate level of security, without the necessity of reading a large number of papers on security hardening and then individually implementing each security element. Without disrupting services and ports in use, this tool protects unused and vulnerable ports, protects against port scans, and implements strong password policies, along with a broad spectrum of other security settings.

The AIX Security Expert tool includes an undo feature, and allows the user to take snapshots of the setting of one system and exactly reproduce those settings on systems throughout the enterprise. AIX Security Expert can also check the security health of the system and report on any settings that have been changed since the last use of the tool.

AIX Security Expert can be run using the **`aixpert`** command line, SMIT (fastpath: **`smit aixpert`**), and the Web-based System Manager GUI. The Web-based System Manager GUI is recommended, as it is especially user friendly in working with the tool.

6.9.1 AIX Security Expert security level settings

AIX Security Expert provides simple menu settings for high-level security, medium-level security, low-level security, and AIX Standard Settings security that integrate over 300 security configuration settings while still providing control over each security element for advanced administrators. AIX Security Expert provides a center for all security settings (TCP, NET, IPSEC, system, and auditing).

The AIX Security Expert view of security levels is derived in part from the National Institute of Standards and Technology document Security Configuration Checklists Program for IT Products - Guidance for CheckLists Users and Developers:

http://csrc.nist.gov/checklists/download_sp800-70.html

However, high, medium, and low level security mean different things to different people. It is important to understand the environment in which your system operates. If you chose a security level that too high, you could lock yourself out of your computer. If you chose a security level that is too low, your computer might be vulnerable to a cyber attack.

The following coarse-grained security settings are available:

High-level security High-level security does not permit **telnet**, **rlogin**, **ftp**, and other common connections that transmit passwords over the network in the clear. These passwords can easily be snooped by someone on the Internet, so a secure method to log in remotely (such as **openssh**) should be used.

Medium-level security Medium-level security applies common security settings, such as port scan protection and password expirations, but leaves the system open to most remote access methods.

Low-level security Low-level security is most appropriate for machines on an isolated secure local network. This system is open for access to a wide variety of services.

Advanced security Custom user-specified security.

AIX Standard Settings Original system default security.

6.9.2 AIX Security Expert groups

The security settings configurable by the AIX Security Expert are grouped according to their respective functions. Refer to the *IBM AIX Information Center* for detailed descriptions and security level setting defaults on each of the following Security Expert groups:

- Password Policy Rules group

  AIX Security Expert provides specific rules for password policy.

- User Group System and Password definitions group

  AIX Security Expert performs specific actions for user, group, and password definitions.

- Login Policy Recommendations group

  AIX Security Expert provides specific settings for login policy.

- Audit Policy Recommendations group

  AIX Security Expert provides specific audit policy settings.

- /etc/inittab Entries group

  AIX Security Expert comments out specific entries in /etc/inittab so that they do not start when the system boots.

- /etc/rc.tcpip Settings group

  AIX Security Expert comments out specific entries in /etc/rc.tcpip so that they do not start when the system boots.

- /etc/inetd.conf Settings group

  AIX Security Expert comments out specific entries in /etc/inetd.conf.

- Disable SUID of Commands group

  By default, the following commands are installed with the SUID bit set. For high, medium, and low security, this bit is unset. For AIX Standard Settings, the SUID bit is restored on these commands.

- Disable Remote Services group

  AIX Security Expert disables unsecure commands for high-level security and medium-level security.

- Remove access that does not require Authentication group

  AIX supports few services that do not require user authentication to log into the network.

- Security Expert Tuning Network Options group

  Tuning network options to the proper values is a large part of security. Setting a network attribute to 0 disables the option, and setting the network attribute to 1 enables the option.

- IPsec filter rules group

  AIX Security Expert provides IPsec filters.

- Miscellaneous group

  AIX Security Expert provides miscellaneous security settings for high-level, medium-level, and low-level security.

6.9.3 AIX Security Expert Undo Security

You can undo some AIX Security Expert security settings and rules. However, there are security settings and rules that cannot be undone. The following are AIX Security Expert non-reversible security settings and rules:

- Check password definitions for high-level security, medium-level security, and low-level security.
- Enable X-Server access for high-level security, medium-level security, and low-level security.
- Check user definitions for high-level security, medium-level security, and low-level security.
- Remove dot from non-root path for high-level security and AIX Standard Settings.
- Check group definitions for high-level security, medium-level security, and low-level security.
- Remove guest account for high-level security, medium-level security, and low-level security.
- TCB update for high-level security, medium-level security, and low-level security.

6.9.4 AIX Security Expert Check Security

AIX Security Expert can generate reports of current system and network security settings.

After AIX Security Expert is used to configure a system, the Check Security option can be used to report the various configuration settings. If any of these settings have been changed outside the control of AIX Security Expert, the AIX Security Expert Check Security option logs these differences in the /etc/security/aixpert/check_report.txt file.

For example, the talkd daemon is disabled in /etc/inetd.conf when you apply low-level security. If the talkd daemon is later enabled and then Check Security is run, this information will be logged in the check_report.txt file, as follows:

```
coninetdconf.ksh: Service talk using protocol udp should be
disabled, however it is enabled now.
```

If the applied security settings have not been changed, the check_report.txt file will be empty.

The Check Security option should be run periodically, and the resulting report should be reviewed to see if any settings have been changed since AIX Security

Expert security settings were applied. The Check Security option should also be run as part of any major system change such as the installation or updating of software.

6.9.5 AIX Security Expert files

AIX Security Expert creates and uses several files.

- /etc/security/aixpert/core/aixpertall.xml

  Contains an XML listing of all possible security settings.
- /etc/security/aixpert/core/appliedaixpert.xml

  Contains an XML list of applied security settings.
- /etc/security/aixpert/core/secaixpert.xml

  Contains an XML listing of selected security settings when processed by the AIX Security Expert GUI.
- /etc/security/aixpert/log/aixpert.log

  Contains a trace log of applied security settings. AIX Security Expert does not use syslog. AIX Security Expert writes directly to /etc/security/aixpert/log/aixpert.log.

The AIX Security Expert XML and log files are created with the following permissions:

- /etc/security/aixpert/
  /etc/security/aixpert/core/
  /etc/security/aixpert/core/appliedaixpert.xml
  /etc/security/aixpert/core/secaixpert.xml
  /etc/security/aixpert/log
  - drwx------
- /etc/security/aixpert/log/aixpert.log
  /etc/security/aixpert/core/secundoaixpert.xml
  /etc/security/aixpert/check_report.txt
  - rw-------
- /etc/security/aixpert/core/aixpertall.xml
  - r--------

6.9.6 AIX Security Expert security configuration copy

AIX Security Expert can be used to take a security configuration snapshot. This snapshot can be used to set up the same security configuration on other systems. This both saves time and ensures that all systems have the proper security configuration in an enterprise environment.

For example, you can apply the security settings on one system with high, medium, low, advanced, or AIX Standard Settings security. After testing this system for compatibility issues within your environment, you can apply the same settings on other AIX systems by copying the /etc/security/aixpert/core/appliedaixpert.xml file to the other system. You then run the following command to set the security of the system to the same security settings as the source system:

```
aixpert -f appliedaixpert.xml
```

6.9.7 The aixpert command

AIX Security Expert can be run from the command line. The full path of the executable is /usr/sbin/aixpert.

The **aixpert** command has the following syntax:

```
aixpert -l high|medium|low|advanced|default [-n -o filename ]
[ -a -o filename ]
aixpert -l h|m|l|a|d [ -n -o filename ] [ -a -o filename ]
aixpert -c
aixpert -u
aixpert [-f filename ] [ -a -o filename ]
aixpert [-v filename]
aixpert -e -i filename -o filename
```

Running aixpert with the only the -l flag set implements the security settings promptly without letting the user configure the settings. For example, running **aixpert -l high** applies all of the high-level security settings to the system automatically. However, running **aixpert -l** with the **-n -o** filename option saves the security settings to a file specified by the filename parameter. The user can then use the -v flag to view the file and view the settings. The -f flag then applies the new configurations.

Note: We recommend that aixpert be rerun after any major systems changes, such as the installation or updates of software. If a particular security configuration item is deselected when aixpert is rerun, that configuration item is skipped.

The following are the useful flags and their definitions:

-a The settings with the associated level security options are written in abbreviated file format to the file specified by the -o flag.

-c Checks the security settings.

-e The settings with the associated level security options are written in expanded file format to the file specified by the -o option.

-f Applies the security settings in the provided filename. For example, aixpert -h -n writes all of the high-level security options to the /etc/security/aixpert/core/secaixpert.xml file. After commenting out any undesired options, you can apply these security settings with the **`aixpert -f /etc/security/aixpert/core/secaixpert.xml`** command. This option also allows for consistent security settings to be applied from system to system by securely transferring and applying an secaixpert.xml file from system to system.

-i Stores security input to the file pointed to by filename. The input file has its read and write permissions set to root as a security precaution. This file should be protected against unwanted access.

-l Sets the system security level to low. When used in conjunction with the -n flag, no action is taken, and the low-level security options are written only to the /etc/security/aixpert/core/secaixpert.xml file. This flag takes the following options:

h high

Specifies high-level security options. When used in conjunction with the -n flag, these security options are not implemented on the system, and the settings are written only to the /etc/security/aixpert/core/secaixpert.xml file. This output can be directed to different output files with the -o flag.

m medium

Specifies medium-level security options. When used in conjunction with the -n flag, these security options are not implemented on the system, and the settings are written only to the /etc/security/aixpert/core/secaixpert.xml file. This output can be directed to different output files with the -o flag.

l low

Specifies low-level security options. When used in conjunction with the -n flag, these security options are not implemented on the system, and the settings are written only to the

/etc/security/aixpert/core/secaixpert.xml file. This output can be directed to different output files with the -o flag.

a advanced

Uses all security rules: high, medium, and low. This option does not provide a higher level of security than the -h flag, but it can be used to view all possible security settings. Some rules may be mutually exclusive. When used in conjunction with the -n flag, these security options are not implemented on the system, and the settings are written only to the /etc/security/aixpert/core/secaixpert.xml file. This output can be directed to different output files with the -o flag.

d default

Uses the default setting, which has no additional security rules, and undoes any configured security settings. When used in conjunction with the -n flag, these security options are not implemented on the system, and the settings are written only to the /etc/security/aixpert/core/secaixpert.xml file. This output can be directed to different output files with the -o flag.

> **Note:** Using the d option can overwrite previously configured security settings that were set through aixpert or independently, and restores the system to its traditional open configuration.

-n The settings with the associated level security options are written only to the /etc/security/aixpert/core/secaixpert.xml file. When used in conjunction with the -o flag, the options are written to the file specified by the -o flag.

-o Stores security output to the file pointed to by filename. The output file has its read and write permissions set to root as a security precaution. This file should be protected against unwanted access.

-u Undoes the security settings that have been applied.

-v Allows for the graphical viewing of the security setting in a particular file.

To start the graphical user interface to step through the security settings in wizard fashion, type:

```
aixpert
```

To write all of the high-level security options to an output file, type:

```
aixpert -l high -o
/etc/security/aixpert/plugin/myPreferredSettings.xml
```

> **Note:** After completing this command, the output file can be edited, and specific security roles can be commented out by enclosing them in the standard xml comment string (<-- begins the comment and -\> closes the comment).

To apply the security settings from a configuration file, type:

```
aixpert -f /etc/security/aixpert/plugin/myPreferredSettings.xml
```

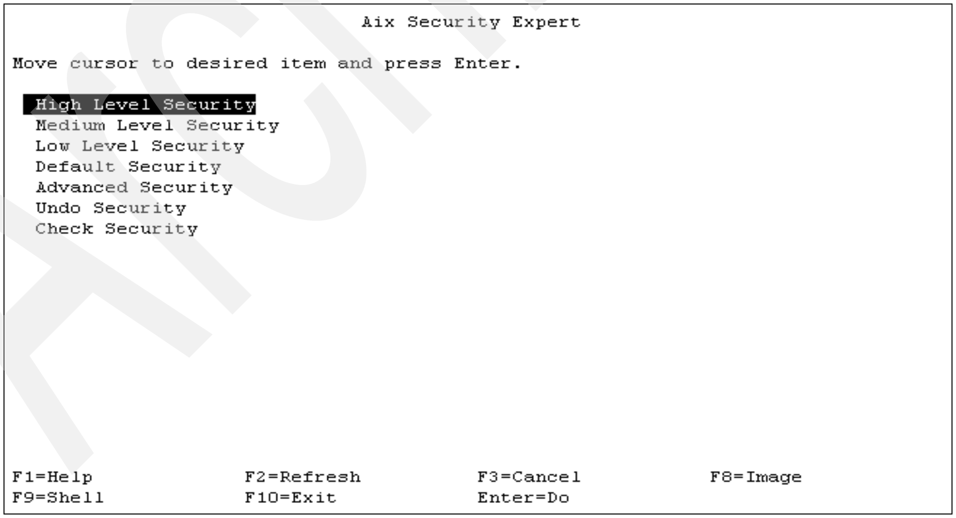
To view the security settings that have been applied to the system, type:

```
aixpert -v /etc/security/aixpert/core/AppliedAixpert.xml
```

AIXpert can be run from SMIT with the fastpath:

```
smit aixpert
```

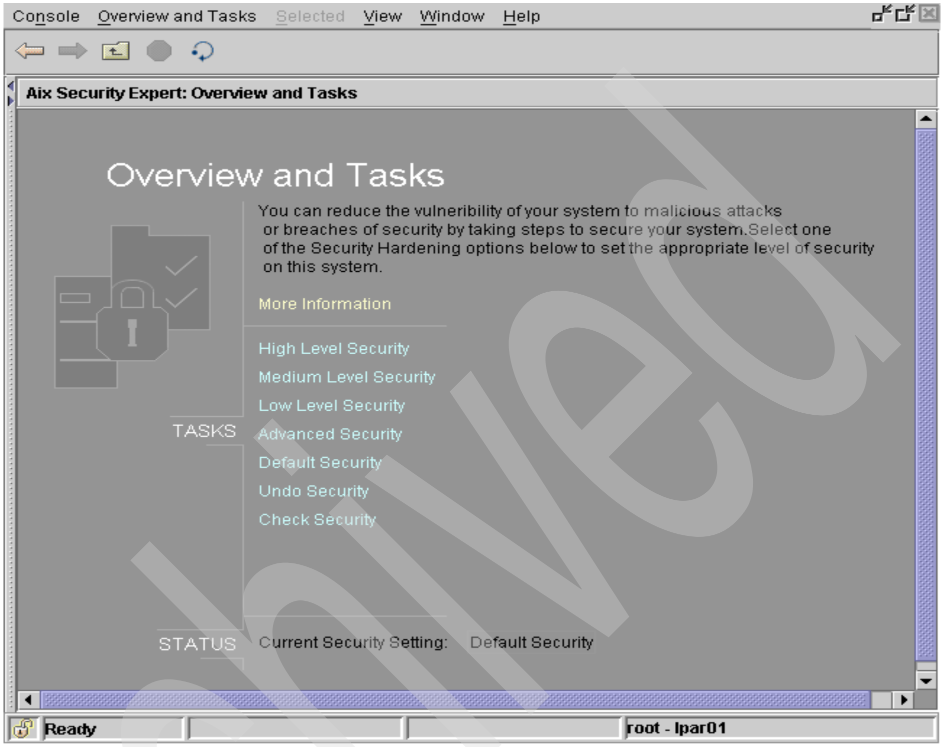
Figure 6-4 shows the AIX Security Expert SMIT menu that will be displayed.



```
                              Aix Security Expert

Move cursor to desired item and press Enter.

  High Level Security
  Medium Level Security
  Low Level Security
  Default Security
  Advanced Security
  Undo Security
  Check Security

F1=Help             F2=Refresh          F3=Cancel           F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 6-4 The smit menu for aixpert

The Web-based System Manager GUI for AIX Security Expert is recommended, as it is especially user friendly in working with the tool, as shown in Figure 6-5.



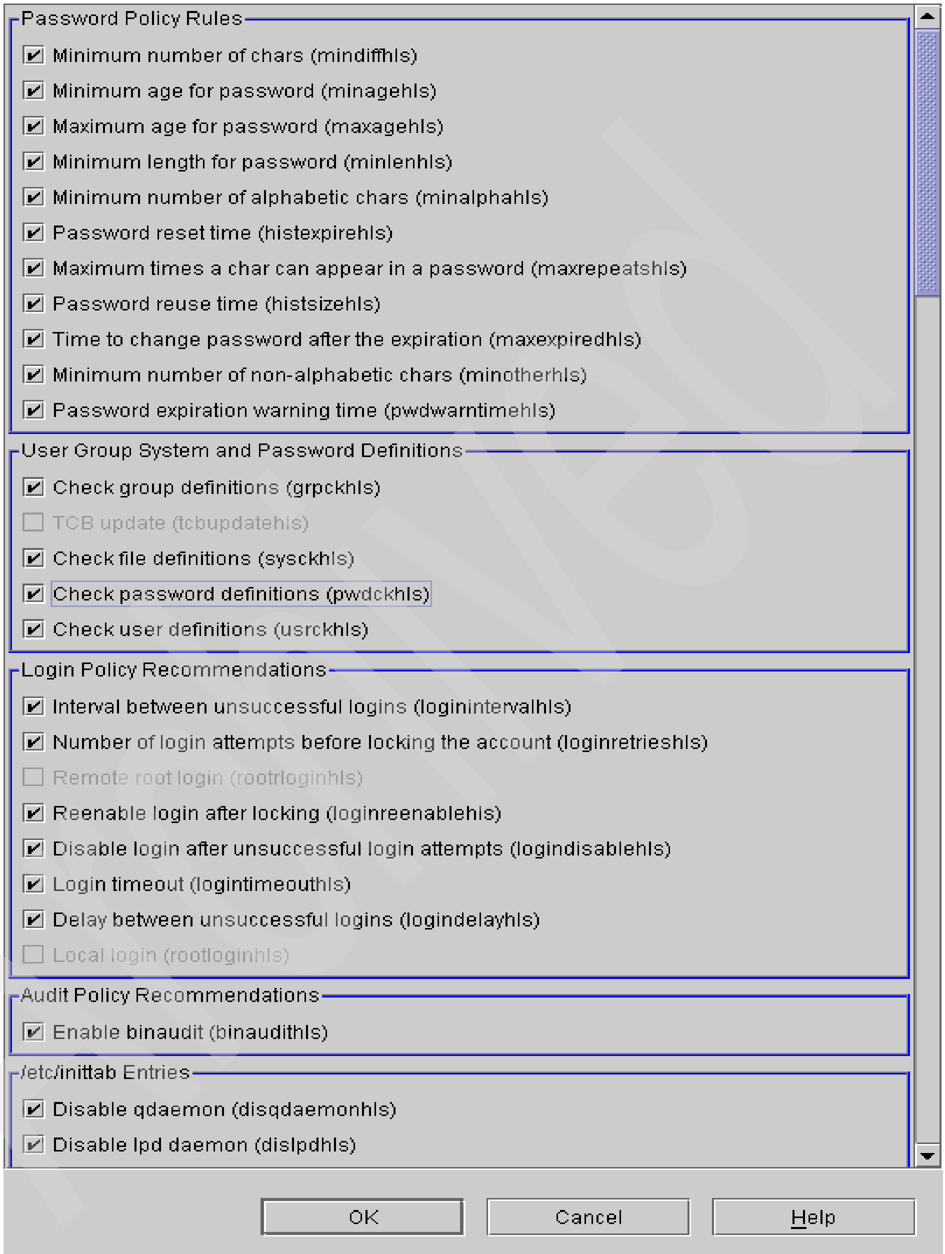
Overview and Tasks

You can reduce the vulneribility of your system to malicious attacks or breaches of security by taking steps to secure your system.Select one of the Security Hardening options below to set the appropriate level of security on this system.

Current Security Setting: Default Security

Figure 6-5 The Security Expert on Web-based System Manager

A detailed view of the options is shown in Figure 6-6.



Password Policy Rules

- ☑ Minimum number of chars (mindiffhls)
- ☑ Minimum age for password (minagehls)
- ☑ Maximum age for password (maxagehls)
- ☑ Minimum length for password (minlenhls)
- ☑ Minimum number of alphabetic chars (minalphahls)
- ☑ Password reset time (histexpirehls)
- ☑ Maximum times a char can appear in a password (maxrepeatshls)
- ☑ Password reuse time (histsizehls)
- ☑ Time to change password after the expiration (maxexpiredhls)
- ☑ Minimum number of non-alphabetic chars (minotherhls)
- ☑ Password expiration warning time (pwdwarntimehls)

User Group System and Password Definitions

- ☑ Check group definitions (grpckhls)
- □ TCB update (tcbupdatehls)
- ☑ Check file definitions (sysckhls)
- ☑ Check password definitions (pwdckhls)
- ☑ Check user definitions (usrckhls)

Login Policy Recommendations

- ☑ Interval between unsuccessful logins (logintervalhls)
- ☑ Number of login attempts before locking the account (loginretrieshls)
- □ Remote root login (rootrloginhls)
- ☑ Reenable login after locking (loginreenablehls)
- ☑ Disable login after unsuccessful login attempts (logindisablehls)
- ☑ Login timeout (logintimeouthls)
- ☑ Delay between unsuccessful logins (logindelayhls)
- □ Local login (rootloginhls)

Audit Policy Recommendations

- ☑ Enable binaudit (binaudithls)

/etc/inittab Entries

- ☑ Disable qdaemon (disqdaemonhls)
- ☑ Disable lpd daemon (dislpdhls)

Figure 6-6 AIXpert individual security settings on Web-based System Manager

The AIX Security Expert tool is installed with the bos.aixpert.cmds file set, which is part of the 5300-05 Technology Level. It is also available as a fix download at the following Web site:

```
http://www7b.boulder.ibm.com/aix/fixes/byCompID/5765G0300/bos.aixpert
/bos.aixpert.cmds.5.3.0.0.bff
```

For more information about AIX Security Expert, refer to the *IBM AIX Information Center* at:

```
http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=
/com.ibm.aix.security/doc/security/aix_sec_expert_pwd_policy_settings.
htm
```

7

Installation, backup, and recovery

This chapter discusses the following major enhancements:

- Installation to disks > 1 TB (5300-05)
- NIM enhancements (5300-05)
- Migrating a NIM client to a POWER5 logical partition
- SUMA enhancements
- Targeting disk for installing AIX (5300-03)
- The multibos command (5300-03)
- mksysb migration support (5300-03)
- mksysb enhancements (5300-01)
- DVD install media support for AIX 5L (5300-02)

7.1 Installation to disks > 1 TB (5300-05)

Previously, AIX 5L could not be installed to devices larger than 1 TB. This restriction has now been lifted.

There are considerations about the amount of disk drive capacity allowed in a single RAID array. Using the 32-bit kernel, there is capacity maximum of 1 TB per RAID array. Using the 64-bit kernel, there is a capacity maximum of 2 TB per RAID array. For RAID adapter and RAID enablement cards, these maximums are enforced by AIX 5L when RAID arrays are created using the PCI-X SCSI Disk Array Manager. The adapters utilizing the PCI-X SCSI Disk Array Manager are:

- PCI-X Dual Channel Ultra320 SCSI RAID Adapter (FC 5703, FC 5711, FC 1975)
- Dual Channel SCSI RAID Enablement Card (FC 5709, FC 5726, FC 1976)
- PCI-X Quad Channel U320 SCSI RAID Adapter (FC 2780)
- PCI-XDDR Dual Channel U320 SCSI RAID Adapter (FC 5737, FC 1913)
- Dual Channel SCSI RAID Enablement Card (FC 5727, FC 5728, FC 1907)
- Dual Channel SCSI RAID Enablement Card (FC 1908)

When creating a RAID array up to 2 TB with the Standalone Diagnostics, ensure that Version 5.3.0.40 or later is used. Previous versions of the Standalone Diagnostics had a capacity limitation of 1 TB per RAID array.

Note: In order to install to devices larger than 1 TB, the 64-bit kernel must be used.

7.2 NIM enhancements (5300-05)

Some of the enhancements that AIX 5L Version 5.3 has made to the NIM environment are:

- Detailed output when creating a NIM lpp_source resource
- Creating Shared Product Object Tree (SPOT) resource from a mksysb
- Restoring SPOT copy function
- Adjustments in NIM to process multiple CD media
- NIM interface to change network attributes

These enhancements along with security ones are shipped with AIX 5.3 base version:

- NIM service handler for client communication — basic **nimsh**
- NIM cryptographic authentication — OpenSSL

These topics are covered in *AIX 5L Differences Guide Version 5.3 Edition*, SG24-7463.

AIX 5L Version 5.3 with TL 5300-05 introduces Thin Server (NIM diskless/dataless client renamed) and common OS image (NIM SPOT resource renamed) Management. This facilitates cloning a common image and performing operations on the clone image, thus preventing operations from interfering with running thin servers. This also allows a thin server to switch to a different common image at a specified time as dictated by the NIM Administrator.

The following sections describe how to configure and use the thin server.

7.2.1 Creating resources to support the thin server on NIM master

The NIM environment can be used for installing and maintaining software on standalone machines and thin servers together. If you create resources for standalone machines, use the smit fastpath:

```
smit nim_mkres_basic_inst
```

The common OS image will be used by the thin server and mounted as the /usr file system. Each time a common OS image is added to the NIM environment, an entry representing the common image as a spot resource is added to the NIM database.

To make a common OS image for thin server, you can use smit fastpath:

```
smit mkcosi
```

To clone a common OS image from an existing one for thin server, you can use smit fastpath:

```
smit cpcosi
```

To manage a common OS image, you can enter:

```
smit chcosi
```

There you can install, update, commit, reject, and remove to/on a common OS image.

Also, you can display information such as status, software content, and logs for a specific common OS image by using:

```
smit lscosi
```

And remove a common OS image by fastpath:

```
smit rmcosi
```

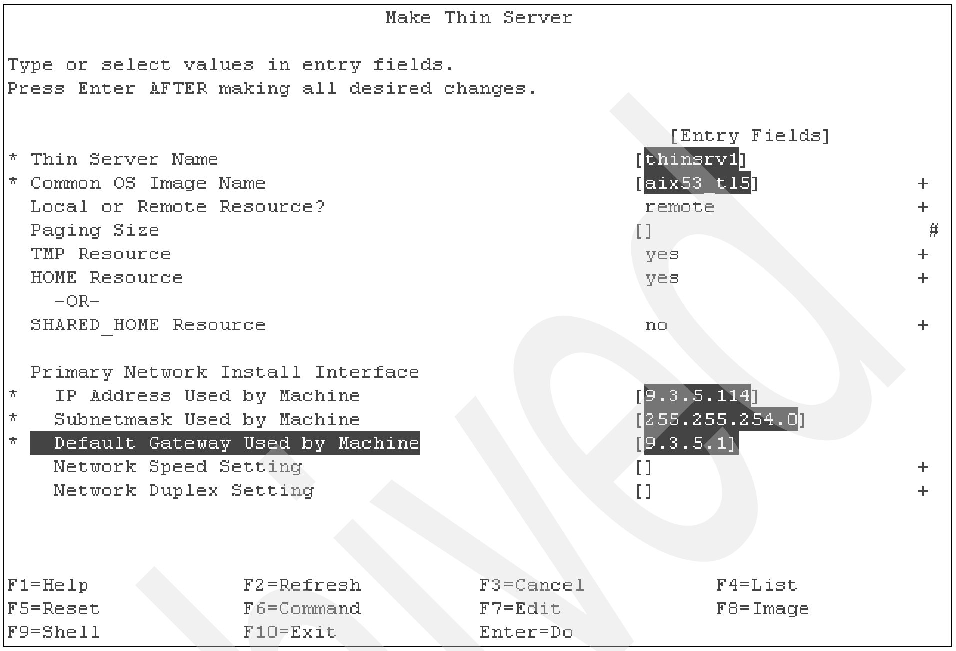
Note: In most NIM environments, the common OS image will already exist. In such environments it is your choice to create a new common OS image or clone one for you.

7.2.2 Adding a thin server to the NIM environment

This procedure is also known as making a thin server. Prior to this step, a common OS image must be created as the resource for the thin server.

1. Use the **`smit mkts`** fast path to add thin server to the NIM environment.
2. Type the host name or IP address of the machine that you want to be a thin server. This machine will be added to the NIM environment and to the NIM database.
3. Select the common OS image Name for the thin server.
4. Provide NIM the network information and press Enter to proceed.

It will take a few minutes for NIM to add the thin server to the database and prepare for other data. See Figure 7-1.



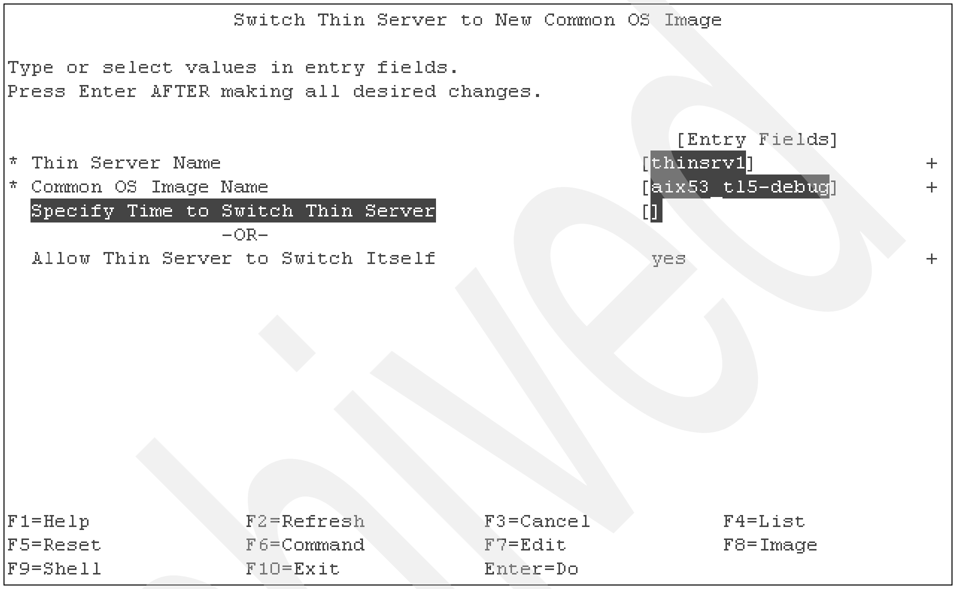
```
                               Make Thin Server

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

                                                         [Entry Fields]
* Thin Server Name                                    [thinsrv1]
* Common OS Image Name                                [aix53_tl5]              +
  Local or Remote Resource?                            remote                  +
  Paging Size                                         []                        #
  TMP Resource                                         yes                     +
  HOME Resource                                        yes                     +
    -OR-
  SHARED_HOME Resource                                 no                      +

  Primary Network Install Interface
*   IP Address Used by Machine                        [9.3.5.114]
*   Subnetmask Used by Machine                        [255.255.254.0]
*   Default Gateway Used by Machine                   [9.3.5.1]
    Network Speed Setting                             []                       +
    Network Duplex Setting                            []                       +



F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 7-1 The smit mkts panel

AIX 5L Version 5.3 with TL 5300-05 provides a useful function that allows a thin server to switch to a different common image. The administrator has the option of allowing the thin server to switch over to a different common OS image now, at a specific time, or at the client's own convenience. You can use the following smit fastpath to achieve this:

```
smit swts
```

Type the time, in the **at** command format, to switch the thin server over to the specified common OS image in the Specify Time to Switch Thin Server field. At the specified time, the thin server will mount the common OS image as its /usr file system. If Specify Time to Switch Thin Server and Allow Thin Server to Switch Itself are not specified, then the switch will occur immediately. See Figure 7-2.



```
                   Switch Thin Server to New Common OS Image

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

                                                          [Entry Fields]
* Thin Server Name                                     [thinsrv1]            +
* Common OS Image Name                                 [aix53_tl5-debug]     +
  Specify Time to Switch Thin Server                   []
                -OR-
  Allow Thin Server to Switch Itself                    yes                  +









F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 7-2 The smit swts panel

7.2.3 Booting a thin server machine

A thin server is also known as a diskless or a dataless client. Booting a thin server is the same as booting a machine from the network.

1. Turn the power switch on the system unit to the On position or press the power on button.
2. Enter the SMS menu by pressing 1 (this may vary on some models).
3. Select the network adapter that you want to boot from as the first boot device and specify the NIM master IP address before exiting.
4. When the server reboots, the server will reboot from the network and work as a thin server (at the same time, a NIM client).

If the thin server is unable to boot successfully, you can use a debug common OS image to get more detailed debug information. You can build a debug common OS image by using:

```
smit dbts
```

7.3 Migrating a NIM client to a POWER5 logical partition

AIX 5L Version 5.3 with ML 5300-03 provides a facility **nim_move_up** command to enable new hardware (namely POWER5 or later servers) support in AIX environments.

7.3.1 Requirement of migrating

The **nim_move_up** application allows you to easily migrate a back-level AIX system onto an logical partition (LPAR) residing on a POWER5 (or newer) server.

The system must meet the following requirements before you can run the **nim_move_up** application properly:

- NIM master requirements
  - A configured NIM master running AIX 5.3 with 5300-03 or later
  - Perl 5.6 or later
  - OpenSSH (obtainable from the Linux Toolbox media)
  - At least one stand-alone NIM client running AIX 4.3.3.75 or later
  - AIX product media Version AIX 5L with 5200-04 or later, or AIX 5L product media Version 5.3 or later, or equivalent lpp_source and SPOT NIM resources
- Server and resource requirements
  - A POWER5 server with sufficient hardware resources to support the target client's equivalent POWER5 configuration.
  - If virtual resources will be used to migrate the clients, an installed and configured Virtual I/O Server is required.
  - HMC controlling the POWER5 server, along with sufficient privileges to start, stop, and create LPARs.
  - Root user authority.

This **nim_move_up** process requires no downtime on the part of the original client. In addition, **nim_move_up** is capable of migrating a client onto virtualized hardware, such as virtual disks, using the Virtual I/O capabilities of the POWER5 server. This migration process can be completed by the **nim_move_up** application in phases to allow more control over the process, or it can be completed all at once without any user interaction required.

With the **nim_move_up** application, you can use a NIM master and its clients as the starting point for a migration that produces the following hardware environment:

- The original NIM master
- LPARs on POWER5 server that correspond to the original NIM clients and are controlled by the NIM master
- HMC to control the LPARs on the POWER5 servers, communicated with by the NIM master through SSH
- The original NIM clients

7.3.2 Migration phases

The **nim_move_up** migration process is completed in the following phases to allow more control over the process:

1. The create NIM resources phase creates the needed NIM resources to perform the migration steps if they do not already exist or are not provided beforehand.
2. The pre-migration software assessment phase performs an assessment on each target client to determine what software is installed and can be migrated. Any software that is missing from the lpp_source will be added from the source of installation images that should be provided to **nim_move_up**.
3. The client hardware and utilization data gathering phase gathers data about each target client's hardware resources and attempts to assess how much of those resources are utilized on average over a given amount of time.
4. The POWER5 resource availability data gathering and client resource data translation phase searches the given managed system for available hardware resources. It uses the data gathered in the previous phase to create an equivalent LPAR configuration that utilizes the managed system's available resources. It creates the client LPARs with virtual I/O resources instead of physical I/O resources if **nim_move_up** was provided with a Virtual I/O Server LPAR to work with. It creates the appropriate adapters and configuration on the Virtual I/O Server as they are needed.
5. The create system backups of target clients phase creates an installable image of each target client and its resources using the **mksysb** command.

6. The migrate each system backup phase uses the **nimadmin** command to migrate the newly created installable images to the new level of AIX.
7. The allocate nim resources to new lpars phase uses the network information provided to the **nim_move_up** application to create NIM standalone client objects for the new LPARs created in the POWER5 resource availability data gathering and client resource data translation phase. It allocates the appropriate NIM resources and runs a bos_inst pull operation (that is, NIM will not attempt to boot the client) on each NIM client.
8. The initiate installation on lpars phase reboots each LPAR via the control host (HMC partition) and initiates the installation.

> **Note:** This phase ends when the installation begins. The actual progress of the installation is not monitored.

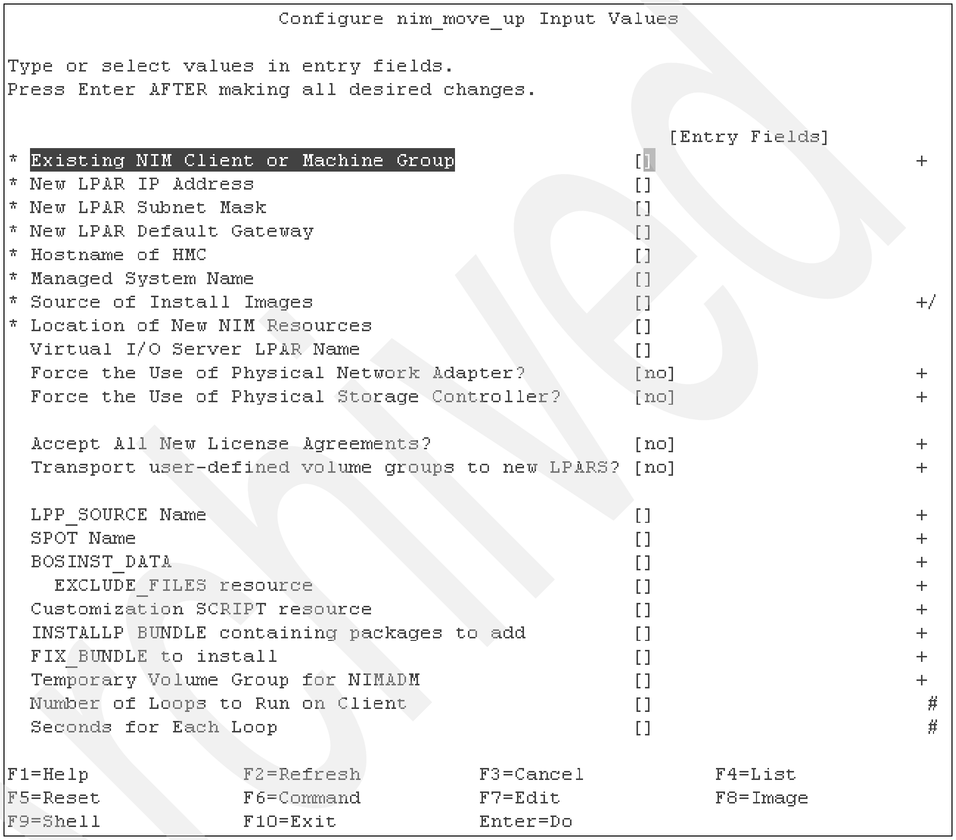
9. The post-migration software assessment phase assesses the overall success of the migration after each installation and reports on any software migration issues. It may be necessary to manually correct the errors reported for file sets that fail to migrate.
10. The post-installation customization phase performs a NIM customization operation on each client with the values provided if an alternate lpp_source, fileset list, or customization script was provided to the nim_move_up application. This allows for the optional installation of additional software applications or for any additional customization that may be needed.

7.3.3 Smit menu for nim_move_up

The SMIT fastpath to the root menu of **nim_move_up** is **smitty nim_move_up**.

After all prerequisites needed to run the **nim_move_up** application have been met, **nim_move_up** performs the migration process in two steps: configuration and phase execution. You can run the **nim_move_up** application from SMIT by completing the following steps:

1. Enter `smitty nim_move_up_config`. The Configure nim_move_up Input Values panel opens, as shown in Figure 7-3.



```
                       Configure nim_move_up Input Values

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

                                                        [Entry Fields]
* Existing NIM Client or Machine Group               []                       +
* New LPAR IP Address                                []
* New LPAR Subnet Mask                               []
* New LPAR Default Gateway                           []
* Hostname of HMC                                    []
* Managed System Name                                []
* Source of Install Images                           []                       +/
* Location of New NIM Resources                      []
  Virtual I/O Server LPAR Name                       []
  Force the Use of Physical Network Adapter?         [no]                     +
  Force the Use of Physical Storage Controller?      [no]                     +

  Accept All New License Agreements?                 [no]                     +
  Transport user-defined volume groups to new LPARS? [no]                     +

  LPP_SOURCE Name                                    []                       +
  SPOT Name                                          []                       +
  BOSINST_DATA                                       []                       +
    EXCLUDE_FILES resource                           []                       +
  Customization SCRIPT resource                      []                       +
  INSTALLP BUNDLE containing packages to add         []                       +
  FIX_BUNDLE to install                              []                       +
  Temporary Volume Group for NIMADM                  []                       +
  Number of Loops to Run on Client                   []                        #
  Seconds for Each Loop                              []                        #

F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 7-3 Configure nim_move_up input values

2. Enter information in the required fields. This information is retained by the **nim_move_up** application, unless the application is reset. You can change this information at any time from the Configure nim_move_up Input Values panel.

3. To begin the actual migration process, enter `smitty nim_move_up_exec`. The Execute nim_move_up Phases panel opens.

4. Provide an appropriate answer to the option Execute All Remaining Phases? on the Execute nim_move_up Phases panel and press Enter.

You can use other panels to interact with the **nim_move_up** application, in addition to the Configure nim_move_up Input Values panel and the Execute nim_move_up Phases panel. The are as follows:

- Display the Current Status of nim_move_up

  Selecting this menu option is equivalent to running **nim_move_up** with the -S flag. The next phase to be executed and a listing of all of the saved options are displayed.

- Configure SSH Keys on Target HMC

  This SMIT panel provides a simple interface to setting up SSH keys on the remote control host (HMC). Using this panel is the equivalent of using the -K command-line option. Configuring SSH keys on the remote control host enables the unattended remote execution of commands from the NIM master.

- Unconfigure nim_move_up

  This SMIT panel provides an interface to unconfiguring the nim_move_up environment. Unconfiguring the environment removes all state information, including what phase to execute next, saved data files generated as a result of the execution of some phases, and all saved input values. Optionally, all NIM resources created through **nim_move_up** can also be removed. Using this panel is the equivalent of using the -r command-line option.

7.4 SUMA enhancements

Service Update Management Assistant (SUMA) allows the automation of fix downloads using the **suma** command. There have been two major enhancements to this since the release of AIX 5L Version 5.3:

- Integration between the Network Installation Manager (NIM) and **suma** has been introduced.
- There have been several refinements to the **suma** functionality itself, most notably integration with the updated AIX 5L release strategy.

7.4.1 NIM and SUMA integration (5300-05)

This function provides the ability to gather and combine client inventories, compare microcode and firmware levels, and generate reports. It also enables the download of AIX 5L fixes to be consolidated. The main features introduced are two new commands, **niminv** and **geninv**, for inventory management. In addition, the existing **suma** and **compare_report** commands can now work with NIM resources.

suma and compare_report NIM integration

The **suma** and **compare_report** commands are now able to work with NIM lpp_source resources.

The suma command

The following **suma** command options can now take an lpp_source as an option:

```
suma -a DLTarget=<value>
suma -a FilterDir=<value>
```

For the DLTarget field, this allows an lpp_source directory to be used as the download target. **suma** will filter and download-only fixes that are not existent in the lpp_source. The FilterDir allows file sets to be downloaded to a different target directory, but filtered against the contents of an existing lpp_source.

The compare_report command

The **compare_report** command is now able to use an lpp_source as follows:

```
compare_report -i <FixDir>
```

This option can now take an lpp_source as the name of the fix repository directory. The file set levels of the images contained in this directory will be used in the comparison.

7.4.2 The suma command enhancements (5300-04)

Starting with AIX 5L Version 5.3 level 5300-04, the maintenance strategy has altered with the introduction of technology levels. More details on these changes can be found in 4.6, “LDAP enhancements (5300-03)” on page 97.

The **suma** command has been integrated with this via the introduction of the new RqType fields, TL and SP for technology level and service pack, respectively. For example, the following command would download the AIX 5L Version 5.3 technology level 5300-04 immediately:

```
suma -x -a RqType=TL -a RqName=5300-04
```

It is also now possible to specify RqLevel fields for the RqTypes APAR, Security, Critical, or Latest. This enables the request of fixes on the specified TL without requiring a move to the next TL. The following command specifies the download of APAR IY12345 for the 5300-04 technology level:

```
suma -x -a RqType=APAR -a RqName=IY12345 -a RqLevel=5300-04
```

> **Note:** The APAR will first be released with 5300-05 at the 5.3.0.50 level. Later it might be released with 5300-04-CSP at the 5.3.0.48 level.

7.4.3 The geninv command (5300-05)

The **geninv** command gathers software and hardware installation version inventories from systems.

For software inventories, the following sources are supported:

- Installp
- rpm (RedHat Package Manager)
- ISMP (Install Shield Multi-Platform)
- Emergency and interim fixes

For hardware inventories, the following sources are supported:

- System Firmware
- Adapter/Component Microcode that can be upgraded

Table 7-1 gives details on the **geninv** command usage, using the following syntax:

```
geninv { -c | -l } [-D] [-P <protocol> | -H <host>]
```

Table 7-1 The geninv command parameter details

Parameter	Description
c	Produces colon-separated output.
l	Produces list-style output.
D	Produces debug output.
P	Uses <protocol> to gather inventory from a remote machine. The <protocol> should contain all attributes including protocol command name and remote hostn ame, for example. The <protocol> will be prepended to all information-gathering commands.
H	Uses rsh protocol to gather inventory from <host>.

Example 7-1 shows **geninv** usage.

Example 7-1 Example of geninv usage

```
#geninv -l -P ssh 9.3.5.111

PROTOCOL=ssh
   Installation Name        Level  Type   State  Description
   ------------------------------------------------------------------
```

```
Java14.sdk             1.4.2.75  F      C      Java SDK 32-bit
Tivoli_Management_Agent.client.rte
                        3.7.1.0  F      C      Management Framework
                                               Endpoint Runtime"
X11.adt.bitmaps         5.3.0.0  F      C      AIXwindows Application
                                               Development Toolkit Bitmap
                                               Files
X11.adt.imake          5.3.0.30  F      C      AIXwindows Application
                                               Development Toolkit imake
X11.adt.include        5.3.0.50  F      C      AIXwindows Application
                                               Development Toolkit Include
                                               Files


.....
.....
......
IHS2                   2.0.47.1  P      C      IBM HTTP Server
AIX-rpm-5.3.0.50-1     5.3.0.50  R      C      Virtual Package for
                                               libraries and shells
                                               installed on system
cdrecord-1.9-7              1.9  R      C      A command line CD/DVD
                                               recording program.
mkisofs-1.13-4             1.13  R      C      Creates an image of an
                                               ISO9660 filesystem.
openssl-0.9.7g-2         0.9.7g  R      C      Secure Sockets Layer and
                                               cryptography libraries and
                                               tools
prngd-0.9.29-1           0.9.29  R      C      Pseudo Random Number
                                               Generator Daemon
hdisk.IC35L07.44543031
                       53323847  H      C      16 Bit LVD SCSI Disk Drive
                                               (73400 MB)
rmt.2107.                  2107  H      C      LVD SCSI Tape Drive (80000
                                               MB)
sisioa.5052414E.030D004f
                       030D004f  H      C      PCI-X Dual Channel U320
                                               SCSI RAID Adapter
sys.system.9113-550
                      SF240_219  H      C      System Object
```

7.4.4 The niminv command (5300-05)

The **niminv** command gathers and compares software and, if applicable, hardware installation version inventorys from NIM objects. It can also download fixes based on these inventorys:

- To get installation inventory:

```
niminv -o invget -a targets=object1,object2,... [ -a location=path ]
[ -a colonsep=yes|no ]
```

- To conglomerate installation inventory:

```
niminv -o invcon -a targets=object1,object2,... [ -a base=
highest|lowest ] [ -a location=path ] [ -a colonsep=yes|no ]
```

- To compare installation inventory:

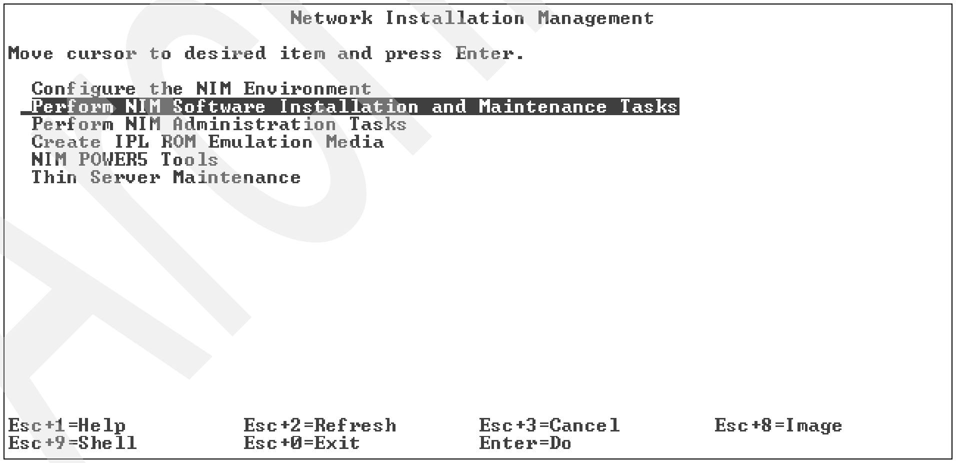
```
niminv -o invcmp -a targets=object1,object2,... [ -a base=object
|any ] [ -a location=path ]
```

- To get fixes based on conglomerate inventory:

```
niminv -o fixget -a targets=object1,object2,.. [-a download=yes|no ]
[ -a lp_source=object ] [ -a location=path ] -a newlppname=name
```

This command is also accessible through the **smit nim** subpanel Installation Inventory. The following command shows the panel provided in Figure 7-4:

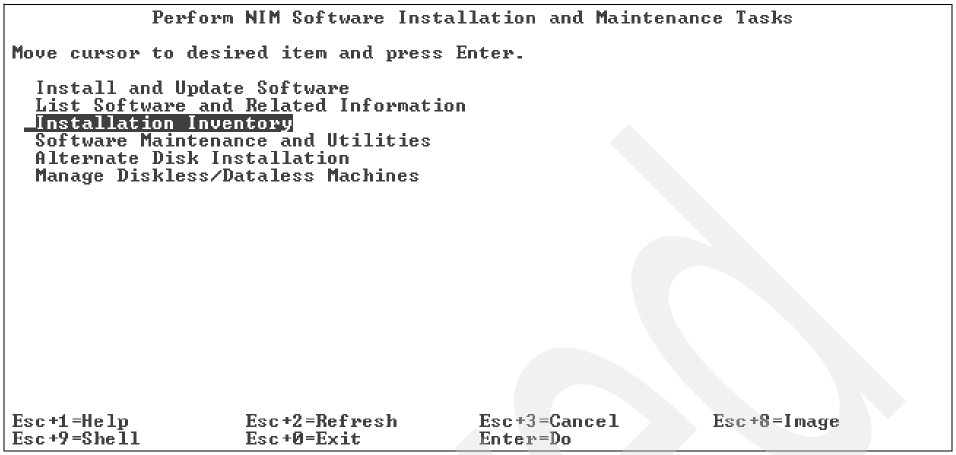
```
#smit nim
```



```
                          Network Installation Management

Move cursor to desired item and press Enter.

  Configure the NIM Environment
  Perform NIM Software Installation and Maintenance Tasks
  Perform NIM Administration Tasks
  Create IPL ROM Emulation Media
  NIM POWER5 Tools
  Thin Server Maintenance

Esc+1=Help          Esc+2=Refresh       Esc+3=Cancel        Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do
```

Figure 7-4 Output of smit nim

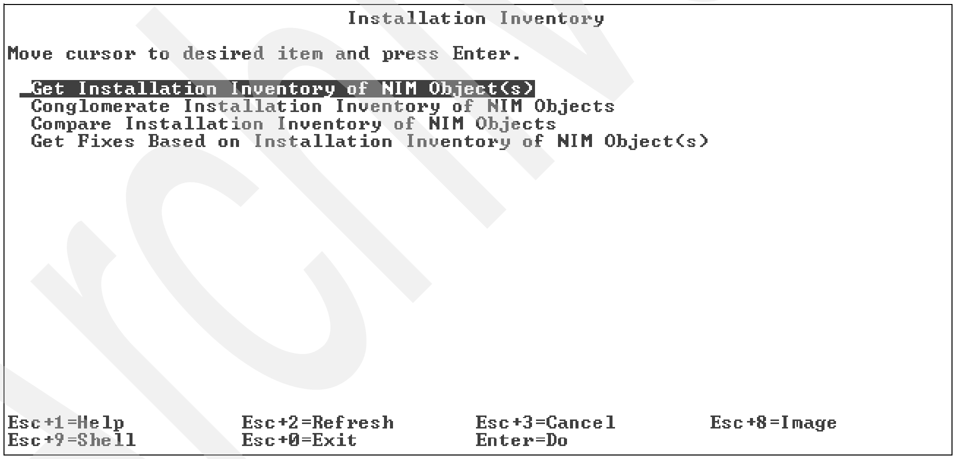
Selecting the highlighted text brings you to the panel shown in Figure 7-5.



```
              Perform NIM Software Installation and Maintenance Tasks

Move cursor to desired item and press Enter.

  Install and Update Software
  List Software and Related Information
  Installation Inventory
  Software Maintenance and Utilities
  Alternate Disk Installation
  Manage Diskless/Dataless Machines

Esc+1=Help          Esc+2=Refresh       Esc+3=Cancel        Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do
```

Figure 7-5 NIM installation inventory

Selecting the highlighted selection brings you to the panel shown in Figure 7-6.



```
                              Installation Inventory

Move cursor to desired item and press Enter.

  Get Installation Inventory of NIM Object(s)
  Conglomerate Installation Inventory of NIM Objects
  Compare Installation Inventory of NIM Objects
  Get Fixes Based on Installation Inventory of NIM Object(s)

Esc+1=Help          Esc+2=Refresh       Esc+3=Cancel        Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do
```

Figure 7-6 NIM installation inventory details

The smit fast path **`smit nim_inventory`** can also be used.

The **`niminv`** command (Network Install Manager Inventory) allows system administrators to accomplish the following tasks:

- Gather installation inventory of several NIM objects.
- Conglomerate installation inventorys of several NIM objects.

- Compare installation inventorys of several NIM objects.
- Download fixes based on the installation inventorys of several NIM objects.
- The **niminv** command can use any NIM object that contains installation information. Examples of such objects include standalone client objects, SPOT objects, lpp_source objects, and mksysb objects.

Using the **niminv** command has the following advantages:

- Hardware installation inventory is gathered alongside the software installation inventory.
- Data files are saved with a naming convention that is easily recognizable.
- All NIM objects that have installation inventory can be used.

Thus, the **niminv** command provides a holistic view of all managed NIM objects.

7.5 Targeting disk for installing AIX (5300-03)

In previous versions of AIX 5L and AIX, if you do not specify the disk on which you want the AIX system installed, the operating system is installed on a disk that was previously installed with AIX. If you have many disks that contain data volume groups, and these data volume groups are discovered before the previous root volume group is found, the installation can be delayed until a suitable disk is found. Beginning with AIX 5L Version 5.3 with the 5300-03 Recommended Maintenance package, you can specify the disk on which you want to install the system.

You can specify the installation disk by using one of the following methods.

Specify the installation disk in the bosinst.data file by physical location code (PHYSICAL_LOCATION) or physical volume identifier (PVID):

- To determine the physical location on a running system, type:

```
# lsdev -F "name physloc" -l hdisk0
hdisk0 U787B.001.DNW108F-P1-T14-L3-L0
```

- To determine the physical volume identifier on a running system, type:

```
# lsattr -E -O -a pvid -l hdisk0
#pvid
00c5e9de205cf5c60000000000000000
```

If you are using a fibre-channel disk for the installation, you can use the following command in the bosinst.data file:

```
SAN_DISKID=worldwide_portname//lun_id
```

Specify the installation disk in the bosinst.data file through either an installation from CD or DVD, or through a network installation.

For a network installation, specify the installation disk in the bosinst.data file by typing the following command:

```
nim -o bos_inst -a bosinst_data=value ...
```

For an installation from CD or DVD, specify the installation disk in the bosinst.data file by following the procedures in *Customizing and using the `bosinst.data` file* at:

http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=/com.ibm.aix.install/doc/insgdrf/bosinst.data_custom.htm

If you do not specify the characteristics of the disk in the bosinst.data file on the target system, the installation disk is chosen based on the flags in the control_flow stanza of the bosinst.data file. Unless you specify EXISTING_SYSTEM_OVERWRITE=no, the first suitable root volume group is chosen for the installation. For overwrite or preserve installations, any root volume group is acceptable. For migration, the volume group must be installed with a version of the operating system that can be migrated to the level being installed. If you specify EXISTING_SYSTEM_OVERWRITE=no in the control_flow stanza of the bosinst.data file, then the installation goes to the first unused disk.

7.6 The multibos command (5300-03)

The **`multibos`** command allows a root-level administrator to create and maintain a second instance of the AIX Base Operating System (BOS) within the running rootvg. Installing maintenance and technology updates to the standby BOS does not change system files on the active BOS. This allows for concurrent update of the standby BOS, while the active BOS remains in production. This feature has been made more robust and more user friendly in 5300-05.

7.6.1 Requirements of the multibos command

The **`multibos`** command has requirements for operating system, space, and logical volumes. The general requirements and limitations are:

- The **`multibos`** command is supported on AIX 5L Version 5.3 with the 5300-03 Recommended Maintenance package and later versions.
- The current rootvg must have enough space for each BOS object copy. BOS object copies are placed on the same disks as the original.

- The **multibos** command supports a total number of copied logical volumes up to 128. The total number of copied logical volumes and shared logical volumes is subject to volume group limits.

> **Note:** When updating the non-running BOS instance, it is best to first update the running BOS instance with the latest available version of multibos (which is in the bos.rte.bosinst file set).

7.6.2 Using the multibos command

The /usr/sbin/multibos command is supplied in AIX 5L with 5300-03 (in the bos.rte.bosinst file set) to create and manage a new instance of the operating system within the running rootvg. The multibos utility provides the root user with operations to set up, access, maintain, update, and customize this new instance of the BOS.

The setup operation creates a standby BOS that boots from a distinct Boot Logical Volume (BLV). This creates two bootable instances of the BOS on a given rootvg. You can boot from either instance of a BOS by specifying the respective BLV as an argument to the **bootlist** command, or using system firmware boot operations.

You can simultaneously maintain two bootable instances of a BOS. The instance of a BOS associated with the booted BLV is the active BOS. The instance of a BOS associated with the BLV that has not been booted is the standby BOS. Only two instances of BOS are supported per rootvg.

The **multibos** command allows you to access, install, maintain, update, and customize the standby BOS either during setup or during any subsequent customization operations. Installing maintenance or technology level updates to the standby BOS does not change system files on the active BOS. This allows for concurrent update of the standby BOS, while the active BOS remains in production.

The multibos utility has the ability to copy or share logical volumes and file systems. By default, the multibos utility copies the BOS file systems (currently the /, /usr, /var, /opt, and /home directories), associated log devices, and the boot logical volume. The administrator has the ability to share or keep private all other data in the rootvg.

> **Note:** As a general rule, shared data should be limited to file systems and logical volumes containing data not affected by an upgrade or modification of private data.

Copies of additional BOS objects (see the –L flag) can also be made. All other file systems and logical volumes are shared between instances of the BOS. Separate log device logical volumes (those not contained within the file system) are not supported for copy and will be shared.

Note: Automatic file system expansion

Run all **multibos** operations with the -X flag auto-expansion feature. This flag allows for automatic file system expansion on any **multibos** command execution if additional space is necessary to perform multibos-related tasks.

Table 7-2 The multibos command flags

Flag	Definition
-a	Specifies the update_all install option. Valid with setup and customization operation.
-B	Build boot image operation. The standby boot image is created and written to the standby BLV using the **bosboot** command.
-b	File specifies the install bundle to be installed during the setup or customization operation. The install bundle syntax should follow **geninstall** conventions.
-c	Performs a customized update of the software in standby BOS.
-e	File lists active BOS files to be excluded during the setup operation in regular expression syntax.
-f	File lists fixes (such as APARs) that are to be installed during the setup or customization operation. The list's syntax follows instfix conventions.
-i	File specifies an optional image.data file to use instead of the default image.data file created from the current rootvg.
-L	File specifies a file that contains a list of additional logical volumes to include in standby BOS.
-l	Device installs device or directory for software update during the setup or customization operation.
-m	Mounts standby BOS.
-N	Skips boot image processing. This flag should only be used by experienced administrators that have a good understanding of the AIX boot process.
-n	Does not perform cleanup upon failure. This option is useful to retain multibos data after a failed operation.

Flag	Definition
-p	Performs a preview of the given operation. Valid with setup, remove, mount, unmount, and customization operations.
-R	Removes all standby BOS objects.
-S	Initiates an interactive shell with **chroot** access to the standby BOS file systems.
-s	Creates an instance of standby BOS.
-t	Prevents multibos from changing the bootlist.
-u	Unmounts standby BOS.
-X	Allows for automatic file system expansion if space is needed to perform tasks related to **multibos**. We recommend that all multibos operations are executed with this flag.

Examples of the multibos command

The following are examples of using the **multibos** command:

- To perform a standby BOS setup operation preview, type:

```
multibos -Xsp
```

- To set up standby BOS, type:

```
multibos -Xs
```

- To set up standby BOS with optional image.data file /tmp/image.data and exclude list /tmp/exclude.list, type:

```
multibos -Xs -i /tmp/image.data -e /tmp/exclude.list
```

- To set up standby BOS and install additional software listed as bundle file /tmp/bundle and located in the images source /images, type:

```
multibos -Xs -b /tmp/bundle -l /images
```

- To execute a customization operation on standby BOS with the update_all install option, type:

```
multibos -Xac -l /images
```

- To mount all standby BOS file systems, type:

```
multibos -Xm
```

- To perform a standby BOS remove operation preview, type:

```
multibos -RXp
```

- To remove standby BOS, type:

  `multibos -RX`

7.6.3 Standby BOS setup operation

The standby BOX setup operation has the following key attributes:

- A structure based on image.data is created.
- Prefixes are added to new LVs and file systems (bos_ for LV and bosinst_ for file systems).
- A file list is generated. The exclude option used.
- Files are copied.
- Additional customization is performed.
- The BLV is created.
- The bootlist is set to standby/active.

Example 7-2 shows the command execution.

Example 7-2 The multibos -s command output to set up standby BOS

```
# multibos -s -X
Initializing multibos methods ...
Initializing log /etc/multibos/logs/op.alog ...
Gathering system information ...

+----------------------------------------------------------------------+
Setup Operation
+----------------------------------------------------------------------+
Verifying operation parameters ...
Creating image.data file ...

+----------------------------------------------------------------------+
Logical Volumes
+----------------------------------------------------------------------+
Creating standby BOS logical volume bos_hd5
Creating standby BOS logical volume bos_hd4
Creating standby BOS logical volume bos_hd2
Creating standby BOS logical volume bos_hd9var
Creating standby BOS logical volume bos_hd10opt

+----------------------------------------------------------------------+
File Systems
+----------------------------------------------------------------------+
Creating all standby BOS file systems ...
Creating standby BOS file system /bos_inst on logical volume bos_hd4
Creating standby BOS file system /bos_inst/usr on logical volume
bos_hd2
Creating standby BOS file system /bos_inst/var on logical volume
bos_hd9var
Creating standby BOS file system /bos_inst/opt on logical volume
bos_hd10opt

+---------------------------------------------------------------------+
Mount Processing
+---------------------------------------------------------------------+
Mounting all standby BOS file systems ...
Mounting /bos_inst
Mounting /bos_inst/usr
Mounting /bos_inst/var
Mounting /bos_inst/opt

+---------------------------------------------------------------------+
BOS Files
+---------------------------------------------------------------------+
Including files for file system /
Including files for file system /usr
Including files for file system /var
Including files for file system /opt

Copying files using backup/restore utilities ...
Percentage of files copied:   0.00%
Percentage of files copied:   1.93%
...
Percentage of files copied: 100.00%

+---------------------------------------------------------------------+
Boot Partition Processing
+---------------------------------------------------------------------+
Active boot logical volume is hd5.
Standby boot logical volume is bos_hd5.
Creating standby BOS boot image on boot logical volume bos_hd5
bosboot: Boot image is 30420 512 byte blocks.

+---------------------------------------------------------------------+
Mount Processing
+---------------------------------------------------------------------+
Unmounting all standby BOS file systems ...
Unmounting /bos_inst/opt
Unmounting /bos_inst/var
Unmounting /bos_inst/usr
Unmounting /bos_inst

+-----------------------------------------------------------------------------+
Bootlist Processing
+-----------------------------------------------------------------------------+
Verifying operation parameters ...
Setting bootlist to logical volume bos_hd5 on hdisk0.
ATTENTION: firmware recovery string for standby BLV (bos_hd5):
boot /pci@800000020000003/pci@2,4/pci1069,b166@1/scsi@0/sd@3:4
ATTENTION: firmware recovery string for active BLV (hd5):
boot /pci@800000020000003/pci@2,4/pci1069,b166@1/scsi@0/sd@3:2

Log file is /etc/multibos/logs/op.alog
Return Status = SUCCESS
```

The **multibos** setup operation, using the -s flag, performs the following steps:

1. The multibos methods are initialized.
2. If you provide a customized image.data file, it is used for the logical volume attributes. Otherwise, a new one is generated. You can use the customized image.data file to change BOS object (logical volume or file systems) attributes. You cannot use the customized image.data file to add or delete BOS logical volumes or file systems.
3. The standby logical volumes are created based on image.data attributes. The active and standby logical volumes are marked with unique tags in the logical volume control block. The **multibos** command uses these tags to identify copied logical volumes. If the active logical volume names are classic names, such as hd2, hd4, hd5, and so on, then the bos_prefix is prepended to create a new standby name. If the active logical volume names have the bos_prefix, the prefix is removed to create a new standby name.

   Note: The Logical Volume Manager (LVM) limits the maximum length of a logical volume name to 15 characters. This means that any logical volume classic name may not exceed 11 characters. You can rename logical volumes that have classic names that exceed 11 characters using the **chlv** command. If the active logical volume name already has the bos_prefix, then the prefix is removed in the standby name.

4. The standby file systems are created based on image.data attributes. The active and standby file systems are marked with unique tags in the hosting logical volume control block and /etc file systems. The multibos utility uses

these tags to identify copied logic volumes. The /bos_inst prefix is prepended to the original active file system name to create the standby file system name. The standby file system name may not exceed the system's PATH_MAX limit. The standby file systems appear as standard entries in the active BOS /etc/filesystems.

5. The standby file systems are mounted.
6. A list of files that will be copied from the active BOS is generated. This list is comprised of the current files in copied active BOS file systems, less any files that you excluded with the optional exclude list (see the -e flag).
7. The list of files generated in the previous step is copied to the standby BOS file systems using the backup and restore utilities.
8. Any optional customization is performed. This can include installation of file set updates or other software.
9. The standby boot image is created and written to the standby BLV using the **`bosboot`** command. You can block this step with the -N flag. Only use the -N flag if you are an experienced administrator and have a good understanding the AIX boot process.
10. The standby BLV is set as the first boot device, and the active BLV is set as the second boot device. You can skip this step using the -t flag.

7.6.4 Rebuilding the standby BOS boot image

The rebuild boot image operation, using the -B flag, enables you to rebuild the standby BOS boot image.

The new boot image will be based on standby BOS system files and written to the standby BLV. The **`multibos`** command build boot image operation performs the following steps:

1. The standby BOS file systems are mounted, if they are not already.
2. The standby boot image is created and written to the standby BLV using the **`bosboot`** command.
3. If the standby BOS file systems were mounted in step 1, they are unmounted.

7.6.5 Mounting the standby BOS

It is possible to access and modify the standby BOS by mounting its file systems over the standby BOS file system mount points. The **`multibos`** command mount operation, using the -m flag, mounts all standby BOS file systems in the appropriate order.

Example 7-3 shows an example of mounting of the standby BOS.

Example 7-3 The multibos -m -X command output

```
# multibos -m -X
Initializing multibos methods ...
Initializing log /etc/multibos/logs/op.alog ...
Gathering system information ...

+-----------------------------------------------------------------------+
BOS Mount Operation
+-----------------------------------------------------------------------+
Verifying operation parameters ...

+-----------------------------------------------------------------------+
Mount Processing
+-----------------------------------------------------------------------+
Mounting all standby BOS file systems ...
Mounting /bos_inst
Mounting /bos_inst/usr
Mounting /bos_inst/var
Mounting /bos_inst/opt

Log file is /etc/multibos/logs/op.alog
Return Status = SUCCESS
```

7.6.6 Customizing the standby BOS

You can use the multibos customization operation, with the -c flag, to update the standby BOS.

The customization operation requires an image source (-l device or directory flag) and at least one installation option (installation by bundle, installation by fix, or **`update_all`** command). The customization operation performs the following steps:

1. The standby BOS file systems are mounted, if not already mounted.
2. If you specify an installation bundle with the -b flag, the installation bundle is installed using the **`geninstall`** command. The installation bundle syntax should follow **`geninstall`** command conventions. If you specify the -p preview flag, **`geninstall`** will perform a preview operation.
3. If you specify a fix list with the -f flag, the fix list is installed using the **`instfix`** command. The fix list syntax should follow instfix conventions. If you specify

the -p preview flag, then the `instfix` command will perform a preview operation.

4. If you specify the `update_all` function with the -a flag, it is performed using the `install_all_updates` command. If you specify the -p preview flag, then the `install_all_updates` command performs a preview operation.

> **Note:** It is possible to perform one, two, or all three of the installation options during a single customization operation.

5. The standby boot image is created and written to the standby BLV using the `bosboot` command. You can block this step with the -N flag. You should only use the -N flag if you are an experienced administrator and have a good understanding the boot process.
6. If standby BOS file systems were mounted in step 1, they are unmounted.

7.6.7 Unmounting the standby BOS

The multibos unmount operation using the -u flag unmounts all standby BOS file systems in the appropriate order.

7.6.8 Using the standby BOS shell operation

The multibos shell operation -S flag enables you to start a limited interactive `chroot` shell with standby BOS file systems.

This shell allows access to standby files using standard paths. For example, /bos_inst/usr/bin/ls maps to /usr/bin/ls within the shell. The active BOS files are not visible outside of the shell unless they have been mounted over the standby file systems. Limit shell operations to changing data files, and do not make persistent changes to the kernel, process table, or other operating system structures. Only use the BOS shell if you are experienced with the `chroot` environment.

The multibos shell operation performs the following steps:

1. The standby BOS file systems are mounted, if they are not already.
2. The `chroot` utility is called to start an interactive standby BOS shell. The shell runs until an exit occurs.
3. If standby BOS file systems were mounted in step 1, they are unmounted.

The following is an example of some operations that can be performed in the multibos shell:

MULTIBOS> lppchk –v	Checks system file set consistency
MULTIBOS> installp -ug bos.games	Removes bos.games
MULTIBOS> oslevel –r	Reports recommended maintenance level for standby BOS

7.6.9 Booting the standby BOS

The **bootlist** command supports multiple BLVs.

As an example, to boot from disk hdisk0 and BLV bos_hd5, you would enter the following:

```
# bootlist –m normal hdisk0 blv=bos_hd5
```

After the system is rebooted from the standby BOS, the standby BOS logical volumes are mounted over the usual BOS mount points, such as /, /usr, /var, and so on.

The set of BOS objects, such as the BLV, logical volumes, file systems, and so on that are currently booted are considered the active BOS, regardless of logical volume names. The previously active BOS becomes the standby BOS in the existing boot environment.

7.6.10 Removing the standby BOS

The remove operation, using the -R flag, deletes all standby BOS objects, such as BLV, logical volumes, and file systems.

You can use the remove operation to make room for a new standby BOS, or to clean up a failed multibos installation. The remove operation performs standby tag verification on each object before removing it. The remove operation will only act on BOS objects that multibos created, regardless of name or label. You always have the option of removing additional BOS objects using standard AIX utilities, such as Rmlv, rmfs, rmps, and so on. The multibos remove operation performs the following steps:

1. All boot references to the standby BLV are removed.
2. The bootlist is set to the active BLV. You can skip this step using the -t flag.
3. Any mounted standby BLVs are unmounted.
4. Standby file systems are removed.
5. Remaining standby logical volumes are removed.

7.6.11 Relevant files and logs

All log files are kept in the /etc/multibos/logs directory. The following are examples of files that may be found in this directory:

- op.alog

  A circular alog file of all multibos operations
- scriptlog.<timestamp>.txt

  A log of commands being run during the current shell operation
- scriptlog.<timestamp>.txt.Z

  A compressed log of commands run during a previous shell operation

In addition, the bootlog contains redundant logging of all multibos operations that occur during boot (for example, the verify that attempts synchronization from inittab).

Multibos private data is stored in the /etc/multibos/data directory, the logical volume control block (LVCB) of logical volumes that were the source or target of a copy, and the /etc/filesystems entries that were the source or target of a copy. The following are examples of files found in the /etc/multibos/data directory:

acttag	The multibos tag for the active BOS
sbyfslist	The list of file systems private to the standby BOS
sbylvlist	The list of logical volumes private to the standby BOS
sbytag	The multibos tag for the standby BOS

The following may be seen in the fs field of the logical volumes that were the source or target of a copy:

```
mb=<TAG>:mbverify=<integer>
```

The following may be seen in /etc/filesystems as an attribute of file systems that were the source or target of a copy:

```
mb = <TAG>
```

The user should not modify multibos private data.

To prevent multibos operations from working simultaneously, the directory /etc/multibos/locks contains lock files. The following is an example of a file that may be found in this directory:

```
global_lock : The process ID (PID) of the currently running multibos
operation.
```

If a multibos operation exited unexpectedly and was not able to clean up, it may be necessary to remove this file. The user should check that the PID is not running before removing this file.

This inittab entry (if runnable) is removed upon removal of the standby BOS using **multibos -R**.

> **Note:** For more detailed information refer to the latest /usr/lpp/bos/README.multibos file and documentation regarding the **multibos** command in the AIX Information Center.

7.7 mksysb migration support (5300-03)

System p5 servers support only AIX 5L Version 5.2 and 5.3, leaving out versions 4.3 and 5.1. A client who has an older system running AIX 4.3 or AIX 5L Version 5.1 has no simple migration path to move to the new hardware and the new release of the operating system. This feature provides a method to restore a **mksysb** and then migrate it to a higher version of the operating system with a single operation using the available base operating system installation mechanisms (for example, tape, optical drive, network.)

A mksysb migration allows you to restore the **mksysb** from an old system to a system that supports AIX 5.3 and then migrate the **mksysb**.

> **Note:** A **mksysb** migration is not intended for systems that you can migrate with a traditional migration. This method allows you to bypass the hardware limitation by restoring the mksysb on the new hardware configuration and migrate it without running AIX 4.3. The resulting system will be running the new level of AIX 5L.

7.7.1 Supported levels of mksysb

Any mksysb at AIX 4330-10 and later is supported.

Table 7-3 lists the migration paths supported by the **mksysb** command.

Table 7-3 Supported migration paths matrix

From target (pre-migrated)	**To 5.1**	**To 5.2**	**To 5.3**
To output (post-migrated)			
FROM 4.3 (4330-10)	Yes	Yes	Yes
FROM 5.1	No	Yes	Yes
FROM 5.2	No	No	Yes
FROM 5.3	No	No	No

7.7.2 Customized bosinst.data file with a mksysb migration

A customized bosinst.data file is required to perform a mksysb migration installation. Your customized bosinst.data file must meet the following requirements to be used with a mksysb migration:

- The file must be provided using the supplementary diskette method.
- The file must be provided using the client file method (NIM).

The supplementary CD or DVD method is not supported for a mksysb migration.

> **Note:** The file must contain a new variable called MKSYSB_MIGRATION_DEVICE. This variable specifies the name of the device that contains the mksysb. For information about the supported values for this variable, see bosinst.data control_flow stanza descriptions.

The following variables in the CONTROL_FLOW stanza must be set as follows:

PROMPT	Must be set to no
INSTALL_METHOD	Must be set to migrate
EXISTING_SYSTEM_OVERWRITE	Must be set to yes
RECOVER_DEVICES	Must be set to no

A **mksysb** migration attempts to recover the sys0 attributed for the source system as specified in the **mksysb** ODM, but no other device-specific data is recovered from the source system.

Any user-supplied values for these variable are ignored.

The file should list the disks to be installed in the TARGET_DISK_DATA stanza to ensure that only those disks are used. A **mksysb** migration is a combination of an overwrite installation and a migration installation. The overwrite portion destroys all of the data on the target disks. The TARGET_DISK_DATA stanza must have enough information to clearly single out a disk. If you supply an empty TARGET_DISK_DATA stanza, the default disk for the platform is used, if available. The following examples show possible values for the TARGET_DISK_DATA stanza.

Example 7-4 shows a record with disk names only (two disks).

Example 7-4 Record with disk names only

```
target_disk_data:
                      PVID =
                      PHYSICAL_LOCATION =
                      CONNECTION =
                      LOCATION =
                      SIZE_MB =
                      HDISKNAME = hdisk0
target_disk_data:
                      PVID =
                      PHYSICAL_LOCATION =
                      CONNECTION =
                      LOCATION =
                      SIZE_MB =
                      HDISKNAME = hdisk1
```

Example 7-5 shows a record with the physical location specified (one disk).

Example 7-5 Record with physical locations only

```
target_disk_data:
                      PVID =
                      PHYSICAL_LOCATION = U0.1-P2/Z1-A8
                      CONNECTION =
                      LOCATION =
                      SIZE_MB =
                      HDISKNAME =
```

Example 7-6 shows a record by physical volume ID (PVID) (two disks).

Example 7-6 Record with PVIDs only

```
   target_disk_data:
                         PVID = 0007245fc49bfe3e
                         PHYSICAL_LOCATION =
                         CONNECTION =
                         LOCATION =
                         SIZE_MB =
                         HDISKNAME =
   target_disk_data:
                         PVID = 00000000a472476f
                         PHYSICAL_LOCATION =
                         CONNECTION =
                         LOCATION =
                         SIZE_MB =
                         HDISKNAME =
```

7.7.3 Performing a mksysb migration with CD or DVD installation

You can perform a **mksysb** migration with a CD or DVD installation of AIX 5.3.

Migration overview

The following is a list of the major tasks performed during a migration:

1. Target system boots from 5300-03.
2. MKSYSB_MIGRATION_DEVICE is found in customized bosinst.data.
3. The device is checked to verify that it exists.
4. The **mksysb migration** banner is shown on the TTY after the console is configured.
5. The image.data is checked for correctness and the target disks are inspected for availability and size.
6. The required mksysb migration files are restored from the **mksysb**, namely, image.data and /etc/filesystems. The user is prompted to swap CD/DVD if required for CD Boot installs with **mksysb** on CD/DVD. After restoration the user is asked to reinsert the product media.
7. The target logical volumes and file systems are created according to the image.data file and mounted according to the /etc/filesystems file.
8. The **mksysb** data is restored. The user is prompted to swap the CD/DVD if required for CD Boot installs with **mksysb** on CD/DVD.

9. The system now looks like an imported **rootvg**. The installation continues as a migration from here on.

Prerequisites

The following are the major prerequisites:

- All requisite hardware, including any external devices (such as tape, CD, or DVD-ROM drives) must be physically connected. For more information about connecting external devices, see the hardware documentation that accompanied your system.
- Before you begin the installation, other users who have access to your system must be logged off.
- Verify that your applications run on AIX 5L Version 5.3. Also, verify that your applications are binary compatible with AIX 5L Version 5.3. If your system is an application server, verify that there are no licensing issues. Refer to your application documentation or provider to verify on which levels of AIX your applications are supported and licensed. You can also check the AIX application availability guide at the following Web address:

  http://www-1.ibm.com/servers/aix/products/ibmsw/list/

- Verify that your hardware microcode is up-to-date.
- There must be adequate disk space and memory available. AIX 5L Version 5.3 requires 256–512 MB of memory and 2.2 GB of physical disk space. For additional release information, see the *AIX 5.3 Release Notes* at:

  http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=/com.ibm.aix.resources/53relnotes.htm

- Make a backup copy of your system software and data. For instructions on how to create a system backup, refer to *Creating system backups* in the *IBM AIX Information Center* at:

  http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=/com.ibm.aix.install/doc/insgdrf/create_sys_backup.htm

  This backup is used during the **mksysb** migration installation to restore your system files prior to migration.

- If the source system is available, run the pre-migration script on it. Ignore any messages that pertain to the hardware configuration of the source system because the migration takes place on the target system. Correct any other problems as recommended by the script.

Step 1: Prepare your system for installation

Prepare for migrating to the AIX 5.3 BOS by completing the following steps:

1. Insert the AIX Volume 1 disk into the media device.

2. Shut down the target system. If your machine is currently running, power it off by following these steps:

   a. Log in as the root user.

   b. Enter:

   ```
   shutdown -F
   ```

3. If your system does not automatically power off, place the power switch in the Off (0) position.

   > **Note:** You must not turn on the system unit until instructed to do so.

4. Turn on all attached external devices. External devices include the following:

   – Terminals

   – CD-ROM drives

   – DVD-ROM drives

   – Tape drives

   – Monitors

   – External disk drives

5. Turning on the external devices first is necessary so that the system unit can identify each peripheral device during the startup (boot) process.

6. If your MKSYSB_MIGRATION_DEVICE is a tape, insert the tape for the **mksysb** in the tape drive. If your MKSYSB_MIGRATION_DEVICE is a CD or DVD, and there is an additional CD or DVD drive on the system (other than the one being used to boot AIX), insert the **mksysb** CD or DVD in the drive to avoid being prompted to swap medias.

7. Insert your customized bosinst.data supplemental diskette in the diskette drive. If the **system** does not have a diskette drive, use the network installation method for mksysb migration.

8. Boot from your installation media.

9. The following steps migrate your current version of the operating system to AIX 5.3. If you are using an ASCII console that was not defined in your previous system, you must define it.

   a. Turn the system unit power switch from Off (0) to On (|).

   b. When the system beeps twice, press F5 on the keyboard (or 5 on an ASCII terminal). If you have a graphics display, you will see the keyboard icon on the panel when the beeps occur. If you have an ASCII terminal (also called a TTY terminal), you will see the word `keyboard` when the beeps occur.

> **Note:** If your system does not boot using the F5 key (or the 5 key on an ASCII terminal), refer to your hardware documentation for information about how to boot your system from an AIX product media.

The system begins booting from the installation media. The **mksysb** migration installation proceeds as an unattended installation (non-prompted) unless the MKSYSB_MIGRATION_DEVICE is the same CD or DVD drive as the one being used to boot and install the system. In this case, the user is prompted to switch the product media for the mksysb CD or DVD to restore the image.data and the /etc/filesystems file. After this happens the user is prompted to reinsert the product media and the installation continues. When it is time to restore the mksysb image, the same procedure repeats.

The BOS menus do not currently support **mksysb** migration, so they cannot be loaded. In a traditional migration, if there are errors that can be fixed by prompting the user for information through the menus, the BOS menus are loaded. If such errors or problems are encountered during **mksysb** migration, the installation asserts an error stating that the migration cannot continue displays. Depending on the error that caused the assertion, information specific to the error might be displayed. If the installation asserts, the LED shows 088.

Step 2: Finish the BOS migration

After the installation process begins, the Installing Base Operating System panel displays.

As the installation progresses, the numbers in the percentage complete field and the elapsed time field increment to indicate the installation status. After the **mksysb** is restored, the base run-time environment is installed, and status information about other software that is being installed displays. After the BOS installation is complete, the system automatically reboots.

After the system has restarted, you are prompted to configure your installation of the BOS.

> **Note:** If there is not enough space to migrate all of the usually migrated software, a collection of software called a migration bundle is available when you install additional software later. You must create additional disk space on the machine where you want to install the migration bundle, and then you can run **smit update_all** to complete the installation where the migration bundle is installed.

If you are not doing the installation from a graphics console, a Graphics_Startup bundle is created. If the pre-migration script ran on the source system, run the post-migration script and verify the output files.

7.7.4 Performing a mksysb migration with NIM installation

You can perform a **mksysb** migration with a NIM installation of AIX 5L:

1. All requisite hardware, including any external devices (such as tape, CD, or DVD-ROM drives) must be physically connected. For more information about connecting external devices, see the hardware documentation that accompanied your system.
2. Before you begin the installation, other users who have access to your system must be logged off.
3. Verify that your applications run on AIX 5.3. Also, verify that your applications are binary compatible with AIX 5.3. If your system is an application server, verify that there are no licensing issues. Refer to your application documentation or provider to verify on which levels of AIX your applications are supported and licensed. You can also check the AIX application availability guide at the following Web address:

   http://www.ibm.com/servers/aix/products/ibmsw/list/
4. Verify that your hardware microcode is up-to-date.
5. There must be adequate disk space and memory available. AIX 5.3 requires 256 MB of memory and 2.2 GB of physical disk space. For additional release information, see the AIX 5.3 Release Notes.
6. Make a backup copy of your system software and data. This backup is used during the **mksysb** migration installation to restore your system files prior to migration.
7. If the source system is available, run the pre-migration script on it. Ignore any messages that pertain to the hardware configuration of the source system because the migration takes place on the target system. Correct any other problems as recommended by the script.

Step 1: Prepare your system for installation

To prepare your system, verify that the following conditions are met:

- The target system must be a defined client to the NIM master.
- The required customized bosinst.data file described in the prerequisites is a NIM bosinst.data resource or supplied using the supplemental diskette method.

- To instruct the NIM master to start an installation of the client run the following command:

  ```
  # nim -o bos_inst -a source=rte -a spot=spot name -a lpp_source=lpp
  source name -a bosinst_data=bosinst_data resource name -a
  mksysb=mksysb name client_name
  ```

- The SPOT file and lpp_source file should be at the AIX 5.3 level.
- Alternatively, the **mksysb** can be allocated to the client first using a separate alloc operation. Then use the command line or **smitty nim** to perform a **bos_inst** operation on the client. If the **mksysb** is allocated to the client prior to the **bos_inst** operation, the specification of the **mksysb** is not required.

Step 2: Boot from your installation media

The following steps migrate your current version of the operating system to AIX 5.3. If you are using an ASCII console that was not defined in your previous system, you must define the console.

After the network boot image is transferred, the system begins booting using the network resources.

The **mksysb** migration installation proceeds as an unattended installation (non-prompted).

The BOS menus do not currently support **mksysb** migration, so they cannot be loaded. In a traditional migration, if there are errors that can be fixed by prompting the user for information through the menus, the BOS menus are loaded. If such errors or problems are encountered during **mksysb** migration, the installation asserts and an error stating that the migration cannot continue displays. Depending on the error that caused the assertion, information specific to the error might be displayed. If the installation asserts, the LED shows 088.

Step 3: Finish the BOS migration

After the installation process begins, the Installing Base Operating System panel displays.

As the installation progresses, the numbers in the percentage complete field and the elapsed time field increment to indicate the installation status. After the mksysb is restored, the base run-time environment is installed, and status information about other software that is being installed displays. After the BOS installation is complete, the system automatically reboots.

After the system has restarted, you are prompted to configure your installation of the BOS.

> **Note:** If there is not enough space to migrate all of the usually migrated software, a collection of software called a migration bundle is available when you install additional software later. You must create additional disk space on the machine where you want to install the migration bundle, and then you can run `smit update_all` to complete the installation where the migration bundle is installed.

If you are not doing the installation from a graphics console, a Graphics_Startup bundle is created.

If the pre-migration script ran on the source system, run the post-migration script and verify the output files.

7.7.5 The nimadm command

The Network Install Manager Alternate Disk Migration (`nimadm`) command is a utility that uses NIM resources to perform the following tasks:

- Create a copy of rootvg to an unused NIM client hdisk and simultaneously migrate the rootvg copy to a new version or release level of AIX 5L.
- Using a copy of rootvg, create a NIM mksysb resource that has been migrated to a new version or release level of AIX 5L.
- Migrate an existing NIM mksysb resource to a new version or release level of AIX 5L.

Enhancements were made to the `nimadm` command in 5300-03 designed to minimize downtime associated with operating system migrations by performing a migration utilizing an AIX Version 4.3 or AIX 5L `mksysb` image.

The `nimadm` command performs an alternate disk migration to a new version or release of AIX using NIM resources. Currently, this utility supports the following scenarios:

- NFS migration
- Disk caching migration

The following are new scenarios supported starting with AIX 5300-03:

- `mksysb` to client
- Client to `mksysb`
- `mksysb` to `mksysb`

Example scenarios

The following sections describe common scenarios you may find handy.

mksysb to client

A defined **mksysb** resource on the NIM environment is restored onto alternate file systems on the NIM Master. The data is then migrated and written out to alternate file systems on the client's alternate disk. Free disks are required on the client.

The syntax for this operation is:

```
nimadm -s<spot> -l<lpp_source> -c <client> -d<disks(s)> -j<cache vg>
-T <mksysb NIM resource>
```

Client to mksysb

Copies of the client's file systems are made on the NIM Master. The data is then migrated and backed up to produce a **mksysb** resource on the NIM environment.

The syntax for this operation is:

```
nimadm -s<spot> -l<lpp_source> -c <client>  -j<cache vg>
-O <mksysb>
```

mksysb to mksysb

A NIM **mksysb** resource is restored to alternate file systems on the master. The data is migrated and then backed up to produce a NIM **mksysb** resource at the new level.

The syntax for this operation is:

```
nimadm -s<spot> -l<lpp_source>  -j<cache vg>
-T <mksysb NIM resource> -O <mksysb NIM resource>
```

Advantages to using nimadm

There are several advantages to using **nimadm** over a conventional migration:

- Reduced migration downtime

  The migration is performed while the system is up and functioning normally. There is no requirement to boot from install media, and the majority of processing occurs on the NIM master.

- Reduced recovery time

  – Using **nimadm** facilitates quick recovery in the event of a migration failure. Since **nimadm** uses the AIX 5L alt_disk_install to create a copy of rootvg, all changes are performed to the copy (**altinst_rootvg**).

  - In the unlikely event of serious migration installation failure, the failed migration is cleaned up and there is no need for the administrator to take further action.
  - In the event of a problem with the new (migrated) level of AIX 5L, the system can be quickly returned to the pre-migration operating system by booting from the original disk.
- Flexibility and customization

  nimadm allows a high degree of flexibility and custimization in the migration process. This is done with the use of optional NIM custimization resources: image_data, bosinst_data, exclude_files, pre-migration script, installp_bundle, and post-migration script.

To use the **nimadm** command, the following requirements must be met:

1. A configured NIM Master running AIX 5L Version 5.3. ML03 or later.
2. The NIM master must have the same level of bos.alt_disk_install.rte installed in its rootvg and the SPOT that will be used to perform the migration.
3. The selected lpp_source NIM resource and the selected SPOT NIM resource must match the AIX 5L level to which you are migrating.
4. The NIM Master must be at the same or higher AIX 5L level as the level being migrated to.
5. The NIM client must be registered with the master as a stand-alone NIM client.
6. The NIM master must be able to execute remote commands on the client using the rshd protocol.
7. A reliable network, which can facilitate large amounts of NFS traffic, must exist between the NIM master and the client. The NIM master and client must be able to perform NFS mounts and read/write operations.
8. The client's hardware and software should support the AIX 5L level that is being migrated to and meet all other conventional migration requirements.

If you cannot meet requirements 1–7, you will need to perform a conventional migration. If you cannot meet requirement 8, then migration is not possible.

When planning to use **nimadm**, ensure that you are aware of the following considerations:

- If the NIM client rootvg has TCB turned on, you will need to either permanently disable it or perform a conventional migration. This situation exists because TCB needs to access file metadata that is not visible over NFS.
- All NIM resources used by **nimadm** must be local to the NIM master.

- Although there is almost no interference with the client's active rootvg during the migration, the NIM client may experience minor performance decrease due to increased disk input/output, NFS biod activity, and some CPU usage associated with alt_disk_install cloning.
- A reliable network and NFS tuning will assist in optimizing **nimadm** network performance.

7.7.6 The nim_move_up command

The **nim_move_up** is a command that facilitates the enablement of new hardware (namely POWER5 or later servers) in AIX environments. For more information see 7.3.3, “Smit menu for nim_move_up” on page 201.

7.7.7 Debugging a mksysb migration installation

If a **mksysb** migration assert takes place, the LED will show 088.

If the error takes place after the console is available (you can see messages on the panel) then enable the BOSINST_DEBUG variable in the customized bosinst.data by changing its value to yes and restart the process.

If the error takes place before the console is available (you see the 088 but nothing has been displayed on the panel), you will need to enable the KDB kernel debugger and capture debug output.

> **Note:** For more information and examples about **mksysb** migration installations, refer to the *IBM AIX Information Center* publication at:
>
> http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=/com.ibm.aix.install/doc/insgdrf/mksysbmigration.htm

7.8 mksysb enhancements (5300-01)

The method used by the **mksysb** command to restore data through system backups has changed. Enhancements were added to more fully restore customized data so that a restored system more closely resembles the system at the time the backup was performed. This occurs when restoring a backup on the system that the backup originated from.

These enhancements were added to reduce the amount of additional work that sometimes needs to occur to restore devices to their customized configuration at the time of backup.

If devices were removed from or replaced on the system after the backup was created, that information is restored when you are installing a backup, and the system shows these devices in a defined state.

These enhancements do not affect installing the backup onto other systems or cloning.

7.9 DVD install media support for AIX 5L (5300-02)

DVD install media support is now available for AIX 5L.

Abbreviations and acronyms

ABI	Application Binary Interface
AC	Alternating Current
ACL	Access Control List
AFPA	Adaptive Fast Path Architecture
AIO	Asynchronous I/O
AIX	Advanced Interactive Executive
APAR	Authorized Program Analysis Report
API	Application Programming Interface
ARP	Address Resolution Protocol
ASMI	Advanced System Management Interface
BFF	Backup File Format
BIND	Berkeley Internet Name Domain
BIST	Built-In Self-Test
BLV	Boot Logical Volume
BOOTP	Boot Protocol
BOS	Base Operating System
BSD	Berkeley Software Distribution
CA	Certificate Authority
CATE	Certified Advanced Technical Expert
CD	Compact Disk
CDE	Common Desktop Environment
CD-R	CD Recordable
CD-ROM	Compact Disk-Read Only Memory
CEC	Central Electronics Complex
CHRP	Common Hardware Reference Platform
CLI	Command Line Interface
CLVM	Concurrent LVM
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CSM	Cluster Systems Management
CUoD	Capacity Upgrade on Demand
DCM	Dual Chip Module
DES	Data Encryption Standard
DGD	Dead Gateway Detection
DHCP	Dynamic Host Configuration Protocol
DLPAR	Dynamic LPAR
DMA	Direct Memory Access
DNS	Domain Naming System
DR	Dynamic Reconfiguration
DRM	Dynamic Reconfiguration Manager
DVD	Digital Versatile Disk
EC	EtherChannel
ECC	Error Checking and Correcting
EOF	End of File
EPOW	Environmental and Power Warning
ERRM	Event Response resource manager
ESS	Enterprise Storage Server®
F/C	Feature Code
FC	Fibre Channel
FCAL	Fibre Channel Arbitrated Loop

FDX	Full Duplex
FLOP	Floating Point Operation
FRU	Field Replaceable Unit
FTP	File Transfer Protocol
GDPS®	Geographically Dispersed Parallel Sysplex™
GID	Group ID
GPFS™	General Parallel File System™
GUI	Graphical User Interface
HACMP	High Availability Cluster Multiprocessing
HBA	Host Bus Adapters
HMC	Hardware Management Console
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
Hz	Hertz
I/O	Input/Output
IBM	International Business Machines
ID	Identification
IDE	Integrated Device Electronics
IEEE	Institute of Electrical and Electronics Engineers
IP	Internetwork Protocol
IPAT	IP Address Takeover
IPL	Initial Program Load
IPMP	IP Multipathing
ISV	Independent Software Vendor
ITSO	International Technical Support Organization
IVM	Integrated Virtualization Manager
JFS	Journaled File System
L1	Level 1
L2	Level 2
L3	Level 3
LA	Link Aggregation
LACP	Link Aggregation Control Protocol
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LED	Light Emitting Diode
LMB	Logical Memory Block
LPAR	Logical Partition
LPP	Licensed Program Product
LUN	Logical Unit Number
LV	Logical Volume
LVCB	Logical Volume Control Block
LVM	Logical Volume Manager
MAC	Media Access Control
Mbps	Megabits Per Second
MBps	Megabytes Per Second
MCM	Multichip Module
ML	Maintenance Level
MP	Multiprocessor
MPIO	Multipath I/O
MTU	Maximum Transmission Unit
NFS	Network File System
NIB	Network Interface Backup
NIM	Network Installation Management
NIMOL	NIM on Linux
NVRAM	Non-Volatile Random Access Memory
ODM	Object Data Manager
OSPF	Open Shortest Path First
PCI	Peripheral Component Interconnect
PIC	Pool Idle Count
PID	Process ID
PKI	Public Key Infrastructure
PLM	Partition Load Manager

POST	Power-On Self-test
POWER	Performance Optimization with Enhanced Risc (Architecture)
PPC	Physical Processor Consumption
PPFC	Physical Processor Fraction Consumed
PTF	Program Temporary Fix
PTX®	Performance Toolbox
PURR	Processor Utilization Resource Register
PV	Physical Volume
PVID	Physical Volume Identifier
PVID	Port Virtual LAN Identifier
QoS	Quality of Service
RAID	Redundant Array of Independent Disks
RAM	Random Access Memory
RAS	Reliability, Availability, and Serviceability
RCP	Remote Copy
RDAC	Redundant Disk Array Controller
RIO	Remote I/O
RIP	Routing Information Protocol
RISC	Reduced Instruction-Set Computer
RMC	Resource Monitoring and Control
RPC	Remote Procedure Call
RPL	Remote Program Loader
RPM	Red Hat Package Manager
RSA	Rivet, Shamir, Adelman
RSCT	Reliable Scalable Cluster Technology
RSH	Remote Shell
SAN	Storage Area Network
SCSI	Small Computer System Interface
SDD	Subsystem Device Driver
SMIT	System Management Interface Tool
SMP	Symmetric Multiprocessor
SMS	System Management Services
SMT	Simultaneous Muli-threading
SP	Service Processor
SPOT	Shared Product Object Tree
SRC	System Resource Controller
SRN	Service Request Number
SSA	Serial Storage Architecture
SSH	Secure Shell
SSL	Secure Socket Layer
SUID	Set User ID
SVC	SAN Virtualization Controller
TCP/IP	Transmission Control Protocol/Internet Protocol
TSA	Tivoli System Automation
UDF	Universal Disk Format
UDID	Universal Disk Identification
VG	Volume Group
VGDA	Volume Group Descriptor Area
VGSA	Volume Group Status Area
VIPA	Virtual IP Address
VLAN	Virtual Local Area Network
VP	Virtual Processor
VPD	Vital Product Data
VPN	Virtual Private Network
VRRP	Virtual Router Redundancy Protocol
VSD	Virtual Shared Disk
WLM	Workload Manager

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 IBM Redbooks” on page 244. Note that some of the documents referenced here may be available in softcopy only.

- *AIX 5L Differences Guide Version 5.3 Edition*, SG24-7463
- *Advanced POWER Virtualization on IBM eServer p5 Servers: Architecture and Performance Considerations*, SG24-5768
- *AIX 5L Practical Performance Tools and Tuning Guide*, SG24-6478
- *Effective System Management Using the IBM Hardware Management Console for pSeries*, SG24-7038
- *IBM System p Advanced POWER Virtualization Best Practices*, REDP-4194
- *Implementing High Availability Cluster Multi-Processing (HACMP) Cookbook*, SG24-6769
- *Introduction to pSeries Provisioning*, SG24-6389
- *Linux Applications on pSeries*, SG24-6033
- *Managing AIX Server Farms*, SG24-6606
- *NIM from A to Z in AIX 5L*, SG24-7296
- *Partitioning Implementations for IBM eServer p5 Servers*, SG24-7039
- *A Practical Guide for Resource Monitoring and Control (RMC)*, SG24-6615
- *Integrated Virtualization Manager on IBM System p5*, REDP-4061

Other publications

These publications are also relevant as further information sources:

- The following types of documentation are located through the Internet

  `http://www.ibm.com/servers/eserver/pseries/library`

  – User guides

  – System management guides

  – Application programmer guides

  – All commands reference volumes

  – Files reference

  – Technical reference volumes used by application programmers

- Detailed documentation about the Advanced POWER Virtualization feature and the Virtual I/O Server

  `https://www14.software.ibm.com/webapp/set2/sas/f/vios/documentation/home.html`

- *AIX 5L V5.3 Partition Load Manager Guide and Reference*, SC23-4883
- *Linux for pSeries installation and administration (SLES 9)*

  `http://www-128.ibm.com/developerworks/linux/library/l-pow-pinstall/`

- *Linux virtualization on POWER5: A hands-on setup guide*

  `http://www-128.ibm.com/developerworks/edu/l-dw-linux-pow-virutal.html`

- *POWER5 Virtualization: How to set up the SUSE Linux Virtual I/O Server*

  `http://www-128.ibm.com/developerworks/eserver/library/es-susevio/`

Online resources

These Web sites and URLs are also relevant as further information sources:

- AIX 5L and Linux on POWER community

  `http://www-03.ibm.com/systems/p/community/`

- Capacity on Demand

  `http://www.ibm.com/systems/p/cod/`

- IBM Advanced POWER Virtualization on IBM System p Web page

  http://www.ibm.com/systems/p/apv/
- IBM eServer pSeries and AIX Information Center

  http://publib16.boulder.ibm.com/pseries/index.htm
- IBM System Planning Tool

  http://www.ibm.com/servers/eserver/support/tools/systemplanningtool/
- IBM Systems Hardware Information Center

  http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/index.jsp
- IBM Systems Workload Estimator

  http://www-912.ibm.com/supporthome.nsf/document/16533356
- Latest *Multipath Subsystem Device Driver User's Guide*

  http://www.ibm.com/support/docview.wss?rs=540&context=ST52G7&uid=ssg1S7000303
- Novell SUSE Linux Enterprise Server information

  http://www.novell.com/products/server/index.html
- SCSI T10 Technical Committee

  http://www.t10.org
- SDDPCM software download page

  http://www.ibm.com/support/docview.wss?uid=ssg1S4000201
- SDD software download page

  http://www.ibm.com/support/docview.wss?rs=540&context=ST52G7&dc=D430&uid=ssg1S4000065&loc=en_US&cs=utf-8&lang=en
- Service and productivity tools for Linux on POWER

  http://techsupport.services.ibm.com/server/lopdiags
- Silicon-on-insulator (SOI) technology

  http://www.ibm.com/chips/technology/technologies/soi/
- VIOS supported environment

  http://techsupport.services.ibm.com/server/vios/documentation/datasheet.html
- Virtual I/O Server documentation

  http://techsupport.services.ibm.com/server/vios/documentation/home.html

- Virtual I/O Server home page

  http://techsupport.services.ibm.com/server/vios/home.html
- Virtual I/O Server home page (alternate)

  http://www14.software.ibm.com/webapp/set2/sas/f/vios/home.html
- Virtual I/O Server supported hardware

  http://www14.software.ibm.com/webapp/set2/sas/f/vios/documentation/datasheet.html
- Virtual I/O Server Support Page

  http://techsupport.services.ibm.com/server/vios/download/home.html

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, 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

AIX 5L Differences Guide Version 5.3 Addendum

(0.2”spine)
0.17”<->0.473”
90<->249 pages

IBM®

AIX 5L Differences Guide Version 5.3 Addendum

Redbooks

Where the AIX 5L Differences Guide left off - ML 5300-01 through TL 5300-05

An expert's guide to the between release enhancements

AIX 5L Version 5.3 enhancements explained

This IBM Redbooks publication focuses on the differences introduced in AIX 5L Version 5.3 since the initial AIX 5L Version 5.3 release. It is intended to help system administrators, developers, and users understand these enhancements and evaluate potential benefits in their own environments.

Since AIX 5L Version 5.3 was introduced, many new features including JFS2, LDAP, trace and debug, installation and migration, NFSv4, and performance tools enhancements were introduced. There are many other improvements offered through updates for AIX 5L Version 5.3, and you can explore them in this book.

For clients who are not familiar with the base enhancements of AIX 5L Version 5.3, a companion publication, *AIX 5L Differences Guide Version 5.3 Edition*, SG24-7463, is available.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-7414-00 ISBN 0738486299