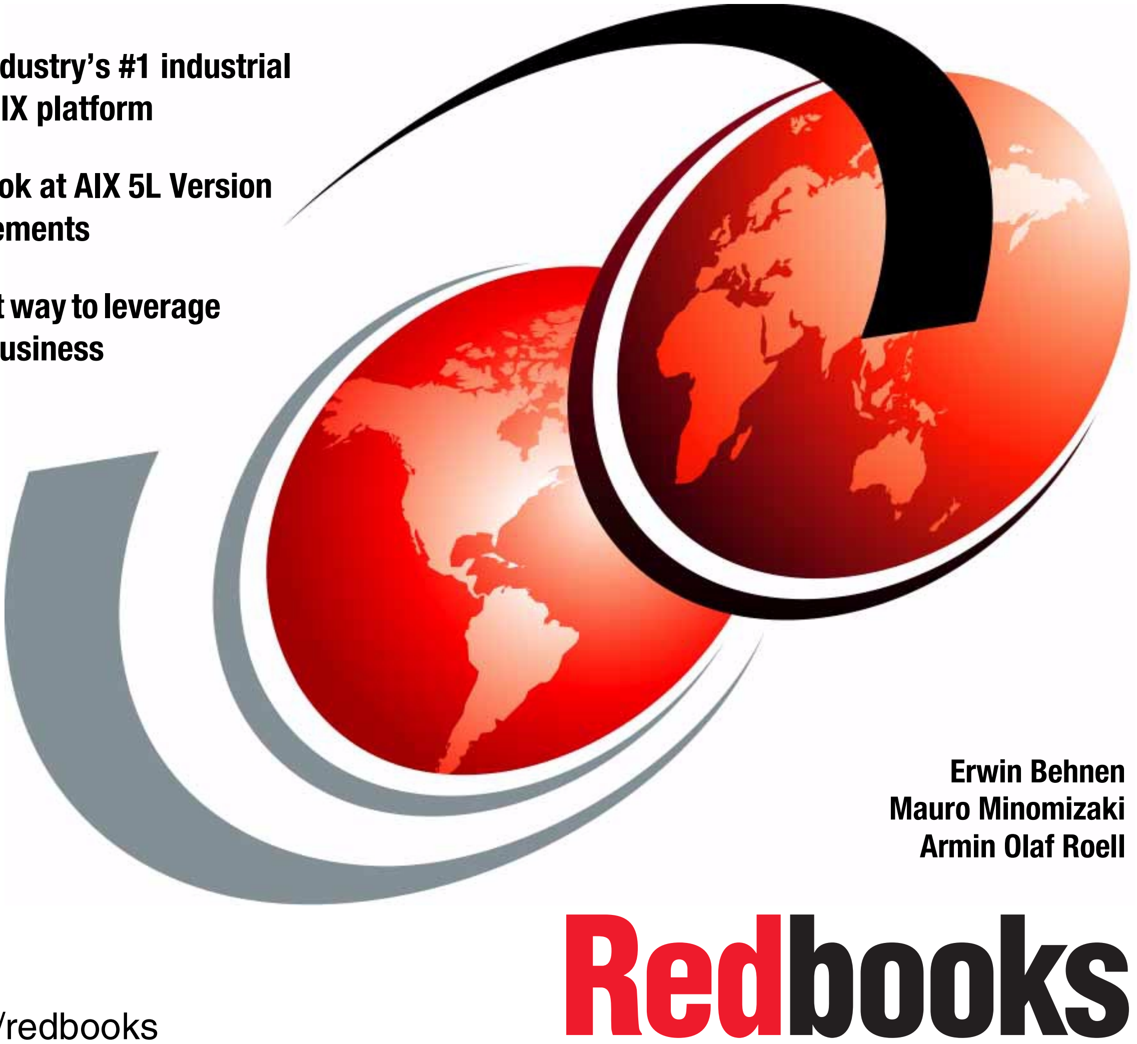
AIX 5L Differences Guide Version 5.0 Edition

AIX - The Industry's #1 industrial strength UNIX platform

An inside look at AIX 5L Version 5.0 enhancements

An excellent way to leverage your UNIX business



Erwin Behnen
Mauro Minomizaki
Armin Olaf Roell

Redbooks

ibm.com/redbooks

SG24-5765-00

International Technical Support Organization

AIX 5L Differences Guide
Version 5.0 Edition

December 2000

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix B, “Special notices” on page 229.

First Edition (December 2000)

This edition applies to AIX 5L for POWER Version 5.0 Early Adopters Release, program number 5799-JOE, available through an i-listed PRPQ and AIX 5L for Itanium-based systems Version 5.0 Software Developers Release available through the beta program.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. JN9B Building 003 Internal Zip 2834
11400 Burnet Road
Austin, Texas 78758-3493

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Figures

Tables

Preface

This redbook focuses on the latest enhancements introduced in AIX 5L Version 5.0. It is intended to help system administrators, developers, and users understand these enhancements and evaluate potential benefits in their own environments.

AIX 5L is available for POWER and Itanium-based systems. The initial offering of AIX 5L for POWER is available as a no-charge i-listed PRPQ. AIX 5L for Itanium-based systems is covered under the beta program. Both platforms were developed from the same common code base.

AIX 5L introduces many new features, including virtual IP, quality of service enhancements, enhanced error logging, dynamic paging space reduction, hot-spare disk management, advanced Workload Manager, JFS2, and others. The availability of an improved Web-based System Manager continues AIX's move towards a standard, unified interface for system tools. There are many other enhancements available with AIX 5L, and you can explore them all in this redbook.

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

The team that wrote this redbook

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

Erwin Behnen is an Advisory Engineer in Germany. He has eight years of experience in supporting VLSI design systems running on the RS/6000 under AIX. He holds a Dr.-Ing. degree in Electrical Engineering from Technische Universität Braunschweig. His areas of expertise include VLSI Custom Circuit Design Tools, Design Environment Architecture, AFS and AIX.

Mauro Minomizaki is an RS/6000 System Engineer in Brazil. He has four years of experience with AIX and the RS/6000 systems field. He holds a degree in Data Processing I/T from the Sao Paulo Institute of Technology (FATEC-SP). He has worked at IBM for six years. His areas of expertise include RS/6000 systems, RS/6000 SP systems, AIX, performance tuning, sizing and capacity planning, and problem determination. He is a co-author on the S-Series Handbook.

Armin Olaf Roell joined IBM Germany in 1995 and works as an RS/6000 system engineer responsible for pre-sales technical support. At present, he is the team leader for a World-Wide Technical Focus Group and specializes in AIX operating system-related matters.

The project that produced this publication was managed by:

Scott Vetter	IBM Austin

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

Andre L. Albot	IBM Austin
Jack Alford	IBM Austin
James P. Allen	IBM Austin
Sandy Amin	IBM Austin
Greg Birgen	IBM Austin
Matt Brandyberry	IBM Austin
Deanna Quigg Brown	IBM Austin
Bill Buros	IBM Austin
Scott Carroll	IBM Austin
Corene Casper	IBM Austin
Ufuk Celikkan	IBM Austin
Daisy Chang	IBM Austin
Carol Christensen	IBM Austin
David Clissold	IBM Austin
Helmut Cossmann	IBM Heidelberg
Richard Cutler	IBM Austin
Saravanan Devendran	IBM Austin
Bimal Doshi	IBM Austin
Greg Flaig	IBM Austin
Nathan Fontenot	IBM Austin
Shevaun Fontenot	IBM Austin
Douglas M. Freimuth	IBM Watson Research
Masahiro Furutera	IBM Japan

Denise Genty	IBM Austin
Nigel Griffiths	IBM U.K.
Michael S. Harrell	IBM Austin
Julianne Haugh	IBM Austin
Duen-wen Hsiao	IBM Austin
Megumi Iwata	IBM Japan
Greg Laib	IBM Poughkeepsie
Jim Lentz	IBM Austin
Susan Machutt	IBM Austin
Michael Mall	IBM Austin
Gerald McBrearty	IBM Austin
Brian McCorkle	IBM Austin
Dave McCracken	IBM Austin
Hye-Young McCreary	IBM Austin
Rajeev Mishra	IBM Austin
Hajime Mita	IBM Tokyo
Steve Nasypany	IBM Austin
Grover Neuman	IBM Austin
Frank L. Nichols III	IBM Austin
Subhrata Parichhah	IBM India
Marilyn Payne	IBM Austin
Steve Peckham	IBM Austin
Rick Poston	IBM Austin
Prasad V. Potluri	IBM Austin
Ruben Ramirez	IBM Austin
Tony Ramirez	IBM Austin
Krystal Rothaupt	IBM Poughkeepsie
Kenneth Rozendal	IBM Austin
Jim Shaffer	IBM Austin
Rakesh Sharma	IBM Austin
Danling Shi	IBM Austin

Johnny Shieh	IBM Austin
Nancy L. Springen	IBM Austin
Randy Swanberg	IBM Austin
Kurt Taylor	IBM Austin
Marvin Toungate	IBM Austin
Kim Tran	IBM Austin
Girish Vazzalwar	IBM Austin
Wade Wallace	IBM Austin
Jason Wu	IBM Austin
Lakshmi Yerneni	IBM Austin
Gina Yuan	IBM Poughkeepsie

Comments welcome

Your comments are important to us!

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

- Fax the evaluation form found in "IBM Redbooks review" on page 255 to the fax number shown on the form.
- Use the online evaluation form found at **ibm.com**/redbooks
- Send your comments in an Internet note to redbook@us.ibm.com

Chapter 1. AIX 5L introduction and overview

The next generation of AIX 5L is a unique and open enterprise class UNIX operating system incorporating technology from software developers around the world. AIX 5L provides customers with the benefits of business flexibility and performance for both IBM POWER and Intel Itanium processors.

The initial offering of AIX 5L on POWER is distributed as a no-charge i-listed PRPQ with limited support. AIX 5L for Itanium-based systems is covered under the terms and conditions of the beta program.

This initial release of AIX 5L for POWER or Itanium-based systems is not part of the software subscription program. The AIX Bonus Pack will not be part of this initial release however a new Expansion Pack will be included for both the POWER and Itanium-based platforms. The AIX Fast Connect file and print server will be included for this time only in the AIX 5L Early Adopters Release on POWER.

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

- AIX 5L kernel and application development differences

  A summary of these differences can be found in Section 1.1, “AIX 5L kernel and application development differences summary” on page 5.

- Development environment and tools enhancements
  - An improved print function for DBX that provides more legible output is explained in Section 2.2, “DBX enhancements” on page 9.
  - Pthread enhancements including application-level access to the pthread debug library, a new method to unregister atfork handlers, and a read/write locking enhancement is explained in Section 2.3, “Pthread differences and enhancements” on page 11.
  - Core file enhancements that allow an application to core dump without termination is discussed in Section 2.4, “Lightweight core file support” on page 12.
  - Enhancements to the KDB kernel debugger including a new way to load it and additional subcommands is discussed in Section 2.5, “KDB kernel debugger and kdb command enhancements” on page 12.
  - Enhancements that allow application level control over the scheduler during critical sections to prevent loss of context is explained in Section 2.6, “Context switch avoidance” on page 15.

- 32-bit application scaling enhancements are discussed in Section 2.7, “Very large program support” on page 16.
- A new kornshell, ksh93, is discussed in Section 2.8, “Kornshell enhancements” on page 17.
- Enhancements in malloc provide faster access to free memory for applications. This is discussed in Section 2.9, “Malloc enhancements” on page 18.
- An improved `restore` command helps you recover sparse database files, as explained in Section 2.10, “Non-sparseness support for the restore command” on page 19.
- The `pax` command includes support for large files, such as dumps greater than 2 GB, as discussed in Section 2.11, “Enhancements for pax” on page 19.
- AIX 5L introduces the IBM AIX Developer Kit, JAVA 2 Technology Edition, Version 1.3.0, as discussed in Section 2.12, “Java currency” on page 20.

- LVM and file system enhancements
  - New LVM hot-spare disk support, new `redefinevg`, `migratelp`, and `recreatevg` commands, new logical track group sizes, and hot spot management are discussed in Section 3.1, “Enhancements for volume group commands” on page 21.
  - The /proc file system is discussed in Section 3.2, “The /proc file system” on page 27.
  - The JFS 2 is introduced in Section 3.3, “Journaled File System 2” on page 31. It provides the capability to store much larger files than JFS, in a more efficient manner.
  - NFS statd, AutoFS, and CacheFS enhancements are discussed in Section 3.4, “NFS statd multithreading” on page 43, Section 3.5, “Multithreaded AutoFS” on page 44, and Section 3.6, “Cache file system enhancements” on page 44.
  - A new passive mirror write consistency check can improve disk mirroring performance as discussed in Section 3.7, “Passive mirror write consistency check” on page 44.
  - Updates to LVM libraries for multithreaded applications are discussed in Section 3.8, “Thread-safe liblvm.a” on page 46.

- System management and utility enhancements
  - An expanded set of devices that allow for simultaneous multiple device configuration during system startup is discussed in Section 4.1, “Fast device configuration enhancement” on page 47.
  - New ways for you to dynamically manage your paging areas, such as deactivating a paging space with the `swapoff` command or decreasing its size is discussed in Section 4.2, “Paging space enhancements” on page 47.
  - Updates to the error log provide a more concise view of system errors, such as a link between the error log and diagnostics, or the elimination of duplicate errors, is described in Section 4.3, “Error log enhancements” on page 50.
  - AIX 5L provides a set of resources to be monitored and actions to be taken at defined events providing automatic monitoring and recovery of select critical system resources. For more information, see Section 4.4, “Resource Monitoring and Control (RMC)” on page 53.
  - Shutdown logging is available, as described in Section 4.5, “Shutdown enhancements” on page 63.
  - New methods to diagnose system errors through dump improvements is described in Section 4.6, “System dump enhancements” on page 63.
  - The ability to recover from certain system hangs is covered in Section 4.7, “System hang detection” on page 65.
  - Enhancements to performance tools, including `truss`, `iostat`, and `vmstat`, are discussed in Section 4.8, “Performance Analysis Tools enhancements” on page 67.
  - Workload Manager continues to receive improvements as discussed in Section 4.10, “Workload Manager” on page 78.
  - The new System V Release 4 print subsystem is discussed in Section 4.11, “System V Release 4 print subsystem” on page 111.
  - Web-Based System Manager receives major usability improvements with a much improved architecture and usability enhancements, such as accelerator keys. A discussion of all the enhancements can be found in Section 4.12, “Web-based System Manager for AIX 5L” on page 139.
  - Security and User authentication and LDAP enhancements are discussed in Sections 4.13, “User and group integration” on page 159, 4.14, “IBM SecureWay Directory Version 3.2” on page 168, and 4.15, “LDAP name resolution enhancement” on page 170.

- A new documentation search engine to handle single and doublebyte searches together is discussed in Section 4.16, “Documentation search-engine enhancement” on page 178.
- AIX is Tivoli ready, as discussed in Section 4.17, “Tivoli readiness” on page 178.
- An updated Welcome Center will teach you what is available for AIX in the CATIA market. For more information, see Section 4.18, “CATIA Welcome Center” on page 179.

- Networking Enhancements
  - The demand for QoS arises from applications such as digital audio/video or real-time applications and the need to manage bandwidth resources for arbitrary administratively-defined traffic classes. For more information, see Section 5.1, “Quality of service support” on page 181.
  - Together, multipath routing and dead gateway detection provide automatic selection of alternate network pathways that provide significant improvements in network availability. For more information, see Section 5.2, “TCP/IP routing subsystem enhancements” on page 186.
  - With Virtual IP Address, the application is bound to a virtual IP address, not a real network interface that can fail. When a network or network interface failure is detected (using routing protocols or other schemes), a different network interface can be used by modifying the routing table without affecting application operation. For more information, see Section 5.3, “Virtual IP address support” on page 207.
  - Enhancements to the Network Buffer Cache and HTTP GET kernel extension provide class leading Web server performance. For more information, see Sections 5.4, “Network Buffer Cache dynamic data support” on page 211, and 5.5, “HTTP GET kernel extension enhancements” on page 214.
  - Applications can be modified to capture network data packets through a new interface, explained in Section 5.6, “Packet capture library” on page 218.
  - To allow more flexible development of firewall software, AIX provides additional hooks, as described in Section 5.7, “Firewall hooks enhancements” on page 220.
  - PC Interoperability using Fast Connect File and Print Services provides support for Windows 2000, improved user and name mapping, share

options, WTS support, better performance, and more as discussed in Section 5.8, “Fast Connect enhancements” on page 222.

- A list of packages and filesets that are not part of the AIX 5L Itanium-based offering is provided in Appendix A, “AIX 5L POWER and Itanium-based fileset differences” on page 227.

1.1 AIX 5L kernel and application development differences summary

The AIX development team made every effort possible to make AIX 5L for the POWER and Itanium-based platforms appear and function identically, there are, however, a few unavoidable differences due to the underlying hardware.

The following list provides a summary of the major differences, from a kernel and application development point of view, between POWER and Itanium-based systems:

- The most influential difference is the use of the IA64 instruction set architecture (ISA). Itanium-based platforms operate in little endian mode. The Itanium-based ISA arranges instructions into bundles and groups. It also contains instruction *predication* to enable explicit parallelism in instruction execution.
- Itanium-based AIX has a 64-bit kernel. There is no 32-bit kernel for Itanium-based systems.
- Common header files contain #ifdef _ia64 to denote differences between Itanium-based and POWER structures.
- Itanium-based systems have a different machine register context. The MST, signal context, and jump buffers all contain different context. The user space debugger for Itanium-based systems also displays the IA64 registers.
- Itanium-based systems have a different application binary interface (ABI) than POWER. Linkage and parameter passing conventions are different from POWER due to the machine register context differences between the platforms.
- The Itanium-based machine architecture contains a register stack engine (RSE). The RSE requires a second stack area for every thread. The RSE stack is allocated by the operating system and programs typically do not need to be aware of it. There are environment variables available to applications which need to control the size of the RSE stack.
- Itanium-based systems do not provide the ptrace() function. Itanium-based AIX provides the /proc file system for debugging and tracing user applications.

- 64-bit Itanium-based applications receive an exception if they try to fetch from or store into address 0. This is different that 64-bit POWER applications which can fetch from location 0. 32-bit Itanium-based applications can fetch from location 0.
- The layout of the user address space is different between Itanium-based AIX and POWER. For example, the addresses where shared libraries and shmat'ed areas reside is different between the platforms. There are also differences in the kernel address space layout.
- Itanium-based AIX uses the ELF object file format for executable programs. ELF object file utilities are provided, such `as`,`ar`, `nm`, `ldd`, and others. The object file utilities generally do not share options with the POWER versions. Shared libraries reside in different directories. Shared libraries end in a .so suffix versus .a for POWER. Lazy binding is the default symbol binding mode. The runtime linker is contained in libc.so.
- There are a number of low level differences in the system which are typically not visible to applications. These differences include booting, system initialization, virtual address translation hardware, I/O interrupt hardware, exception interrupts, DMA, Intel firmware callbacks, machine check support, and others.
- Itanium-based AIX has a different kernel debugger than POWER. The Itanium-based kernel debugger commands are different from POWER.

1.2 AIX 5L 64-bit kernel overview

AIX 5L provides a new, scalable, 64-bit kernel that:

- Provides simplified data and I/O device sharing for multiple applications on the same system.
- Provide more scalable kernel extensions and device drivers that make full use of the kernel’s system resources and capabilities.
- Allow for future hardware development that will provide even larger single image systems ideal for server consolidation or workload scalability.

This following sections provide a general understanding of the new 64-bit kernel.

1.2.1 64-bit kernel considerations

There are some points for consideration for this new 64-bit kernel.

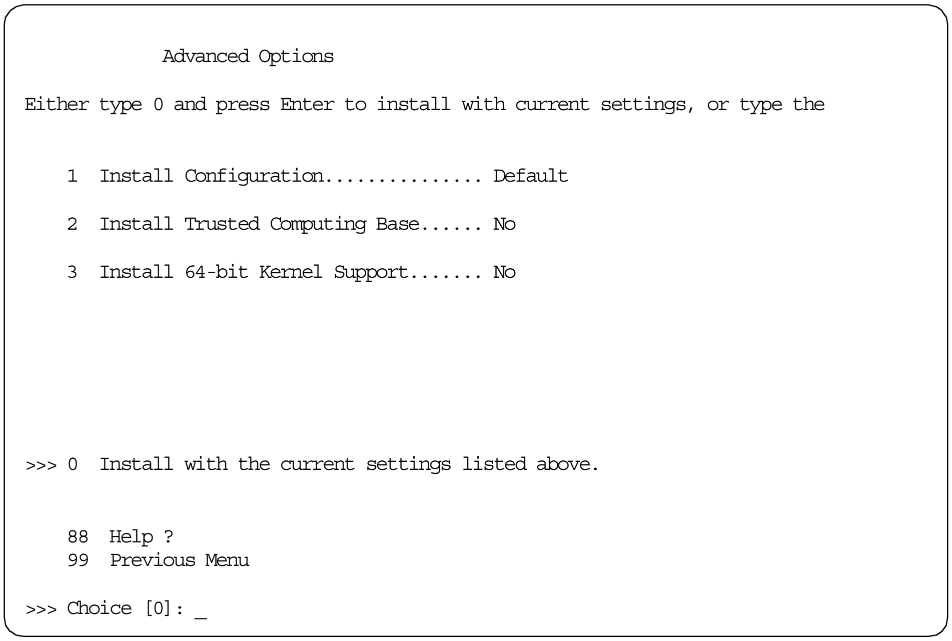
- The 64-bit kernel is the only kernel for Itanium-based systems
- Both 32-bit and 64-bit kernels are available for the POWER platform.

- Only 64-bit CHRP-compliant PowerPC machines are supported for the 64-bit kernel on the POWER platform.
- Only Itanium-based machines are supported for the IA-64 platform.
- Only 64-bit kernel extensions are supported; that means no existing 32-bit kernel extensions (in the case of POWER) can be reused for the 64-bit kernel.
- Kernel extensions and device drivers must be compiled in 64-bit mode to be loaded into the 64-bit kernel.
- The 32-bit and 64-bit application environments are available on all 64-bit platforms (POWER and Itanium-based).

1.2.2 Selecting the 64-bit kernel on POWER systems

AIX 5L for POWER now provides a 64-bit kernel in addition to the previously available 32-bit kernel. In addition, some data types have been enlarged to support this 64-bit kernel.

The installation of this new 64-bit kernel is selectable through the Advanced Option screen during the initial AIX installation. As shown in the following panel, you need to toggle option three to install this new kernel.



```
               Advanced Options

Either type 0 and press Enter to install with current settings, or type the


    1  Install Configuration.............. Default

    2  Install Trusted Computing Base...... No

    3  Install 64-bit Kernel Support....... No







>>> 0  Install with the current settings listed above.


    88  Help ?
    99  Previous Menu

>>> Choice [0]: _
```

You can also install this kernel later by installing the bos.mp64 fileset.

If your system has 64-bit processors, the 64-bit kernel is automatically installed with the base operating system. However, the 64-bit kernel is only enabled if you set the Install 64-bit Kernel option to yes during the initial AIX installation. However, you can switch between the 32-bit and 64-bit kernels using the following procedure.

As root, enter the following commands, substituting 64 for 64-bit kernel or mp for 32-bit multiprocessor kernel for ??.

```
ln -sf /usr/lib/boot/unix_?? /unix
ln -sf unix_?? /usr/lib/boot/unix
bosboot -ad /dev/ipldevice
shutdown -r
```

With a similar flow of actions, you can re-activate the 32-bit kernel again.

Note that this does not affect the 64-bit application environment, which is supported running either the 32-bit or 64-bit kernel. The 64-bit application environment can be enabled/disabled from SMIT under System Environments.

Chapter 2. Development environment and tool enhancements

AIX 5L provides several enhancements in the that assist you in developing your own software. This chapter is dedicated to them.

Refer to the AIX Application programmer guides in the Online Documentation Library for detailed information about many of these new functions.

2.1 Large data type support - binary compatibility

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

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

For the Itanium-based platform, this feature is the standard programming model.

2.2 DBX enhancements

The `print` subcommand in DBX is enhanced to provide an easier to read display output. In AIX Version 4.3.3 and previous releases, array elements and structure or union fields are printed serially, one after the other, on a single line, which sometimes makes it hard to understand.

This feature is only available on the POWER platform. For Itanium-baed systems, a SUI/PICL application debugger should be used.

A sample output of the dbx print output subcommand in AIX Version 4.3 follows:

```
(dbx) print x
(op = O_CONT, nodetype = (nil), value = union:(sym = 0x20076d88, name
= 0x20076d88, lcon = 0x20076d88, dash = 0x20076d88, llcon = 0x20076d88
00000000, addrcon = 0x20076d8800000000, fcon = 2.1841616996348188e-154
, qcon = (val = (2.1841616996348188e-154, 0.0)), kcon = (real = 2.1841
616996348188e-154, imag = 0.0), qkcon = (real = (val = (2.184161699634
8188e-154, 0.0)), imag = (val = (1.605837571007193e-154, 1.72522746112
82083e-314))), scon = "", fscon = (scon = "", strsize = 0x0), arg = (0
x20076d88, (nil), (nil), (nil), 0x20013980), trace = (exp = 0x20076d88
, place = (nil), cond = (nil), inst = false, event = 0x20013980, actio
ns = (nil)), step = (source = 537357704, skipcalls = false), examine =
 (mode = "", beginaddr = (nil), endaddr = (nil), count = 0x0), procret
urn = (proc = 0x20076d88, retLocation = 0x0, caller_fp = 0x20013980000
00000), funcList = 0x20076d88), touch = '^A', refcount = '\0')
```

You can enable the new print subcommand style using `set $pretty="on"`, this mode will use indentation to represent static scope of each value. A sample output is provided below:

```
(dbx) print a
{
    NamedObject::identity = {
        name = "0"
        number = 0x20008528
    }
    id = 0x1
    motion[0] = {
        ColoredObject::color = yellow
        a = 48.0
        b = 1000.0
        c = 0.0
    }
    motion[1] = {
        ColoredObject::color = indigo
        a = 2.0
        b = 100.0
        c = 0.0
    }
    motion[2] = {
        ColoredObject::color = orange
        a = 0.0
        b = 5.0
        c = 0.0
    }
}
```

Another output style can be enabled. The verbose mode will use qualified names instead of indentation to represent the static scope. To enable verbose mode, set the variable as: `set $pretty="verbose"`. A sample output for verbose mode is provided below:

```
(dbx) print a
NamedObject::identity.name = "0"
NamedObject::identity.number = 0x20008528
```

```
id = 0x1
motion[0].ColoredObject::color = yellow
motion[0].a = 48.0
motion[0].b = 1000.0
motion[0].c = 0.0
motion[1].ColoredObject::color = indigo
motion[1].a = 2.0
motion[1].b = 100.0
motion[1].c = 0.0
motion[2].ColoredObject::color = orange
motion[2].a = 0.0
motion[2].b = 5.0
motion[2].c = 0.0
```

2.3 Pthread differences and enhancements

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

2.3.1 Pthread debug library

In AIX Version 4.3.3 and previous releases, dbx was the only debugger that could access information about pthreads library objects. In AIX 5L, the pthreads debug library provides a set of functions that allows application developers to examine and modify pthread library objects.

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

For a complete list of functions as well the initialization process, refer to the product documentation.

2.3.2 Pthread unregister atfork handler

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

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

2.3.3 Multiple read/write lock read owners

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

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

2.4 Lightweight core file support

AIX 5L supports lightweight core files (lwcf) that consist of stack tracebacks from each thread and process. This enhancement assists large parallel jobs that need a way of collecting and displaying the state of all threads and processes when the job is abnormally terminated.

This enhancement provides two new routines, mt__trce() and install_lwcf_handler(), to be used by programs to generate a lightweight core file. This lightweight core file provides traceback information for each thread in each process of a potentially distributed application for debugging purposes.

Core files can be generated without process termination to increase application availability.

2.5 KDB kernel debugger and kdb command enhancements

The KDB kernel debugger and `kdb` command are enhanced. For AIX 5L and subsequent releases, the KDB Kernel Debugger is the standard kernel debugger and is included in the unix_up and unix_mp kernels, which may be found in /usr/lib/boot. This enhancement is only available on the POWER platform. For Itanium-based systems, use the `iadb` debugger.

2.5.1 KDB kernel debugger introduction

The KDB Kernel Debugger must be loaded at boot time. This requires that a boot image be created with the debugger enabled. To enable the KDB Kernel Debugger in AIX 5L, the `bosboot` command must be invoked with options set to enable the KDB Kernel Debugger. The kernel debugger can be enabled using either the -I or -D options of `bosboot`.

Examples of `bosboot` commands:

- `bosboot -a -d /dev/ipldevice`
- `bosboot -a -d /dev/ipldevice -D`
- `bosboot -a -d /dev/ipldevice -I`

2.5.2 KDB new functions and enhancements

New subcommands were added to KDB in order to provide some functions already present in the `crash` command.

Dcal and hcal

The KDB kernel debugger and `KDB` command dcal and hcal subcommands will be modified to include the additional operators ^, %, and ().

Conv

The conv subcommand performs base conversions. The syntax for this command will be:

```
conv [-bdox | -axx] num
```

where num is the value to be converted and the optional flags indicate the base for num:

- -b = binary
- -d = decimal (default)
- -o = octal
- -x = hex
- -axx = base xx (2 to 36)

The input value is then displayed in binary, octal, decimal, and hex.

Dump

The dump subcommand performs exactly the same function as the dump subcommand in `crash`, to dump the contents of storage.

Errpt

The errpt subcommand prints all error log entries not picked up by the errdemon and allows the printing of a user-specified number of entries that have been picked up by the errdemon (the default is 3).

Inode

The inode subcommand has two additional options. A -c flag displays the reference count of an inode. The second flag is -d. This flag requires that the next three arguments to the subcommand specify the major and minor device numbers and the inode number to be displayed. These changes will be made for both the KDB kernel debugger and the KDB command.

Lke

Option -n name is added to the lke subcommand to allow specification of a substring that is required to occur within a loader entry name (for it to be displayed).

Mbuf

A new -n option allows following the chain for the m_next element until the end of the chain. This chain is the collection of mbufs for a single packet. The -a option allows following the chain of m_act entries. This chain is a group of packets linked together. The -a and -n options can be used together. When both options are used, information for the mbufs within each packet is displayed; then the display proceeds to the next packet. These options were added to both the KDB kernel debugger and KDB command.

Netm

The netm subcommand displays the most recent net_malloc_police record when invoked without any arguments. It may be invoked with an -a option to display all net_malloc_police records. It may also be invoked with an address to display records whose address or caller fields match the given address.

Proc

A new -s option is added to the KDB proc subcommand. This option will be available for use in conjunction with the "*" option, which displays a summary of all processes. The -s option will limit output to processes that are in the state specified following the -s flag.

Sock

An additional function is added to the KDB sock subcommand. This function is available through the use of the -p flag and may be used to limit the output from the socket subcommand to just sockets associated with a specific process.

Sr64

A new -n option is added to the sr64 subcommand. This option may be used to indicate that uadnode information is to be displayed for the uadnodes associated with the segment information displayed.

Status

A new subcommand status is added to both the KDB kernel debugger and KDB command. For each CPU, the CPU number and the thread ID, thread slot, process ID, process slot, and process name for the current thread are displayed.

Th

The th subcommand has the addition of a -r and -p flag. The -r flag displays only runable threads. The -p flag requires that a process table entry be specified and will display all threads for the indicated process.

Varrm

The varrm subcommand is added to both the KDB kernel debugger and command, and it allows user-defined variables to be cleared. A variable will be cleared by issuing the varrm subcommand and specifying the variable name as a parameter. Clearing a variable deletes the variable from the list of user-defined variables, freeing the slot for use by another user-defined variable.

Varlist

The varlist subcommand is added to the KDB kernel debugger and command, and it lists the names and values for any user defined variables.

2.6 Context switch avoidance

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

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

If a thread is still in a critical section at the end of the extra time slice, it loses its scheduling benefit for one time slice. At the end of that time slice, it is eligible again for another slice benefit. If a thread never leaves a critical section, it cannot be stopped by a debugger or control-Z from the parent shell.

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

2.7 Very large program support

AIX 5L now supports a more flexible way for 32-bit programs to make maximum use of the eight available data segments as either heap or shared memory. At the time of writing, this feature is only available on the POWER platform.

With Very Large Program Support programs can specify the size of the heap they want to use with the -bmaxdata option for the `ld` command. The following command compiles and links a program to allow up to eight segments to be used for the data heap with Very Large Program Support.

```
cc sample.c -bmaxdata:0x80000000/dsa
```

The new support in AIX 5L offers a dynamic segment allocation (DSA) algorithm that it uses to create the segments for the data heap dynamically. The command shown in the example specifies that the program is allowed to grow its data heap up to eight segments. Segments that are not used by the data heap are available to the program to be used for other purposes such as memory mapped files. Once a segment is claimed by the data heap though, it is no longer available for other purposes. In addition, the behavior of system calls such as mmap() and shmat() are changed to start allocating from the top of the address space and work down if the DSA flag is specified.

AIX also allows you to change the maxdata value of the XCOFF file at program loading time. The environment valuable LDR_CNTRL will be used as the ld option for this purpose. For example,

```
export LDR_CNTRL=MAXDATA=0x40000000
```

tells the AIX loader to override the maxdata field of the XCOFF file for execution to use four data segments.

2.8 Kornshell enhancements

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

2.8.1 ksh93

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

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

This new version of ksh has the following functional enhancements:

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

For a detailed description of the new features, consult the official KornShell Web site at `http://www.kornshell.com`.

2.8.2 New value for shell attribute

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

2.9 Malloc enhancements

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

2.9.1 Malloc multiheap

The multiheap malloc was introduced in AIX Version 4.3.3 as part of the service stream and it may not be well known. It is available on both the POWER and Itanium-based platforms.

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

2.9.2 Malloc buckets

Malloc buckets was introduced in AIX Version 4.3.3 as part of the service stream. It is available on both the POWER and Itanium-based platforms.

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

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

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

2.10 Non-sparseness support for the restore command

In AIX 5L the `restore` command has a new -e flag, which preserves the sparseness or non-sparseness of files created with the `backup` command.

A file is a sequence of indexed blocks of arbitrary size. The indexing is accomplished through the use of direct mapping or indirect index blocks from the files inode. Each index within a file's address range is not required to map to an actual data block.

A file that has one or more indexes that are not mapped to a data block is referred to as being sparsely-allocated or a sparse file. A sparse file will have a size associated with it, but it will not have all of the data blocks allocated to fulfill the size requirements. To identify if a file is sparsely-allocated, use the `fileplace` command. It will indicate all blocks in the file that are not currently allocated.

Such files are common in a database application's sparse files. The blocks with the NULL values are also often called holes. The default behavior of the `restore` command is to save disk space and therefore to create sparse files (if possible). This is the correct behavior, if the original file is also a sparse file, but incorrect in the case of the backup of a non-sparse file.

This enhancement restores the non sparse files as non sparse as they were archived by the name format of `backup` command for both packed and unpacked files. It is necessary to know the sparseness/non-sparseness of the file(s) before archiving the files, since enabling this flag restores the sparse files as non-sparse.

This flag should be enabled only if files are to be restored are non sparse consisting of more than 4K NULLs. If the -e flag is specified during restore, it successfully restores all normal files normally and non-sparse database files as non sparse.

2.11 Enhancements for pax

In AIX 5L, the `pax` command is enhanced to support a 64-bit POSIX-defined data format, which is used by default. The objective of this command is to allow archiving of large files, such as dumps. The commands `cpio` and `tar` do not support files used as input larger than 2 GB, because they are limited by their 32-bit formats. There are no plans to enhance these programs to support this situation in the future.

If you have to archive files larger than 2 GB, the only available option is the `pax` command. Suppose you have several `tar` archives with a size in total exceeding the 2 GB limit. With the following command, you can create an archive for all of them:

```
# pax -x pax -wvf soft.pax ./soft?.tar
```

The default mode for pax (without the -x option) is to behave as tar. The -x option will allow pax the ability to work with files larger than 2 GB, a behavior tar does not have.

This enhancement is also available on AIX Version 4.3.3 service releases.

2.12 Java currency

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

There is no link from the /usr/bin directory to the installed version of Java, which is completely self-contained in the directory /usr/java130. This allows you to install any other version of Java in parallel without jeopardizing the correct function of certain AIX subsystems, such as the Web-based System Manager, which relies on this specific version of Java. The most important information is the readme file, located in the /usr/java130 directory.

Java installed by default on AIX 5L is the 32-bit Java 1.3.0. The 64-bit Java 1.3.0 will also run on AIX 5L, but will not be installed by default and will be in a different directory when it is installed.

The Web site specifically for Java on AIX is
`http://www.ibm.com/java/jdk/aix/index.html`

Chapter 3. LVM and file system enhancements

AIX 5L introduces several new features for the logical volume manager and supports the second generation journaled file system (JFS2) and the /proc pseudo file system.

3.1 Enhancements for volume group commands

The following enhancements to volume group commands in AIX 5L will be discussed in this section.

- The `redefinevg` command
- Read-only `varyonvg`
- LVM hot spare disk in a volume group
- Support for different logical track group sizes
- LVM hot-spot management
- The `migratelp` command
- The `recreatevg` command

Note

Because the physical volume and volume group identifiers have been changed from 16 characters to 32 characters, you can only access a volume group created on AIX 5L from a AIX Version 4.3.3 system after you have applied the appropriate fixes from the Fall 2000 AIX Version 4.3.3 Update CD. You can access a volume group created on AIX Version 4.3.3 on an AIX 5L system, but using any of the new features, like setting a different logical track group size, will change some of the volume group identification internal data structures in a way that the volume group becomes unusable on an AIX V4.3.3 system or lower.

3.1.1 The redefinevg command

The command `redefinevg` is rewritten in C to improve performance.

3.1.2 Read-only varyonvg

The command `varyonvg` now supports an -r flag that allows a volume group to be varied-on in read-only mode.

3.1.3 LVM hot spare disk in a volume group

The chpv and the chvg commands are enhanced with a new -h flag that allows you to designate disks as hot spare disks in a volume group and to specify a policy to be used in the case of failing disks. These commands are not replacements for the sparing support available with SSA disks; they complement it. You can also use it with SSA disks when you add one to your volume group.

Note

These new options have an effect only if the volume group has mirrored logical volumes.

There is a new -s flag for the chvg command that is used to specify synchronization characteristics.

The following command marks hdisk1 as a hot spare disk:

```
# chpv -hy hdisk1
```

This is only successful if there are not already allocated logical partitions on this disk. Using n instead of y would again remove the hot spare disk marker. If you add a physical volume to a volume group (to mark it as a hot spare disk), the disk has to have, at least, the same capacity as the smallest disk already in the volume group.

After you have marked one or more disks as hot spare disks, you have to decide which policy to use in case a disk is starting to fail. There are four different policies you can specify with the -h flag, shown using the following syntax:

```
# chvg -hhotsparepolicy -ssyncpolicy VolumeGroup
```

The following four values are valid for the hotsparepolicy argument:

y This policy automatically migrates partitions from one failing disk to one spare disk. From the pool of hot spare disks, the smallest one which is big enough to substitute for the failing disk will be used.

Y This policy automatically migrates partitions from a failing disk, but might use the complete pool of hot spare disks.

n No automatic migration will take place. This is the default value for a volume group.

r This value removes all disks from the pool of hot spare disks for this volume group.

The syncpolicy argument can only use the values y and n.

y This will automatically try to synchronize stale partitions.

n This will not automatically try to synchronize stale partitions.

The latter argument is also the default for a volume group.

After setting this up, Volume Group Status Area (VGSA) write failures and Mirror Write Consistency (MWC) write failures will mark a physical volume missing and start the migration of data to the hot spare disk.

3.1.4 Support for different logical track group sizes

AIX 5L now supports different logical track group (LTG) sizes. In previous versions of AIX, the only supported LTG size was 128 KB. This is still the default for the creation of new volume groups, even under AIX 5L. You can change this value when you create a new volume group with the `mkvg` command, or later for an existing volume group with the `chvg` command.

The LTG corresponds to the maximum allowed transfer size for disk I/O (many disks today support sizes larger than 128 KB). To take advantage of these larger transfer sizes and get a better disk I/O performance, AIX 5L now accepts values of 128 KB, 256 KB, 512 KB, and 1024 KB for the LTG size, now and possibly even larger values in the future. The maximum allowed value is the smallest maximum transfer size supported by all disks in a volume group. The `mkvg` SMIT screen shows all four values in the selection dialog for the LTG. The `chvg` SMIT screen shows only the values for the LTG supported by the disks. The supported sizes are discovered using an ioctl(IOCINFO) call.

The following command shows how to change the LTG size for testvg from the default of 128 KB to 256 KB.

```
# chvg -L256 testvg
```

To ensure the integrity of the volume group, this command varies off the volume group during the change. The `mkvg` command supports the same new -L flag.

To find out what the maximum supported LTG size of your hard disk is, you can use the lquerypv command with the -M flag. The output gives the maximum LTG size in KB, as can be seen from the following lines:

```
# /usr/sbin/lquerypv -M hdisk0
256
```

You can list the values for all the new options (LTG size, AUTO SYNC, and HOT SPARE) with the lsvg command. Note that the volume identifier has been widened from 16 to 32 characters.

```
# lsvg rootvg
VOLUME GROUP:   rootvg                   VG IDENTIFIER:  000bc6fd00004c00000000e10fdd7f52
VG STATE:       active                   PP SIZE:        16 megabyte(s)
VG PERMISSION:  read/write               TOTAL PPs:      1084 (17344 megabytes)
MAX LVs:        256                      FREE PPs:       1032 (16512 megabytes)
LVs:            11                       USED PPs:       52 (832 megabytes)
OPEN LVs:       10                       QUORUM:         2
TOTAL PVs:      2                        VG DESCRIPTORS: 3
STALE PVs:      0                        STALE PPs:      0
ACTIVE PVs:     2                        AUTO ON:        yes
MAX PPs per PV: 1016                     MAX PVs:        32
LTG size:       128 kilobyte(s)          AUTO SYNC:      yes
HOT SPARE:      yes (one to one)
```

3.1.5 LVM hot-spot management

Two new commands, lvmstat and migratelp, help you to identify and remedy hot-spot problems within your logical volumes. You have a hot-spot problem if some of the logical partitions on your disk have so much disk I/O that your system performance noticeably suffers. By default, no statistics for the logical volumes are gathered. The gathering of statistics has to be enabled first with the lvmstat command for either a logical volume or an entire volume group.

The complete command syntax for lvmstat is as follows:

```
lvmstat { -l | -v } Name [ -e | -d ] [ -F ] [ -C ] [ -c Count ] [ -s ] [
Interval [ Iterations ] ]
```

The meaning of the flags are as follows.

-e Enables the gathering of statistics about the logical volume.

-d Disables the gathering of statistics.

-l Specifies the name of a logical volume to work on.

-v Specifies the name of a volume group to work on. You can also enable, in the first step, a volume group and selectively disable afterwards some logical volumes you are not working with.

-F Separates the output of the statistics by colons (to make it easier for parsing by other scripts).

The first use of `lvmstat`, after enabling, displays the counter values since system reboot. Each usage thereafter displays the difference from the last call. With the -C flag, you can clear the counter for the specified logical volume or volume group.

With the -c flag, you specify how many lines from the top you want to have listed. The -s flag suppresses the header lines for subsequent outputs if you are using the interval and iteration arguments. In the case of interval and iteration, only values for logical volumes for which there was a change in the last interval will be outputted. If there was no change at all, only a . (period) will be printed to the console.

The following example is a session where data was copied from /unix to /tmp:

```
# lvmstat -v rootvg -e
# lvmstat -v rootvg -C
# lvmstat -v rootvg

Logical Volume       iocnt   Kb_read   Kb_wrtn      Kbps
  hd8                    4         0        16      0.00
  paging01               0         0         0      0.00
  lv01                   0         0         0      0.00
  hd1                    0         0         0      0.00
  hd3                    0         0         0      0.00
  hd9var                 0         0         0      0.00
  hd2                    0         0         0      0.00
  hd4                    0         0         0      0.00
  hd6                    0         0         0      0.00
  hd5                    0         0         0      0.00
```

The previous output shows that, basically, all counters have been reset to zero. Before the following example, data was copied from /unix to /tmp:

```
# cp -p /unix /tmp
# lvmstat -v rootvg

Logical Volume       iocnt   Kb_read   Kb_wrtn      Kbps
  hd3                  296         0      6916      0.04
  hd8                   47         0       188      0.00
  hd4                   29         0       128      0.00
  hd2                   16         0        72      0.00
  paging01               0         0         0      0.00
  lv01                   0         0         0      0.00
  hd1                    0         0         0      0.00
  hd9var                 0         0         0      0.00
  hd6                    0         0         0      0.00
  hd5                    0         0         0      0.00
```

As shown, there is activity on the hd3 logical volume, which is mounted on /tmp, on hd8, which is the jfslog logical volume, on hd4, which is / (root), on hd2, which is /usr, and on hd9var, which is /var. The following output provides details on hd3 and hd2:

```
# lvmstat -l hd3

Log_part  mirror#  iocnt   Kb_read   Kb_wrtn      Kbps
       1       1     299         0      6896      0.04
       3       1       4         0        52      0.00
       2       1       0         0         0      0.00
       4       1       0         0         0      0.00
# lvmstat -l hd2

Log_part  mirror#  iocnt   Kb_read   Kb_wrtn      Kbps
       2       1       9         0        52      0.00
       3       1       9         0        36      0.00
       7       1       9         0        36      0.00
       4       1       4         0        16      0.00
       9       1       1         0         4      0.00
      14       1       1         0         4      0.00
       1       1       0         0         0      0.00
```

The output for a volume group provides a summary for all the I/O activity of a logical volume. It is separated into the number of I/O requests (iocnt), the kilobytes read and written (Kb_read and Kb_wrtn, respectively) and the transferred data in KB/s (Kbps). If you request the info for a logical volume you receive the same information, but for each logical partition separately. If you have mirrored logical volumes, you receive statistics for each of the mirror volumes. In the previous sample output, several lines for logical partitions without any activity were omitted. The output is always sorted in decreasing order on the iocnt column.

3.1.6 The migratelp command

With the output of the `lvmstat` command described in the previous section, it is easy to identify the logical partitions with the heaviest traffic. If you have several logical partitions with heavy usage on one physical disk and want to balance these across the available disks, you can use the new `migratelp` command to move these logical partitions to other physical disks.

Note

The `migratelp` command will not work with partitions of striped logical volumes.

The `migratelp` command uses the following syntax

```
migratelp lvname/lpartnum[/copynum] destpv[/ppartnum]
```

This command uses, as parameters, the name of the logical volume, the number of the logical partition (as it is displayed in the lvmstat output), and an optional number for a specific mirror copy. If information is omitted, the first mirror copy is used. You have to specify the target physical volume for the move; in addition, you can specify a target physical partition number. If successful, the output will appear similar to the following:

```
# migratelp hd3/1 hdisk1/109
migratelp: Mirror copy 1 of logical partition 1 of logical volume
        hd3 migrated to physical partition 109 of hdisk1.
```

3.1.7 The recreatevg command

The recreatevg command is used when you have a disk to disk copy to perform, but you want to create a unique volume and not an exact mirror. A direct dd copy would create a problem because all the information such as VGDAs and LVs, in one disk are copied to the other. Duplicate volume group, logical volume, and file system mount points are prevented by using the recreatevg command. Command options allow you to specify a logical volume name, a prefix label to uniquely define the VG. Automatic name generation is the default.

3.2 The /proc file system

AIX 5L provides support of the /proc file system. This pseudo file system maps processes and kernel data structures to corresponding files. The output of the mount and df commands showing /proc is shown in the following examples:

```
# mount
  node       mounted        mounted over    vfs       date        options
-------- ---------------  ---------------  ------ ------------ ---------------
         /dev/hd4         /                jfs    Sep 11 16:52 rw,log=/dev/hd8
         /dev/hd2         /usr             jfs    Sep 11 16:52 rw,log=/dev/hd8
         /dev/hd9var      /var             jfs    Sep 11 16:52 rw,log=/dev/hd8
         /dev/hd3         /tmp             jfs    Sep 11 16:52 rw,log=/dev/hd8
         /dev/hd1         /home            jfs    Sep 11 16:53 rw,log=/dev/hd8
         /proc            /proc            procfs Sep 11 16:53 rw

# df
Filesystem    512-blocks      Free %Used    Iused %Iused Mounted on
/dev/hd4           65536     27760   58%     2239    14% /
/dev/hd2         1507328    242872   84%    22437    12% /usr
/dev/hd9var        32768     16432   50%      448    11% /var
/dev/hd3          557056    538008    4%      103     1% /tmp
/dev/hd1           32768     31608    4%       47     2% /home
/proc                  -         -    -         -     -  /proc
```

The entry in the /etc/vfs file appears as follows:

```
# lsvfs procfs
procfs  6       none    none
```

Each process is assigned a directory entry in the /proc file system with a name identical to its process ID. In this directory, several files and subdirectories are created corresponding to internal process control data structures. Most of these files are read-only, but some of them can also be written to and be used for process control purposes. The interface to these files are the standard C language subroutines open(), read(), write(), and close(). It is possible to have several concurrent readers, but for reliability reasons, the first write access should use the exclusive flag, so that subsequent opens for write access fail. The description of the data structures used can be found in /usr/include/sys/procfs.h. The ownership of the files in the /proc file system is the same as for the processes they represent. Therefore, regular users can only access /proc files that belong to their own processes.

A simple example illustrates this further. Suppose a process's is waiting for standard input (the information in the process data structures is basically static). If you look at an active process, a lot of the information would constantly change:

```
# ls -l /proc/19082/
total 0
dr-xr-xr-x   1 root     system           0 Sep 15 15:12 .
dr-xr-xr-x   1 root     system           0 Sep 15 15:12 ..
-rw-------   1 root     system           0 Sep 15 15:12 as
-r--------   1 root     system         128 Sep 15 15:12 cred
--w-------   1 root     system           0 Sep 15 15:12 ctl
dr-xr-xr-x   1 root     system           0 Sep 15 15:12 lwp
-r--------   1 root     system           0 Sep 15 15:12 map
dr-x------   1 root     system           0 Sep 15 15:12 object
-r--r--r--   1 root     system         448 Sep 15 15:12 psinfo
-r--------   1 root     system        1024 Sep 15 15:12 sigact
-r--------   1 root     system        1520 Sep 15 15:12 status
-r--r--r--   1 root     system           0 Sep 15 15:12 sysent
```

Table 1 provides the function of the pseudo files listed in the previous output.

Table 1. Function of pseudo files in /proc/<pid> directory

Pseudo file name	Function
as	Read/write access to address space
cred	Credentials
ctl	Write access to control process. For example: stop or resume
map	Virtual address map
psinfo	Information for the `ps` command; readable by everyone

Pseudo file name	Function
sigact	Signal status
status	Process state information, such as address, size of heap or stack
sysent	Information about system calls

The pseudo file, named *as*, allows you to access the address space of the process, and as it can be seen by the rw (read/write) access flags, you can read and write to the memory belonging to the process.

It should be understood that only the user regions of the process' address can be written to under /proc. Also, a copy of the address space of the process is made while tracing under /proc. This is the address space that can be modified. This is done so when the as file is closed, the original address space is unmodified.

The cred file provides information about the credentials associated with this process. Writing to the ctl file allows you to control the process; for example, to stop or to resume it. The map file allows to access the virtual address map of the process. Information usually shown by the `ps` command can be found in the psinfo file, which is readable for all system users. The current status of all signals associated with this process are recorded in the sigact file. State information for this process, such as the address and size of the process heap and stack (among others), can be found in the status file. Finally, the sysent file allows you to check for the system calls available to this process.

The object directory contains files with names as they appear in the map file. These files correspond to files mapped in the address space of the process. For example, the content of this directory appears as follows:

```
# ls -l /proc/19082/object
total 13192
dr-x------   1 root     system          0 Sep 15 15:09 .
dr-xr-xr-x   1 root     system          0 Sep 15 15:09 ..
-r-xr-xr-x   1 bin      bin          6264 Aug 24 21:16 a.out
-rwxr-xr-x   1 bin      bin         14342 Aug 22 22:37 jfs.10.5.10592
-r-xr-xr-x   2 bin      bin       6209308 Aug 24 13:03 jfs.10.5.2066
-r--r--r--   1 bin      bin        118267 Aug 24 15:06 jfs.10.5.2076
-r-xr-xr-x   1 bin      bin         11009 Aug 24 14:59 jfs.10.5.4129
-r--r--r--   1 bin      bin        377400 Aug 24 15:05 jfs.10.5.4161
-r-xr-xr-x   1 bin      bin          6264 Aug 24 21:16 jfs.10.5.6371
```

The a.out file always represents the executable binary file for the program running in the process itself. Because the example program is written in C and must use the C runtime library, it can be concluded from the size of the entry named jfs.10.5.2066 that this corresponds to the /usr/ccs/lib/libc.a file. Checking this file reveals that the numbers in the file name are the major and minor device numbers, and the inode number, respectively. This can be seen in the following output, where /usr corresponds to /dev/hd2 and the `ncheck` command is used to find a file belonging to an inode in a specific file system:

```
# ls -l /dev/hd2
brw-rw----   1 root     system    10,  5 Sep 20 16:09 /dev/hd2
# ncheck -i 2066 /dev/hd2
/dev/hd2:
2066    /ccs/lib/libc.a
```

The lwp directory, finally, has subdirectory entries for each kernel thread running in the process. The term *lwp* stands for lightweight process and is the same as the term thread used in the AIX documentation. It is used in the context of the /proc file system to keep a common terminology with the /proc implementation of other operating systems. The names of the subdirectories are the thread IDs. The test program has only one thread with the ID 54891, as shown in the output of the `ps` command. Therefore, only the content of this one thread directory is shown:

```
# ps -mo THREAD -p 19082
    USER   PID  PPID    TID ST  CP PRI SC    WCHAN        F     TT BND COMMAND
    root 19082 20678      - A    0  83  1 700e6244   200001  pts/3   - wc
       -     -     -  54891 S    0  83  1 700e6244    10400      -   - -
# ls -l /proc/19082/lwp/54891
total 0
dr-xr-xr-x   1 root     system            0 Sep 15 15:03 .
dr-xr-xr-x   1 root     system            0 Sep 15 15:03 ..
--w-------   1 root     system            0 Sep 15 15:03 lwpctl
-r--r--r--   1 root     system          120 Sep 15 15:03 lwpsinfo
-r--------   1 root     system         1200 Sep 15 15:03 lwpstatus
```

The lwpctl, lwpsinfo, and lwpstatus files contain thread specific information to control this thread, for the `ps` command, and about the state, similar to the corresponding files in the /proc/<pid> directory.

As an example of what can be obtained from reading these files, the following lines show the content of the cred file (after the use of the `od` command):

```
# ls -l /proc/19082/cred
-r--------   1 root     system          128 Sep 15 15:07 /proc/19082/cred
# od -x /proc/19082/cred
0000000  0000 0000 0000 0000 0000 0000 0000 0000
*
0000160  0000 0000 0000 0007 0000 0000 0000 0000
0000200  0000 0000 0000 0002 0000 0000 0000 0003
0000220  0000 0000 0000 0007 0000 0000 0000 0008
0000240  0000 0000 0000 000a 0000 0000 0000 000b
```

```
0000260
```

The output shows, in the left most column, the byte offset the file in octal representation. The remainder of the lines are the actual content of the file in hexadecimal notation. Even if the directory listing shows the size of the file to be 128 bytes or 0200 bytes in octal, the actual output is 0260 or 176 bytes in size. This is due to the dynamic behavior of the last field in the corresponding structure. The digit 7 in the line with the number 0160 specifies the number of groups the user ID running this process belongs to. Because every user ID is at least part of its primary group, but belongs possibly to a number of other groups which cannot be known in advance, only space for the primary group is reserved in the cred data structure. In this case the primary group ID is zero, because the user ID running this process is root. Reading the complete content of the file, nevertheless, reveals all the other group IDs the user currently belongs to. The group IDs in this case (2, 3, 7, 8, 0xa (10), and 0xb (11)) map to the groups bin, sys, security, cron, audit, and lp. This is exactly the set of groups the user ID root belongs to by default.

3.3 Journaled File System 2

The Journaled File System 2 (JFS2) is an enhanced and updated version of the JFS on AIX Version 4.3 and previous releases. The journaled file system JFS and JFS2 are native to the AIX operating system. The file system links the file and directory data to the structure used by storage and retrieval mechanisms.

Both JFS (the default) and JFS2 are available on POWER systems. Only JFS2 is supported on Itanium-based systems.

The Journaled File System 2 (JFS2) is intended to provide a robust, quickly restartable, transaction-oriented, log-based, and scalable byte-level file system implementation for the AIX environments. While tailored primarily for the high throughput and reliability requirements of servers, JFS2 is also applicable to client configurations where performance and reliability are desired.

JFS2 has new features that includes extent based allocation, sorted directories, and dynamic space allocation for file system objects.

3.3.1 What's new in JFS2?

Table 2 provides a comparison chart between the JFS2 and the standard JFS.

Table 2. Journaled File System specifications

Function	JFS2	JFS
Fragments/Block Size	512-4096 Block Sizes	512-4096 Fragments
Architectural Maximum File	4 PB[1]	64 GB
Architectural Maximum File System Size	4 PB	1 TB[2]
Maximum File Size Tested	1 TB	64 GB
Maximum File System Size	1 TB	1 TB
Number of Inodes	Dynamic, limited by disk space	Fixed, set at file system creation
Directory Organization	B-tree	Linear
Online Defragmentation	Yes	Yes
Compression	No	Yes
Default Ownership at Creation	root.system	sys.sys
SGID of Default File Mode	SGID=off	SGID=on
Quotas	No	Yes
Available on Itanium-based Architecture	Yes	No
Available on Power Architecture	Yes	Yes

[1] PB stands for PetaBytes which is equal to 1,048,576 GigaBytes.
[2] TB stands for TeraBytes which is equal to 1,024 GigaBytes.

3.3.1.1 Journaling

JFS2 provides improved structural consistency and recoverability and much faster restart times than non-journaled file systems. These other file systems are subject to corruption in the event of system failure, since a logical file operation often takes multiple media I/Os to accomplish and may not be totally reflected on the media at any given point in time. These file systems rely on restart time utilities (for example, `fsck`), which examine all of a file system's metadata (for example, directories and disk addressing structures) to detect and repair structural integrity problems. This is a time-consuming and error prone process which, in the worst case, can lose or misplace data.

In contrast, JFS2 uses techniques originally developed for databases to log information about operations performed into file system metadata as atomic transactions. In the event of a system failure, a file system is restored to a consistent state by replaying the log and applying log records for the appropriate transactions. The recovery time associated with this log-based approach is much faster, since the replay utility need only examine the log records produced by recent file system activity, rather than examine all file system metadata.

3.3.1.2 Extent based addressing structures

JFS2 uses extent based addressing structures, along with aggressive block allocation policies, to produce compact, efficient, and scalable structures for mapping logical offsets within files to physical addresses on disk.

An extent is a sequence of contiguous blocks allocated to a file as a unit and is described by a triple, consisting of <logical offset, length, physical address>. The addressing structure is a B+-tree populated with extent descriptors (the triples above), rooted in the inode, and keyed by logical offset within the file.

3.3.1.3 Variable block size

JFS2 supports block sizes of 512, 1024, 2048, and 4096 bytes on a per file system basis, allowing users to optimize space utilization based upon their application environment. Smaller block sizes reduce the amount of internal fragmentation within files and directories and are more space efficient. However, small blocks can increase path length, since block allocation activities will occur more often than if a larger block size were used. The default block size is 4096 bytes, since performance, rather then space utilization, is generally the primary consideration for server systems.

3.3.1.4 Dynamic disk inode allocation

JFS2 dynamically allocates space for disk inodes as required, freeing the space when it is no longer required. This support avoids the traditional approach of reserving a fixed amount of space for disk inodes at file system creation time, thus eliminating the need for customers to estimate the maximum number of files and directories that a file system will contain.

3.3.1.5 Directory organization

Two different directory organizations are provided. The first organization is used for small directories and stores the directory contents within the directory's inode. This eliminates the need for separate directory block I/O as well as the need for separate storage allocation. Up to eight entries may be stored in-line within the inode, excluding the self (.) and parent (..) directory entries, which are stored in a separate area of the inode.

The second organization is used for larger directories and represents each directory as a B+-tree keyed on name. The intent is to provide faster directory lookup, insertion, and deletion capabilities when compared to traditional unsorted directory organizations.

3.3.1.6 On-line file system free space defragmentation

JFS2 supports the defragmentation of free space in a mounted and actively accessed file system. Once a file system's free space has become fragmented, defragmenting the file system allows JFS2 to provide more I/O-efficient disk allocations and to avoid some out of space conditions.

Defragmentation support is provided in two pieces. The first piece is a user space JFS2 utility which examines the file system's metadata to determine the extent of free space fragmentation and to identify the file system reorganization activities required to reduce or eliminate the fragmentation. The second piece is integrated into the JFS2 kernel extension and is called by the user space utility. This second piece actually performs the reorganization activities, under the protection of journaling and with appropriate serialization to maintain file system consistency.

3.3.2 Compatibility

In this section, how the JFS2 interacts with the JFS1 environment is described.

3.3.2.1 Mixed volumes compatibility

In some cases, there will be many servers coexisting with different levels of AIX in a data center. From the JFS point of view, you can only import volume groups and mount file systems from AIX 4.X to AIX 5L servers. It is not possible to mount the JFS2 file system on AIX 4.X machines.

AIX 5L servers importing volume groups with JFS file systems.

Figure 1 shows an example of AIX Version 4.X machine exporting a volume group, and a AIX 5L machine importing this volume group and mounting a file system.

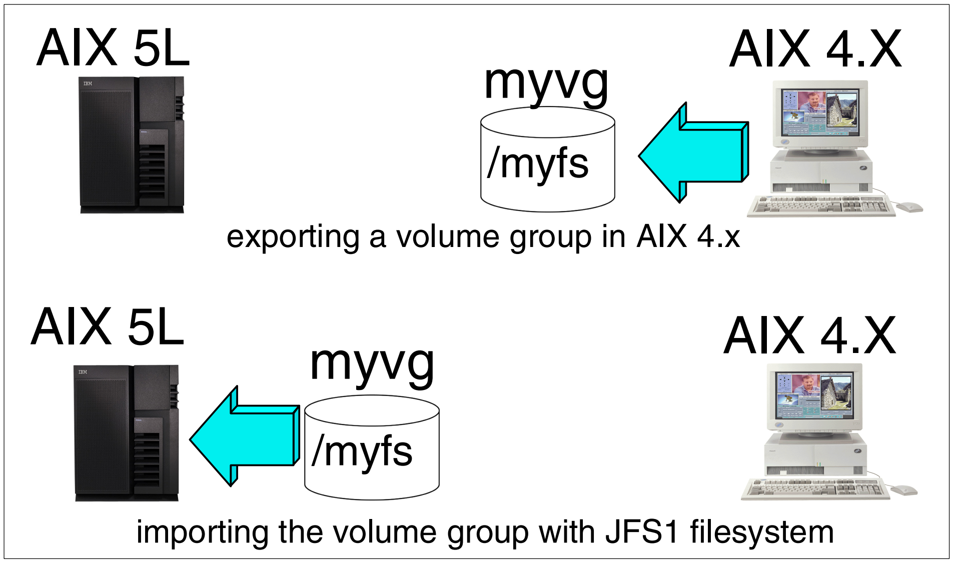

AIX 5L
myvg
AIX 4.X
/myfs
exporting a volume group in AIX 4.x
AIX 5L
myvg
AIX 4.X
/myfs
importing the volume group with JFS1 filesystem

Figure 1. Example of a server with AIX 5L importing and mounting JFS1 volumes

JFS-type migration note

In a case of JFS-type migration (for example, for performance or security reasons), a backup/restore approach is required. There is no LVM nor JFS command that migrates JFS volumes automatically.

It is possible to migrate JFS volumes in two different ways:

1. Backing up the file system, removing it and recreating it in the JFS2 type, then restoring the backup above the new file system.
2. If there is enough disk space available in the volume group, it is possible to create a new JFS2 file systems structure with the same attributes, and just copy all the files from one file system to another.

3.3.2.2 NFS mounting compatibility

There are two possible scenarios when mounting NFS file systems across different versions of JFS:

1. An AIX 5L JFS2 machine NFS mounting a remote JFS1 file system, as shown in Figure 2.

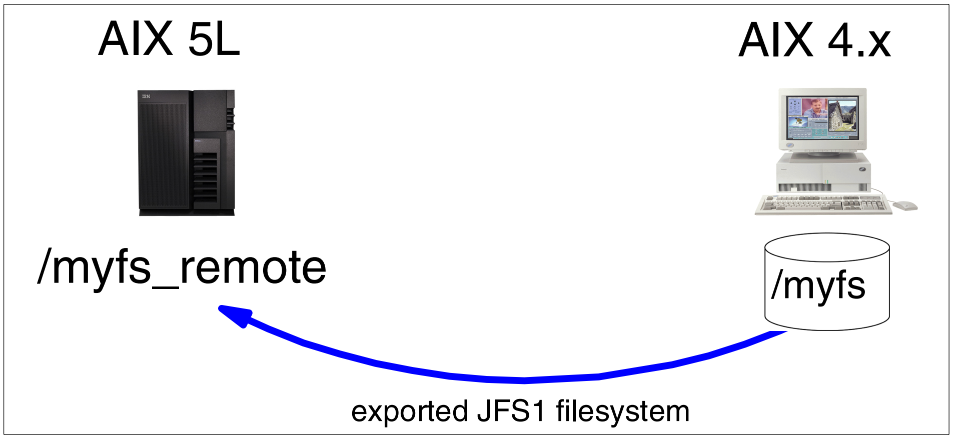


Figure 2. AIX 5.0 JFS2 machine NFS mounting a JFS1 file system

2. An AIX 4.X JFS1 machine NFS mounting a remote JFS2 file system, as can shown in Figure 3.

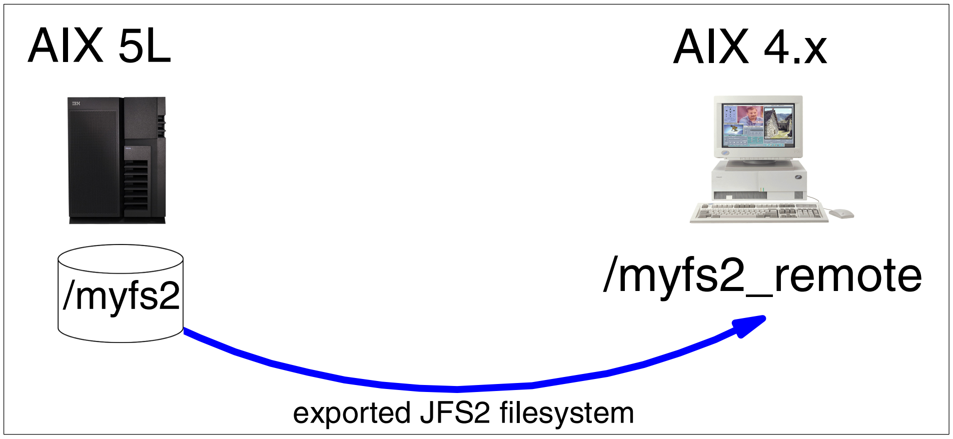


Figure 3. AIX 4.x JFS1 machine NFS mounting a JFS2 file system

Both scenarios have no compatibility issues.

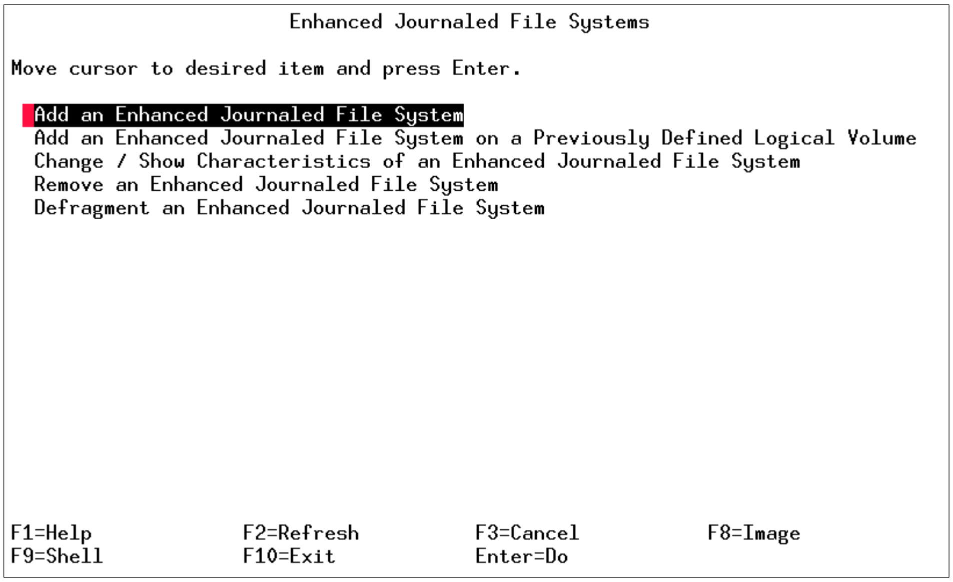
3.3.3 Commands and utilities changes

There is a set of new commands included in AIX for JFS2 management, and a set of JFS1 commands that are updated to handle JFS2 file systems.

In this section, a brief explanation about these JFS commands is provided.

3.3.3.1 Creating a JFS2 file system

The easiest way to create a JFS2 file system is through SMIT. Using the SMIT JFS2 fast path will show a JFS2 management menu, as shown in Figure 4 on page 37.



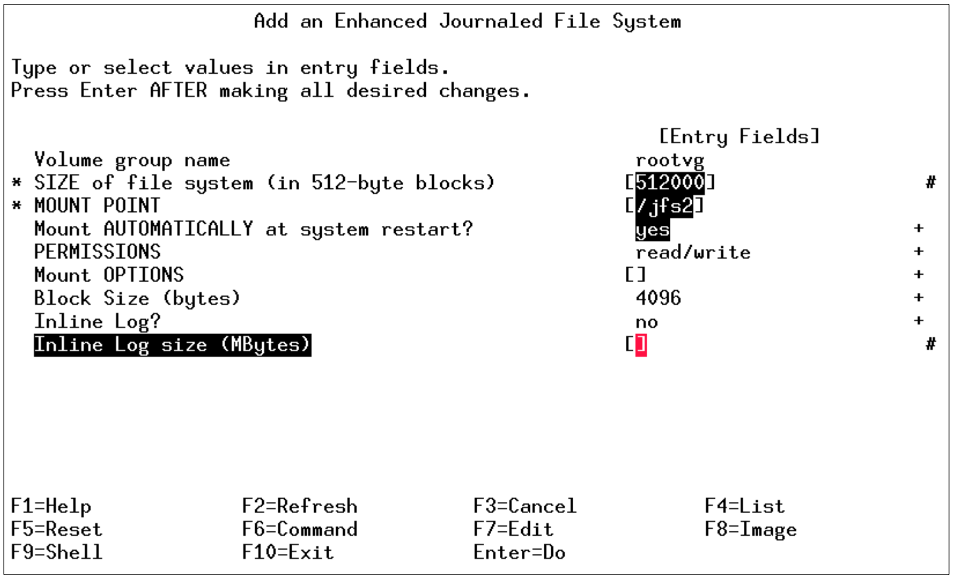
```
                        Enhanced Journaled File Systems

Move cursor to desired item and press Enter.

  Add an Enhanced Journaled File System
  Add an Enhanced Journaled File System on a Previously Defined Logical Volume
  Change / Show Characteristics of an Enhanced Journaled File System
  Remove an Enhanced Journaled File System
  Defragment an Enhanced Journaled File System

F1=Help             F2=Refresh          F3=Cancel           F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 4. SMIT panel for JFS2 management

Using the SMIT menu, the first option, **Add an Enhanced Journaled File System**, creates the JFS2 file system, and the second option, **Add an Enhanced File System on a Previously Defined Logical Volume**, creates a JFS2 file system above a previously created logical volume, which may be needed to organize or by the application for instance.

Add an enhanced file system

This option in the SMIT JFS2 menu allows the creation of a JFS2 file system with a size of 512-byte blocks and the mount point, as shown in Figure 5.



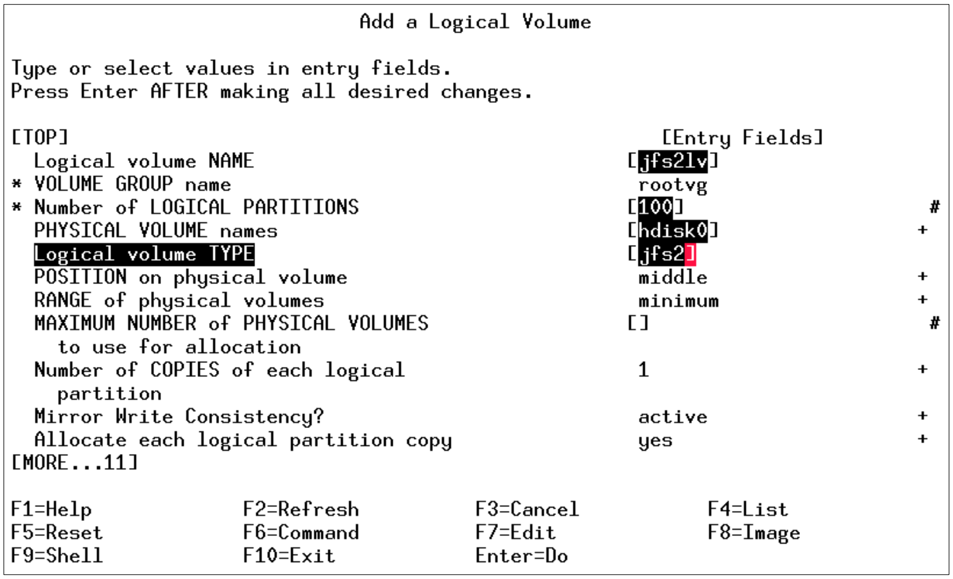
```
                     Add an Enhanced Journaled File System

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

                                                       [Entry Fields]
  Volume group name                                   rootvg
* SIZE of file system (in 512-byte blocks)           [512000]             #
* MOUNT POINT                                        [/jfs2]
  Mount AUTOMATICALLY at system restart?              yes                +
  PERMISSIONS                                         read/write         +
  Mount OPTIONS                                      []                  +
  Block Size (bytes)                                  4096               +
  Inline Log?                                         no                 +
  Inline Log size (MBytes)                           []                   #






F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 5. Example of JFS2 file system creation through SMIT

Add on a Previously Defined Logical Volume

If a non-default logical volume is needed for the JFS2 file system creation, this logical volume must be defined prior to the file system creation.

The Logical Volume type must be assigned as JFS2; otherwise, it will not appear as a selectable logical volume in the file system creation, as shown in Figure 6.

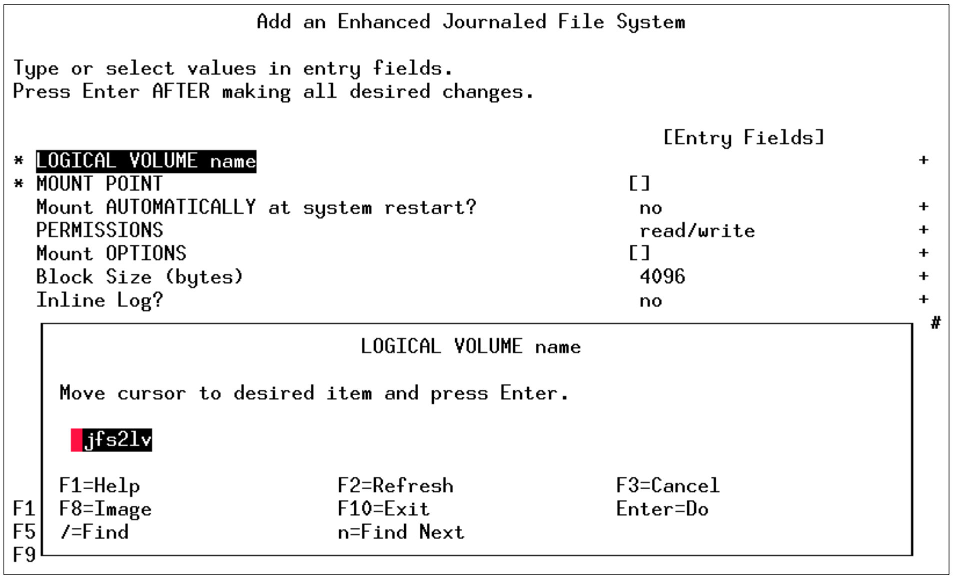


```
                              Add a Logical Volume

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

[TOP]                                                    [Entry Fields]
  Logical volume NAME                                 [jfs2lv]
* VOLUME GROUP name                                    rootvg
* Number of LOGICAL PARTITIONS                        [100]                   #
  PHYSICAL VOLUME names                               [hdisk0]               +
  Logical volume TYPE                                 [jfs2]
  POSITION on physical volume                          middle                +
  RANGE of physical volumes                            minimum               +
  MAXIMUM NUMBER of PHYSICAL VOLUMES                  []                      #
    to use for allocation
  Number of COPIES of each logical                     1                     +
    partition
  Mirror Write Consistency?                            active                +
  Allocate each logical partition copy                 yes                   +
[MORE...11]

F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 6. Logical volume type entry field

After creating the logical volume, you must associate this logical volume with the file system to be created. Go to the SMIT JFS2 panel and choose the second option.

If the logical volume was created correctly, it must appear as a selectable logical volume, as shown in Figure 7.



```
                         Add an Enhanced Journaled File System

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

                                                         [Entry Fields]
* LOGICAL VOLUME name                                                       +
* MOUNT POINT                                          []
  Mount AUTOMATICALLY at system restart?                no                  +
  PERMISSIONS                                           read/write          +
  Mount OPTIONS                                        []                   +
  Block Size (bytes)                                    4096                +
  Inline Log?                                           no                  +
                                                                             #
                                 LOGICAL VOLUME name

    Move cursor to desired item and press Enter.

      jfs2lv

    F1=Help                  F2=Refresh                F3=Cancel
F1  F8=Image                 F10=Exit                  Enter=Do
F5  /=Find                   n=Find Next
F9
```

Figure 7. SMIT panel showing the logical volume selection

After selecting the correct logical volume, you have to complete the relevant SMIT fields.

3.3.3.2 Command Line Interface

It is also possible to create the JFS2 file system using the command line interface (CLI). An additional VFS type was added to the `crfs` command.

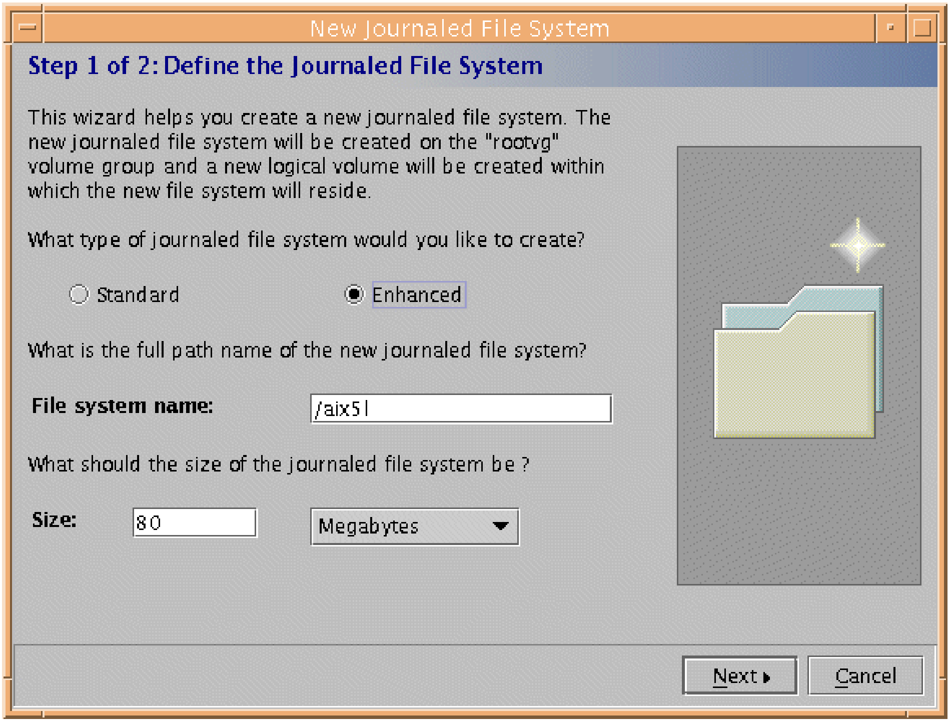
When using CLI operations, the `crfs` command requires a *-v jfs2* flag in order to create a JFS2-type file system.

```
# crfs -v jfs2 -g rootvg -a size=1 -m /jfs2 -A yes -p rw -a agblksize=4096
mkfs completed successfully.
16176 kilobytes total disk space.
New File System size is 32768.
```

The output above illustrates a `crfs` command used to create a /jfs2 file system using JFS2.

3.3.3.3 Web-based System Manager

You can manage JFS2 file systems from the Web-based System Manager interface. It is possible to create, enlarge, remove, and monitor JFS2 file systems from this management tool, as shown in Figure 8.


New Journaled File System

Step 1 of 2: Define the Journaled File System

This wizard helps you create a new journaled file system. The new journaled file system will be created on the "rootvg" volume group and a new logical volume will be created within which the new file system will reside.

What type of journaled file system would you like to create?

○ Standard ● Enhanced

What is the full path name of the new journaled file system?

File system name: /aix5l

What should the size of the journaled file system be ?

Size: 80 Megabytes

Figure 8. Web-based System Manager panel for file system creation

3.3.3.4 Check and Recover File System

The `fsck` utility was enhanced to also handle JFS2-type file systems. This utility checks the file system for consistency and repairs problems found.

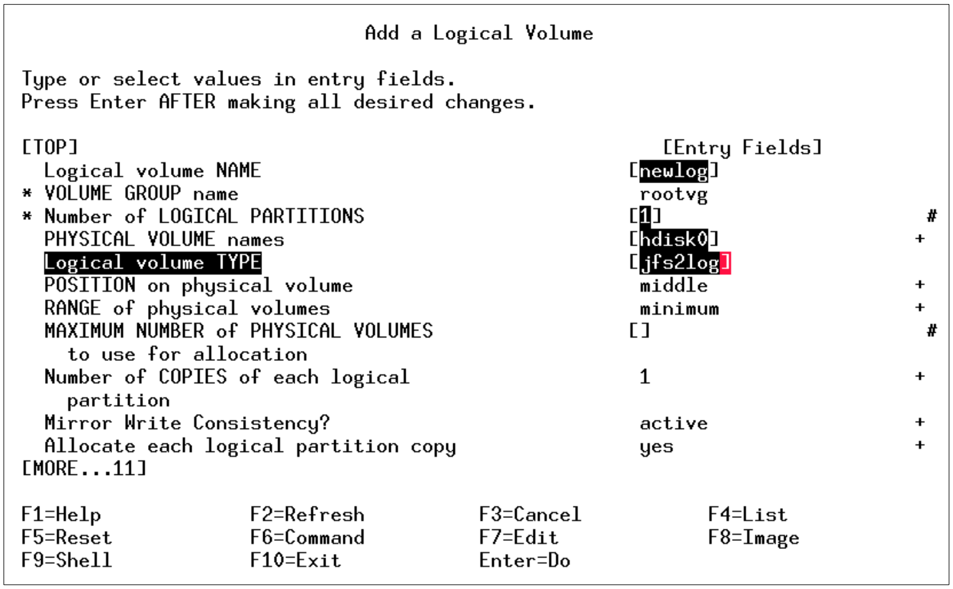
```
# fsck -V jfs2 /myfs
****************
The current volume is: /dev/lv01
File system is clean.
All observed inconsistencies have been repaired.
```

If the -V flag is not specified, `fsck` will figure out the JFS type by the VFS type specified for this file system and work in the assumed way:

```
# fsck /myfs
****************
The current volume is: /dev/lv01
File system is clean.

All observed inconsistencies have been repaired.
```

3.3.3.5 Creating a JFS2 Log Device

If you need to create a separate log device for a JFS2 file system, you must specify JFS2LOG as the logical volume type, as shown in Figure 9.



```
                         Add a Logical Volume

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

[TOP]                                                  [Entry Fields]
  Logical volume NAME                                [newlog]
* VOLUME GROUP name                                   rootvg
* Number of LOGICAL PARTITIONS                       [1]               #
  PHYSICAL VOLUME names                              [hdisk0]         +
  Logical volume TYPE                                [jfs2log]
  POSITION on physical volume                         middle          +
  RANGE of physical volumes                           minimum         +
  MAXIMUM NUMBER of PHYSICAL VOLUMES                 []                #
    to use for allocation
  Number of COPIES of each logical                    1               +
    partition
  Mirror Write Consistency?                           active          +
  Allocate each logical partition copy                yes             +
[MORE...11]

F1=Help            F2=Refresh          F3=Cancel           F4=List
F5=Reset           F6=Command          F7=Edit             F8=Image
F9=Shell           F10=Exit            Enter=Do
```

Figure 9. Adding a logical volume as a jfs2log device

Otherwise, you will not be able to format the log device and use it as a log for a JFS2 file system.

3.3.3.6 Format a JFS2 Log Device

If you need to format a separate log device for a JFS2 file system, keep in mind that the `logform` command is set to -V jfs2 flag in order to create a correct type of log device. For example:

```
# logform -V jfs2 /dev/jfs2log
logform: destroy /dev/jfs2log (y)?y
```

If the `-V` flag is not specified, the `logform` command will try to determine what kind of log device will be created through the VFS information encountered in the logical volume.

To verify the VFS type of a logical volume, you must check the output of the following command:

```
# lslv newlog | grep TYPE
TYPE:              jfs2log              WRITE VERIFY:   off
```

3.3.3.7 Inline log

A new type of log can be created for JFS2 type file systems. An inline log is a feature specific to JFS2 file systems that allows you to create the log within the same data logical volume.

With an inline log, each JFS2 file system can have its own log device without having to share this device. For a scenario with multiples of hot swap disk devices and large number of file systems, this feature can be used to improve RAS if a system loses a single disk that contains the log device for multiple file systems. See Figure 5 on page 38 for the SMIT panel with inline log enablement.

In the following example, the output for the `mount` command shows the logical volume and log device as the same device.

```
# mount
node       mounted        mounted over    vfs       date        options
----- ---------------  ---------------  ------ ------------ ---------------
      /dev/hd4         /                jfs    Sep 01 11:32 rw,log=/dev/hd8
      /dev/hd2         /usr             jfs    Sep 01 11:32 rw,log=/dev/hd8
      /dev/hd9var      /var             jfs    Sep 01 11:32 rw,log=/dev/hd8
      /dev/hd3         /tmp             jfs    Sep 01 11:32 rw,log=/dev/hd8
      /dev/hd1         /home            jfs    Sep 01 11:33 rw,log=/dev/hd8
      /proc            /proc            procfs Sep 01 11:33 rw
      /dev/lv02        /jfs22           jfs2   Sep 05 10:00 rw,log=/dev/lv02
```

3.4 NFS statd multithreading

In AIX 5L, the NFS statd daemon is multithreaded. In AIX Version 4.3, when the statd daemon is detecting whether the clients are up or not, it hangs and waits for a time out when a client can not be found. If there are a large number of clients that are offline, it can take a long time to time out all of them sequentially.

With a multithreading design, stat requests run in parallel to solve the time-out problem. The server statd monitors clients and the client’s statd monitors the server if a client has multiple mounts. Connections are dropped if the remote partner cannot be detected without affecting other stat operations. The following example is an output from the `ps -mo THREAD` command that shows three different threads for rpc.statd daemon.

```
# ps -mo THREAD -p 17570
    USER   PID  PPID    TID ST  CP PRI SC    WCHAN        F     TT BND COMMAND
  daemon 17570  6456      - A    0  60  3        -   240001      -   - /usr/sbin
/rpc.statd
       -     -     -  20409 S    0  60  1        -   418400      -   - -
       -     -     -  26065 Z    0  60  1        -   c00001      -   - -
       -     -     -  26579 Z    0  60  1        -   c00001      -   - -
```

3.5 Multithreaded AutoFS

In AIX 5L, the automountd daemon implementing the AutoFS function is now multithreaded, as can be seen from the following output of the ps command.

```
# ps -fmo THREAD -p 19134
    USER   PID  PPID    TID ST  CP PRI SC    WCHAN        F     TT BND COMMAND
    root 19134  6456      - A    0  60  2 e60056a0   240001      -   - /usr/sbin
/automountd
       -     -     -  35747 S    0  60  1        -   418400      -   - -
       -     -     -  44443 S    0  60  1 e60056a0  8410400      -   - -
```

With this new feature, the AutoFS mounter daemon remains responsive, even if one of the servers from which it tries to mount file systems becomes unavailable. As a single-threaded application, it would not be possible for the kernel to switch to the corresponding process if that process waits for a network connection to an unresponsive server.

3.6 Cache file system enhancements

In AIX 5L,the cache file system (cachefs) allows 64-bit operations. In both a 64-bit and 32-bit environment, cachefs now handles files larger than 2 GB. In AIX Version 4.3.3 and earlier releases, cachefs only runs on a 32-bit system and all files must be 2 GB at a maximum.

When making the transition from a 32-bit POWER kernel to a 64-bit POWER kernel, there is no need to recreate the cache directory.

3.7 Passive mirror write consistency check

AIX 5L introduces a new passive mirror write consistency check (MWCC) algorithm for mirrored logical volumes. This option only applies to big volume groups.

Previous versions of AIX used a single MWCC algorithm which is now called the active MWCC algorithm to distinguish it from the new algorithm. With active MWCC, records of the last 62 distinct logical transfer groups (LTG) written to disk are kept in memory and also written to a separate checkpoint area on disk. Because only new writes are tracked, if new MWCC tracking tables have to be written out to the disk checkpoint area, the disk performance can degrade if there are a lot of random write requests issued. The purpose of the MWCC is to guarantee the consistency of the mirrored logical volumes in case of a crash. After a system crash, the logical volume manager will use the LTG tables in the MWCC copies on disk to make sure that all mirror copies are consistent.

The new passive MWCC algorithm does not use an LTG tracking table, but sets a dirty bit for the mirrored logical volume as soon as the volume is opened for writes. This bit gets cleared only if the volume is successfully synced and is closed. In the case of a system crash, the entire mirrored logical volume will undergo a background re-synchronization spawned during vary-on of the volume group, because the dirty bit has not been cleared. Once the background re-synchronization completes, the dirty bit is cleared, but can be reset at any time if the mirrored logical volume is opened. It should be noted that the mirrored logical volume can be used immediately after system reboot, even though it is undergoing background re-synchronization.

The trade-off for the new passive MWCC algorithm compared to the default active MWCC algorithm is better performance during normal system operations. However, there is additional I/O that may slow system performance during the automatic background re-synchronization that occurs during recovery after a crash.

The `lslv` and `chlv` commands have been changed accordingly. Instead of outputting just an off or on in the MIRROR WRITE CONSISTENCY field, the value now reads on/ACTIVE or on/PASSIVE, as shown in the following example:

```
# lslv lv00
LOGICAL VOLUME:     lv00                   VOLUME GROUP:   software
LV IDENTIFIER:      000bc6fd00004c00000000e1b374aba8.2 PERMISSION:
read/write
VG STATE:           active/complete        LV STATE:       opened/syncd
TYPE:               jfs                    WRITE VERIFY:   off
MAX LPs:            512                    PP SIZE:        8 megabyte(s)
COPIES:             1                      SCHED POLICY:   parallel
LPs:                62                     PPs:            62
STALE PPs:          0                      BB POLICY:      relocatable
INTER-POLICY:       minimum                RELOCATABLE:    yes
INTRA-POLICY:       middle                 UPPER BOUND:    32
MOUNT POINT:        /software              LABEL:          /software
MIRROR WRITE CONSISTENCY: on/ACTIVE
EACH LP COPY ON A SEPARATE PV ?: yes
```

The -w flag for the `chlv` command now accepts either an a or y option to turn on active mirror write consistency checking, or a p option to use the new passive MWCC algorithm. The n option turns off mirror write consistency checking.

3.8 Thread-safe liblvm.a

In AIX 5L, the libraries implementing query functions of the logical volume manager (LVM) functions (liblvm.a) is now thread-safe. Because LVM commands must be able to run even when the system is booting or getting installed, the LVM library cannot rely on the availability of the pthread support library. Therefore, the internal architecture of the liblvm.a library ensures that the library is thread safe.

The following libraries are now thread safe:

- lvm_querylv
- lvm_querypv
- lvm_queryvg
- lvm_queryvgs

Chapter 4. System management and utility enhancements

AIX 5L provides many enhancements in the area of system management and utilities. This chapter discusses these enhancements.

4.1 Fast device configuration enhancement

AIX 4.3.3 introduced a new device configuration methodology in order to reduce the time needed to detect and configure all the devices attached to the system. The `cfgmgr` command was changed so that it can run device configuration methods in parallel rather than sequentially (one at a time). This function did not support every device on every bus type.

AIX 5L adds support for parallel configuration of Fiber Channel (FC) adapters and devices and an expanded list of devices and bus types (provided below):

- Fiber Channel adapters and devices (POWER platform)
- PCI buses on CHRP systems (POWER platform)
- PCI SCSI adapters on CHRP and PReP systems (POWER platform)
- PCI async adapter and their concentrators on CHRP and PReP systems (POWER platform)
- SCSI disks on any POWER platform
- TTYs on any POWER platform

This feature is only available on the POWER platform.

4.2 Paging space enhancements

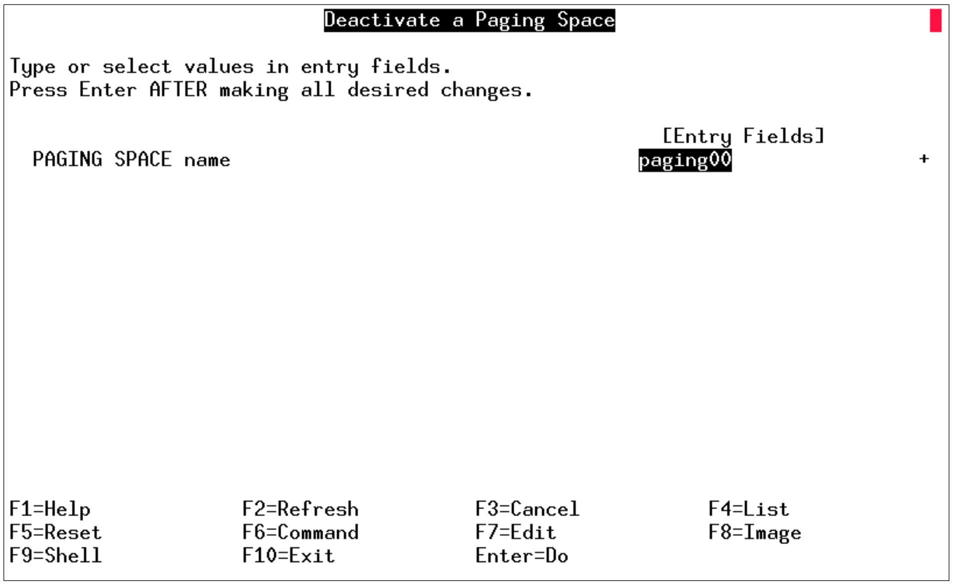
AIX 5L provides two enhancements for managing paging space. A new command, `swapoff`, allows you to deactivate a paging space. The -d flag, for the `chps` command, provides the ability to decrease the size of a paging space. For both commands, a system reboot is no longer required.

4.2.1 Deactivating a paging space

To deactivate a paging space with the `swapoff` command, you can either use:

```
# swapoff device name { device name ... }
```

or the SMIT panel (fast path swapoff), as shown in Figure 10.



```
                        Deactivate a Paging Space

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

                                                        [Entry Fields]
  PAGING SPACE name                                    paging00          +

F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 10. Deactivate a paging space

This command may fail due to

- Paging space size constraints
- I/O errors

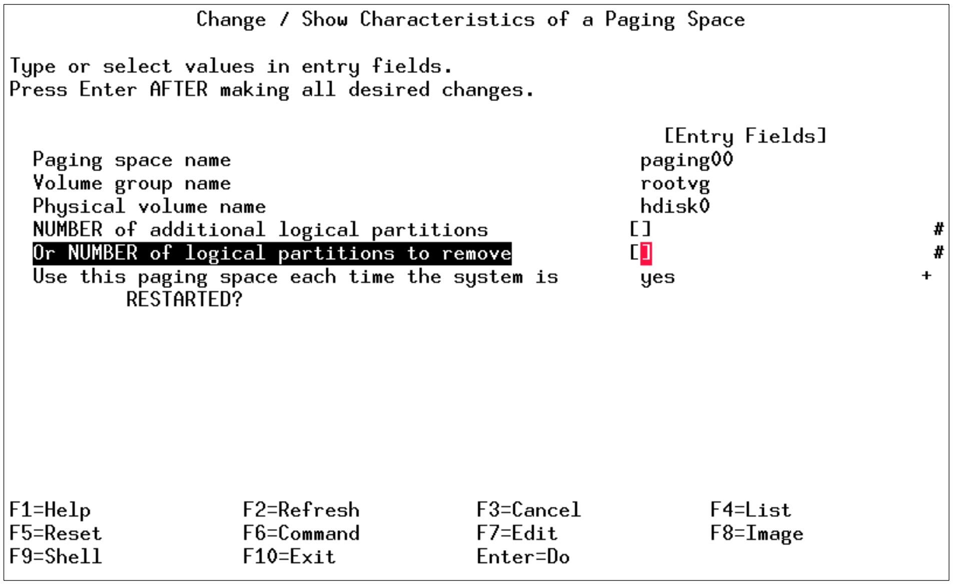
Because it is necessary to move all pages (in use on the paging space) to be deactivated to other paging spaces, there must be enough space available in the other active paging spaces. Basically, this command pages in all active pages (after marking the paging space to be deactivated as unavailable) and allows the AIX VMM to page these pages out again to the other available paging spaces. In the case of I/O errors, you should check the error log, deactivate the paging space you are working on for the next system reboot with the `chps` command, and reboot the system. Do not try to reactivate paging spaces with I/O errors before you have checked the corresponding disk with the appropriate diagnostic tools. The `lsps` command will display, in this case, the string `I/O err` in the column with the heading Active.

4.2.2 Decreasing the size of a paging space

By using the new -d flag, you can decrease the size of an existing paging space using the `chps` command as follows:

```
# chps -dLogicalPartitions PagingSpace
```

or specify it on the SMIT panel (fast path chps), as shown in Figure 11.



```
                Change / Show Characteristics of a Paging Space

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

                                                     [Entry Fields]
  Paging space name                                  paging00
  Volume group name                                  rootvg
  Physical volume name                               hdisk0
  NUMBER of additional logical partitions           []                    #
  Or NUMBER of logical partitions to remove         []                    #
  Use this paging space each time the system is      yes                 +
          RESTARTED?

F1=Help              F2=Refresh           F3=Cancel            F4=List
F5=Reset             F6=Command           F7=Edit              F8=Image
F9=Shell             F10=Exit             Enter=Do
```

Figure 11. Decreasing the size of a paging space

The actual processing is done by the shell script `shrinkps`. In the case of decreasing the size of an active paging space, `shrinkps` will create a temporary paging space, move all pages from the paging space to be decreased to this temporary one, delete the old paging space, recreate it with the new size, move all the pages back, and finally delete the temporary paging space. This temporary paging space is always created in the same volume group as the one you try to decrease. It is therefore necessary to have enough space available in the volume group for this temporary paging space. If you decrease the size of a deactivated paging space, the creation of a temporary paging space is not necessary and therefore omitted.

The following example shows the commands needed to remove one logical partition from paging01:

```
# lsps -a
Page Space  Physical Volume   Volume Group    Size   %Used  Active  Auto  Type
paging01    hdisk0            rootvg          48MB       1     yes   yes    lv
hd6         hdisk0            rootvg          32MB      11     yes   yes    lv
# chps -d 1 paging01
shrinkps: Temporary paging space paging00 created.
shrinkps: Paging space paging01 removed.
shrinkps: Paging space paging01 recreated with new size.
# lsps -a
Page Space  Physical Volume   Volume Group    Size   %Used  Active  Auto  Type
paging01    hdisk0            rootvg          32MB       1     yes   yes    lv
hd6         hdisk0            rootvg          32MB      12     yes   yes    lv
```

As you can imagine from the above description, the deactivation or decrease in size of an active paging space can result in a noticeable performance degradation, depending on the size and usage of the paging space and the current system workload. But the main advantage is that there is no system reboot necessary to rearrange the paging space.

If you are working with the primary paging space (usually hd6), this command will prevent you from decreasing the size below 32 MB or actually deleting it. If you decrease the primary paging space, a temporary boot image and a temporary /sbin/rc.boot pointing to this temporary primary paging space will be created to make sure the system is always in a state where it can be safely rebooted.

Note

These command enhancements are not available through the Web-based System Manager. The Web-based System Manager allows you, by default, to specify the increase in size for a paging space in the Megabytes field.

4.3 Error log enhancements

AIX 5L provides three enhancements in the area of error logging. First, you can specify a time threshold that treats identical errors arriving closer than this threshold as duplicates and counts them only once. Second, with the `errpt` command, you can now request an intermediate format that removes seldom needed data from the detailed error report format. A third enhancement, the diagnostic tool, will now put additional information into the error log entry.

4.3.1 Elimination of duplicate errors

The errdemon command was enhanced in AIX 5L to support four additional flags. The flags -D and -d specify if duplicate error log entries are to be removed or not. The default is the -D flag, which instructs the command to remove the duplicates. With the -t and -m flags, you can control what is considered a duplicate error log entry. A value in the range 1 to 2^{31} - 1 specifies the time in milliseconds within which an error identical to the previous one is considered a duplicate. The default value for this flag is 100 or 0.1 seconds. The -m flag sets a count, after which the next error is no longer considered a duplicate of the previous one. The range for this value is 1 to 2^{31} - 1 with a default of 1000.

The following command increases the time threshold to 1 second and the number of duplicates after which the same error would again be counted as a new one to 100000:

```
# /usr/lib/errdemon -m 100000 -t 1000
```

The errpt command also has a new -D flag, which consolidates duplicate errors. In conjunction with the -a flag, only the number of duplicate errors and the timestamps for the first and last occurrence are reported. This is complemented by a new -P flag, which displays only the duplicate errors logged by the new mechanisms of errdemon mentioned previously.

4.3.2 Enhancements for the errpt command

In addition to the two new flags (-D and -P) mentioned in the previous section, errpt now supports an intermediate output format using the -A flag, in addition to the summary and the details already provided. In this intermediate format, only the values for LABEL, Date/Time, Type, Resource Name, Description, and Detail Data are displayed.

The following lines show the output of the errpt command for one specific error using the summary, intermediate, and detailed option, respectively:

```
# errpt -j 9DBCFDEE
IDENTIFIER TIMESTAMP  T C RESOURCE_NAME  DESCRIPTION
9DBCFDEE   0919101600 T O errdemon       ERROR LOGGING TURNED ON
# errpt -A -j 9DBCFDEE
---------------------------------------------------------------------------
LABEL:          ERRLOG_ON
Date/Time:       Tue Sep 19 10:16:41 CDT
Type:            TEMP
Resource Name:   errdemon
Description
ERROR LOGGING TURNED ON
# errpt -a -j 9DBCFDEE
---------------------------------------------------------------------------
LABEL:          ERRLOG_ON
```

```
IDENTIFIER:     9DBCFDEE

Date/Time:       Tue Sep 19 10:16:41 CDT
Sequence Number: 1
Machine Id:      000BC6FD4C00
Node Id:         localhost
Class:           O
Type:            TEMP
Resource Name:   errdemon

Description
ERROR LOGGING TURNED ON

Probable Causes
ERRDEMON STARTED AUTOMATICALLY

User Causes
/USR/LIB/ERRDEMON COMMAND

        Recommended Actions
        NONE
```

4.3.3 Link between error log and diagnostics

When the diagnostic tool runs, it automatically tries to diagnose hardware errors it finds in the error log. Starting with AIX 5L, the information generated by the `diag` program is put back into the error log entry, so that it is easy to make the connection between the error event and, for example, the FRU number required to repair failing hardware. The following lines show an example of this process: first the header of the error log entry is shown, and then the information added by the diagnostic tool:

```
LABEL:EPOW_SUS_CHRP
IDENTIFIER:BE0A03E5

Date/Time:       Wed Sep 20 13:47:27 CDT
Sequence Number: 14
Machine Id:      000BC6DD4C00
Node Id:         server3
Class:           H
Type:            PERM
Resource Name:   sysplanar0
Resource Class:  planar
Resource Type:   sysplanar_rspc
Location:        00-00
...
Diagnostic Analysis
Diagnostic Log sequence number:8
Resource tested:sysplanar0
Resource Description:System Planar
Location:P1
SRN:     651-812
```

```
Description:System shutdown due to: 1) Loss of AC power, 2)
                            Power button was pushed without proper
                            system shutdown, 3) Power supply failure.
```

4.4 Resource Monitoring and Control (RMC)

In AIX 5L, a new Resource Monitoring and Control (RMC) subsystem is available that is comparable in function to the Reliable Scalable Cluster Technology (RSCT) on the IBM SP type of machines. Therefore, the two terms RMC and RSCT may be used as though they are the same, but RMC is the new subsystem available as part of base AIX 5L, while RSCT has a broader, more general meaning. This subsystem allows you to associate predefined responses with predefined conditions for monitoring system resources. An example is to broadcast a message when the /tmp file system becomes 90 percent full to summon the attention of a dutiful system administrator.

At the time of writing, this feature is only available on the POWER platform.

4.4.1 Packaging and installation

The RMC subsystem is installed by default and is delivered in one bundle named rsct.core containing nine different filesets with the following names:

```
# lslpp -L "*rsct*"
  Fileset                      Level  State  Description
  ----------------------------------------------------------------------------
  rsct.core.auditrm          2.1.0.0    C    RSCT Audit Log Resource Manager
  rsct.core.errm             2.1.0.0    C    RSCT Event Response Resource
                                             Manager
  rsct.core.fsrm             2.1.0.0    C    RSCT File System Resource
                                             Manager
  rsct.core.gui              2.1.0.0    C    RSCT Graphical User Interface
  rsct.core.hostrm           2.1.0.0    C    RSCT Host Resource Manager
  rsct.core.rmc              2.1.0.0    C    RSCT Resource Monitoring and
                                             Control
  rsct.core.sec              2.1.0.0    C    RSCT Security
  rsct.core.sr               2.1.0.0    C    RSCT Registry
  rsct.core.utils            2.1.0.0    C    RSCT Utilities
```

All executables and related items are installed into the /usr/sbin/rsct directory, while the log files and other temporary data is located in /var/ct. The following entry is located in /etc/inittab:

```
ctrmc:2:once:/usr/bin/startsrc -s ctrmc > /dev/console 2>&1
```

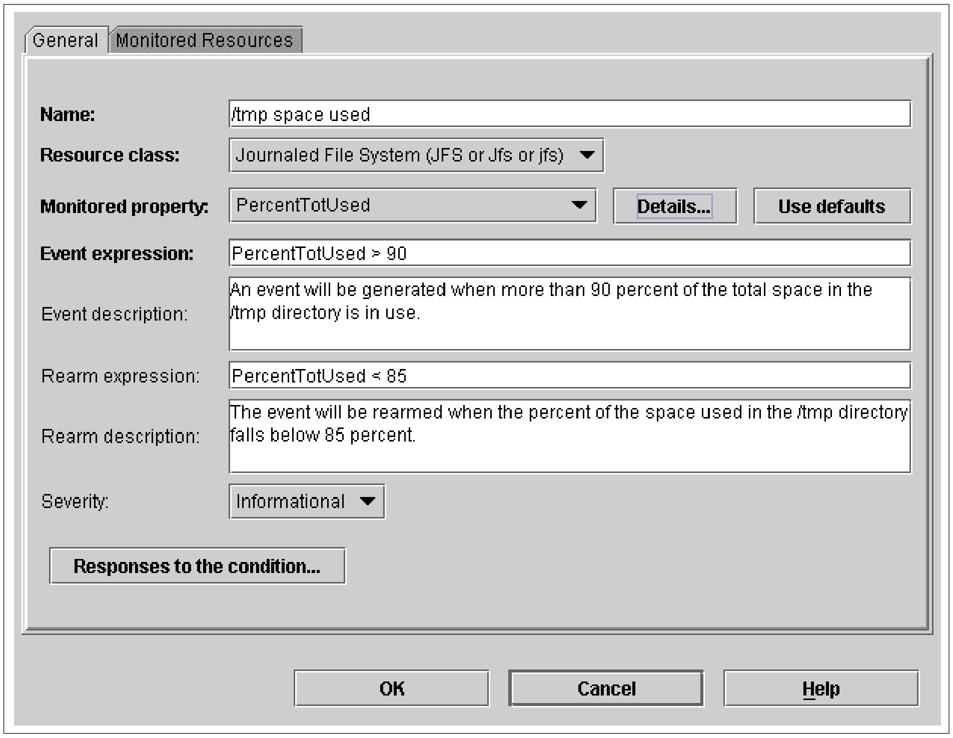
Due to this entry, the RMC subsystem is also automatically started. This subsystem can be controlled using the SRC commands, but it also has its own control command (`/usr/sbin/rsct/bin/rmcctrl`), which is the preferred way to stop and start it. Due to the number of available options on this subsystem, it can only be controlled through the Web-based System Manager. A SMIT interface is not available at the time of publication.

4.4.2 Concepts of RMC

The basic function of RMC is based on two concepts: conditions and responses. To provide you a ready-to-use system, 84 conditions and 8 responses are predefined for you. You can use them as they are, customize them, or use them as templates to define your own conditions and responses. To monitor a condition, simply associate one or more responses with the condition.

A condition monitors a specific property, such as total percentage used, in a specific resource class, such as JFS. You can monitor the condition for one, or more, or all the resources within the monitored property, such as /tmp, or /tmp and /var, or all the file systems. Each condition contains an event expression to define an event and an optional rearm expression to define a rearm event. The event expression is a combination of the monitored property, mathematical operators, and some numbers, such as PercentTotUsed > 90 in the case of a file system. The rearm expression is a similar entity; for example, PercentTotUsed < 85.

The following figures provide an example of a condition property dialog with two tabs: General (Figure 12) and Monitored Resource (Figure 13 on page 56).



Name: /tmp space used

Resource class: Journaled File System (JFS or Jfs or jfs)

Monitored property: PercentTotUsed

Event expression: PercentTotUsed > 90

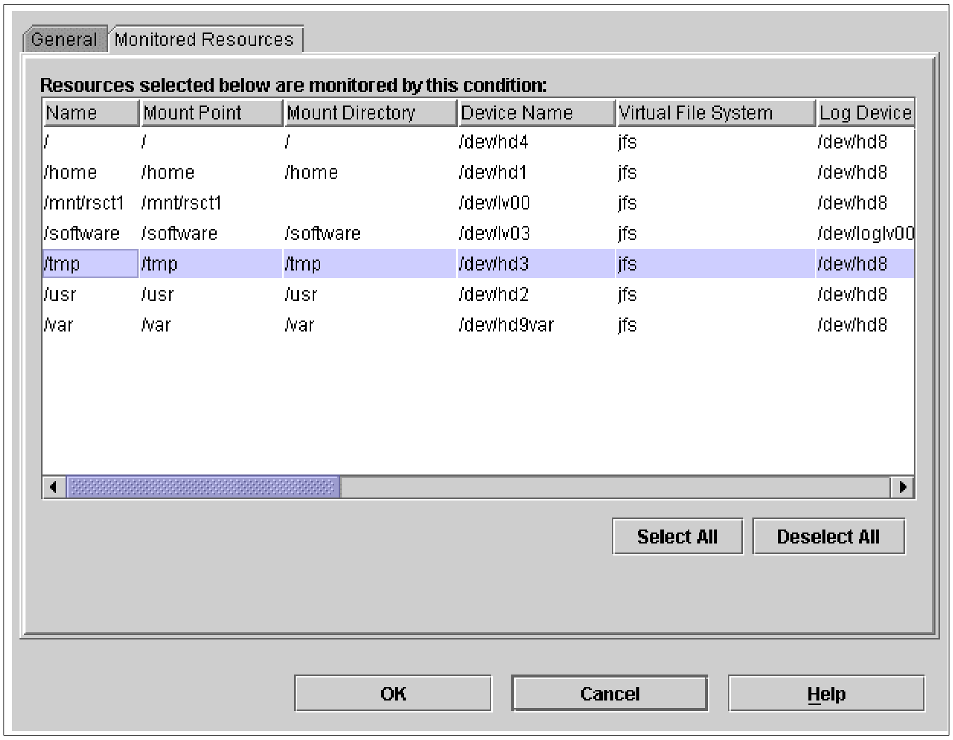
Event description: An event will be generated when more than 90 percent of the total space in the /tmp directory is in use.

Rearm expression: PercentTotUsed < 85

Rearm description: The event will be rearmed when the percent of the space used in the /tmp directory falls below 85 percent.

Severity: Informational

Figure 12. Condition Properties dialog - General tab

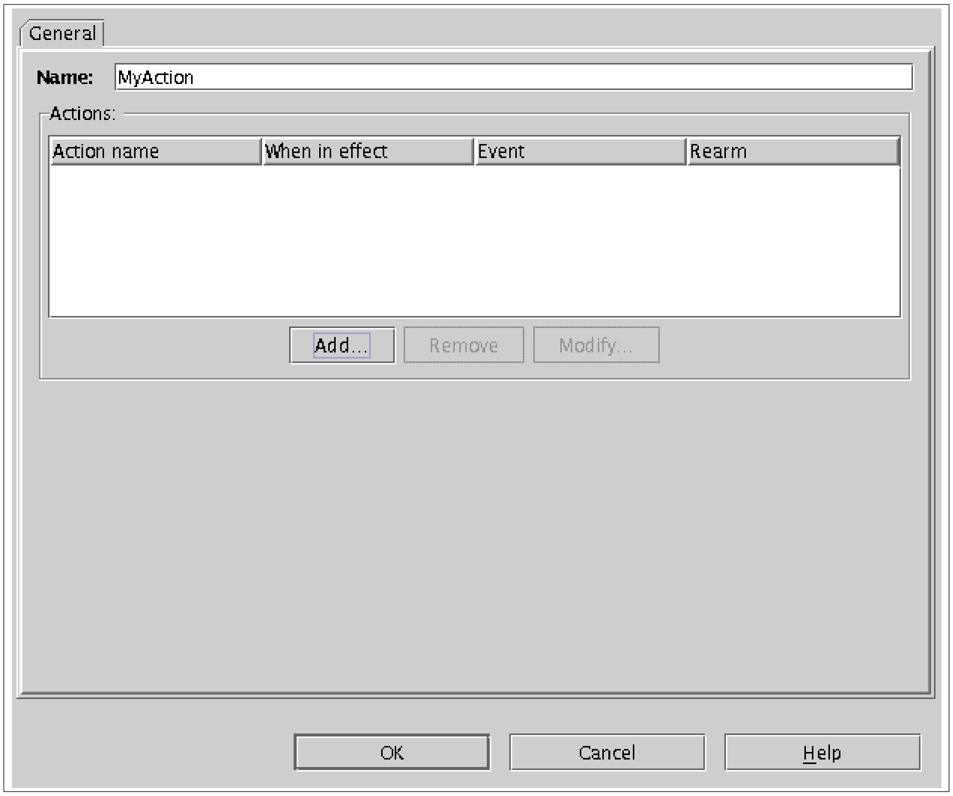


Resources selected below are monitored by this condition:

Name	Mount Point	Mount Directory	Device Name	Virtual File System	Log Device
/	/	/	/dev/hd4	jfs	/dev/hd8
/home	/home	/home	/dev/hd1	jfs	/dev/hd8
/mnt/rsct1	/mnt/rsct1		/dev/lv00	jfs	/dev/hd8
/software	/software	/software	/dev/lv03	jfs	/dev/loglv00
/tmp	/tmp	/tmp	/dev/hd3	jfs	/dev/hd8
/usr	/usr	/usr	/dev/hd2	jfs	/dev/hd8
/var	/var	/var	/dev/hd9var	jfs	/dev/hd8

Figure 13. Condition Properties dialog - Monitored Resources tab

Each response can consist of one or more actions. Figure 14 on page 57 provides an example for a Response Properties dialog.



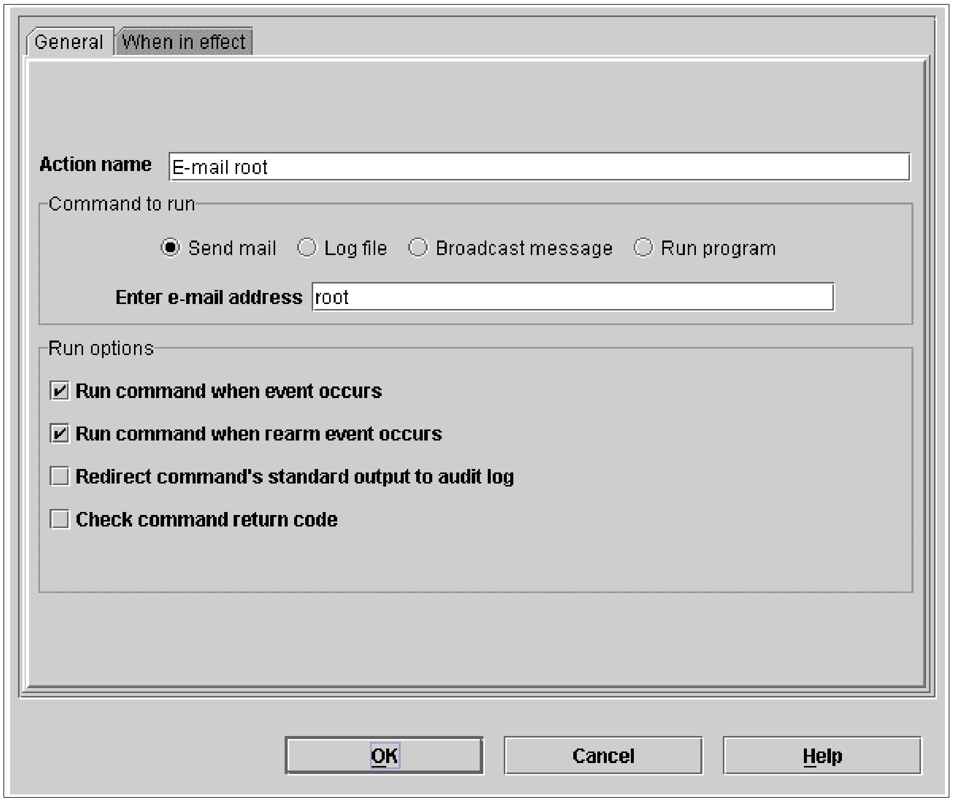
Name: MyAction

Figure 14. Response Properties dialog - General tab

The Add or Modify buttons launch an Action Properties dialog.

To define an action, you can choose one of the three predefined commands Send mail, Broadcast a message, or Log an entry to a file, or you can specify an arbitrary program or a script of your own by using the Run program option. The action can be active for an event only, for a rearm event only, or for both. You can also specify a time window in which the action is active, such as always, or only during on-shift on weekdays.

The following figures provide an example of an Action Properties dialog with two tabs: General (Figure 15 on page 58) and a When in Effect (Figure 16 on page 59).



Action name: E-mail root

Command to run

● Send mail ○ Log file ○ Broadcast message ○ Run program

Enter e-mail address: root

Run options

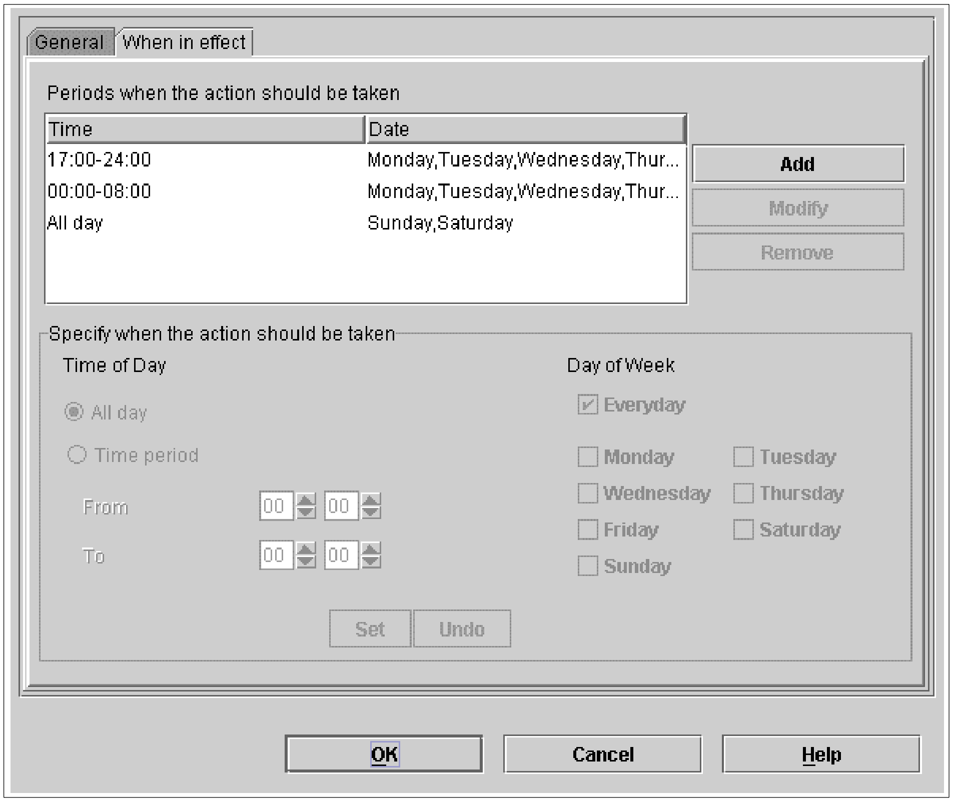
☑ Run command when event occurs

☑ Run command when rearm event occurs

□ Redirect command's standard output to audit log

□ Check command return code

Figure 15. Action Properties dialog - General tab



Periods when the action should be taken

Time	Date
17:00-24:00	Monday,Tuesday,Wednesday,Thur...
00:00-08:00	Monday,Tuesday,Wednesday,Thur...
All day	Sunday,Saturday

Specify when the action should be taken

Time of Day

● All day

○ Time period

From 00 00

To 00 00

Day of Week

☑ Everyday

□ Monday □ Tuesday

□ Wednesday □ Thursday

□ Friday □ Saturday

□ Sunday

Figure 16. Action Properties dialog - When in Effect tab

The previously mentioned predefined commands are using the notifyevent, wallevent, and logevent scripts, respectively, in the /usr/sbin/rsct/bin subdirectory. These command scripts capture events through the Event Response Resource Manager (ERRM) environment variables and notify you of the events through e-mails, broadcast messages, and logs. Do not modify these predefined command scripts. However, you can copy these predefined commands as templates to create your own scripts and use them for the “Run program” option.

Note that singe the logevent script uses the `alog` command to log events to the files you designate, the content of these files can be listed with the `alog` command.

If the event expression of a condition is evaluated to be true, an event occurs, and the ERRM checks all responses associated with the condition and executes the event actions defined in these responses. Only after the rearm expression becomes true and the ERRM has executed the corresponding rearm event actions defined in the responses, can the event and the event actions be generated again.

For each of the event and rearm events, the actions taken in response to them, and the success or failure of any commands running in these actions are logged by the Audit Log Resource Manager (AuditRM) to the audit log. The standard error of a run command, if any, is always logged to the audit log. The standard output of a run command is logged to the audit log only if the “Redirect command’s standard output to audit log” option is selected for the command in the Action Properties dialog. The audit log records can be listed with the `lsaudrec` command and removed from the log file with the `lsaudrec` command and removed from the log file with the `rmaudrec` command.

4.4.3 How to set up an efficient monitoring system

The following steps are provided to assist you in setting up an efficient monitoring system.

1. Review the predefined conditions of your interests. Use them as they are, customize them to fit your configurations, or use them as templates to create your own.
2. Review the predefined responses. Customize them to suit your environment and your working schedule. For example, the response "Critical notifications" is predefined with three actions, log events to /tmp/criticalEvents, e-mail to root, and broadcast message to all logged-in users anytime when an event or a rearm event occurs. You may modify the response such as to log events to a different file anytime when events occur, e-mail to you during non-working hours, and add a new action to page you only during working hours. With such as setup, different notification mechanisms can be automatically switched based on your working schedule.
3. Re-use the responses for conditions. For example, you can customize the three severity responses, "Critical notifications," "Warning notifications," and "Informational notifications" to take actions in response to events of different severities, and associate the responses to the conditions of respective severities. With only 3 notification responses, you can get notified of all the events with respective notification mechanisms based on their urgencies.
4. Once the monitoring is setup, your system continues being monitored whether your Web-based System Manager session is running or not. To know the system status, you may bring up a Web-based System Manager

session and view the Events plug-in, or simply use the lsaudrec command from the command line interface to view the audit log.

4.4.4 Resources

The resources that can be monitored are managed by two resource managers: the File System Resource Manager (FSRM), and the Host Resource Manager (HostRM).

The FSRM monitors all local JFSs on a machine and checks for the status (offline, online), the total percentage used, and the percentage of inodes used in the file system. It is currently not possible to monitor JFS2.

The HostRM supports nine different resource classes. The network adapter resource classes (Ethernet Device, Token Ring Device, ATM Device and FDDI Device) each monitor five different properties, such as receive error rates and others. There is one resource class (Physical Volume) supporting the monitoring of hard disk. It checks for four different properties, such as percentage of time the device was busy between two consecutive observations. The percentage of free paging space is currently the only supported property of the resource class Paging Device. The Processor resource class monitors processor utilization by checking, for example, for the idle time property and others.

The Host resource class supports 46 different properties that represent all different areas, in order to get a system-wide status of your machine. This includes, among others, properties such as the size of the system run queue, sizes and change in size of various memory buffer pools in the kernel, and overall utilization of all processors in the system.

The last resource class (Program) checks if a specific program is running or the number of processes for a specific program is changing. The predefined condition in this resource class checks to see if the sendmail daemon is running. You can restrict this condition by specifying a filter expression, which can use the various fields supported by the ps command. This allows, for example, monitoring of only programs running with a specific user ID.

All resource classes support, in addition to their specific properties, a general configuration change property. With this property, you can send a mail to root or any other specified user whenever the configuration of a device changes. The JFS, PagingDevice, and Processor resource classes support the operational state property.

The RMC subsystem is comprised of several multithreaded daemons, as shown in the following output:

```
# ps -mo THREAD -p 5948,20388,21942,23792,25348
USER   PID  PPID    TID ST  CP PRI SC    WCHAN        F     TT BND COMMAND
    root  5948  6456      - A    0  60  3 e6004020   340001      -   -
/usr/sbin/rsct/bin/rmcd -c
       -     -     -   7497 S    0  60  1        -   418410      -   - -
       -     -     -  29165 S    0  60  1        -  2400400      -   - -
       -     -     -  32771 S    0  60  1 e6004020  8c10410      -   - -
    root 20388  6456      - A    0  60 13        *   240001      -   -
/usr/sbin/rsct/bin/IBM.ERrmd
       -     -     -  29441 S    0  60  1 e60039a0  8410410      -   - -
       -     -     -  30481 S    0  60  1 7006686c   410410      -   - -
       -     -     -  31741 S    0  60  1 7005c06c   400410      -   - -
       -     -     -  31761 S    0  60  1        -   418410      -   - -
       -     -     -  32037 S    0  60  1        -  2400400      -   - -
       -     -     -  32513 S    0  60  1 7038ca6c   410410      -   - -
       -     -     -  33033 S    0  60  1 e60040a0  8410410      -   - -
       -     -     -  37155 S    0  60  1 e60048a0  8410410      -   - -
       -     -     -  37413 S    0  60  1 e6004920  8410410      -   - -
       -     -     -  37671 Z    0  61  1        -   c00001      -   - -
       -     -     -  41837 S    0  60  1 e60051a0  8c10410      -   - -
       -     -     -  50319 Z    0  60  1        -   c00001      -   - -
       -     -     -  51191 Z    0  60  1        -   c00001      -   - -
    root 21942  6456      - A    0  60  9        *   240001      -   -
/usr/sbin/rsct/bin/IBM.AuditRMd
       -     -     -  33809 S    0  60  1 70062e6c   410410      -   - -
       -     -     -  34073 S    0  60  1        -   418410      -   - -
       -     -     -  34595 S    0  60  1        -  2400400      -   - -
       -     -     -  34833 S    0  60  1 70179e6c   400410      -   - -
       -     -     -  35091 S    0  60  1 e60044a0  8410410      -   - -
       -     -     -  36125 S    0  60  1 e60046a0  8410410      -   - -
       -     -     -  36381 S    0  60  1 e6004720  8c10410      -   - -
       -     -     -  36639 S    0  60  1 e60047a0  8c10410      -   - -
       -     -     -  36897 S    0  60  1 e6004820  8c10410      -   - -
    root 23792  6456      - A    0  60  8        *   240001      -   -
/usr/sbin/rsct/bin/IBM.FSrmd
       -     -     -  41677 S    0  60  1 e6005120  8410410      -   - -
       -     -     -  43371 S    0  60  1 70126c6c   410410      -   - -
       -     -     -  43641 S    0  60  1        -  2400400      -   - -
       -     -     -  44409 S    0  60  1 70317c6c   400410      -   - -
       -     -     -  47101 S    0  60  1 e6005ba0  8410410      -   - -
       -     -     -  50589 S    0  60  1        -   418410      -   - -
       -     -     -  52393 S    0  60  1 e6006620  8c10410      -   - -
       -     -     -  52659 S    0  60  1 e60066a0  8c10410      -   - -
    root 25348  6456      - A    0  60  7        *   240001      -   -
/usr/sbin/rsct/bin/IBM.HostRMd
       -     -     -  42359 S    0  60  1 e60052a0  8410410      -   - -
       -     -     -  43031 S    0  60  1 e6005420  8c10410      -   - -
       -     -     -  48793 S    0  60  1        -   418410      -   - -
       -     -     -  50879 S    0  60  1        -  2400400      -   - -
       -     -     -  57321 S    0  60  1 e6006fa0  8430410      -   - -
       -     -     -  57831 S    0  60  1 7022786c   410410      -   - -
       -     -     -  58583 S    0  60  1 70391a6c   400410      -   - -
```

The main control daemon (rmcd), the event response daemon (IBM.ERrmd), and the audit daemon (IBM.AuditRMd) run as soon as the RMC subsystem is activated. The file system IBM.FSrmd and host daemon IBM.HostRMd are only active if a file system or host condition, respectively, is monitored.

4.5 Shutdown enhancements

AIX 5L enhances the `shutdown` command with a -l flag to log the output (from select actions during the shutdown) to the file /etc/shutdown.log. The contents of this file appears similar to the following:

```
# cat /etc/shutdown.log

Fri Aug 25 13:21:30 CDT 2000
shutdown:  THE SYSTEM IS BEING SHUT DOWN NOW

User(s) currently logged in:
 root


Stopping some active subsystems...

0513-044 The dpid2 Subsystem was requested to stop.
0513-044 The hostmibd Subsystem was requested to stop.
0513-044 The qdaemon Subsystem was requested to stop.
0513-044 The writesrv Subsystem was requested to stop.
0513-044 The wsmrefserver Subsystem was requested to stop.

Unmounting the file systems...

/usr/local unmounted successfully.
 /proc unmounted successfully.
 /home unmounted successfully.
 /tmp unmounted successfully.

Bringing down network interfaces:

detached en0 from the network interface list
detached en1 from the network interface list
detached et0 from the network interface list
detached lo0 from the network interface list
detached tr0 from the network interface list
```

The output of consecutive shutdowns (if the -l flag is used) are appended to the /etc/shutdown.log file. Therefore, this information is available even if there are problems with booting the system and the machine had to be shut down several times. The log file continues to grow until the system administrator intervenes.

4.6 System dump enhancements

AIX 5L provides the following enhancements in the area of system dumps:

- A new command, `dumpcheck`, checks if the dump device and the copy directory for the dump are large enough to actually accept a system dump.
- The creation of a core file for a process without terminating the process.
- Minor enhancements to the `snap` command.

4.6.1 The dumpcheck command

The new dumpcheck command has the following syntax:

```
/usr/lib/ras/dumpcheck [ [ -l ] [ -p ] [ -t Time ] [ -P ] ] | [ -r ]
```

By default, dumpcheck is started by a crontab entry each afternoon at 3:00 PM. The output of the command will be logged in the system error log. With the -p flag, you can request a dumpcheck at any time and get the result printed to stdout. The output would look similar to the following example:

```
# /usr/lib/ras/dumpcheck -p
There is not enough free space in the file system containing the copy directory
to accommodate the dump.
File system name         /var/adm/ras
Current free space in kb         14360
Current estimated dump size in kb        25600
```

The -l flag logs the command output into the system error log and is the default parameter if no other parameter is specified. With the -t flag, you can determine, with a time value in crontab format enclosed in single or double quotation marks, at what time this check will be run by the cron facility. The -P flag updates the crontab entry to reflect whatever parameters are specified with it. The cron facility mails the standard output of a command to the user who runs this command (in this case root). If you use the -p flag in the crontab entry, root will be sent a mail with the standard output of the dumpcheck command.

Note

Currently, the command redirection (>/dev/null 2>&1) will not automatically be removed, which prevents the cron facility from sending the mail. You have to remove this redirection manually.

The -r flag removes the corresponding crontab entry. This flag cannot be used together with any other flag.

4.6.2 The coredump() system call

An application can now create a core file by using the new coredump() system call. This call takes, as a single parameter, a pointer to a coredumpinfop structure that sets the path and file name for the core file to be generated.

To use coredump(), you must compile your source with the -bM:UR options. The -b flag is for ld, M: is to specify a Module type, and UR indicates to save the user registers on system calls.

4.6.3 Snap enhancements

The snap command in AIX 5L uses the pax command instead of the tar command to create the snap archive file. This is necessary to manage the ever increasing sizes of the dump files, as file sizes larger than 2 GB are only supported by the pax command. The snap utility also links the dump file to the directory structure it creates instead of copying it into the structure, which wastes disk space. The data needed most for analyzing the situation (that is, what caused the dump) is written out first, so that it has a good chance to be part of the archive file created by snap even if the dump is only partially successful. For other enhancements to pax, see Section 2.11, “Enhancements for pax” on page 19.

4.7 System hang detection

A new feature called *system hang detection* provides a SMIT-configurable mechanism to detect system hands and initiate the configured action. It relies on a new daemon named shdaemon and a corresponding configuration program named shconf.

In the case where applications adjust their process or thread priorities using system calls, there is the potential problem that their priorities will become so high that regular system shells are not scheduled. In this situation, it is difficult to distinguish a system that really hangs (it is not doing any meaningful work anymore) from a system that is so busy that none of the lower priority tasks, such as user shells, have a chance to run.

The new system hang detection feature uses a shdaemon entry in the /etc/inittab file with an action field that is set to off by default. Using the shconf command or SMIT (fastpath shd), you can enable this daemon and configure the actions it takes when certain conditions are met. The following flags are allowed with the shconf command.

```
shconf [ -d ] [ -R |-D [ -O] | -E [ -O ] | [ [ -a Attribute ] ...] -l prio
[ -H ]
```

The only existing detection name is prio, which means that the system hang daemon will always compare the priorities of all running processes to a set threshold, and will take one of the five supported actions, each of a different priority, when the entire system fails to run a process below the specified priority any time in the time-out period.

The -d flag displays the current status of the shdaemon. The -R flag restores the system default values. With the -D and -E flags, you can display either the default or the effective values of the configuration parameters. The -H flag adds an optional header to this output. You can request a more concise output by using the -O flag together with either the -D or -E flags (in this case, the -H flag is not allowed). It displays two lines: one with the colon-separated names, and one with the colon-separated values of the configuration parameters. With the -a flag and a name/value pair, you can change the parameter values.

After a new default system installation that has effective values that are identical to the default values occurs, the output of the `shconf` command appears as follows:

```
# shconf -d
sh_pp=disable
# shconf -E -l prio -H
attribute  value        description

sh_pp      disable      Enable Process Priority Problem
pp_errlog  disable      Log Error in the Error Logging
pp_eto     2            Detection Time-out
pp_eprio   60           Process Priority
pp_warning disable      Display a warning message on a console
pp_wto     2            Detection Time-out
pp_wprio   60           Process Priority
pp_wterm   /dev/console Terminal Device
pp_login   enable       Launch a recovering login on a console
pp_lto     2            Detection Time-out
pp_lprio   56           Process Priority
pp_lterm   /dev/tty0    Terminal Device
pp_cmd     disable      Launch a command
pp_cto     2            Detection Time-out
pp_cprio   60           Process Priority
pp_cpath   /            Script
pp_reboot  disable      Automatically REBOOT system
pp_rto     5            Detection Time-out
pp_rprio   39           Process Priority
```

The ss_pp parameter determines the availability of the system hang detection feature. Enabling it with the default configuration may generate the following error:

```
# shconf -l prio -a sh_pp=enable
shconf:Enable to configure the emergency login.
shconf: Configuration method error.
```

You have to disable to pp_login action, enable the system hang detection, and then configure the desired actions. The output of these commands appear as follows:

```
# shconf -l prio -a sh_pp=disable
shconf: Priority Problem Conf has changed.
# shconf -l prio -a pp_login=disable
shconf: Priority Problem Conf has changed.
# shconf -l prio -a sh_pp=enable
shconf: Priority Problem Conf has changed.
shconf: WARNING: Priority Problem Detection is enabled with all actions disabled.
```

The last command shown in the previous output toggles the action field of the shdaemon entry in /etc/inittab to respawn and starts the /usr/sbin/shdaemon program. After enabling (for example, the errlog action), the priority of the shdaemon process is 0, the highest possible value. This is shown in the following example:

```
# ps lwx 19580
     F S UID   PID  PPID   C PRI NI ADDR  SZ  RSS   WCHAN    TTY  TIME CMD
240001 A   0 19580     1   0  60 20 fa5e 192  236   EVENT      -  0:00 /usr/sbin/shdaemon
# shconf -l prio -a pp_errlog=enable
shconf: Priority Problem Conf has changed.
# ps lwx 19584
     F S UID   PID  PPID   C PRI NI ADDR  SZ  RSS   WCHAN    TTY  TIME CMD
240001 A   0 19584     1   0   0 20 fa5e 33000 33044   EVENT      -  0:00
/usr/sbin/shdaemon
```

This action makes sure that the shdaemon is always scheduled and can evaluate the current machine status and take the configured actions when appropriate. The available actions include the following:

errlog Generates a entry in the error log.

warning Displays a warning message on a console; the default is /dev/console.

login Enables a login shell with priority 0 on a serial terminal; the default is /dev/tty0.

cmd Starts a command with priority 0.

reboot Automatically reboots the machine.

4.8 Performance Analysis Tools enhancements

The Performance Analysis Tools in AIX 5L adds the following new tools:

- truss allows the tracing of all system calls made and signals received by a command or an existing process. Section 4.8.1, “Process system call tracing with truss” on page 68 describes truss in more detail.
- alstat is a new tool which reports alignment exception statistics. This tool can be used to detect performance degradations caused by misalignment data or code (POWER only).

For AIX 5L on POWER and Itanium-based systems, the following tools and commands are available, tprof, gennames, truss, iostat, vmstat, sar, prof, and gprof.

The following analysis tools are available on POWER only: emstat, alstat, filemon, fileplace, netpmon, pprof, rmss, svmon, and topas.

The following tools have been withdrawn in AIX 5L: bf (bigfoot), bfrpt, stem, and syscalls. Consult the man pages for svmon and truss to locate similar function.

4.8.1 Process system call tracing with truss

AIX 5L now supports the truss command, which allows you to trace system calls executed by a process as well as record the received signals and the occurrence of machine faults.

The application to trace is either specified on the command line of the truss command or truss can be attached to one or more already running processes by using the -p flag with a list of process IDs. The complete list of flags supported by the truss command is:

```
# truss
Usage:  [ -f ] [ -c ] [ -a ] [ -e ] [ -i ] [ - [ tx ] [ ! ] syscall [
,syscall ] ] [ -s [ ! ] signal [ ,signal ] ] [ -m [ ! ] fault [ ,fault ] ]
[-[ rw ] [ ! ] fd [ ,fd ] ] [ -o outfile ] { command | -p pid [. . .] }
```

If the -o flag that redirects the output of truss to a file is not used, the truss output goes to standard out and can be mixed with the output of the command truss is tracing. Before describing the other flags, the following lines show a simple example of running the date command under truss:

```
# truss -e -o truss.out date
Thu Sep 14 15:28:20 CDT 2000
# cat truss.out
execve("/usr/bin/date", 0x2FF22C44, 0x2FF22C4C)  argc: 1
 envp: _=/usr/bin/truss LANG=en_US LOGIN=root
  NLSPATH=/usr/lib/nls/msg/%L/%N:/usr/lib/nls/msg/%L/%N.cat
  PATH=/usr/bin:/etc:/usr/sbin:/usr/ucb:/usr/bin/X11:/sbin
  LC__FASTMSG=true WINDOWID=4194317
  CGI_DIRECTORY=/var/docsearch/cgi-bin LOGNAME=root
  MAIL=/usr/spool/mail/root LOCPATH=/usr/lib/nls/loc USER=root
  DOCUMENT_SERVER_MACHINE_NAME=localhost AUTHSTATE=compat
  DISPLAY=9.3.240.103:0.0 SHELL=/usr/bin/ksh ODMDIR=/etc/objrepos
  DOCUMENT_SERVER_PORT=49213 HOME=/ TERM=xterm
  MAILMSG=[YOU HAVE NEW MAIL] ITECONFIGSRV=/etc/IMNSearch PWD=/
  DOCUMENT_DIRECTORY=/usr/docsearch/html TZ=CST6CDT
  ITECONFIGCL=/etc/IMNSearch/clients ITE_DOC_SEARCH_INSTANCE=search
  A__z=! LOGNAME
sbrk(0x00000000)                                = 0x20001C50
brk(0x20011C50)                                 = 0
getuidx(4)                                      = 0x00000000
getuidx(2)                                      = 0x00000000
getuidx(1)                                      = 0x00000000
```

```
getgidx(4)                                      = 0
getgidx(2)                                      = 0
getgidx(1)                                      = 0
__loadx(0x01000080, 0x2FF1E8E0, 0x00003E80, 0x2FF22870, 0x00000000,
0x00000000, 0x80000000, 0x7F7F7F7F) = 0xD0072130
__loadx(0x01000180, 0x2FF1E8D0, 0x00003E80, 0xF0133E10, 0xF0133D40,
0x00000000, 0xFFFFFFFD, 0xD0074388) = 0xF02885B8
__loadx(0x07080000, 0xF0133DE0, 0xFFFFFFFF, 0xF02885B8, 0x00000000,
0x6000C018, 0x600078AF, 0x00000000) = 0xF02892BC
__loadx(0x07080000, 0xF0133D20, 0xFFFFFFFF, 0xF02885B8, 0x00000000,
0x6000C018, 0x600078AF, 0x00000000) = 0xF02892C8
__loadx(0x07080000, 0xF0133DF0, 0xFFFFFFFF, 0xF02885B8, 0x00000000,
0x6000C018, 0x600078AF, 0x00000000) = 0xF02892F8
__loadx(0x07080000, 0xF0133D30, 0xFFFFFFFF, 0xF02885B8, 0x00000000,
0x6000C018, 0x600078AF, 0x00000000) = 0xF0289304
__loadx(0x07080000, 0xF0133DB0, 0xFFFFFFFF, 0xF02885B8, 0x00000000,
0x6000C018, 0x600078AF, 0x00000000) = 0xF02892D4
__loadx(0x07080000, 0xF0133D60, 0xFFFFFFFF, 0xF02885B8, 0x00000000,
0x6000C018, 0x600078AF, 0x00000000) = 0xF02892EC
__loadx(0x07080000, 0xF0133DC0, 0xFFFFFFFF, 0xF02885B8, 0x00000000,
0x6000C018, 0x600078AF, 0x00000000) = 0xF0289310
__loadx(0x07080000, 0xF0133DD0, 0xFFFFFFFF, 0xF02885B8, 0x00000000,
0x6000C018, 0x600078AF, 0x00000000) = 0xF0289340
__loadx(0x07080000, 0xF0133D50, 0xFFFFFFFF, 0xF02885B8, 0x00000000,
0x6000C018, 0x600078AF, 0x00000000) = 0xF0289328
__loadx(0x07080000, 0xF0133D70, 0xFFFFFFFF, 0xF02885B8, 0x00000000,
0x6000C018, 0x600078AF, 0x00000000) = 0xF02892F8
__loadx(0x07080000, 0xF0133D30, 0xFFFFFFFF, 0xF02885B8, 0x00000000,
0x6000C018, 0x600078AF, 0x00000000) = 0xF0289304
__loadx(0x07080000, 0xF0133DB0, 0xFFFFFFFF, 0xF02885B8, 0x00000000,
0x6000C018, 0x600078AF, 0x00000000) = 0xF02892D4
__loadx(0x07080000, 0xF0133D60, 0xFFFFFFFF, 0xF02885B8, 0x00000000,
0x6000C018, 0x600078AF, 0x00000000) = 0xF02892EC
__loadx(0x07080000, 0xF0133DC0, 0xFFFFFFFF, 0xF02885B8, 0x00000000,
0x6000C018, 0x600078AF, 0x00000000) = 0xF0289310
__loadx(0x07080000, 0xF0133DD0, 0xFFFFFFFF, 0xF02885B8, 0x00000000,
0x6000C018, 0x600078AF, 0x00000000) = 0xF0289340
__loadx(0x07080000, 0xF0133D50, 0xFFFFFFFF, 0xF02885B8, 0x00000000,
0x6000C018, 0x600078AF, 0x00000000) = 0xF0289328
__loadx(0x07080000, 0xF0133D70, 0xFFFFFFFF, 0xF02885B8, 0x00000000,
0x6000C018, 0x600078AF, 0x00000000) = 0xF028934C
access("/usr/lib/nls/msg/en_US/date.cat", 0)    = 0
_getpid()                                       = 19528
kioctl(1, 22528, 0x00000000, 0x00000000)        = 0
kwrite(1, 0xF018ABD8, 29)                       = 29
kfcntl(1, F_GETFL, 0xF0170918)                  = 2
kfcntl(2, F_GETFL, 0xF0170918)                  = 2
```

```
_exit(0)
```

The -e flag is responsible for the display of the environment content in the truss output file. By default, truss does not trace forked processes; the -f flag will force truss to go into forked processes. Interruptible sleeping system calls are displayed once on completion if the -i flag is used. The -c flag generates a summary file instead of the detailed report shown previously. The -c flag also gives a count for how often a specific system call was executed and the overall time spent in total in it.

The other flags allow the inclusion (or exclusion, if the ! (exclamation point) is used) by name of specific system calls, signals, machine faults, or the data read from or written to specific file descriptors. By default, truss displays symbolic constants from the appropriate system header files as the arguments of the system calls; this can be forced to always display hexadecimal values by using the -x flag. These four flags accept the symbol all to include all possible system calls, signals, and so forth. The return value of the system call is shown on the right hand side of the equal sign.

For this simple date command (shown in the previous output), the truss output file is already about 10 KB. You need to reduce the number of system calls you are tracing, or attach truss to a running process only for a limited amount of time, to keep the size of the truss output file within a manageable range.

4.8.2 Emulation and alignment detection

A new tool was added in the *perfagent.tools* fileset; in addition to the existing emstat command, alstat will count alignment interrupts while emstat will display emulation statistics.

Both commands can use the -v flag, which will display the statistics per CPU in SMP systems.

This feature is only available on the POWER platform.

4.8.3 Performance monitor API

A new set of APIs is available to provide access to Performance Monitor data on selected processor types, namely 604, 604e, POWER3, POWER3-II, RS64-II, and RS64-III. Other processors of the POWER platform not listed are not supported by this API.

This feature is only available on the POWER platform.

Refer to “Performance Monitor API Programming Concepts” section in Chapter 10 “Programming on Multiprocessor Systems” of the Programming Guides publication in the Online Documentation Library for a complete list of API calls, as well sample subroutines and load and unload Performance Monitor API kernel extension commands.

4.8.4 Enhancements to vmstat

The `vmstat` utility has two new flags in AIX 5L; these new flags add new controls and improve monitoring.

The -I flag outputs a report with the new columns *fi* and *fo;* these columns indicate the number of file pages in (*fi*) and out (*fo*). In this report, the *re* and *cy* columns are not displayed. A new *p* column displays the number of threads waiting for a physical I/O operation.

```
# vmstat -I 1 3
  kthr      memory             page              faults        cpu
-------- ----------- ------------------------ ------------ -----------
 r  b  p   avm   fre  fi  fo  pi  po  fr  sr   in   sy  cs us sy id wa
 0  0  0 46391   228   0   0   0   0   0    2 108  156  20  1  0 99  0
 0  1  0 46391   226   0   0   0   0   0    0 432 8080  53  1  1 98  0
 0  1  0 46391   226   0   0   0   0   0    0 424   91  50  0  0 99  0
```

The -t flag shows a time stamp at the end of each line.

```
# vmstat -t 1 3
kthr     memory             page              faults        cpu        time
----- ----------- ------------------------ ------------ ----------- --------
 r  b   avm   fre  re  pi  po  fr   sr  cy  in   sy  cs us sy id wa hr mi se
 0  0 46905  5752   0   0   0   0    2   0 108  156  20  1  0 99  0 11:46:28
 0  1 46905  5749   0   0   0   0    0   0 429 7264  72  1  1 98  0 11:46:29
 0  1 46905  5749   0   0   0   0    0   0 434  165  60  0  0 99  0 11:46:30
```

4.8.5 Enhancements to iostat

The `iostat` command is enhanced with new parameters that will provide a better presentation of the generated reports.

The -s flag adds a new line to the header of each statistics data that reports the sum of all activity on the system.

```
# iostat -s 1 3
System: server1.itsc.austin.ibm.com
                           Kbps      tps    Kb_read   Kb_wrtn
                         9405.3     2351.3      28216         0

Disks:        % tm_act     Kbps      tps      Kb_read   Kb_wrtn
hdisk0          46.7     4693.3     1173.3      14080         0
hdisk1          24.0     2356.0     588.7       7068          0
hdisk2           0.0       0.0       0.0          0           0
hdisk3          24.3     2356.0     589.3       7068          0
hdisk4           0.0       0.0       0.0          0           0
cd0              0.0       0.0       0.0          0           0
```

The -a flag produces an output similar to the s- flag output, with the difference that it provides an adapter basis sum of activities. After displaying the adapter activity, it provides a per-disk basis set of statistics.

```
# iostat -a 1 3
tty:      tin        tout   avg-cpu:  % user    % sys     % idle    % iowait
          0.0       923.7              13.2     41.6       30.9      14.2

Adapter:                 Kbps      tps      Kb_read   Kb_wrtn
scsi0                  7030.4     1757.6      7048         0

Disks:        % tm_act     Kbps      tps      Kb_read   Kb_wrtn
hdisk0          43.9     4684.3     1171.1      4696         0
hdisk1          24.9     2346.1     586.5       2352         0
hdisk2           0.0        0.0       0.0          0         0
cd0              0.0        0.0       0.0          0         0

Adapter:                 Kbps      tps      Kb_read   Kb_wrtn
scsi1                  2346.1     585.5       2352         0

Disks:        % tm_act     Kbps      tps      Kb_read   Kb_wrtn
hdisk3          19.0     2346.1     585.5       2352         0
hdisk4           0.0        0.0       0.0          0         0
```

4.8.6 Enhancements to netpmon and filemon

These utilities receive the following enhancement introduced with AIX 5L:

- New offline support that allows you to generate netpmon reports with a normal trace report file and a gennames output for improved use and scalability on target systems.

At the time of writing, this feature is only available on the POWER platform.

To use this the new support, you must generate a normal trace output (for example, through smit trace and then start trace), and then generate an unformatted trace file through the output trace file, as shown in the following example:

```
# trcrpt -r /var/adm/ras/trcfile > /tmp/newtrcfile
```

Immediately following the collection of the trace file, you should also run the gennames command save its output:

```
# gennames > /tmp/gennames.out
```

When both files are correctly set, you can generate your offline report using the -i and -n flags, as shown in the following netpmon example:

```
# netpmon -i /tmp/newtrcfile -n /tmp/gennames.out
```

4.8.7 Enhancements to svmon

The svmon command has been enhanced to display information about different superclasses and subclasses introduced with the Workload Manager in AIX 5L update.

This feature is only available on the POWER platform.

Four new flags, discussed in the following sections, can be used in order to make use of this new function.

4.8.7.1 The -W flag

The -W flag is used to collect statistics for either an entire superclass or only a specific subclass. The following example is an output generated for a superclass:

```
# svmon -W sv
Superclass                           Inuse      Pin     Pgsp  Virtual
sv                                    2039        8        0      231

    Vsid     Esid Type Description                   Inuse   Pin Pgsp Virtual
    5f4b        - pers /dev/hd2:43509                 1082     0    -     -
    48e8        - pers /dev/hd2:47134                  182     0    -     -
    e099        - work                                  69     0    0    70
    48ac        - work                                  61     0    0    62
```

To display subclass information, you must use class.subclass for syntax.

```
# svmon -W sv.sv_sub
Class                                Inuse      Pin     Pgsp  Virtual
sv.sv_sub                             1929        6        0      124

    Vsid     Esid Type Description                   Inuse   Pin Pgsp Virtual
    5f4b        - pers /dev/hd2:43509                 1082     0    -     -
    48e8        - pers /dev/hd2:47134                  182     0    -     -
    c8bc        - work                                  74     2    0    73
    2f45        - pers /dev/hd2:47128                   54     0    -     -
```

4.8.7.2 The -e flag

The -e flag reports the statistics for the subclass of a superclass. It only applies to superclasses or tiers. The -e flag is only allowed with -T and -W. A sample output is shown in the following example:

```
Superclass                           Inuse      Pin     Pgsp  Virtual
sv                                    1867        4        0       74

================================================================================
Class                                Inuse      Pin     Pgsp  Virtual
sv.sv_sub                             1769        0        0        0

    Vsid     Esid Type Description                   Inuse   Pin Pgsp Virtual
    5f4b        - pers /dev/hd2:43509                 1082     0    -     -
    48e8        - pers /dev/hd2:47134                  182     0    -     -
    2f45        - pers /dev/hd2:47128                   54     0    -     -
================================================================================
Class                                Inuse      Pin     Pgsp  Virtual
```

```
sv.Default                              98        4        0       74

    Vsid     Esid Type Description                   Inuse   Pin Pgsp Virtual
    28c0        - work                                  23     0    0    15
    710b        - work                                  21     0    0    13
    e0f9        - work                                  21     0    0    15
    3043        - work                                  14     2    0    14
    3103        - work                                  12     2    0    12
    6109        - work                                   7     0    0     5

===============================================================================
Class                                Inuse      Pin     Pgsp  Virtual
sv.Shared                                0        0        0        0
```

4.8.7.3 The -T flag

The -T flag reports the statistics of all the classes in a tier. If a parameter is passed to the -T, then only the classes belonging to the tier will be analyzed. A list of tiers can be provided. When no parameter is specified, all the defined tiers of the class will be analyzed. Examples of flag interaction and command response follows.

- -T flag with no parameter.

```
# svmon -T

===============================================================================
Tier                                 Inuse      Pin     Pgsp  Virtual
   0                                 87112     6650    11462    29167

===============================================================================
Superclass                           Inuse      Pin     Pgsp  Virtual
System                               72109     6616     9197    25124
Shared                                6535        0      878     2530
Unclassified                          5950       10        5       20
Default                               2518       24     1382     1493
Unmanaged                                0        0        0        0
random                                   0        0        0        0
sequential                               0        0        0        0

===============================================================================
Tier                                 Inuse      Pin     Pgsp  Virtual
   1                                  1853        2        0       74

===============================================================================
Superclass                           Inuse      Pin     Pgsp  Virtual
sv                                    1853        2        0       74
```

- -T flag with a specific tier value.

```
# svmon -T 1

===============================================================================
Tier                                 Inuse      Pin     Pgsp  Virtual
   1                                  1902        4        0      130

===============================================================================
Superclass                           Inuse      Pin     Pgsp  Virtual
sv                                    1902        4        0      130
```

- -T flag with the -a flag indicating a specific superclass. All the subclasses of the indicated superclass in the tier *tiernumber* will be reported.

```
# svmon -a sv -T 1

===============================================================================
Tier Superclass                      Inuse      Pin     Pgsp  Virtual
   1 sv                               2037       10        0      245

===============================================================================
Class                                Inuse      Pin     Pgsp  Virtual
sv.sv_sub                             1769        0        0        0
sv.Default                             268       10        0      245
```

- -T flag with the -x flag will report all the superclasses segment statistics of the specific tier.

```
# svmon -T 0 -x
Tier                                 Inuse      Pin     Pgsp  Virtual
   0                                 88106     6659    11462    30028

===============================================================================
Superclass                           Inuse      Pin     Pgsp  Virtual
System                               73095     6625     9197    25982

    Vsid     Esid Type Description                   Inuse   Pin Pgsp Virtual
    db99        - pers large file /dev/lv04:23       27702     0    -     -
    8010        - work misc kernel tables             3287     0 1210  3289
       0        - work kernel seg                     3134  1635 1919  3379
    8811        - work kernel pinned heap             3087  1222 1226  3187
    8af0        - pers /dev/hd2:112665                2316     0    -     -
```

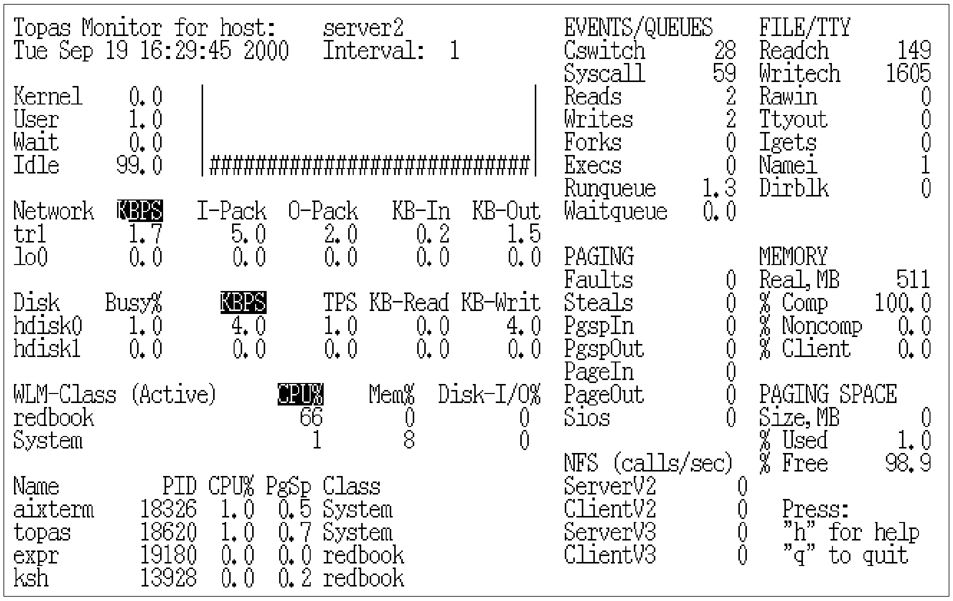
As shown in the preceding examples, you can mix different flags to obtain different outputs. Refer to the `svmon` command man pages to check for other combinations.

4.8.8 Enhancements to topas

Topas is a performance monitor tool that was introduced in AIX Version 4.3.3. In AIX 5L, it has several new enhancements, including Workload Manager support, an improved set of CPU usage panels, several new column sort options, NFS statistics, lock statistics, and per disk or adapter breakdown of network and disk usage.

This feature is only available on the POWER platform.

Figure 17 provides a sample `topas` main screen. This section is too brief to demonstrate all the features. It is recommended that the `topas` tool is given a complete exploration through hands-on use.



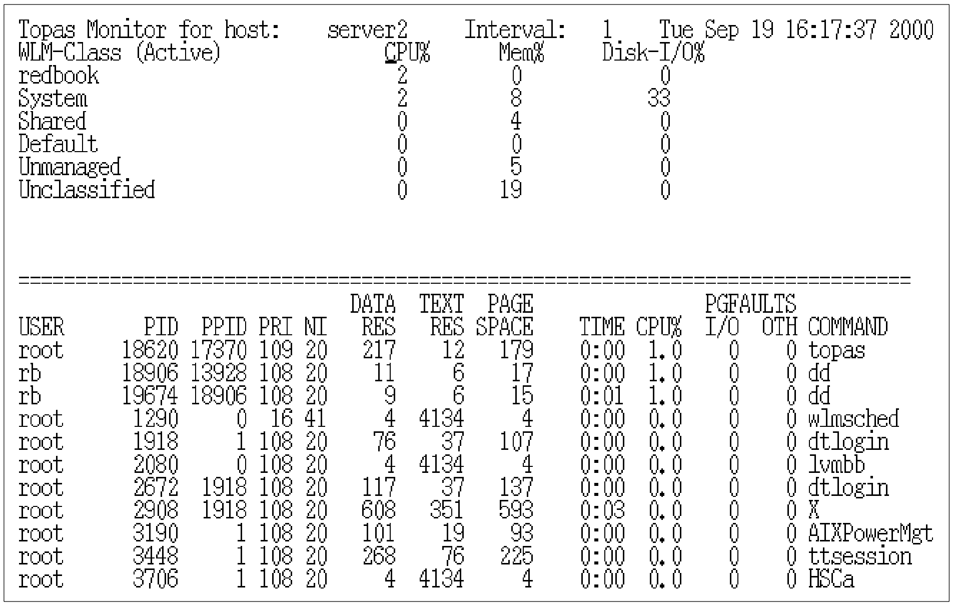
```
Topas Monitor for host:    server2                EVENTS/QUEUES    FILE/TTY
Tue Sep 19 16:29:45 2000   Interval:  1           Cswitch      28  Readch      149
                                                  Syscall      59  Writech    1605
Kernel    0.0  |                             |    Reads         2  Rawin         0
User      1.0  |                             |    Writes        2  Ttyout        0
Wait      0.0  |                             |    Forks         0  Igets         0
Idle     99.0  |#############################|    Execs         0  Namei         1
                                                  Runqueue    1.3  Dirblk        0
Network  KBPS    I-Pack  O-Pack   KB-In  KB-Out   Waitqueue   0.0
tr1       1.7      5.0     2.0     0.2     1.5
lo0       0.0      0.0     0.0     0.0     0.0    PAGING           MEMORY
                                                  Faults        0  Real,MB     511
Disk    Busy%     KBPS     TPS KB-Read KB-Writ    Steals        0  % Comp    100.0
hdisk0    1.0      4.0     1.0     0.0     4.0    PgspIn        0  % Noncomp   0.0
hdisk1    0.0      0.0     0.0     0.0     0.0    PgspOut       0  % Client    0.0
                                                  PageIn        0
WLM-Class (Active)        CPU%    Mem%  Disk-I/O%  PageOut       0  PAGING SPACE
redbook                     66       0          0  Sios          0  Size,MB       0
System                       1       8          0                   % Used      1.0
                                                  NFS (calls/sec)  % Free     98.9
Name            PID  CPU% PgSp Class              ServerV2       0
aixterm       18326  1.0  0.5 System              ClientV2       0    Press:
topas         18620  1.0  0.7 System              ServerV3       0    "h" for help
expr          19180  0.0  0.0 redbook             ClientV3       0    "q" to quit
ksh           13928  0.0  0.2 redbook
```

Figure 17. Topas main screen

4.8.8.1 Workload manager support

Topas displays the CPU, disk, and block I/O usage for each class. By default, it will display the top two classes. Two new commands were added to `topas` to change the Workload Manager monitoring. The `w` (lower case) command will toggle the top two classes on or off, and, the `W` (upper case) command will switch to a full Workload Manager classes monitoring screen.

The example shown in Figure 17 on page 76 has the top two classes enabled, while Figure 18 shows the entire set of classes being monitored by `topas`.

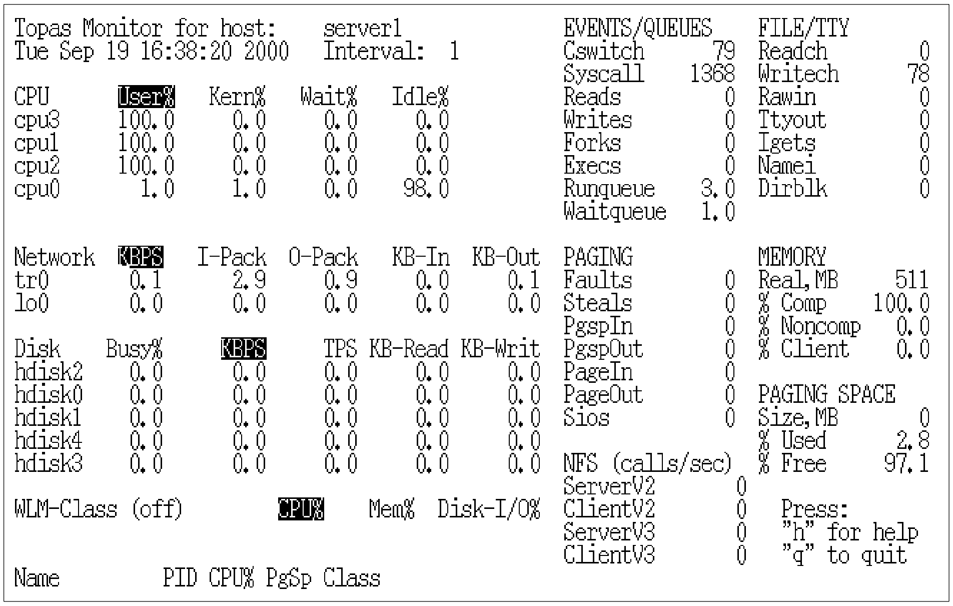


```
Topas Monitor for host:    server2     Interval:   1    Tue Sep 19 16:17:37 2000
WLM-Class (Active)              CPU%       Mem%     Disk-I/O%
redbook                          2          0           0
System                           2          8          33
Shared                           0          4           0
Default                          0          0           0
Unmanaged                        0          5           0
Unclassified                     0         19           0



=============================================================================
                              DATA  TEXT  PAGE                PGFAULTS
USER       PID  PPID PRI NI    RES   RES SPACE    TIME CPU%  I/O  OTH COMMAND
root     18620 17370 109 20    217    12   179    0:00  1.0    0    0 topas
rb       18906 13928 108 20     11     6    17    0:00  1.0    0    0 dd
rb       19674 18906 108 20      9     6    15    0:01  1.0    0    0 dd
root      1290     0  16 41      4  4134     4    0:00  0.0    0    0 wlmsched
root      1918     1 108 20     76    37   107    0:00  0.0    0    0 dtlogin
root      2080     0 108 20      4  4134     4    0:00  0.0    0    0 lvmbb
root      2672  1918 108 20    117    37   137    0:00  0.0    0    0 dtlogin
root      2908  1918 108 20    608   351   593    0:03  0.0    0    0 X
root      3190     1 108 20    101    19    93    0:00  0.0    0    0 AIXPowerMgt
root      3448     1 108 20    268    76   225    0:00  0.0    0    0 ttsession
root      3706     1 108 20      4  4134     4    0:00  0.0    0    0 HSCa
```

Figure 18. Workload Manager screen using W subcommand

4.8.8.2 CPU display

By default, `topas` will display cumulative CPU usage as in previous releases. However, the `c` (lower case) command can toggle to a per CPU usage view on SMP systems. The `c` command also toggles CPU monitoring off (see Figure 19 on page 78).



```
Topas Monitor for host:    server1                EVENTS/QUEUES    FILE/TTY
Tue Sep 19 16:38:20 2000   Interval:  1           Cswitch      79  Readch        0
                                                  Syscall    1368  Writech      78
CPU      User%    Kern%   Wait%   Idle%           Reads         0  Rawin         0
cpu3     100.0      0.0     0.0     0.0           Writes        0  Ttyout        0
cpu1     100.0      0.0     0.0     0.0           Forks         0  Igets         0
cpu2     100.0      0.0     0.0     0.0           Execs         0  Namei         0
cpu0       1.0      1.0     0.0    98.0           Runqueue    3.0  Dirblk        0
                                                  Waitqueue   1.0

Network  KBPS   I-Pack  O-Pack   KB-In  KB-Out    PAGING           MEMORY
tr0       0.1      2.9     0.9     0.0     0.1    Faults        0  Real,MB     511
lo0       0.0      0.0     0.0     0.0     0.0    Steals        0  % Comp    100.0
                                                  PgspIn        0  % Noncomp   0.0
Disk    Busy%      KBPS     TPS KB-Read KB-Writ   PgspOut       0  % Client    0.0
hdisk2    0.0       0.0     0.0     0.0     0.0   PageIn        0
hdisk0    0.0       0.0     0.0     0.0     0.0   PageOut       0  PAGING SPACE
hdisk1    0.0       0.0     0.0     0.0     0.0   Sios          0  Size,MB       0
hdisk4    0.0       0.0     0.0     0.0     0.0                    % Used      2.8
hdisk3    0.0       0.0     0.0     0.0     0.0   NFS (calls/sec)  % Free     97.1
                                                  ServerV2       0
WLM-Class (off)          CPU%    Mem%  Disk-I/O%  ClientV2       0   Press:
                                                  ServerV3       0   "h" for help
                                                  ClientV3       0   "q" to quit
Name          PID CPU% PgSp Class
```

Figure 19. Topas with per CPU usage enabled

4.9 FDPR code duplication optimization

FDPR is a tool, first introduced in AIX 3.2, that optimizes binaries generated from xl compilers. It contains two major components: `aopt`, which is used for instrumenting and reordering of AIX XCOFF executables, and `fdpr`, which is a more user-friendly interface to the `aopt` command.

This feature is only available on the POWER platform.

A new improvement introduced with AIX 5L is the Code Duplication optimization. Code Duplication optimization eliminates the need to invoke the store and restore functions of small, but frequently used functions in the *Link Register*, which were not suitable for optimization, by creating a new copy of the callee function and redirecting the calling instructions to its duplicated copy.

4.10 Workload Manager

AIX Workload Manager (WLM) is an operating system feature introduced in AIX Version 4.3.3. It is a part of the operating system kernel at no additional charge.

In AIX 5L, WLM provides additional controls that fill out many of the capabilities of Workload Manager.

Keep in mind that the discussion of AIX Version 4.3.3 and previous POWER platform editions in this section is only for historical reference. AIX 5L for Itanium-based systems benefit from all the enhancements made in previous POWER platform releases as the cumulative function was ported.

4.10.1 Overview

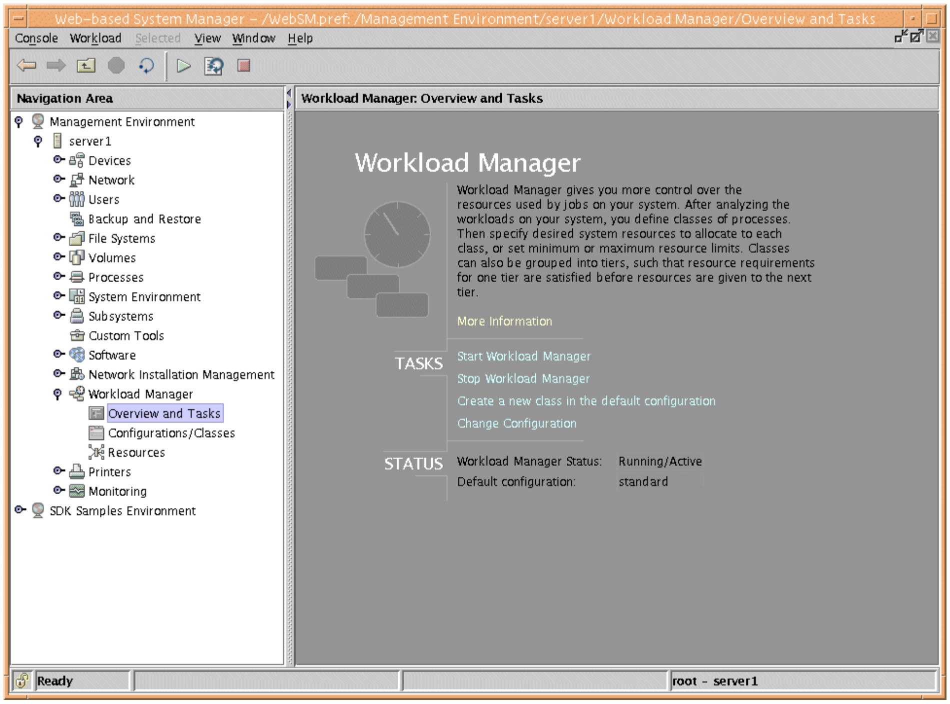
WLM is designed to give the system administrator greater control over how the scheduler and Virtual Memory Manager (VMM) allocate CPU and physical memory resources to processes. It can be used to prevent different jobs from interfering with each other and to allocate resources based on the requirements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

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

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

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

WLM configuration is performed through the preferred interface, the Web-based System Manager (Figure 20 on page 80), through a text editor and AIX commands, or through the AIX administration tool SMIT.



Workload Manager gives you more control over the resources used by jobs on your system. After analyzing the workloads on your system, you define classes of processes. Then specify desired system resources to allocate to each class, or set minimum or maximum resource limits. Classes can also be grouped into tiers, such that resource requirements for one tier are satisfied before resources are given to the next tier.

Workload Manager Status: Running/Active
Default configuration: standard

Figure 20. Web-based System Manager Overview and Tasks dialog

4.10.2 Workload Manager enhancements history

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

4.10.2.1 AIX Version 4.3.3

In AIX Version 4.3.3, WLM was able to allocate CPU and physical memory resources to classes of jobs and allowed processes to be assigned to classes based on user, group, or application.

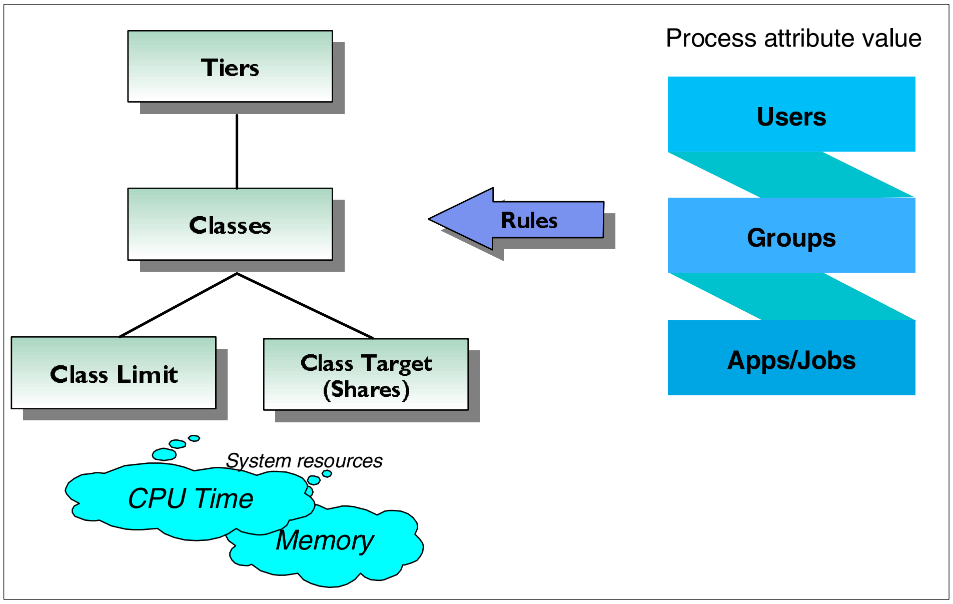


Figure 21. Basic Workload Manager elements in AIX Version 4.3

4.10.2.2 AIX Version 4.3.3 with Maintenance Level 2

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

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

4.10.2.3 AIX 5L

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

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

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

Note

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

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

4.10.3 Concepts and architectural enhancements

The following section outlines the concepts provided with WLM on AIX 5L.

4.10.3.1 Classes

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

Hierarchy of classes

WLM allows system administrators to set up a hierarchy of classes with two levels by defining superclasses and subclasses. The main difference between superclasses and subclasses is the resource control (shares and limits):

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

The system administrator can delegate the administration of the subclasses of each superclass to a *superclass administrator*, thus having the option of allocating a portion of the system resources to each superclass and then letting superclass administrators distribute the allocated resources among the users and applications they manage.

WLM supports 32 superclasses (27 user defined plus five predefined). In turn, each superclass can have 12 subclasses (10 user defined and two predefined, as shown in Figure 22 on page 84). Depending on the needs of the organization, a system administrator can decide to use only superclasses or both superclasses and subclasses. An administrator can also use subclasses only for some of the superclasses.

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

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

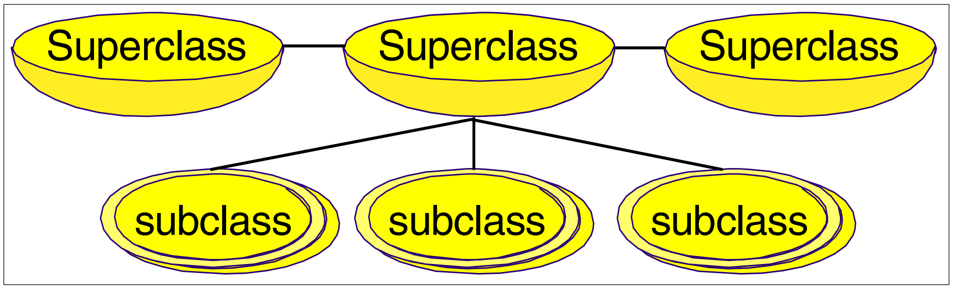


Figure 22. Hierarchy of Classes

Superclasses

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

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

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

that superclass. The pages are placed in the Shared superclass only when a process from a different superclass accesses the shared memory region or file. This superclass can have only physical memory shares and limits applied to it. It cannot have shares or limits for the other resource types, subclasses, or assignment rules specified.

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

Subclasses

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

Up to 10 subclasses can be defined by the system administrator or by the superclass administrator for each superclass. In addition, two special subclasses, Default and Shared, are always defined in each superclass:

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

4.10.3.2 Tiers

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

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

Tier separation, in terms of prioritization, is much more enforced in AIX 5L than in the previous release. A process in tier 1 will never have priority over a process in tier 0, since there is no overlapping of priorities in tiers. It is unlikely that classes in tier 1 will acquire any resources if the processes in tier 0 are consuming all the resources. This occurs because the control of leftover resources is much more restricted than in the AIX Version 4.3.3 release of WLM, as shown in Figure 23.

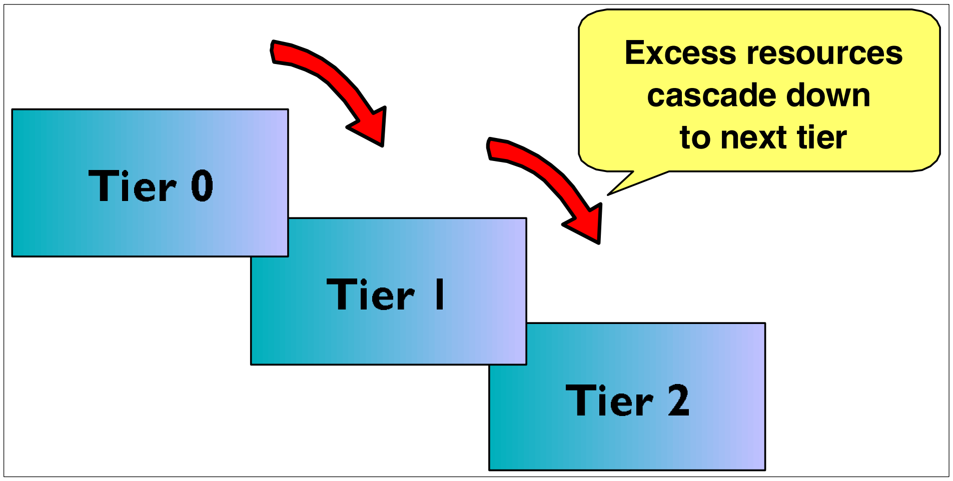
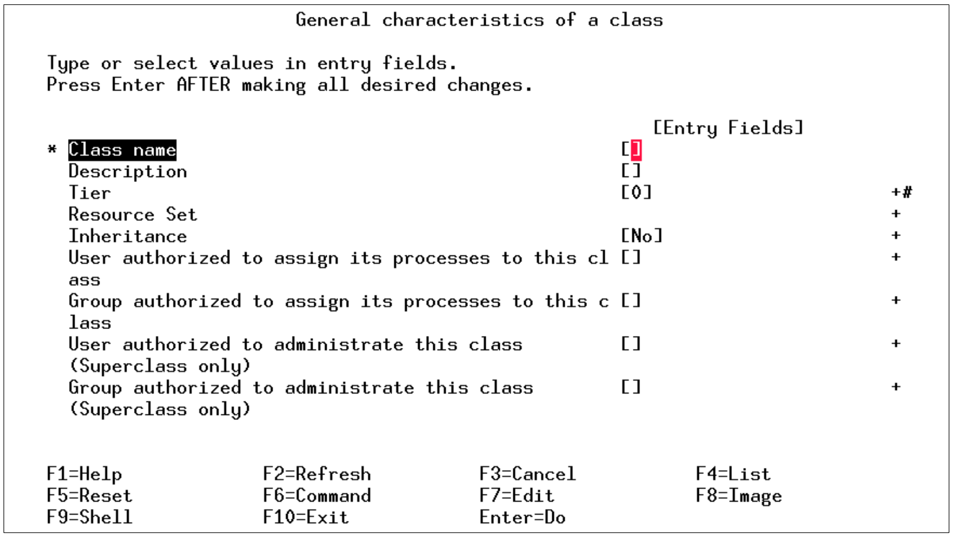


Figure 23. Resources cascading through tiers

4.10.3.3 Class attributes

In order to create a class, there are different attributes that are needed to have an accurate and well organized group of classes. Figure 24 on page 87 shows the SMIT panel for Class attributes.



```
                          General characteristics of a class

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

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


    F1=Help             F2=Refresh          F3=Cancel           F4=List
    F5=Reset            F6=Command          F7=Edit             F8=Image
    F9=Shell            F10=Exit            Enter=Do
```

Figure 24. SMIT with the class creation attributes screen

The sequence of attributes within a class (as shown in Figure 24) are outlined below:

Class name

It is a unique class name with up to 16 characters. It can contain uppercase and lowercase letters, numbers, and underscores (_).

Description

An optional brief description about this class.

Tier

A number between 0 and 9, for class priority ranking. It will be the tier that this class will belong to. An explanation about tiers can be found in Section 4.10.3.2, “Tiers” on page 86.

Resource Set

This attribute is used to limit the set of resources a given class has access to in terms of CPUs (processor set). The default, if unspecified, is *system*, which gives access to all the CPU resources available on the system.

Inheritance

The inheritance attribute indicates whether or not a child process should inherit its parent’s class or get classified according to the automatic assignment rules upon `exec`. The possible values are *yes* or *no*; the default is *no*. This attribute can be specified at both superclass and subclass level.

User and Group authorized to assign its processes to this class

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

User and Group authorized to administrate this class

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

4.10.3.4 Classification process

There are two ways to classify processes in WLM:

- Automatic assignment when a process calls the system call exec, using assignment rules specified by a WLM administrator. This automatic assignment is always in effect (cannot be turned off) when WLM is active.

This is the most common method of assigning processes to the different classes.

- Manual assignment of a selected process or group of processes to a class by a user with the required authority on both the process and the target class. This manual assignment can be done either by a WLM command, which could be invoked directly or through SMIT or Web-based System Manager, or by an application, using a function of the WLM Application Programming Interface. Manual assignment overrides automatic assignment.

Automatic assignment

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

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

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

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

- If the field in the rule has a value of hyphen (-), any value of the corresponding process attribute is a match.
- If the value of the process attribute (for all the attributes except *type)* matches one of the values in the list in a rule, and it is not excluded (prefaced by an exclamation point (!)), it is considered a match.

- When one of the values for *type* attribute in the rule is comprised of two or more values separated by a plus (+) sign, a process will be a match for this value only if its characteristics match all the values mentioned above.

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

Manual assignment

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

The manual assignment can be made or canceled separately at the superclass level, the subclass level, or both. In order to manually assign processes to a class or cancel an existing manual assignment, a user must have the right level of privilege (that is, must be the root user, adminuser, or admingroup for the superclass or authuser and authgroup for the superclass or subclass). A process can be manually assigned to a superclass only, a subclass only, or to a superclass and a subclass of the superclass. In the latter case, the dual assignment can be done simultaneously (with a single command or API call) or at different times, possibly by different users.

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

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

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

- The root user can assign any process to any class.

- A user with administration privileges on the subclasses of a given superclass (that is, the user or group name matches the attributes adminuser or admingroup of the superclass) can manually reassign any process from one of the subclasses of this superclass to another subclass of the superclass.
- A user can manually assign their own processes (same real or effective user ID) to a superclass or a subclass for which they have manual assignment privileges (that is, the user or group name matches the attributes authuser or authgroup of the superclass or subclass).

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

Class assignment rules

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

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

Class name

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

Reserved

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

User

The user name (as specified in the /etc/passwd file, LDAP, or in NIS) of the user owning a process can be used to determine the class to which the process belongs. This attribute is a list of one or more user names, separated by a comma (,). Users can be excluded by using an exclamation point (!)

prefix. Patterns can be specified to match a set of user names using full Korn shell pattern matching syntax.

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

Group

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

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

Application path names

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

Process type

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

Table 3. List of process types

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

Application tags

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

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

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

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

Table 4 is an example of classes with their respectives attributes for assignment rules.

Table 4. Examples of class assignment rules

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

In Table 4, the rule for Default class is omitted from display, though this class's rule is always present in the configuration. The rule for System is explicit, and has been put first in the file. This is deliberate so that all processes started by root will be assigned to the System superclass. By moving the rule for the System superclass further down in the rules file, the system administrator could have chosen to assign the root processes which would not be assigned to another class (because of the application executed, for instance) to System only. In Table 4, with the rule for System on top, if root executes a program in /usr/oracle/bin/db* set, the process will be classified as System. If the rule for the System class were after the rule for the db2 class, the same process would be classified as db1 or db2, depending on the tag.

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

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

4.10.3.5 Backward compatibility issues

As mentioned earlier, in the first release of WLM, the system default for the resource shares was one share. In AIX 5L, it is (-), which means that the resource consumption of the class for this particular resource is not regulated by WLM. This changes the semantics quite a bit, and it is advised that system administrators review their existing configurations and consider if the new default is good for their classes, or if they would be better off either setting up a default of one share (going back to the previous behavior) or setting explicit values for some of the classes.

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

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

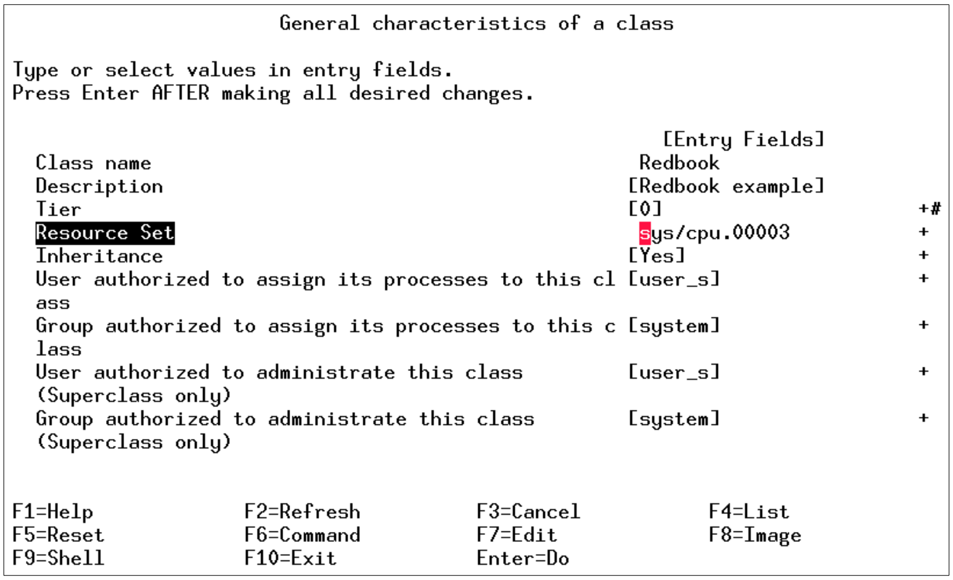
- shares = -
- min = 0 percent, softmax = 100 percent, hardmax = 100 percent.

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

4.10.4 Resource sets

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

Figure 25 on page 95 shows the SMIT panel where a resource set can be specified for a specific class.



```
                        General characteristics of a class

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

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


F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 25. Resource set definition to a specific class

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

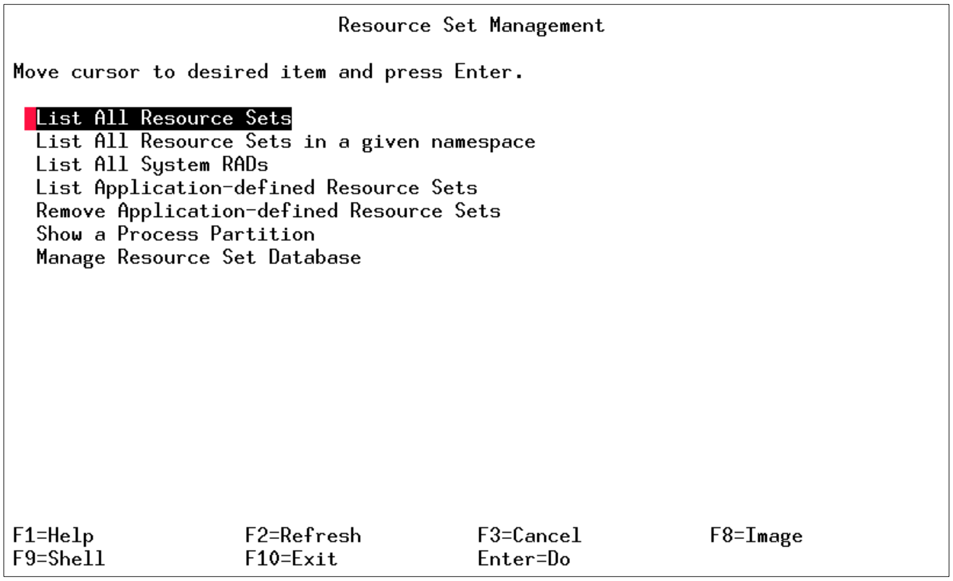
```
# lsrset -av
T  Name               Owner   Group   Mode    CPU  Memory  Resources
r  sys/sys0           root    system  r-----    4     511  sys/sys0
sys/node.00000 sys/mem.00000 sys/cpu.00003 sys/cpu.00002 sys/cpu.00001
sys/cpu.00000
r  sys/node.00000     root    system  r-----    4     511  sys/sys0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

4.10.4.1 Rset registry

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

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

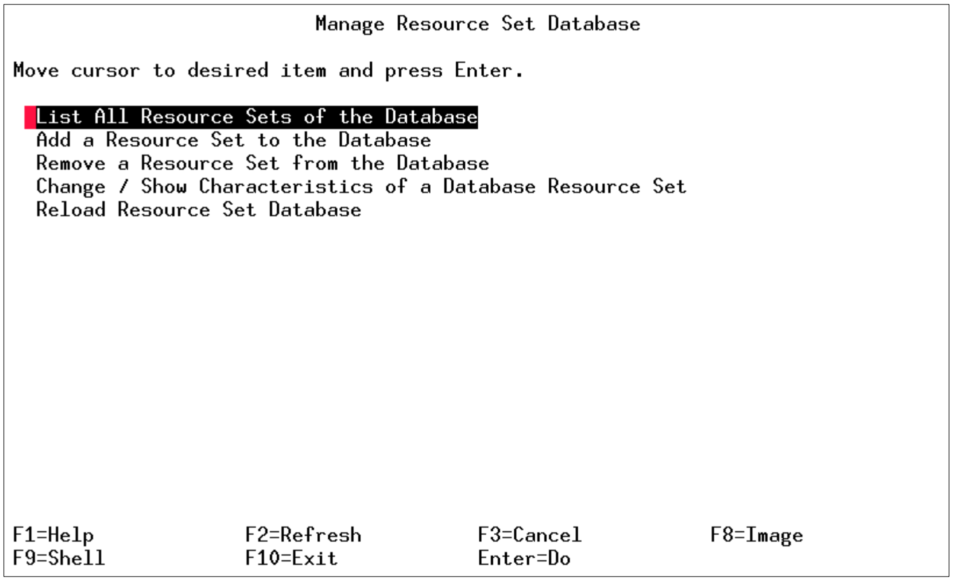
The `SMIT rset` command has options to list, remove, or show a specific resource set used by a process and the management tools, as shown in Figure 26.



```
                              Resource Set Management

Move cursor to desired item and press Enter.

  List All Resource Sets
  List All Resource Sets in a given namespace
  List All System RADs
  List Application-defined Resource Sets
  Remove Application-defined Resource Sets
  Show a Process Partition
  Manage Resource Set Database
```

Figure 26. SMIT main panel for resource set management

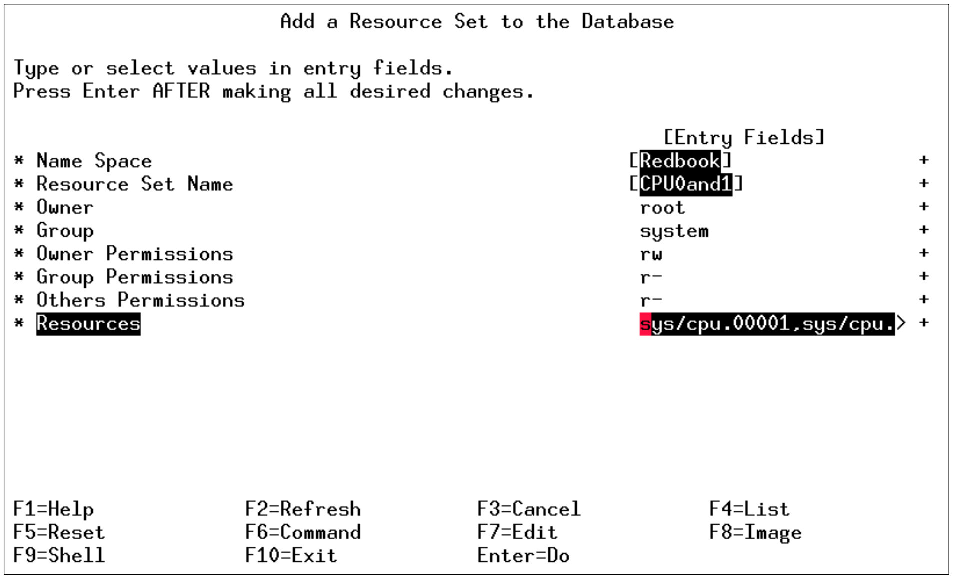
To create, delete, or change a resource set in the rset registry, you must select the Manage Resource Set Database item in the SMIT panel. In this panel, it is also possible to reload the rset registry definitions to make all changes available to the system. Figure 27 on page 98 shows the SMIT panel for rset registry management.



```
                         Manage Resource Set Database

Move cursor to desired item and press Enter.

  List All Resource Sets of the Database
  Add a Resource Set to the Database
  Remove a Resource Set from the Database
  Change / Show Characteristics of a Database Resource Set
  Reload Resource Set Database
```

Figure 27. SMIT panel for rset registry management

To add a new resource set, you must specify a name space, a resource set name, and the list of resources. It is also possible to change the permissions for the owner and group of this rset. In addition, permissions for the owner, groups and others can also be specified. Figure 28 on page 99 shows the SMIT panel for this task.



```
                         Add a Resource Set to the Database

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

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









F1=Help              F2=Refresh           F3=Cancel            F4=List
F5=Reset             F6=Command           F7=Edit              F8=Image
F9=Shell             F10=Exit             Enter=Do
```

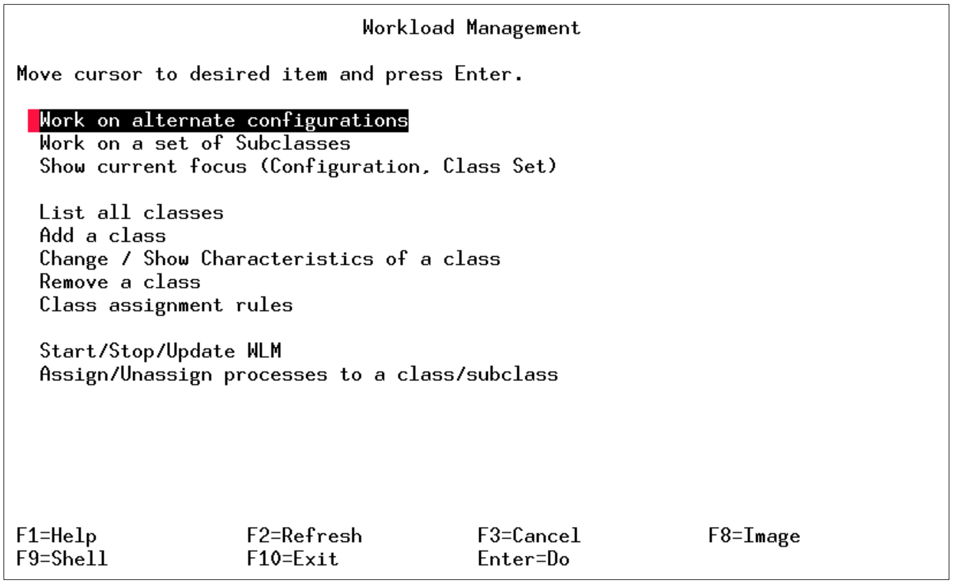
Figure 28. SMIT panel to add a new resource set

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

4.10.5 WLM configuration enhancements

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

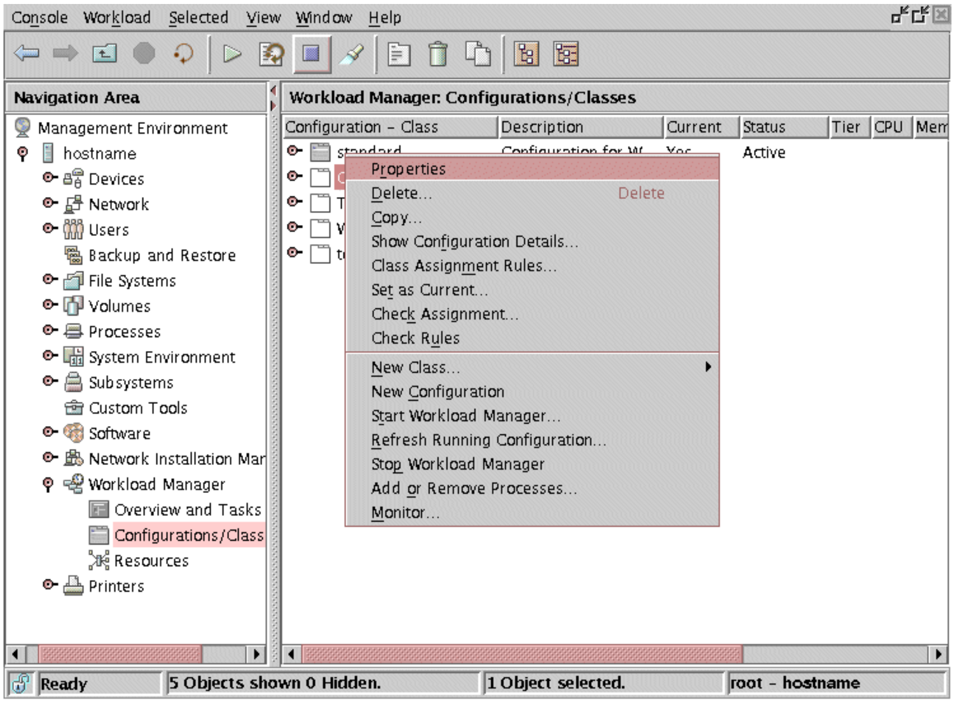
Figure 29 on page 100 shows a SMIT character-based main panel for Workload Manager.



```
                              Workload Management

Move cursor to desired item and press Enter.

  Work on alternate configurations
  Work on a set of Subclasses
  Show current focus (Configuration, Class Set)

  List all classes
  Add a class
  Change / Show Characteristics of a class
  Remove a class
  Class assignment rules

  Start/Stop/Update WLM
  Assign/Unassign processes to a class/subclass





F1=Help             F2=Refresh          F3=Cancel           F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 29. SMIT main panel for Workload Manager configuration

It is also possible, to view, modify or create Workload Manager through the Web-based System Manager, as shown on Figure 30 on page 101.



Ready

5 Objects shown 0 Hidden.

1 Object selected.

root - hostname

Figure 30. Web-based System Manager options for Workload Manager

4.10.5.1 Work on alternate configurations

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

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

If this option is selected in the SMIT panel, it will open a new submenu with some additional options:

Show all configurations

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

```
COMMAND STATUS

Command: OK           stdout: yes          stderr: no

Before command completion, additional instructions may appear below.

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

Copy a configuration

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

Create a configuration

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

Select a configuration

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

Enter configuration description

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

Remove a configuration

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

4.10.5.2 Work on a set of Subclasses

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

In Figure 31 on page 103, user in Superclasses was selected as the class set, and the operation created a new class produced the DB Subclass for user in Superclasses.

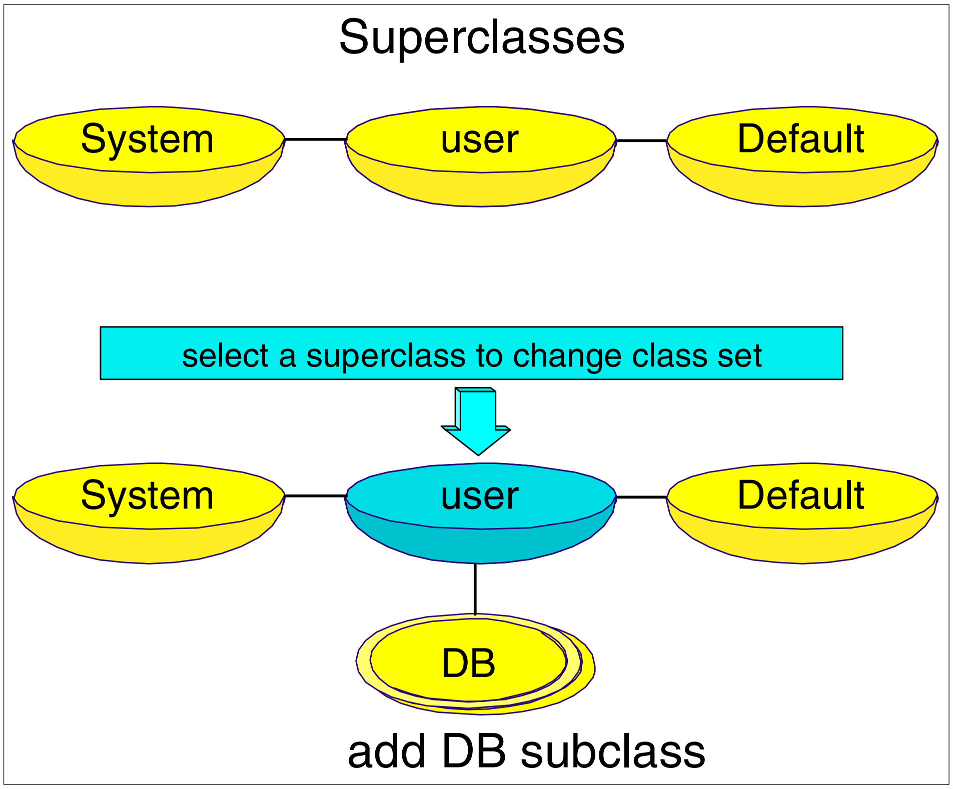


Figure 31. An example of adding a subclass to a superclass

4.10.5.3 Show current focus

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

```
COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

Configuration: redbook
Class set: Subclasses of user/

current -> redbook
```

4.10.5.4 List all classes

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

```
COMMAND STATUS

Command: OK           stdout: yes          stderr: no

Before command completion, additional instructions may appear below.

Default
Shared
db
```

4.10.5.5 Add a class

This option can be used to add a new Superclass or Subclass. Section 4.10.3.3, “Class attributes” on page 87 gives a detailed description of all the fields for this panel.

4.10.5.6 Change/Show Characteristics of a class

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

General characteristics of a class

It is possible to change all the characteristics of a class; see Section 4.10.3.3, “Class attributes” on page 87 for a list of attributes that can be modified with this option. Figure 24 on page 87 shows the SMIT panel for this option.

CPU resource management

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

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

```
CPU resource management

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

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

Memory resource management

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

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

```
Memory resource management

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

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

Disk I/O resource management

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

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

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

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

```
diskIO resource management

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

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

4.10.5.7 Remove a class

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

4.10.5.8 Class assignment rules

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

List all Rules

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

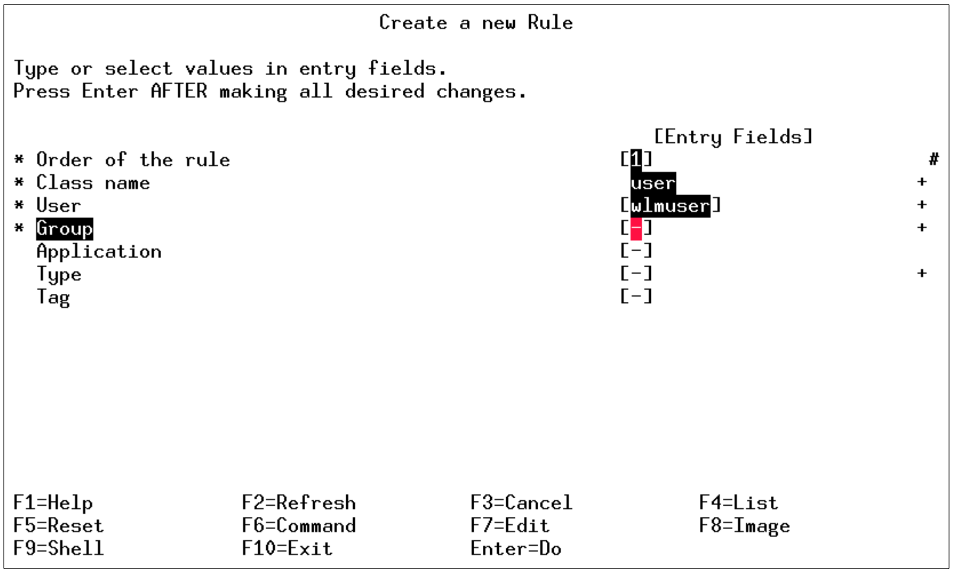
```
COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

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

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

Create a new Rule

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

Figure 32 shows the SMIT panel for creating a new Rule.



```
                              Create a new Rule

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

                                                      [Entry Fields]
* Order of the rule                                 [1]                   #
* Class name                                         user                 +
* User                                              [wlmuser]             +
* Group                                             [-]                   +
  Application                                       [-]
  Type                                              [-]                   +
  Tag                                               [-]









F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 32. Example of SMIT Panel for creating a new Rule

Below is a discussion of the fields to fill out for Rule Order. Order of the Rules and class name are mandatory fields; all others are optional.

Order of the rule Defines the rule order among other rules. The rule number one (001) is the first preferred order.

Class name Specifies which class will be affected by the rule.

User If specified, it will affect the user processes that match the pattern provided.

Group If specified, it will affect the group processes that match the pattern provided.

Application Affects a specific application, or you can use wildcards to affect a certain range of applications. For example, /tmp/wlm/* will affect any application under the /tmp/wlm directory.

Type Only defined types of applications will be affected.

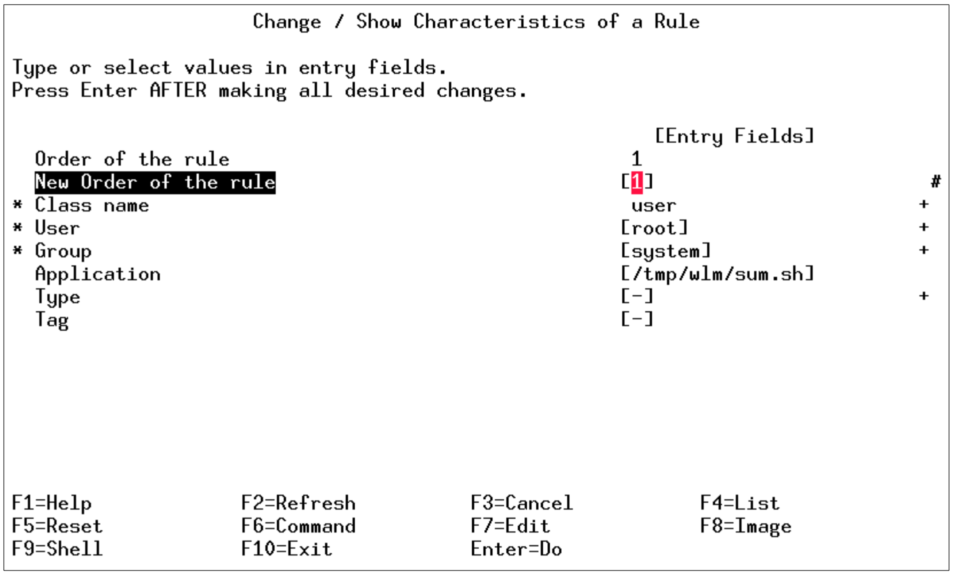
Tag Affects specific applications that have a tag that matches.

Note

Note that Section 4.10.3.4, “Classification process” on page 88 has a detailed architectural approach about Assignment Rules.

Change/Show Characteristics of a Rule

It is possible to change all characteristics established for a Rule, including order and class. Figure 33 shows a SMIT panel used for this item.



```
                      Change / Show Characteristics of a Rule

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

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







F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 33. Fields that can be modified for a specific Rule

Delete a Rule

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

Note

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

4.10.5.9 Start, Stop, or Update WLM

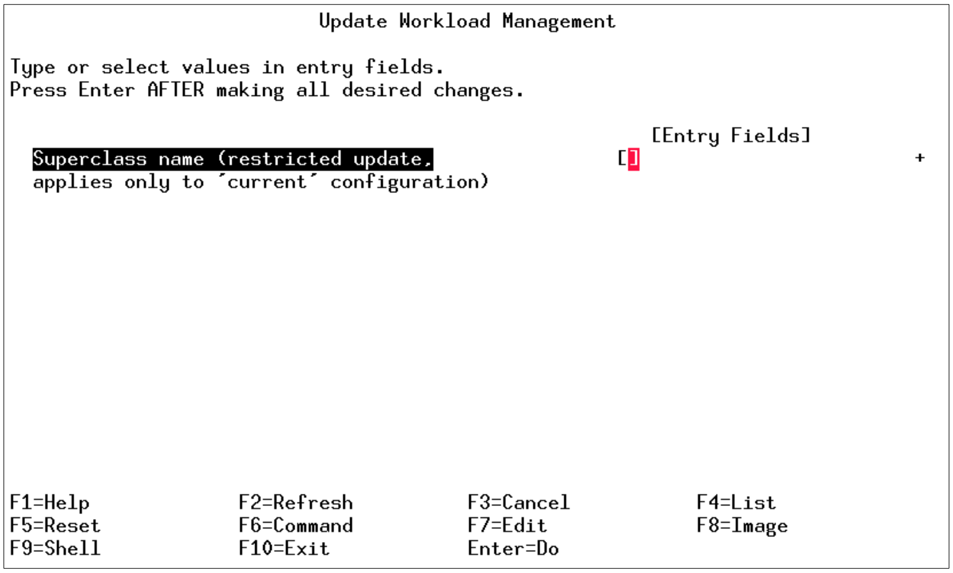
In this option, it is possible to Start and Stop WLM. Or, if you modified, created, or removed any component on WLM, you can update so that the changes take effect. Another function of this option is to show the WLM status.

Update Workload Management

The update function (as shown in Figure 34) allows you to create classes, change assignment Rules, and perform many other functions that were not updated in earlier releases.

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

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



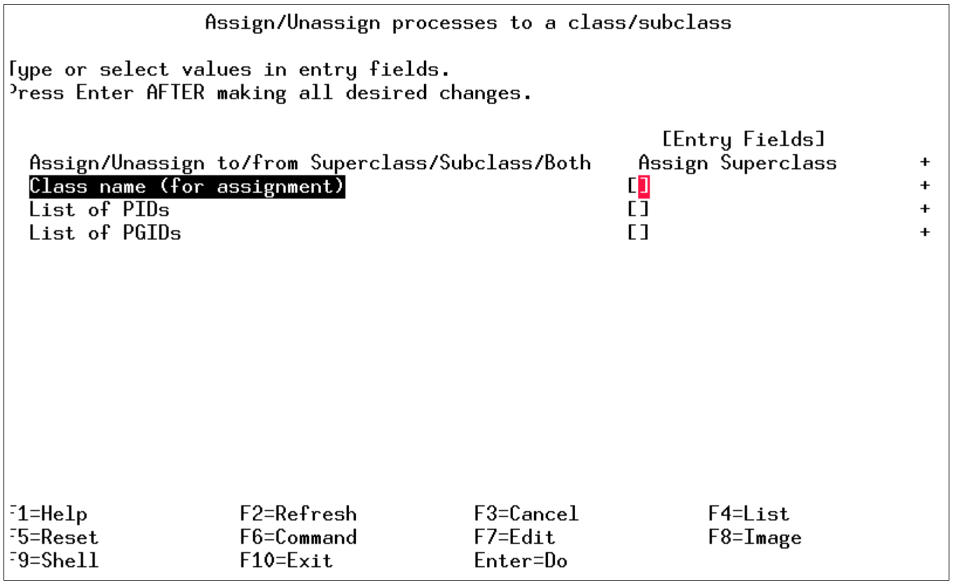
```
                          Update Workload Management

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

                                                     [Entry Fields]
  Superclass name (restricted update,               []                    +
  applies only to 'current' configuration)













F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 34. SMIT Panel for Update Workload Management

4.10.5.10 Assign/Unassign processes to a class/subclass

The “Manual assignment” portion of Section 4.10.3.4, “Classification process” on page 88 describes this process from an architectural point of view.

In terms of administration, Figure 35 shows a SMIT panel with all attributes for this task listed.



```
                  Assign/Unassign processes to a class/subclass

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

                                                     [Entry Fields]
  Assign/Unassign to/from Superclass/Subclass/Both    Assign Superclass      +
  Class name (for assignment)                        []                      +
  List of PIDs                                       []                      +
  List of PGIDs                                      []                      +









F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 35. SMIT panel for manual assignment of processes

Assign/Unassign to/from Superclass/Subclass/Both

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

All the options for this field and their respective description are as follows:

Assign Superclass All desired processes will be assigned to a specific Superclass.

Assign Subclass All desired processes will be assigned to a specific Subclass.

Assign Both All desired processes will be assigned to both Superclass and Subclass levels.

Unassign Superclass All desired processes will be unassigned from a Superclass

Unassign Subclass	All desired processes will be unassigned from a Subclass.
Unassign Both	All desired processes will be unassigned from both Superclass and Subclass.

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

List of PIDs
It is possible to select multiple processes at once. A comma (,) must be used as a separator between each PID.

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

4.11 System V Release 4 print subsystem

On the Itanium-based platform:

- The classic AIX print subsystem is not available.
- The System V Release 4 print subsystem is the default

On the POWER platform:

- Both the AIX and the System V Release 4 print subsystems are available.
- The AIX print subsystem is the default.

When the AIX Print Subsystem was created, it was designed to combine the features of the System V and Berkeley Software Distribution (BSD) printing standard, along with some unique features found only in AIX. This design had some distinct advantages in the past:

- Easy transition to AIX

  To provide an easy transition from another operating system to AIX, many of the commands traditionally used for printing were provided. For instance, BSD users can still print using the same `lpr` command they had become accustomed to. Also, scripts that were used to print did not necessarily need to be changed.

- Powerful and versatile print drivers

  The print drivers used to drive specific printers were designed in such a way that most printing options available on the printer could be used by selecting one or more of the many flags known to the backend. In addition, the print data stream could easily be modified with user and system defined filters and formatters.

- Limits fields

  Limits fields gave users a valid range of choices for each option would prohibit a user from using an incorrect value, and would send a message to the user stating the reason for the resulting print job rejection.

However, the same features that gave AIX printing an advantage over other UNIX operating systems also served to make the AIX print subsystem less compliant to widely used standards. With the onset of the development of AIX 5L for Itanium-based platforms, it was necessary to look for an alternative print solution that provides a more standard, less complex print subsystem that potentially embodies the concept of directory enablement, and lets the source code of AIX 5L for POWER and AIX 5L for Itanium-based systems intersect as much as possible. The AIX 5L for Itanium-based systems' development team had chosen the System V Release 4 (SVR4) print subsystem as the printing solution, and this print subsystem was added to AIX 5L for POWER with the long-term goal of making it the default print solution for AIX. In AIX 5L for Itanium-based systems, it will be the only print subsystem offered. Section 4.11.1, "Understanding the System V print service" on page 113 provides a brief overview of the print request processing of the newly implemented System V print subsystem in AIX 5L for POWER, and Section 4.11.3, "System V print subsystem management" on page 124 describes the commands which are available to manage the System V printer services. System administrators who prefer to use graphical system management tools will find useful information in Section 4.11.5, "User interface for AIX and System V print subsystems" on page 129.

If the code for both print subsystems is installed, the base operating system of the current AIX 5L release uses the traditional AIX print subsystem by default and the System V print subsystem is not active. Section 4.11.2, "Packaging and installation" on page 115 covers the details about fileset packaging and the installation of the System V print subsystem support in AIX 5L.

AIX 5L provides a command menu, a SMIT menu, and a Web-based System Manager menu, which allows the system administrator to switch between the AIX and the System V print subsystems, but will not allow both print subsystems to be active at the same time. Section 4.11.7, "Switching between AIX and System V print subsystems" on page 136 gives in-depth information about the switching process and the related commands.

Supplemental information about the user interface specification, the terminfo database, and the supported printers can be found in the Section 4.11.4, "User interface specifications" on page 127., and Section 4.11.6, "Terminfo and supported printers" on page 134.

4.11.1 Understanding the System V print service

The System V print subsystem was ported from SCO's UnixWare 7 to AIX 5L. The print subsystem, as such, supports local printing (parallel and serial), remote printing using BSD's lpd protocol (RFC 1179), and network printing using Hewlett-Packard's (HP) JetDirect. The code was internationalized to conform to and to comply with the AIX international standards and requirements.

The System V print service is a collection of utilities that assists you, as system administrator (or printer administrator), to configure, monitor, and control the printers on your system.

The print service:

- Receives files users want to print.
- Filters the files (if needed), so they can print correctly.
- Schedules the work of one or more printers.
- Starts programs that interface with the printers.
- Keeps track of the status of jobs.
- Alerts you to printer problems.
- Keeps track of mounting forms and filters.
- Issues error messages when problems arise.

Figure 36 on page 114 shows an overview of the processing of a print request, illustrates the following explanations, and helps to understand the overall concept.

When a user sends a file to a printer, the print service assigns to the request (print job) a unique name, the request ID.

The request ID consists of the name of the printer on which the file is to be printed and a unique number identifying the file. Use this request ID to find out the status of the print job or to cancel the print job. The print service keeps track of all the print requests in an associated request log.

The print job is spooled, or lined up, with other print jobs to be sent to a printer. Each print job is processed and waits its turn in line to be printed. This line of pending print jobs is called a print queue.

Each printer has its own queue; you can hold jobs in the queue, move jobs up in a queue, or transfer jobs to another queue.

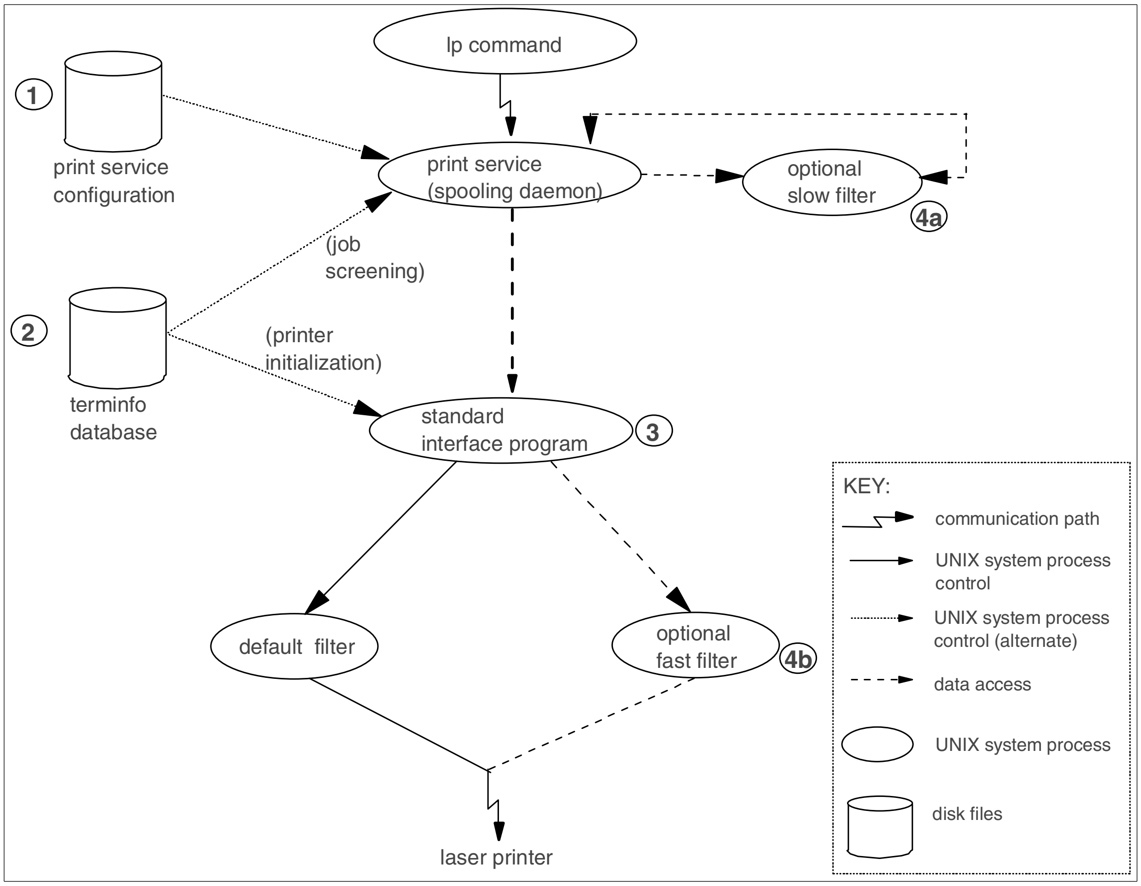


Figure 36. Overview of print request processing

Each print request is sent to a spooling daemon (`lpsched`) that keeps track of all the jobs. The daemon is created when you start the print service. The spooling daemon is also responsible for keeping track of the status of the printers and slow filters; when a printer finishes printing a job, the daemon starts printing another job if one is queued.

You can customize the print service by adjusting or replacing some of the items shown in Figure 36. The following numbers are explanations of the keys used in the diagram:

1. For most printers, you need only to change the printer configuration stored on disk. For further details, refer to the `lpadmin` command documentation for adding or modifying a local printer.
2. The print service relies on the standard interface script and the terminfo database to initialize each printer and set up a selected page size, character pitch, line pitch, and character set. For printers that are not represented in the terminfo database, you can add a new entry that describes the capabilities of the printer. The print service uses the terminfo database in two parallel capacities: screening print requests to ensure that those requests can be handled by the desired printer, and setting the printer so it is ready to print the requests. For example, if the terminfo database does not show a printer capable of setting a page length requested by a user, the spooling daemon rejects the request. However, if it does show it to be capable, then the interface program uses the same information to initialize the printer.
3. If you have a particularly complicated printer or if you want to use features not provided by the print service, you can change the interface script. This script is responsible for managing the printer: it prints the banner page, initializes the printer, and invokes a filter to send copies of the user's files to the printer.
4. To provide a link between the applications used on your system and the printers, you can add slow and fast filters. Each type of filter can convert a file into another form (for example, mapping one set of escape sequences into another), and can provide a special setup by interpreting print modes requested by a user. Slow filters are run separately by the spooling daemon to avoid tying up a printer. Fast filters are run so their output goes directly to the printer; thus, they can exert control over the printer.

4.11.2 Packaging and installation

The AIX and System V print subsystems are both packaged with the base operating system, but it specifically depends on the hardware configuration of your system which filesets are installed during the initial base installation. The option chosen for the Installation Configuration (default/minimal) under the Advanced Options menu during the base system installation process do not have any impact on the selection and installation of the print subsystem filesets.

The filesets given below provide the core function of the AIX print subsystem:

bos.rte.printers	Front End Printer Support
printers.rte	Printer Backend
printers.msg.xx_XX.rte	Printer Backend Messages for the system specific locale indicated by xx_XX in the fileset name. The initial release of AIX 5L limits the National Language Support (NLS) for the System V print subsystem to the ISO8859-1 English (United States) en_US locale.

The frontend printer support, bos.rte.printers, is part of the bos.rte file package, and therefore is always installed on the system. This fileset provides frontend print commands, such as `qprt`, `lpr`, `enq`, `mkque`, and `rmque`, that allow a user or the system administrator to interact with the qdaemon's spooler queues. For compatibility and usability reasons, the traditional AIX print subsystem maps several System V and BSD print commands to the AIX specific print commands. For example, the `lp` command used to be nothing more than a program which translates the System V `lp` flags to their counterparts of the `enq` AIX command, and after all the command line arguments were processed, the translated list of flags is finally used to call the `enq` command. As far as the frontend is concerned, the System V commands affected are `cancel`, `lp`, and `lpstat`. For BSD, the relevant frontend commands are `lpq`, `lpr`, and `lprm`.

In AIX 5L, the System V and BSD frontend print commands are still in the /usr/bin directory but, by default, they are now linked to the traditional AIX print command wrappers in the /usr/aix/bin directory:

```
# ls -l /usr/bin | grep aix
lrwxrwxrwx   1 root system          19 Sep 06 15:46 cancel -> /usr/aix/bin/cancel
lrwxrwxrwx   1 root system          15 Sep 06 15:46 lp -> /usr/aix/bin/lp
lrwxrwxrwx   1 root system          16 Sep 06 15:46 lpq -> /usr/aix/bin/lpq
lrwxrwxrwx   1 root system          16 Sep 06 15:46 lpr -> /usr/aix/bin/lpr
lrwxrwxrwx   1 root system          17 Sep 06 15:46 lprm -> /usr/aix/bin/lprm
lrwxrwxrwx   1 root system          19 Sep 06 15:46 lpstat -> /usr/aix/bin/lpstat
```

The AIX printer backend is a collection of programs called by the spooler's `qdaemon` command to manage a print job that is queued for printing. The printer backend performs the following functions:

- Receives from the `qdaemon` command a list of one or more files to be printed.
- Uses printer and formatting attribute values from the database; overridden by flags entered on the command line.
- Initializes the printer before printing a file.

- Runs filters as necessary to convert the print data stream to a format supported by the printer.
- Provides filters for simple formatting of ASCII documents.
- Provides support for printing national language characters.
- Passes the filtered print data stream to the printer device driver.
- Generates header and trailer pages.
- Generates multiple copies.
- Reports paper out, intervention required, and printer error conditions.
- Reports problems detected by the filters.
- Cleans up after a print job is canceled.
- Provides a print environment that a system administrator can customize to address specific printing needs.

The AIX printer backend fileset printers.rte belongs to several of the default system bundles which are located in the /usr/sys/inst.data/sys_bundle directory. These bundles include:

App-Dev.bnd Application Development Bundle: A collection of software products for developing application programs.

Client.bnd Client Bundle: A collection of software products for single user systems running in a stand-alone or networked client environment.

Pers-Prod.bnd Personal Productivity Bundle: A collection of software products for graphical desktop systems running AIX and PC applications.

Server.bnd Server Bundle: A collection of software products for multi-user systems running in a stand-alone or networked environment.

The fact that the bundles listed above belong to the default system bundle category does not imply that any of these bundles are installed by default. They are predefined and supplied for your convenience, but the system administrator would have to intentionally initiate the installation of any of the bundles.

Furthermore, the printers.rte fileset is not listed in any of the default system bundles, which are used during the base installation process:

ASCII.autoi An ASCII terminal system bundle file that lists filesets to install if the console is not a Low Function Terminal (LFT).

BOS.autoi A system bundle file that lists the group of packages and filesets that will always be installed when the **Default** Installation Configuration under the Advanced Options menu (during the base system installation process) was specified.

MIN_BOS.autoi A system bundle file that lists the group of packages and filesets that will always be installed when the **Minimal** Installation Configuration under the Advanced Options menu (during the base system installation process) was specified.

GOS.autoi A graphics system bundle file that lists filesets to install if the console is an LFT and when the **Default** Installation Configuration was chosen (during the base system installation process).

MIN_GOS.autoi A graphics system bundle file that lists filesets to install if the console is an LFT and when the **Minimal** Installation Configuration was chosen (during the base system installation process).

Since printers.rte is not explicitly included in any of the bundle files with the autoi extension, the requisite for printers.rte of other filesets determines whether or not the backend support for the AIX print subsystem is installed. The fileset dependencies are defined by the multi-volume .toc file in the /usr/sys/mvCD directory of the installation media, and at the time of publication, four fileset dependencies designated printers.rte as a required fileset for installation. These fileset dependencies include:

bos.txt.tfs	Text Formatting Services Commands
printers.ibmNetPrinter.attach	en_US IBM Network Printer Attachment
printers.ibmNetColor.attach	en_US IBM Network Color Printer Attachment
printers.hpJetDirect.attach	en_US Hewlett-Packard JetDirect Network Printer

The most significant fileset of the ones listed above is bos.txt.tfs. The text formatting services are included in GOS.autoi and MIN_GOS.autoi and are also directly required by the X11.Dt.rte fileset for the AIX Common Desktop Environment (CDE) support.

Table 5 summarizes the different combinations for the AIX print subsystem backend support. These combinations' parts include the HW configuration, installation configuration, and system administrators intervention.

Table 5. AIX print subsystem backend support

HW Graphics Support	Installation Configuration	Installation Initiation and Process	AIX Print Backend Support
no	minimal	NA	no
no	default	NA	no
yes	minimal	BOS installation: MIN_GOS.autoi	yes
yes	default	BOS installation: GOS.autoi	yes
no	minimal/ default	Manual Installation: printers.rte	yes
no	minimal/ default	Manual Installation: App-Dev.bnd Cleint.bnd Pers.Prod.bnd Server.bnd	yes

As mentioned before, the traditional AIX print subsystem maps several System V and BSD print commands to the AIX specific print commands. As far as the backend print support is concerned, the only two System V commands affected are `disable` and `enable`. In AIX 5L, these specific System V backend print commands are still in the /usr/bin directory, but by default they are now linked to the traditional AIX print command wrappers in the /usr/aix/bin directory:

```
# ls -l /usr/bin | grep -E "\/enable|disable"
lrwxrwxrwx   1 root system          20 Sep 05 13:46 disable -> /usr/aix/bin/disable
lrwxrwxrwx   1 root system          19 Sep 05 13:46 enable -> /usr/aix/bin/enable
```

In addition to the AIX print command wrappers for System V and BSD print commands in the /usr/aix/bin directory, a new lock file _AIX_print_subsystem gets installed under the /usr/aix directory. The existence of the lock file indicates that the AIX print subsystem is active. For reference, a full listing of the /usr/aix directory is provided below:

```
# ls -lR /usr/aix
total 8
-rw-rw-r--   1 root     system           0 Sep 01 18:02 _AIX_print_subsystem
drwxr-xr-x   2 bin      bin            512 Sep 05 13:46 bin
/usr/aix/bin:
```

```
total 576
-r-xr-xr-x   1 bin      bin           33648 Aug 24 21:22 cancel
-r-xr-x---   1 root     printq        33488 Aug 24 21:22 disable
-r-xr-x---   1 root     printq        33376 Aug 24 21:22 enable
-r-xr-xr-x   1 bin      bin           34228 Aug 24 21:22 lp
-r-xr-xr-x   1 bin      bin           33916 Aug 24 21:22 lpq
-r-xr-xr-x   1 bin      bin           35236 Aug 24 21:22 lpr
-r-xr-xr-x   1 bin      bin           34312 Aug 24 21:22 lprm
-r-xr-xr-x   1 bin      bin           35368 Aug 24 21:22 lpstat
```

The package of the System V print subsystem is named bos.svprint and consists of four filesets:

bos.svprint.fonts	System V Print Fonts
bos.svprint.hpnp	System V Hewlett-Packard JetDirect
bos.svprint.ps	System V Print Postscript
bos.svprint.rte	System V Print Subsystem

These filesets are supplemented by the locale specific message support and the System V printer terminal definitions:

bos.msg.xx_XX.svprint System V Print Subsystem Messages for the system specific locale indicated by xx_XX in the fileset name. The initial release of AIX 5L limits the National Language Support (NLS) for the System V print subsystem to the ISO8859-1 English (United States) en_US locale.

bos.terminfo.svprint.data System V Printer Terminal Definitions

The filesets bos.svprint.* and bos.terminfo.svprint.data are included in the BOS.autoi system bundle and will be installed by default on all AIX 5L systems. The main script which handles the system installation tasks, /usr/lpp/bosinst/bi_main, also ensures that the locale-specific message support is available through bos.msg.xx_XX.svprint.

All System V and BSD commands that are mapped by the executables in the /usr/aix/bin directory to the AIX print subsystem specific commands have their native System V or BSD counterpart in the /usr/sysv/bin directory. During a switch from the AIX to the System V print subsystem, the respective duplicate commands will be handled by removing the inactive print subsystems command's symbolic links and adding new symbolic links for the active commands. The following directory listing reflects this configuration on a system where the initially active AIX print subsystem was deactivated and switched to the System V print subsystem by the use of the newly introduced `switch.prt` command:

```
ls -l /usr/bin | grep sysv
lrwxrwxrwx 1 root system           20 Sep 12 18:58 cancel -> /usr/sysv/bin/cancel
lrwxrwxrwx 1 root system           21 Sep 12 18:58 disable -> /usr/sysv/bin/disable
lrwxrwxrwx 1 root system           20 Sep 12 18:58 enable -> /usr/sysv/bin/enable
lrwxrwxrwx 1 root system           16 Sep 12 18:58 lp -> /usr/sysv/bin/lp
lrwxrwxrwx 1 root system           17 Sep 12 18:58 lpq -> /usr/sysv/bin/lpq
lrwxrwxrwx 1 root system           17 Sep 12 18:58 lpr -> /usr/sysv/bin/lpr
lrwxrwxrwx 1 root system           18 Sep 12 18:58 lprm -> /usr/sysv/bin/lprm
lrwxrwxrwx 1 root system           20 Sep 12 18:58 lpstat -> /usr/sysv/bin/lpstat
```

Once the System V print subsystem is active, the new lock file _SYS5_print_subsystem will be present in the /usr/sysv directory and the AIX print subsystem lock file /usr/aix/_AIX_print_subsystem will no longer exist. You will find the recursive listing for the /usr/sysv directory in the following example (note the differences in user and group ownership in comparison to the executables in the /usr/aix/bin directory):

```
# ls -lR /usr/sysv
total 8
-r--r--r--   1 root     system  0 Sep 12 16:13 _SYS5_print_subsystem
drwxr-xr-x   2 bin      bin     512 Dec 31 1969 bin
/usr/sysv/bin:
total 2136
---x--x--x   1 lp       lp      112506 Aug 24 21:21 cancel
---s--x---   1 root     lp      113034 Aug 24 21:22 disable
---s--x---   1 root     lp      113034 Aug 24 21:22 enable
---x--x--x   1 lp       lp      137338 Aug 24 21:21 lp
-r-sr-xr-x   1 lp       lp      166690 Aug 24 21:22 lpq
-r-xr-xr-x   1 bin      bin     27182  Aug 24 21:22 lpr
-r-xr-xr-x   1 bin      bin     116930 Aug 24 21:22 lprm
---x--x--x   1 lp       lp      189442 Aug 24 21:21 lpstat
```

AIX 5L introduces a new user with the user named lp and a related group named the same.

The user lp is added to the /etc/passwd file for ownership of a majority of the files, which belong to the bos.svprint package. The entry in the /etc/passwd file is similar to the following example:

```
lp:*:11:11::/var/spool/lp:/bin/false
```

The group lp is added to the /etc/group file for group ownership of a majority of the files, which belong to the bos.svprint package. The entry in the /etc/group file is similar to the following example:

```
lp:!:11:root,lp,printq
```

Furthermore, the lp group is added to the formerly existing printq group so that a system administrator with lp group permissions can also have printq permissions, and the entry for the printq group in /etc/group is now similar to the following example:

```
printq:!:9:lp
```

The group definitions for the lp group and the printq group allow administrators with either printq or lp group authority to have access to the print commands of both print subsystems. In other words, an administrator with printq authority is able to use a command with lp group permissions, and vice-versa.

The AIX Print Subsystem is active by default. For both print subsystems, the active frontend commands are located and accessible as always through links in the /usr/bin directory. The commands for the frontend which are not active are not located in the directories which are normally accessible to users through the standard definition of the PATH environment variable. To use the inactive frontend, it must be switched using a command, or, preferably, by the use of the System Management Interface Tool (SMIT), or by the Web-based System Management Tool. More details about switching between the different print subsystems are given in Section 4.11.7, “Switching between AIX and System V print subsystems” on page 136. Only one frontend can be active at any moment.

The remainder of this section provides a set of comprehensive listings of files, directories, user and administrative commands, and internal programs that are installed or created on your system in order to support System V printing. For each entity, the file mode, ownership, group ownership, and the fully qualified pathname is given. Separate listings account for the differences, which depend on the type of the active print subsystem, and some comments are given for further explanation.

Changes and additions, which were applied to the bos.rte.printers fileset, are as follows:

```
File Mode    Owner   Group      Pathname
==========   =====   =====      ================================================
drwxr-xr-x   bin     bin        /usr/aix/bin                           (AIX)
-rwxr-xr-x   bin     bin        /usr/aix/bin/cancel                    (AIX)
-rwxr-xr-x   bin     bin        /usr/aix/bin/lp                        (AIX)
-rwxr-xr-x   bin     bin        /usr/aix/bin/lpq                       (AIX)
-rwxr-xr-x   bin     bin        /usr/aix/bin/lpr                       (AIX)
-rwxr-xr-x   bin     bin        /usr/aix/bin/lprm                      (AIX)
-rwxr-xr-x   bin     bin        /usr/aix/bin/lpstat                    (AIX)
-r-sr-x---   root    system     /usr/sbin/switch.prt                   (AIX)
-rwx------   root    system     /usr/sbin/switch.prt.subsystem         (AIX)
```

During the installation of the AIX 5L, the bos.rte.printers fileset and the newly introduced directory /usr/aix/bin are created. They hold the AIX print subsystem BSD compatibility executables. The switch.prt executable and switch.prt.subsystem script allow switching to the System V print subsystem.

Links and the lock file, that were created during the base operating system installation process, are as follows:

```
File Mode    Owner   Group      Pathname
==========   =====   =====      ================================================
lrwxrwxrwx   root    system     /usr/bin/cancel -> /usr/aix/bin/cancel
lrwxrwxrwx   root    system     /usr/bin/lp -> /usr/aix/bin/lp
lrwxrwxrwx   root    system     /usr/bin/lpq -> /usr/aix/bin/lpq
lrwxrwxrwx   root    system     /usr/bin/lpr -> /usr/aix/bin/lpr
lrwxrwxrwx   root    system     /usr/bin/lprm -> /usr/aix/bin/lprm
lrwxrwxrwx   root    system     /usr/bin/lpstat -> /usr/aix/bin/lpstat
lrwxrwxrwx   root    system     /usr/bin/disable -> /usr/aix/bin/disable
lrwxrwxrwx   root    system     /usr/bin/enable -> /usr/aix/bin/enable
-rwxrwx---   root    system     /usr/aix/_AIX_print_subsystem         (AIX)
```

The listed links and the lock file are only present when the traditional AIX print subsystem is active, and they are created during the BOS installation process by the function Add_Printer_Links of the bi_main script. For your reference, an excerpt of the relevant section in the bi_main script is provided in the following example:

```
...
# Add_Printer_Links
# Adds links and touches a file, to support
# the repackaging of printer filesets.
# This is only called for product installs ($PT=yes).
#
function Add_Printer_Links
{
...

    ln -s /usr/aix/bin/cancel /usr/bin/cancel
    ln -s /usr/aix/bin/lp /usr/bin/lp
    ln -s /usr/aix/bin/lpstat /usr/bin/lpstat
    ln -s /usr/aix/bin/lpq /usr/bin/lpq
    ln -s /usr/aix/bin/lpr /usr/bin/lpr
    ln -s /usr/aix/bin/lprm /usr/bin/lprm

    touch /usr/aix/_AIX_print_subsystem
    return 0
}
...
```

Changes and additions, which were applied to the printers.rte fileset, appear as follows:

```
File Mode    Owner   Group      Pathname
==========   =====   =====      ================================================
-r-xr-x---   root    printq     /usr/aix/bin/disable                  (AIX)
-r-xr-x---   root    printq     /usr/aix/bin/enable                   (AIX)
lrwxrwxrwx   root    system     /usr/bin/disable -> /usr/aix/bin/disable (AIX)
lrwxrwxrwx   root    system     /usr/bin/enable -> /usr/aix/bin/enable (AIX)
```

The links /usr/bin/disable and /usr/bin/enable are created during the printers.rte post installation phase.

A list of all files and directories in bos.svprint.rte are as follows:

```
File Mode    Owner   Group      Pathname
==========   =====   =====      ================================================
drwxrwxr-x   lp      lp         /usr/lib/lp
drwxrwxr-x   lp      lp         /usr/lib/lp/bin
```

```
drwxrwxr-x   lp      lp         /usr/lib/lp/model
drwxrwxr-x   root    system     /usr/lib/lp/objrepos
drwxr-xr-x   bin     bin        /usr/sysv
drwxr-xr-x   bin     bin        /usr/sysv/bin

-r-xr-xr-x   bin     bin        /usr/bin/lpc
-r--r--r--   lp      lp         /usr/lib/lp/bin/alert.proto
---x--x--x   lp      lp         /usr/lib/lp/bin/drain.output
---x--x--x   lp      lp         /usr/lib/lp/bin/lp.cat
---x--x--x   lp      lp         /usr/lib/lp/bin/lp.lvlproc
---x--x--x   lp      lp         /usr/lib/lp/bin/lp.pr
---x--x--x   lp      lp         /usr/lib/lp/bin/lp.set
---x--x--x   lp      lp         /usr/lib/lp/bin/lp.tell
-r-xr-xr-x   lp      lp         /usr/lib/lp/bin/slow.filter
---s--x---   root    lp         /usr/lib/lp/lpsched
---s--x---   root    lp         /usr/lib/lp/lpNet
--x--x--x-   lp      lp         /usr/lib/lp/model/B2
-r-xr-xr-x   lp      lp         /usr/lib/lp/model/B2.banntrail
-r-xr-xr-x   lp      lp         /usr/lib/lp/model/B2.job
-rwxrwxr-x   lp      lp         /usr/lib/lp/model/PS
-rwxr-xr-x   lp      lp         /usr/lib/lp/model/standard
---s--x---   root    lp         /usr/sbin/accept
---s--x---   root    lp         /usr/sbin/lpadmin
---s--x---   root    lp         /usr/sbin/lpfilter
---s--x---   root    lp         /usr/sbin/lpforms
---s--x---   root    lp         /usr/sbin/lpmove
---s--x---   root    lp         /usr/sbin/lpshut
---s--x---   root    lp         /usr/sbin/lpsystem
---s--x---   root    lp         /usr/sbin/lpusers
---s--x---   root    lp         /usr/sbin/reject
---x--x--x   lp      lp         /usr/sysv/bin/cancel
---s--x---   root    lp         /usr/sysv/bin/disable
---s--x---   root    lp         /usr/sysv/bin/enable
---x--x--x   lp      lp         /usr/sysv/bin/lp
-r-sr-xr-x   lp      lp         /usr/sysv/bin/lpq
-r-xr-xr-x   bin     bin        /usr/sysv/bin/lpr
-r-xr-xr-x   bin     bin        /usr/sysv/bin/lprm
---x--x--x   lp      lp         /usr/sysv/bin/lpstat
```

Links and files which are exclusively present when the System V print subsystem is active are as follows:

```
File Mode    Owner   Group      Pathname
==========   =====   =====      ================================================
lrwxrwxrwx   root    system     /usr/bin/cancel -> /usr/sysv/bin/cancel
lrwxrwxrwx   root    system     /usr/bin/lp -> /usr/sysv/bin/lp
lrwxrwxrwx   root    system     /usr/bin/lpq -> /usr/sysv/bin/lpq
lrwxrwxrwx   root    system     /usr/bin/lpr -> /usr/sysv/bin/lpr
lrwxrwxrwx   root    system     /usr/bin/lprm -> /usr/sysv/bin/lprm
lrwxrwxrwx   root    system     /usr/bin/lpstat -> /usr/sysv/bin/lpstat
lrwxrwxrwx   root    system     /usr/bin/disable -> /usr/sysv/bin/disable
lrwxrwxrwx   root    system     /usr/bin/enable -> /usr/sysv/bin/enable
[Created on the fly when switching to System V print subsystem]
-rwxrwx---   root    lp         /usr/sysv/_SYS5_print_subsystem
```

4.11.3 System V print subsystem management

In general, print administrators should use the Web-based System Manager to manage the System V print service. For further details about the Web-based System Manager support for the System V print service management, refer to Section 4.11.5, “User interface for AIX and System V print subsystems” on page 129. If you need to manage your print service from the command line, the remainder of this section provides a brief summary of the System V print service command line interface. All listed commands are fully documented in the AIX product documentation library.

Table 6 lists the print service commands available to all users. All commands are located in the /usr/bin directory.

Table 6. Print service commands available to all users

Command	Description
`cancel`	The `cancel` command allows users to cancel print requests previously sent with the `lp` command. The command permits cancellation of requests based on their request-ID or based on the login-ID of their owner.
`lp`	The `lp` command arranges for the named files and associated information (collectively called a request) to be printed. If file names are not specified on the command line, the standard input is assumed. Alternatively, the `lp` command is used to change the options for a request submitted previously. The print request identified by the request-ID is changed according to the print-options specified with this command.
`lpstat`	The `lpstat` command displays information about the current status of the print service. If no options are given, `lpstat` displays the status of all print requests made by the user.

The administrator can give users the ability to disable and enable a printer, so that when a printer is malfunctioning, the user can turn the printer off without having to call the administrator. (However, in your printing environment, it might not be reasonable to allow regular users to disable a printer.)

Table 7 provides a summary of the print service commands available only to the system or print administrator. To use the administrative commands, you must have root user authority or be member of either the printq or the lp group. All of the administrative print service commands listed in Table 7 are located in the /usr/sbin directory with two exceptions: the `lpsched` program resides in the /usr/lib/lp directory, and the `enable` and `disable` commands are found in the /usr/bin directory.

Table 7. Administrative print service commands

Command	Description
`accept` `reject`	`accept` allows the queuing of print requests for the named destinations. A destination can be either a printer or a class of printers. `reject` prevents queuing of print requests for the named destinations.
`enable` `disable`	The `enable` command activates the named printers, enabling them to print requests submitted by the lp command. If the printer is remote, the command will only enable the transfer of requests to the remote system. The `disable` command deactivates the named printers, disabling them from printing requests submitted by lp.
`lpadmin`	`lpadmin` configures the LP print service by defining printers and devices. It is used to add and change printers, to remove printers from service, to set or change the system default destination, to define alerts for printer faults, to mount print wheels, and to define printers for remote printing services.
`lpfilter`	The `lpfilter` command is used to add, change, delete, and list a filter used with the LP print service. These filters are used to convert the content type of a file to a content type acceptable to a printer.
`lpforms`	The `lpforms` command is used to administer the use of preprinted forms, such as company letterhead paper, with the System V print service.
`lpmove`	`lpmove` moves requests that were queued by lp between destinations (printers or classes of printers).
`lpsched`	`lpsched` allows you to start the System V print service.
`lpshut`	`lpshut` shuts down the print service. All printers that are printing at the time lpshut is invoked will stop printing.
`lpsystem`	The `lpsystem` command is used to define parameters for the LP print service, with respect to communication (via a high-speed network like TCP/IP) with remote systems.
`lpusers`	The `lpusers` command is used to set limits to the queue priority level that can be assigned to jobs submitted by users of the System V print service.

The administrative print service commands listed in Table 7 are supplemented by three default printer filters used by interface programs, which are located in the /usr/lib/lp/bin directory: `lp.cat`, `lp.set`, `lp.tell`. The `lp.cat` program reads the file to be printed on its standard input and writes it to the device to be printed on. Interface programs may call `lp.set` to set the character pitch, line pitch, page width, page length, and character set on the printer. Also, interface programs can use `lp.tell` to forward descriptions of printer faults to the print service. `lp.tell` sends everything that it reads on its standard input to the print service. The print service forwards the message as an alert to the print administrator

Finally, the four BSD compatibility commands (`lpc`, `lpr`, `lpq`, and `lprm`) are available in the /usr/bin directory for users and administrators.

A comprehensive listing of the file modes, ownership, group ownership and the fully qualified pathname for each of the commands mentioned in this paragraph are given in Section 4.11.4, “User interface specifications” on page 127.

4.11.4 User interface specifications

The user interface specifications for the System V print subsystem are the man pages for the printing and associated commands. Table 8 provides an overview of the available commands for the System V print subsystem. BSD system compatibility commands are also included in the list and noted accordingly.

In previous AIX releases, some System V and BSD print commands were mapped to AIX print subsystem commands to enhance compatibility and usability of the AIX print services. The executables of these commands were nothing more than wrappers, which called the AIX print subsystem specific `enq` command after all command line arguments had been translated to a list of `enq` specific flags. Since AIX 5L offers the possibility to use the System V print subsystem as an alternative to the traditional AIX print subsystem, the relevant commands have to be supplied in two different versions. The traditional AIX print subsystem command wrappers for the System V and BSD print executables are kept in the /usr/aix/bin directory, while the native System V print subsystem counterparts are collectively located in the /usr/sysv/bin directory. The relevant commands are referenced by symbolic links in the /usr/bin directory. The symbolic links always point to the version of the executable related to the type of the active print subsystem. The duplicate commands are marked below with an asterisk (*), but as far as the user interface specification for the System V print subsystem is concerned, only the native BSD compatibility executables in the /usr/sysv/bin directory are relevant.

Table 8. System V printing: user and administrative commands

accept	lp.set	lpmove	lpstat *
cancel *	lp.tell	lpq * (BSD)	lpsystem
disable *	lpadmin	lpr * (BSD)	lpusers
enable *	lpc (BSD)	lprm* (BSD)	reject
lp *	lpfilter	lpsched	
lp.cat	lpforms	lpshut	

For more detailed information about specific commands, refer to Section 4.11.3, “System V print subsystem management” on page 124 and the AIX documentation library.

AIX 5L provides a script (located at /usr/aix/bin/switch.prt.subsystem) that is used to switch from one active print subsystem to the other, and to display the active print subsystem. This script is called by the executable /usr/aix/bin/switch.prt for permission reasons. The executable and the script have the same syntax characteristics which are displayed if you run the commands from the command line without any arguments:

```
# switch.prt.subsystem
Usage: switch.prt.subsystem [-s AIX | SystemV ] [-d]
 -s switches to AIX print system or System V print system.
 -d displays current subsystem.

# switch.prt
Usage: switch.prt [-s AIX | SystemV ] [-d]
 -s switches to AIX print system or System V print system.
 -d displays current subsystem.
```

The valid values for the print_subsystem keyword are AIX and System V. The -d flag causes the command to display the current print subsystem. This command is intended to be executed only by the Web-based System Manager or SMIT, but will work from the command line with the proper permissions. The `switch.prt` command and the associated script are not documented in the AIX documentation library.

At the end of this section, a set of comprehensive listings of properties that are associated with the user interface commands and their related directories is provided. For each entity, the file mode, ownership, group ownership and the fully qualified pathname is given.

Properties of System V user interface commands and related directories appear as follows:

```
File Mode    Owner   Group      Pathname
```

```
==========   =====   =====      ===============================================
drwxrwxr-x   lp      lp         /usr/lib/lp
drwxrwxr-x   lp      lp         /usr/lib/lp/bin
drwxr-xr-x   bin     bin        /usr/sysv
drwxr-xr-x   bin     bin        /usr/sysv/bin

-r-xr-xr-x   bin     bin        /usr/bin/lpc
---x--x--x   lp      lp         /usr/lib/lp/bin/lp.cat
---x--x--x   lp      lp         /usr/lib/lp/bin/lp.set
---x--x--x   lp      lp         /usr/lib/lp/bin/lp.tell
---s--x---   root    lp         /usr/lib/lp/lpsched
---s--x---   root    lp         /usr/sbin/accept
---s--x---   root    lp         /usr/sbin/lpadmin
---s--x---   root    lp         /usr/sbin/lpfilter
---s--x---   root    lp         /usr/sbin/lpforms
---s--x---   root    lp         /usr/sbin/lpmove
---s--x---   root    lp         /usr/sbin/lpshut
---s--x---   root    lp         /usr/sbin/lpsystem
---s--x---   root    lp         /usr/sbin/lpusers
---s--x---   root    lp         /usr/sbin/reject
-r-sr-x---   root    system     /usr/sbin/switch.prt
-rwx------   root    system     /usr/sbin/switch.prt.subsystem
---x--x--x   lp      lp         /usr/sysv/bin/cancel
---s--x---   root    lp         /usr/sysv/bin/disable
---s--x---   root    lp         /usr/sysv/bin/enable
---x--x--x   lp      lp         /usr/sysv/bin/lp
-r-sr-xr-x   lp      lp         /usr/sysv/bin/lpq
-r-xr-xr-x   bin     bin        /usr/sysv/bin/lpr
-r-xr-xr-x   bin     bin        /usr/sysv/bin/lprm
---x--x--x   lp      lp         /usr/sysv/bin/lpstat
```

Links and files, which are only present when the System V print subsystem is active, appear as follows:

```
File Mode    Owner   Group      Pathname
==========   =====   =====      ===============================================
lrwxrwxrwx   root    system     /usr/bin/cancel -> /usr/sysv/bin/cancel
lrwxrwxrwx   root    system     /usr/bin/lp -> /usr/sysv/bin/lp
lrwxrwxrwx   root    system     /usr/bin/lpq -> /usr/sysv/bin/lpq
lrwxrwxrwx   root    system     /usr/bin/lpr -> /usr/sysv/bin/lpr
lrwxrwxrwx   root    system     /usr/bin/lprm -> /usr/sysv/bin/lprm
lrwxrwxrwx   root    system     /usr/bin/lpstat -> /usr/sysv/bin/lpstat
lrwxrwxrwx   root    system     /usr/bin/disable -> /usr/sysv/bin/disable
lrwxrwxrwx   root    system     /usr/bin/enable -> /usr/sysv/bin/enable
[Created on the fly when switching to System V print subsystem]
-rwxrwx---   root    lp         /usr/sysv/_SYS5_print_subsystem    (AIX S5 mode)
```

4.11.5 User interface for AIX and System V print subsystems

In the current release of AIX 5L, the Web-based System Manager provides the graphical user interface that will be used for the most common functions of the System V print subsystem. For more advanced functions, or to use less common features, users and administrators have to rely on the command line interfaces.

Note

There are no SMIT menus for the System V print subsystem In the initial AIX 5L Version 5.0. The only exception to this is a menu that switches between the traditional AIX and the System V print subsystem on the POWER platform.

The System V print subsystem management tasks to be performed by the Web-based System Manager application include:

- Adding new printers or classes of printers (parallel, serial, remote, and network).
- Setting the default printer.
- Removing printers or classes of printers.
- Switching to AIX print subsystem.

The status information to be displayed by the Web-based System Manager application includes:

- Showing the default printer.
- Displaying the requests on the default printer.
- Displaying the printers defined on the system.
- Displaying the stopped printers on the system.
- Showing the printers that currently have problems.

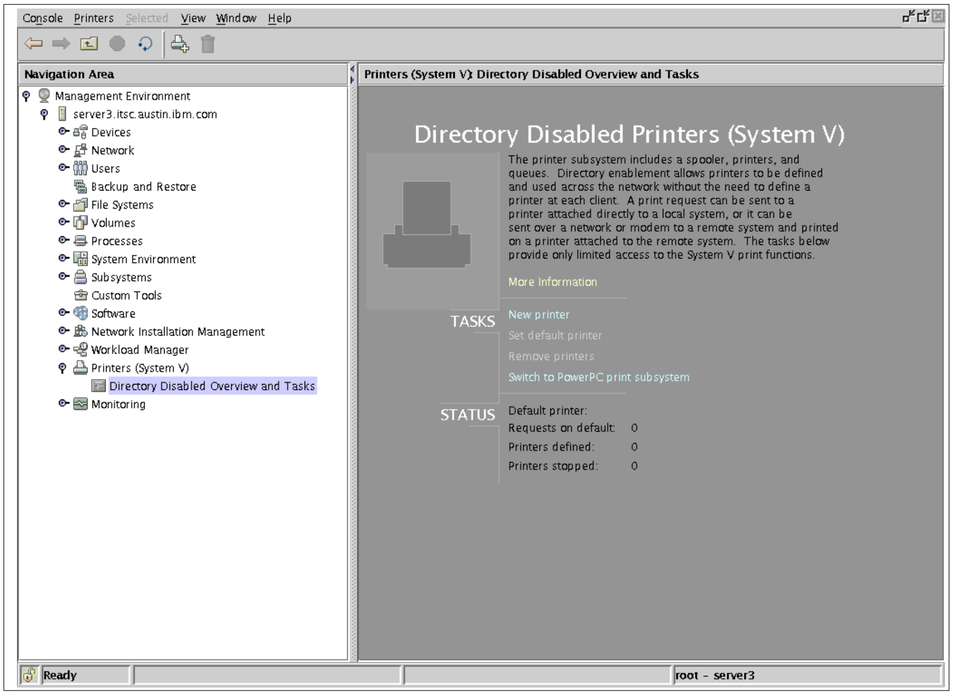
Before you can use the Web-based System Manager environment that supports System V printing, you have to switch from the AIX to the System V print subsystem. You can either utilize the `switch.prt -s SystemV` command, as described in Section 4.11.7, "Switching between AIX and System V print subsystems" on page 136, or use the following sequence of menu selections and operations with the Web-based System Manager tool:

Select--> **Printers** --> **Overview and Tasks**. Select the **Switch to System V print subsystem** task.

After the task has been completed, the Printer container icon is replaced by the Printers (System V) container icon. The Web-based System Manager environment for System V printing is now accessible through the following sequence of menu selections on the Web-based System Manager console:

Select --> **Printers (System V)** --> **Directory Disabled Overview and Tasks**.

Figure 37 on page 131 shows the Web-based System Manager menu for System V print subsystem management tasks.



Printers (System V): Directory Disabled Overview and Tasks

Directory Disabled Printers (System V)

The printer subsystem includes a spooler, printers, and queues. Directory enablement allows printers to be defined and used across the network without the need to define a printer at each client. A print request can be sent to a printer attached directly to a local system, or it can be sent over a network or modem to a remote system and printed on a printer attached to the remote system. The tasks below provide only limited access to the System V print functions.

More Information

TASKS

New printer

Set default printer

Remove printers

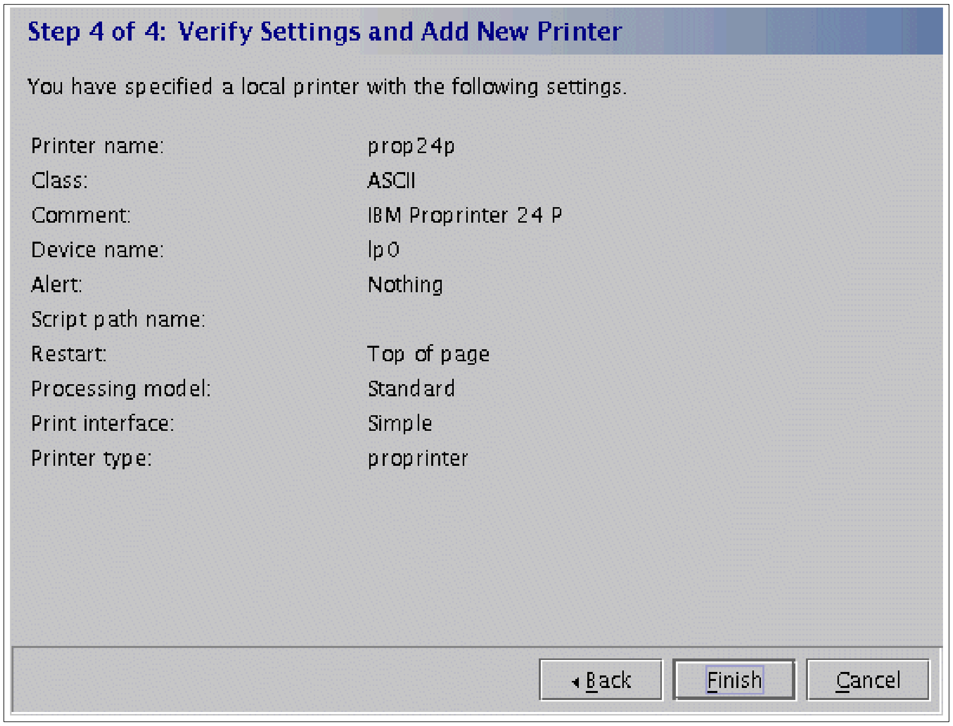
Switch to PowerPC print subsystem

STATUS

Default printer:
Requests on default: 0
Printers defined: 0
Printers stopped: 0

Figure 37. Web-based System Manager menu for System V print subsystem

If, for example, you would like to define a local print queue named prop24p for your predefined IBM Proprinter 24 P print device /dev/lp0, you select the **New printer** task and follow the instructions of the Add New Printer wizard. Figure 38 on page 132 shows Step 4 of 4: Verify Settings and Add New Printer window, which is displayed by the Add New Printer wizard just right before you have the option to complete the task by clicking on **Finish**. Note that the device support for the printer must be installed on the system and that the configuration for `lp0` must be completed before you engage in the System V print queue configuration. The printer type can be selected from the pull-down menu next to the field What is the printer type? in Step 3 of 4: Specify Printer Options wizard menu.



Step 4 of 4: Verify Settings and Add New Printer

You have specified a local printer with the following settings.

Printer name:	prop24p
Class:	ASCII
Comment:	IBM Proprinter 24 P
Device name:	lp0
Alert:	Nothing
Script path name:	
Restart:	Top of page
Processing model:	Standard
Print interface:	Simple
Printer type:	proprinter

Figure 38. Add New Printer Web-based System Manager wizard: Step 4 of 4

If the user-defined printer class ASCII does not already exist, it will be created during the final command execution of the Web-based System Manager wizard. Also, the final commands executed by the Web-based System Manager Add New Printer wizard allow the newly configured prop24p printer to accept (`accept` command) queuing requests and enable (`enable` command) the printer to print requests submitted by the `lp` command. The printer will not be defined as the system default print destination. If the user-defined class did not exist before, the wizard creates the class, but will not allow queueing of requests to the class as the print destination.

System administrators who prefer the command line interface to the System V print subsystem can configure the same print queue using the following command sequence:

```
# lpadmin -p prop24p -v /dev/lp0 -D "IBM Proprinter 24P" -c ASCII -I simple -m standard
   -T proprinter
# accept prop24p
# enable prop24p
```

The new printer can optionally be defined as the system default print destination and the /etc/host file may be submitted as the first test for the System V local print queue:

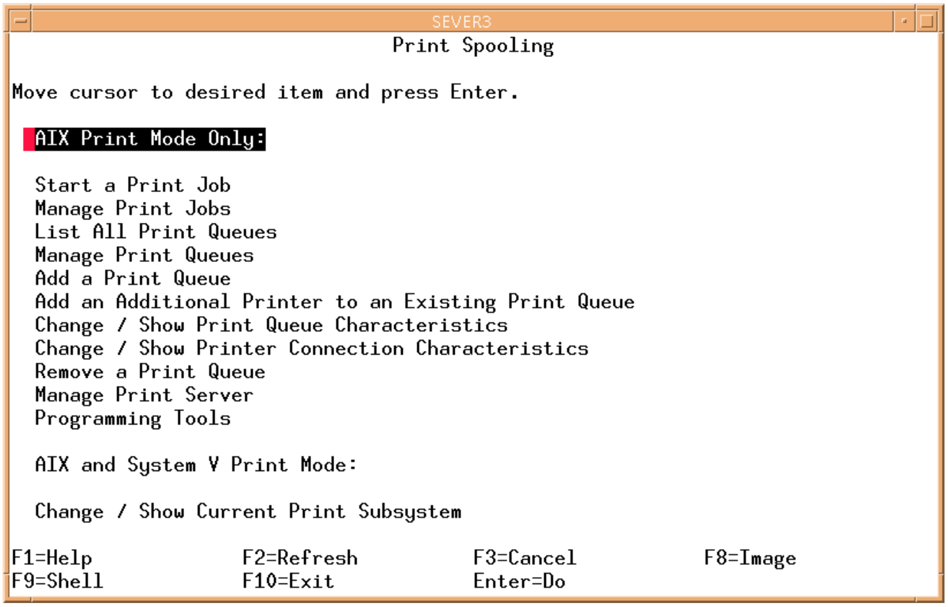
```
# lpadmin -d prop24p
# lp /etc/hosts
```

The `lpstat -t` command, entered immediately after the submission of the print request, gives comprehensive status information about the System V print subsystem:

```
# lpstat -t
scheduler is running
system default destination: prop24p
members of class ASCII:
        prop24p
device for prop24p: /dev/lp0
ASCII not accepting requests since Mon Sep 25 20:02:47 2000 -
        new destination
prop24p accepting requests since Mon Sep 25 20:03:08 2000
printer prop24p now printing prop24p-9. enabled since Mon Sep 25 20:03:15 2000.available.
prop24p-9              root              1439   Mon Sep 25 20:09:18 2000 on prop24p
```

It was previously mentioned that the System V print subsystem management tasks are currently not supported through the SMIT tool. However, some changes and additions have been made to account for the introduction of the System V print subsystem feature.

The Print Spooling menu of the SMIT tool was changed to show that most of the menu choices that now exist are only valid for the AIX Print Subsystem. The AIX print subsystem menu items will still be displayed if the System V print subsystem is active, but they will not work properly, since most of the underlying AIX print subsystem commands and daemons are turned off or disabled in some manner by the `switch.prt.subsytem` script during the switch from the AIX to the System V print subsystem. In addition, one new menu item has ben added at the bottom of the Print Spooling menu; it is valid for AIX and System V printing. The name of this item is Change/Show Current Print Subsystem and it can be used for either displaying the current running print subsystem or for changing from one to the other. Figure 39 on page 134 shows the new Print Spooling menu of SMIT.



```
SEVER3
                                   Print Spooling

Move cursor to desired item and press Enter.

  AIX Print Mode Only:

  Start a Print Job
  Manage Print Jobs
  List All Print Queues
  Manage Print Queues
  Add a Print Queue
  Add an Additional Printer to an Existing Print Queue
  Change / Show Print Queue Characteristics
  Change / Show Printer Connection Characteristics
  Remove a Print Queue
  Manage Print Server
  Programming Tools

  AIX and System V Print Mode:

  Change / Show Current Print Subsystem

F1=Help             F2=Refresh          F3=Cancel           F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 39. Print Spooling menu of SMIT

4.11.6 Terminfo and supported printers

Since System V printing depends heavily on extracting information from the terminfo database to configure and initialize printers, one file has been added which contains the terminfo definitions for all of the printers supported by this subsystem. The name of the file is svprint.ti, and it is located in the /usr/lib/terminfo directory. The file is compiled and stored in the respective terminfo directories at install time. The printers supported in the terminfo data base are listed in Table 9:

Table 9. Supported printers in the terminfo data base

- AP1337-e
- AP1337-i
- AP1339-e
- AP1339-i
- AP1357-e
- AP1357-i
- AP1359-e
- AP1359-i
- AP1371-e
- AP1371-i
- AP9210-i
- AP9210-lj
- AP9210-ljplt
- AP9215-d
- AP9215-e
- AP9215-i
- AP9215-lj
- AP9310-lj
- AP9312-lj
- AP9316-lj
- AP9415-lj
- PS
- PS-b
- PS-br
- PS-r
- bj-10ex
- bj-130e
- bj-200
- bj-300
- bj-330
- lq-870
- oki-320
- oki-390
- oki-ol400
- oki-ol800
- deskjet
- dfx-5000
- dfx-8000
- epl-7500
- fx-1050
- fx-850
- hplaserjet
- kx-p1124
- kx-p1180
- kx-p1624
- kx-p1695
- lq-1170
- lq-570
- paintjet
- proprinter
- unknown

Since many printers can be supported by the same terminfo file, the list of printers that are officially supported by System V printing is much larger. In addition, many printer manufacturers support their own printers for System V and send the support out with the printers. This greatly increases the total number. The list of manufacturers includes, but is not limited to the IBM Printer Division and Lexmark International. In later releases, more printers will be supported and shipped with AIX. The current list of supported printers is given in Table 10:

Table 10. Printer Support by the System V print subsystem in AIX 5L

Canon Bubble Jet 10ex
Canon Bubble Jet 130e
Canon Bubble Jet 200
Canon Bubble Jet 300
Canon Bubble Jet 330
Epson FX 850
Epson FX 1050
Epson DFX 5000
Epson DFX 8000
Epson LQ 570
Epson LQ 870
Epson LQ 1170
Epson EPL 7500
HP LaserJet (PCL)
HP LaserJet (Postscript)
HP LaserJet II (PCL)
HP LaserJet II (Postscript)
HP LaserJet III (PCL)
HP LaserJet III (Postscript)
HP LaserJet IIIsi (PCL)
HP LaserJet IIIsi (Postscript)
HP LaserJet 4 (PCL)
HP LaserJet 4 (Postscript)
HP LaserJet 4L/4ML (PCL)
HP LaserJet 4L/4ML (Postscript)
HP LaserJet 4P/4MP (PCL)
HP LaserJet 4P/4MP (Postscript)
HP LaserJet 4M/4M (PCL)
HP LaserJet 4M/4M (Postscript)
HP LaserJet 4Si/4Si MX (PCL)
HP LaserJet 4Si/4Si MX (Postscript)
HP LaserJet 4 Plus/4M Plus (PCL)
HP LaserJet 4 Plus/4M Plus (Postscript)
HP LaserJet 4V/4MV (PCL)
HP LaserJet 4V/4MV (Postscript)
HP LaserJet 5 (PCL)
HP LaserJet 5 (Postscript)
HP LaserJet 5L/5ML (PCL)
HP LaserJet 5L/5ML (Postscript)
HP LaserJet 5P/5MP (PCL)
HP LaserJet 5P/5MP (Postscript)
HP LaserJet 5Si/5Si MX (PCL)
HP LaserJet 5Si/5Si MX (Postscript)
HP LaserJet 5Si Mopier (PCL)
HP LaserJet 5Si Mopier (Postscript)
HP LaserJet 6P (PCL)
HP LaserJet 6P (Postscript)
HP LaserJet 6L (PCL)
HP LaserJet 6L (Postscript)
HP DeskJet 500
HP DeskJet 1200C/1200CPS
HP DeskJet 1600C/1600CM
HP Paint Jet
IBM ProPrinter
Oki 320
Oki 390
Oki OL 400
Oki OL 800
Panasonic KX-P1180
Panasonic KX-P1695
Panasonic KX-P1124
Panasonic KX-P1624
PostScript (Serial)
PostScript (Parallel)
PostScript (Serial w/ page reversal)
PostScript (Parallel w/ page reversal)
Unisys AP1337 - Epson emulation
Unisys AP1337 - IBM emulation
Unisys AP1339 - Epson emulation
Unisys AP1339 - IBM emulation
Unisys AP1357 - Epson emulation
Unisys AP1357 - IBM emulation
Unisys AP1359 - Epson emulation
Unisys AP1359 - IBM emulation
Unisys AP1371 - Epson emulation
Unisys AP1371 - IBM emulation
Unisys AP9205 - IBM emulation
Unisys AP9205 - HP Laserjet emulation
Unisys AP9205 - HP Laserjet Plotter emulation
Unisys AP9210 - IBM emulation
Unisys AP9210 - HP Laserjet emulation
Unisys AP9210 - HP Laserjet Plotter emulation
Unisys AP9215 - Epson emulation
Unisys AP9215 - Diablo emulation
Unisys AP9215 - IBM emulation
Unisys AP9215 - HP Laserjet emulation
Unisys AP9310 - HP Laserjet emulation
Unisys AP9312 - HP Laserjet emulation
Unisys AP9316 - HP Laserjet emulation
Unisys AP9415 - HP Laserjet emulation
Other

4.11.7 Switching between AIX and System V print subsystems

The current default print subsystem on AIX is the traditional AIX print subsystem. The System V print subsystem is offered as an alternate method of printing. At install time, the AIX print subsystem will always be set as the active one, and System V will always be set as the inactive one. They can not both be set to the active state at the same time using the normal procedures. However, there is nothing to prevent an administrator from overriding this manually (at their own risk).

AIX provides a command, accessible through SMIT and the Web-based System Manager, which will allow a system administrator to display the current active print subsystem, and to switch between the active and inactive one. The command is intended to be executed only by the Web-based System Manager or SMIT, but will work from the command line with the proper permissions. That command, located in /usr/sbin, is `switch.prt [ -s print_subsystem] [ -d ]`. The valid values for the print_subsystem keyword are AIX and System V. Running the command with the `-d` flag will display the current print subsystem; if you do not specify any flag, a brief help message is displayed on the screen, as shown below:

```
# switch.prt
Usage:  [-s AIX | SystemV ] [-d]
 -s switches to AIX print system or SystemV print system.
 -d displays current subsystem.
```

For security reasons, the switch.prt command serves as a front-end to the script /usr/sbin/switch.prt.subsystem, which actually does the real work.

The basic logic of the script for switching from the traditional AIX to the System V print subsystem is outlined in the following paragraph. The tasks that have to be performed by switching in the reverse direction from the System V to the traditional AIX print subsystem are similar, and you are encouraged to examine the code of the original script.

```
# Switch from AIX to System V

# sflag indicates the print subsystem to be switch to
# and the internal variable PRINTSUBSYSTEM refers to
# the type of the currently active print subsystem

else if sflag = SystemV && PRINTSUBSYSTEM = AIX
     then if (active print jobs)
           then echo "All print jobs must be terminated
                       before you can switch to $PRINTSUBSYSTEM"
               exit 1
     else
           Stop qdaemon
           Stop writesrv
           Stop lpd

           Change the action field of the inittab entries for
           qdaemon, writesrv, lpd, and piobe to prevent the unwanted
           start of this subsystems at system boot.

           # The following disables the smit menus as much as
           # possible
           mv /usr/lib/lpd/pio/etc/*.attach files to *.attach.AIX

           # Change the lock files from AIX to System V
           rm /usr/aix/_AIX_print_subsystem
           touch /usr/sysv/_SYS5_print_subsystem

           #force System V links over the existing AIX links for the
           #duplicate commands between them
```

```
ln -sf /usr/bin/cancel -> /usr/sysv/bin/cancel
ln -sf /usr/bin/enable -> /usr/sysv/bin/enable
ln -sf /usr/bin/disable -> /usr/sysv/bin/disable
ln -sf /usr/bin/lp -> /usr/sysv/bin/lp
ln -sf /usr/bin/lpstat -> /usr/sysv/bin/lpstat
ln -sf /usr/bin/lpq -> /usr/sysv/bin/lpq
ln -sf /usr/bin/lpr -> /usr/sysv/bin/lpr
ln -sf /usr/bin/lprm -> /usr/sysv/bin/lprm

#remove symbolic links from the tcbck database
tcbck -d /usr/bin/cancel
tcbck -d /usr/bin/enable
tcbck -d /usr/bin/disable
tcbck -d /usr/bin/lp
tcbck -d /usr/bin/lpstat
tcbck -d /usr/bin/lpq
tcbck -d /usr/bin/lpr
tcbck -d /usr/bin/lprm

#add the new symbolic links to the tcbck database
tcbck -a /usr/bin/cancel symlinks=/usr/sysv/bin/cancel
tcbck -a /usr/bin/enable symlinks=/usr/sysv/bin/enable
tcbck -a /usr/bin/disable symlinks=/usr/sysv/bin/disable
tcbck -a /usr/bin/lp symlinks=/usr/sysv/bin/lp
tcbck -a /usr/bin/lpstat symlinks=/usr/sysv/bin/lpstat
tcbck -a /usr/bin/lpq symlinks=/usr/sysv/bin/lpq
tcbck -a /usr/bin/lpr symlinks=/usr/sysv/bin/lpr
tcbck -a /usr/bin/lprm symlinks=/usr/sysv/bin/lprm

#start lpsched
/usr/lib/lp/lpsched
echo System V Print Subsystem Started

#Update the inittab to start the System V Print Subsystem at system boot

exit 0
```

A closer examination of the switch.prt. subsystem script reveals that the /var/spool/lpd/qdir is probed for files with file names beginning with the letter *n* or *r*, which indicate the existence of pending print jobs. If the search yields a positive result, the script is terminated with an appropriate error message. Consequently, the method provided to switch from one print subsystem to the other does not migrate any pending print jobs.

If no pending print jobs could be identified, the system resource controller command `stopsrc` is used to stop the `qdaemon`, `writesrv`, `and lpd` daemons which control the AIX print subsystem. After that, the action field for the related inittab entries are changed by the `chitab` command from wait to off and the respective inittab entry for the piobe print subsystem backend process is treated in the same fashion.

For the time being, there are no SMIT menus provided to assist users and system administrators perform System V print subsystem related tasks. Therefore, the AIX print subsystem SMIT menus are not replaced by System

V specific entities, but merely hidden by appending the AIX suffix to the menu definition files in /usr/lib/lpd/pio/etc directory.

Since the operating system determines (by the name of the relevant lock file) the type of the active print subsystem, the script replaces the lock file /usr/aix/_AIX_print_subsystem (of the traditional AIX print subsystem) with the lock file /usr/sysv/_SYS5_print_subsystem (of the System V print subsystem).

In AIX 5L, the System V and BSD print commands are still in the /usr/bin directory, but are now either linked to the traditional AIX print command wrappers in the /usr/aix/bin directory or to the appropriate executables in the /usr/sysv/bin (if the System V print subsystem is active). Consequently, the switch.prt.subsystem forces the System V links to take precedence over the AIX links when the system administrator switches from the AIX to the System V print subsystem.

If the Trusted Computing Base (TCB) feature is installed on the system, additional measures have to be taken in order to preserve the integrity of the /etc/security/sysck.cfg TCB file definition data base. The `tcbck -d` command is used to remove the current symbolic links from the configuration during a switch, and the `tcbck -a` command adds the new symbolic link, including the proper user and group ownership attributes, to the file definition database. If the `tcbck` command audits the security state of the system by checking the installation of the files defined in /etc/security/sysck.cfg, no mismatch between the file attributes in the trusted computing base and the actual system configuration will be reported.

Finally, if the `lpsched` daemon is started, and if an entry for `lpsched` exists in inittab, then the related action state is changed from off to wait; otherwise, a new entry will be added after the cron entry.

4.12 Web-based System Manager for AIX 5L

The Web-based System Manager is enhanced in AIX 5L. This section provides an in-depth look at what has changed from previous versions.

Keep in mind that the discussion of AIX Version 4.3.3 and previous POWER platform editions in this section is only for historical reference. AIX 5L for Itanium-based systems benefit from all the enhancements made in previous POWER platform releases as the cumulative function was ported.

Note

For more information about AIX System Management, or the Web-based System Manager architecture and previous releases features, refer to *AIX Version 4.3 Differences Guide*, SG24-2014, third edition.

It is also possible to press F1 during a Web-based System Manager session, and the main help window will be displayed.

4.12.1 Web-based System Manager Architecture

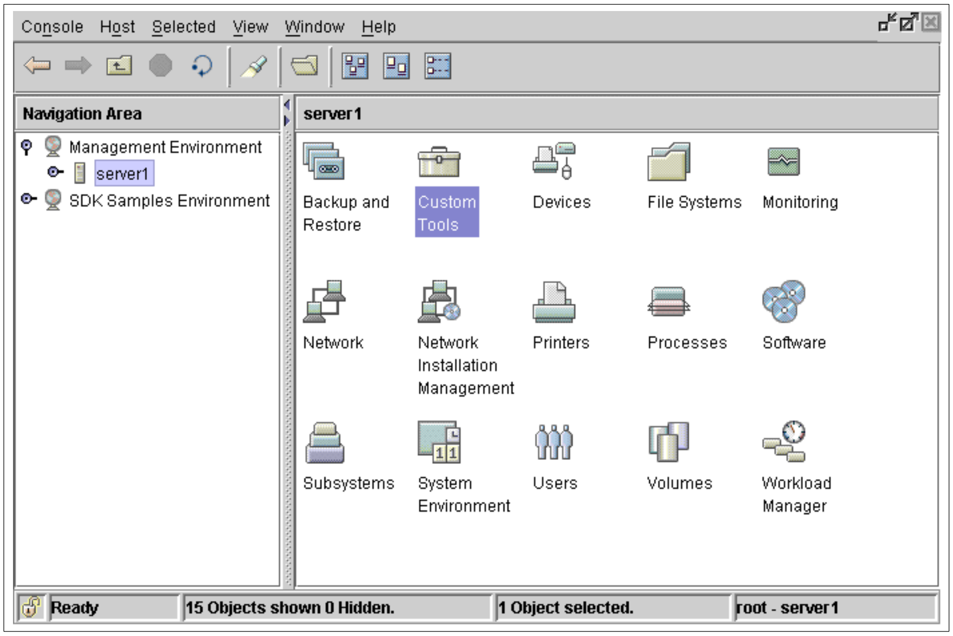
The Web-based System Manager enables a system administrator to manage AIX machines either locally from a graphics terminal or remotely from a PC or RS/6000 client. Information is entered through the GUI components on the client side. The information is then sent over the network to the Web-based System Manager server, which runs the necessary commands to perform the required action.

The Web-based System Manager is implemented using the Java programming language. The implementation of Web-based System Manager in Java provides:

- Cross-platform portability: Any client platform with a Java 1.3-enabled Web browser is able to run a Web-based System Manager client object.
- Distributed processing: A Web-based System Manager client is able to issue commands to AIX machines remotely through the network.
- Multiple launch points: The Web-based System Manager can be launched either in a Java Application Mode locally within the machine to manage both a local and remote system, through a Java Applet mode through a system with a Web browser with Java 1.3, and in Windows PC Client mode, where client code is downloaded from an AIX host.

4.12.1.1 User interface

The User Interface has improved noticeably; the console provides a convenient and familiar interface for managing multiple AIX hosts. The console window is divided into two panes: a Navigation Area, on the left, for displaying the hierarchy of host computers and management applications and a Contents Area, on the right, for displaying the contents of each level in the navigation hierarchy, as shown with the optional SDK Samples Environment installed in Figure 40.



Ready | 15 Objects shown 0 Hidden. | 1 Object selected. | root - server1

Figure 40. Web-based System Manager user interface

4.12.1.2 Plug-in architecture

As shown in Figure 40, the Navigation Area, on the left, has the host names of the servers to be administered, and each server contains a list of items that the Web-based System Manager can handle.

Each item contains a name and an icon. Each icon in this area is a *plug-in*. When the user selects a plug-in icon in the Navigation Area, the plug-in displays its contents in the Contents Area, updates the menu bar and tool bar with its actions, and updates the Tips Area with links for help on relevant tasks. Plug-ins are somewhat analogous to applications; they encapsulate a collection of management functions in the form of managed objects, collections of managed objects, tasks, and actions. A plug-in can consist of:

- An overview panel
- One or more sub plug-ins
- An overview and one or more sub plug-ins
- A collection of managed objects
- A panel for launching management interfaces in a window external to the console

The Web-based System Manager plug-in architecture is designed to provide a high degree of flexibility in the design of client applications. Both object and task-oriented plug-in models are provided, as well as the ability to integrate applications developed outside of the Web-based System Manager framework. The object-oriented design of the framework supports consistency across plug-ins while enabling the flexibility to extend and customize plug-in classes. The Web-based System Manager supports the classes of plug-ins discussed in the following sections.

Container

Container plug-ins are the most common type of plug-in used in the Web-based System Manager user interface. Container plug-ins are somewhat analogous to directories in a file system (or *folders* in a graphical file system manager). They contain other plug-ins, managed objects, or combinations of plug-ins and managed objects. Figure 41 shows a Container plug-in example.

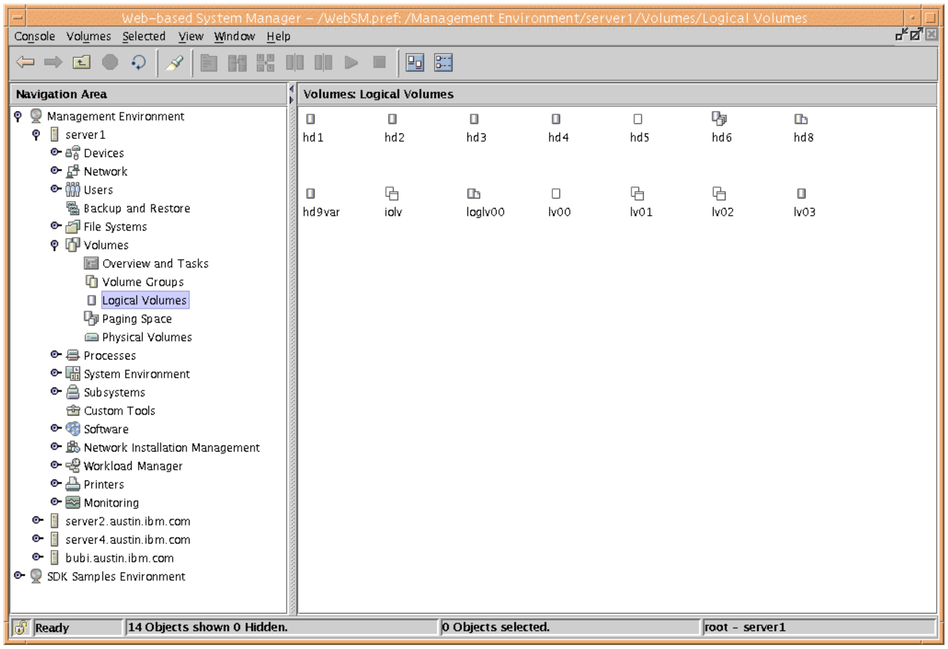
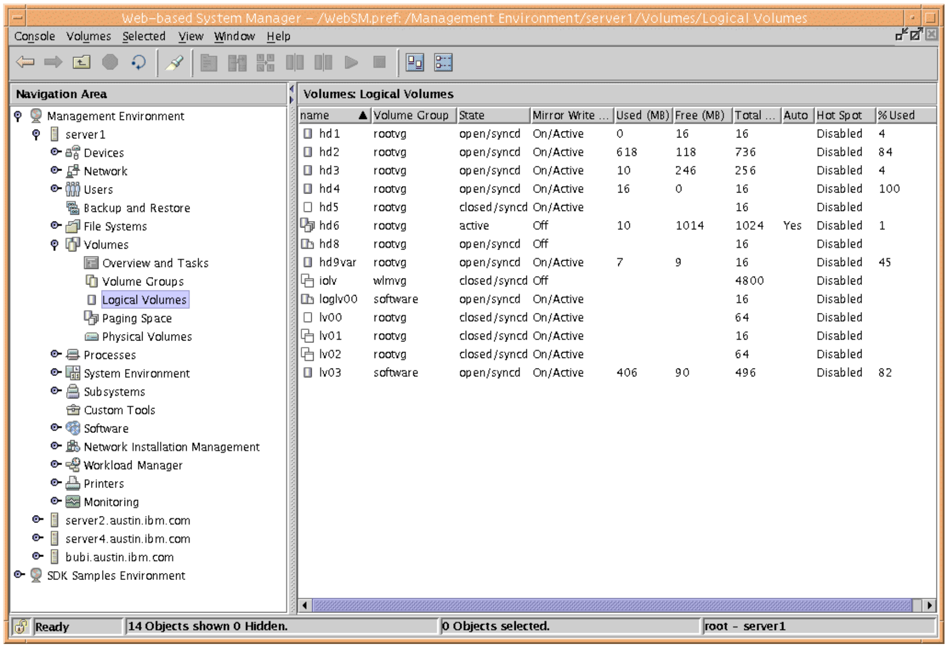


Figure 41. Container plug-in example

Containers present objects in views. The Web-based System Manager supports the typical object views (Large Icon, Small Icon, and Details), as well as two hierarchical views (Tree and Tree-Details). Figure 41 on page 143 shows an example of a Container plug-in used in the Large Icon view; and Figure 42 illustrates the detail view.



Volumes: Logical Volumes

name ▲	Volume Group	State	Mirror Write ...	Used (MB)	Free (MB)	Total ...	Auto	Hot Spot	% Used
hd1	rootvg	open/syncd	On/Active	0	16	16		Disabled	4
hd2	rootvg	open/syncd	On/Active	618	118	736		Disabled	84
hd3	rootvg	open/syncd	On/Active	10	246	256		Disabled	4
hd4	rootvg	open/syncd	On/Active	16	0	16		Disabled	100
hd5	rootvg	closed/syncd	On/Active			16		Disabled	
hd6	rootvg	active	Off	10	1014	1024	Yes	Disabled	1
hd8	rootvg	open/syncd	Off			16		Disabled	
hd9var	rootvg	open/syncd	On/Active	7	9	16		Disabled	45
iolv	wlmvg	closed/syncd	Off			4800		Disabled	
loglv00	software	open/syncd	On/Active			16		Disabled	
lv00	rootvg	closed/syncd	On/Active			64		Disabled	
lv01	rootvg	closed/syncd	On/Active			16		Disabled	
lv02	rootvg	closed/syncd	On/Active			64		Disabled	
lv03	software	open/syncd	On/Active	406	90	496		Disabled	82

Ready | 14 Objects shown 0 Hidden. | 0 Objects selected. | root – server1

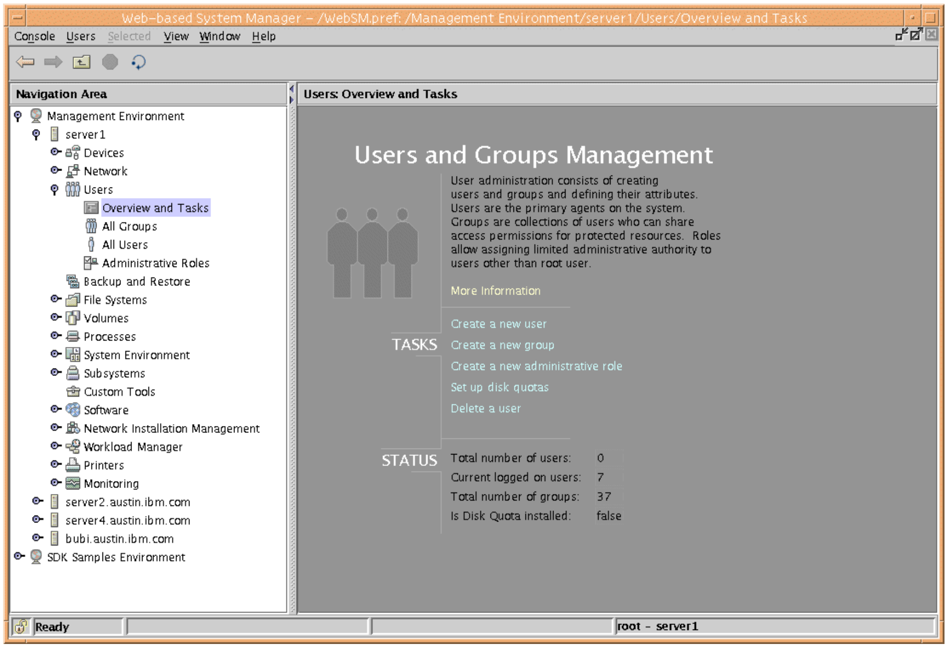
Figure 42. Example of Container, logical volumes container in detail view

Overview

Overview plug-ins are panel interfaces that appear in the contents area of a console child window. The primary functions of overviews are to:

- Explain the function provided by an application plug-in.
- Provide a launch point for routine or *getting started* tasks
- Summarize the status of one or more management functions.

In addition, because overviews are task-based rather than object based, they can be used to provide quicker and easier access to some functions than container views. In cases where a management function does not lend itself to an object-oriented design (for example, backup and restore), the entire application can be implemented using one or more overview plug-ins.



User administration consists of creating users and groups and defining their attributes. Users are the primary agents on the system. Groups are collections of users who can share access permissions for protected resources. Roles allow assigning limited administrative authority to users other than root user.

Total number of users:	0
Current logged on users:	7
Total number of groups:	37
Is Disk Quota installed:	false

Figure 43. Overview plug-in example, users and groups overview

Launch

Launch plug-ins serve as a mechanism for launching applications that were implemented outside of the Web-based System Manager framework. By using a launch plug-in, these *external* applications may be integrated into the Web-based System Manager console. The launch plug-in provides an overview-like panel with title, description area, a link to browser-based help, and a task link for launching the external application.

4.12.1.3 Standard plug-ins for Web-based System Manager

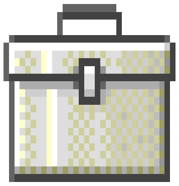
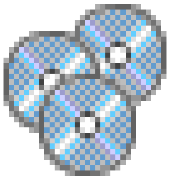
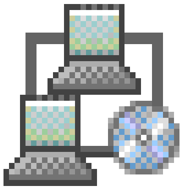
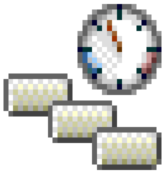
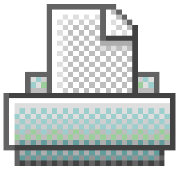
When you first run Web-based System Manager using the new graphical interface, keep in mind that all navigation is performed on the left side of the user interface.

Even if you have more than one server registered, each server will have standard plug-ins, as shown in Table 11.

Table 11. List of standard plug-ins in Web-based System Manager

Plug-In	Containers	Action
Devices	Overview and Tasks All Devices Communication Storage Devices Printers, Display Input Devices Multimedia System Devices	All hardware devices, related actions like add, remove, and change and show.
Network	Network Overview TCP/IP (IPv4 or IPv6) Point-to-Point (PPP) NIS NIS+ SNMP: included in AIX 5L. Virtual Private Networks	All network related actions, such as TCP/IP network, basic configuration, remove network interface, and NIS.
Users	Overview and Tasks All Groups All Users Administrative Roles	Users and Groups related actions, as well administrative roles for users authorization.
Backup and Restore	No containers, all options are located in the overview panel.	Performs actions related to backup, such as Image Backup, incremental backup, and restore.

Plug-In	Containers	Action
File Systems	Overview and Tasks Journaled File Systems Network File Systems Exported Directories CD-ROM File Systems Cache File Systems	All File Systems related tasks, such as add and remove a file system.
Volumes	Overview and Tasks Volume Groups Logical Volumes Paging Space Physical Volumes	All logical volume manager related actions, including Volume Groups and Physical Volumes, can be performed.
Processes	Overview and Tasks All Processes	Process related action, such as changing priority, kill a process, and list all process.
System Environment	Overview and Tasks Settings	System Environment will handle operations such as shutdown and broadcast messages, as well as licenses and Kerberos settings. License manager container is a new option.
Subsystems	Overview and Tasks All Subsystems	All subsystems related tasks can be done through this option, such as list, start, or kill a subsystem.

Plug-In	Containers	Action
Custom Tools 	No containers, just a Custom Tools helps icon. Additional icons will be added for each Custom Tool created.	Custom Tools allows you to integrate any command or Web application into Web-based System Manager.
Software 	Overview and Tasks Installed Software	All software related tasks, such as List and Install new software.
NIM 	Overview and Tasks	Network Installation Manager (NIM) can be set up from this option, as well as NIM administration.
Workload Manager 	Overview and Tasks Configurations/Classes Resources	All Workload Manager related tasks, such as Create class assignment rules, Update, and Stop Workload Manager. Incorporates all new enhancements for AIX 5L.
Printers 	Overview and Tasks All Printers	All printing related tasks, such as add a printer, remove a printer queue, and list all printers. Includes System V printing subsystem.

Plug-In	Containers	Action
Monitoring	Overview and Tasks Conditions Responses Events	All monitoring related tasks, such as create new conditions, list responses and events. It is a new option in Web-based System Manager.

A Security plug-in, not available with a default install, will be made available once you install the Expansion Pack. It is part of the base system, however.

4.12.1.4 Modes of operation

As in previous releases, the Web-based System Manager can be launched from a variety of launch points. For example:

- Java application mode through the `wsm` command in AIX command line on the system being managed
- Java application mode where the console is running on one AIX system, but managing remote systems. Called client-server mode.
- Management Console icon on CDE
- Java applet mode through Java 1.3-enabled Web browser
- Windows PC Client mode.

  The Windows PC client code is downloaded from an AIX host, then installed permanently on the PC. Because all the Java code is native on the PC, startup time and performance are exceptionally good compared to applet mode, and even compared to native AIX application mode (I don't know if that comment was appropriate for this book, but it's true...).

  The user can start Web-based System Manager PC Client in several ways.

  - Double click on the Web-based System Manager icon that was installed on the system desktop.
  - Select the Web-based System Manager entry in the Programs menu.
  - Locate the wsm.exe executable in Windows Explorer by changing to the install directory and double-clicking.
  - Change to the install directory within an MS-DOS window and type wsm.exe

This flexibility allows you to perform administrative tasks across multiple servers regardless where you perform them. From a mode of operation point of view, the Web-based System Manager can be managed from three different ways, as discussed in the following sections:

Local

AIX systems with a Graphical User Interface (GUI) can use this mode to perform local tasks. This mode is enabled by default.

Figure 44 shows the Management Console icon that starts the Web-based System Manager on CDE.

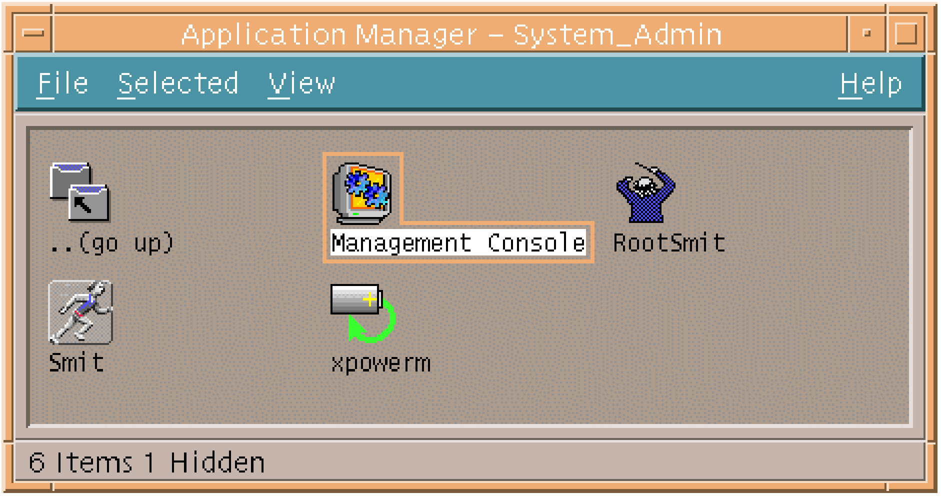


Figure 44. Web-based System Manager icon on CDE user interface

Client-server mode

The administrator can add hosts, represented by icons, to additional Internet-attached hosts in the Navigation Area of the console. The list of hosts and user interface preferences are stored in a console preferences file. The console preferences file can be stored on a specific host that will serve as the contact host or in a distributed file system (to allow it to be accessed directly from multiple hosts). When multiple hosts are set up to be managed from a single console, the Web-based System Manager operates in client-server mode. The first machine contacted by the client acts as the managing host while the other hosts in the navigation area are managed hosts.

Applet mode

In applet or browser mode, the administrator can manage one or more AIX hosts remotely from the client platform’s Web-browsers with Java 1.3. To access the console in this manner, an AIX host need only be configured with a Web-server (provided on the AIX Bonus or Expansion Pack CDs). Once the Web-server is installed and configured, the host can serve the console to the client. The administrator simply enters a URL, `hostname/wsm.html`, into the browser. A Web page is then served to the browser that prompts the user for a user name and password. Once authenticated to the server, the console launches into a separate window frame. In Web-based System Manager applet mode, the browser is used only for logging in and launching the console. Once running, the console is relatively independent of the browser.

4.12.2 Web-based System Manager Enhancements for AIX 5L

Table 12 provides a comparison list of new enhancements on the Web-based System Manager presented with AIX 5L.

Table 12. Comparison chart with the new enhancements

AIX Version 4.3	AIX 5L Version
Launch Pad and multiple windows	Management Console
Single host management	Point-to-Point multiple host management
Java 1.1	Java 1.3
Back end shell script execution	Shell script and API execution interface
Stateless User Interface	Dynamic User Interface
Session UI customization	Persistent UI preferences
AIX on POWER	AIX on POWER and Itanium-based systems
SSL security option	SSL security option
	Kerberos V5 integration in AIX
	Monitoring, notification, and control

4.12.2.1 Monitoring

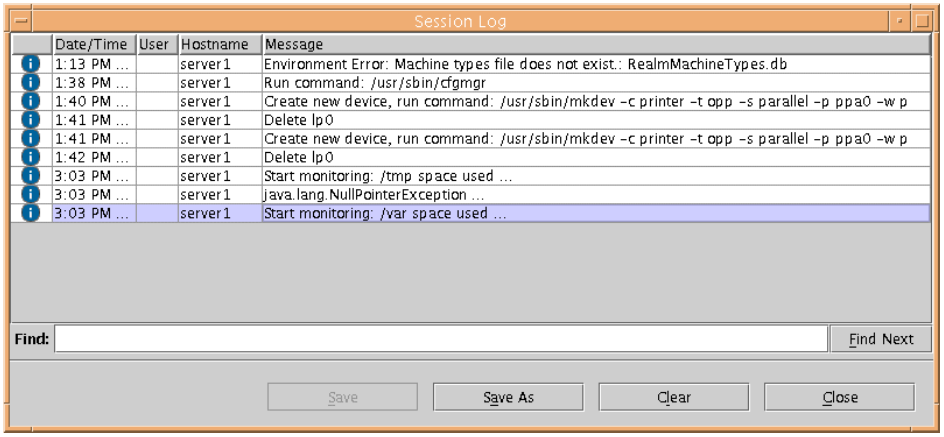
Refer to Section 4.4, “Resource Monitoring and Control (RMC)” on page 53 for monitoring details.

4.12.2.2 Session log

A new feature introduced in Web-based System Manager for AIX 5L is the Session Log. This log is located on the Console menu, and will log the following events:

- All actions performed in any managed host.
- Success or Failure messages.
- Security Level messages.

Figure 45 shows a sample output from a Session Log.



	Date/Time	User	Hostname	Message
ⓘ	1:13 PM ...		server1	Environment Error: Machine types file does not exist.: RealmMachineTypes.db
ⓘ	1:38 PM ...		server1	Run command: /usr/sbin/cfgmgr
ⓘ	1:40 PM ...		server1	Create new device, run command: /usr/sbin/mkdev -c printer -t opp -s parallel -p ppa0 -w p
ⓘ	1:41 PM ...		server1	Delete lp0
ⓘ	1:41 PM ...		server1	Create new device, run command: /usr/sbin/mkdev -c printer -t opp -s parallel -p ppa0 -w p
ⓘ	1:42 PM ...		server1	Delete lp0
ⓘ	3:03 PM ...		server1	Start monitoring: /tmp space used ...
ⓘ	3:03 PM ...		server1	java.lang.NullPointerException ...
ⓘ	3:03 PM ...		server1	Start monitoring: /var space used ...

Figure 45. An example of an output from a Session Log

When this log is opened, you will discover the following controls:

Find — Will search for a particular string or sentence among the messages already logged.

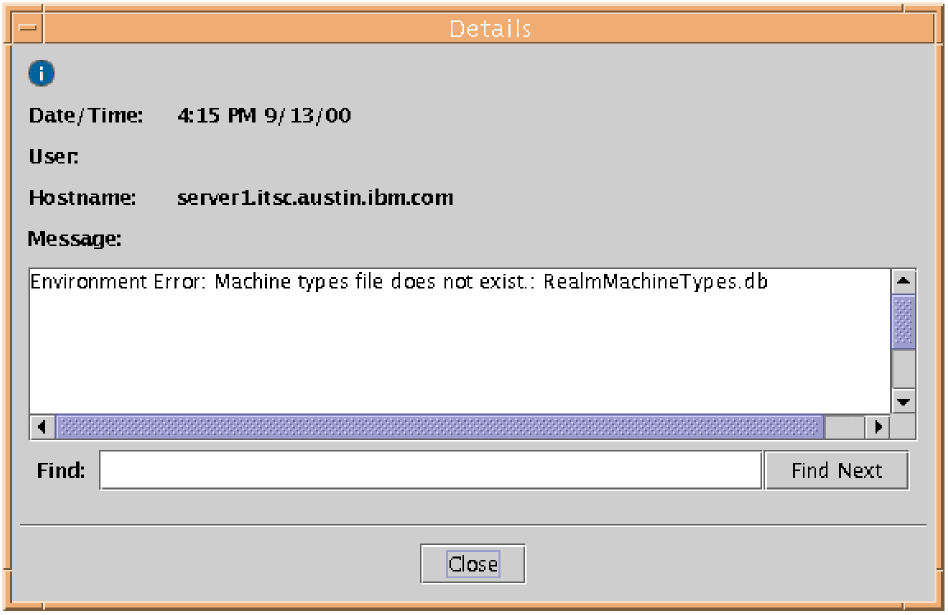
Save — Will save any new entry in the log table, and will append to the log file specified in the Save as option.

Save as — Will save all entries in the log table, and will store them in a new file, or will create the default file in /tmp/websm.log.

Clear — Will remove all entries in the log table.

Close — Will close the Session Log window.

If you double-click any entry in the log table, a new window will popup with detailed information on that specific entry. An example is shown in Figure 46.



Date/Time: 4:15 PM 9/13/00

User:

Hostname: server1.itsc.austin.ibm.com

Message:

Environment Error: Machine types file does not exist.: RealmMachineTypes.db

Figure 46. An example of Session Log detailed entry

4.12.2.3 Custom tools

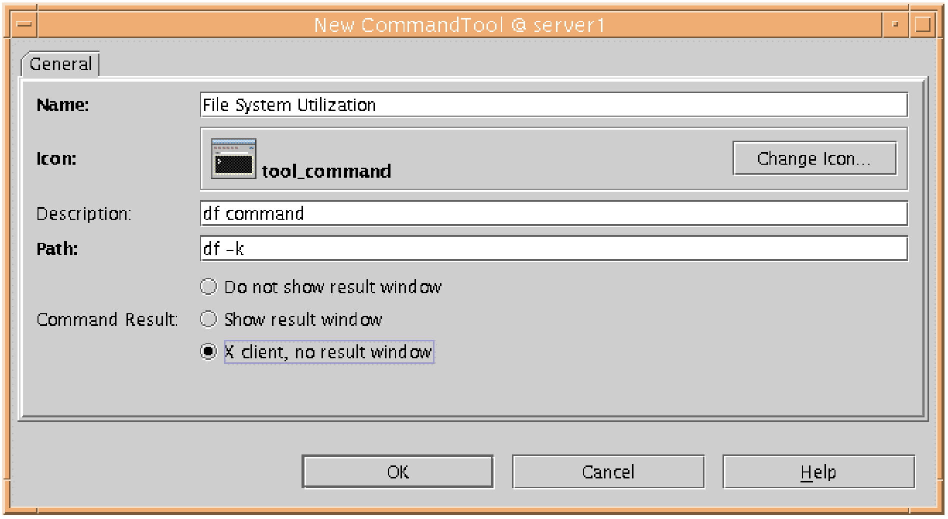
It is possible to integrate other administration applications into Web-based System Manager. Custom Tools extends the capabilities of the Registered Applications tool in previous releases. As before, URL-based applications can be added, but in addition, a new Command Tool option allows any tool that can be invoked through the command line to be integrated into Web-based System Manager.

There are two different types of Custom Tools:

- Web Tools, which are the URL-based applications to be integrated.
- Command Tools, which are the shell executable-based applications to be integrated.

The Web Tool acts exactly the same way as in previous the Web-based System Manager release.

Figure 47 on page 154 shows a screenshot of a Command Tool creation.



New CommandTool @ server1

General

Name: File System Utilization

Icon: tool_command

Description: df command

Path: df -k

Command Result:
○ Do not show result window
○ Show result window
● X client, no result window

Figure 47. Command Tool creation dialog

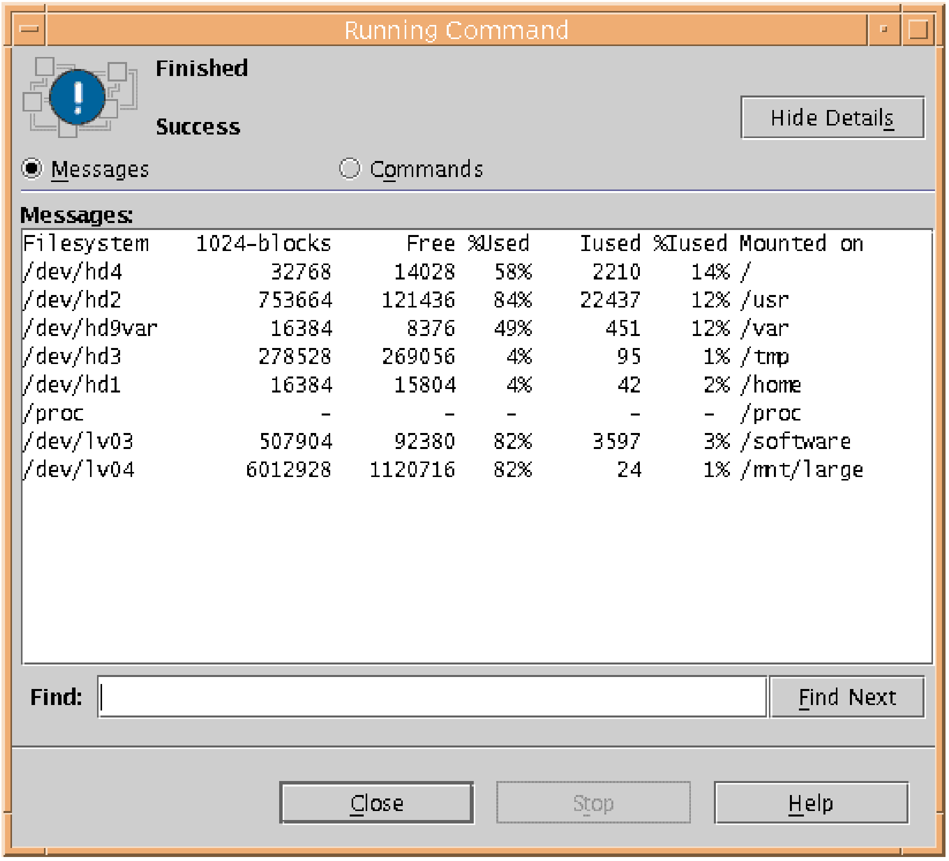
The Command Tool is a new option that allows you to integrate virtually any command line executable into the Web-based System Manager. To create a Command Tool, you need to specify the name of the Tool (a default icon is provided, but you can specify an alternate icon in GIF format), an optional description of the Tool, the complete path to the command, and a chosen result type. The result type can be one of the following:

Do not show the result window: Executes the command, but will not display the results of this command.

Show result window: Opens a new window with output generated by the specified command.

X client, no result window: The tool is an X client application. It will display its own GUI interface as the result window.

Figure 48 on page 155 shows the sample output of a Command Tool that chose Show result window as the result type.



```
Filesystem    1024-blocks      Free %Used    Iused %Iused Mounted on
/dev/hd4            32768     14028   58%     2210    14% /
/dev/hd2           753664    121436   84%    22437    12% /usr
/dev/hd9var         16384      8376   49%      451    12% /var
/dev/hd3           278528    269056    4%       95     1% /tmp
/dev/hd1            16384     15804    4%       42     2% /home
/proc                   -         -     -        -     -  /proc
/dev/lv03          507904     92380   82%     3597     3% /software
/dev/lv04         6012928   1120716   82%       24     1% /mnt/large
```

Figure 48. Example of result type Show result window

4.12.2.4 Tips area

Any container that you select on the Navigation Area will bring you tips on the related topic if *Show Tips Bar* is enabled. To enable it, you need to select **View** in the menu bar and then **Show**, and enable **Tips Bar**.

Figure 49 shows an example of a tip.

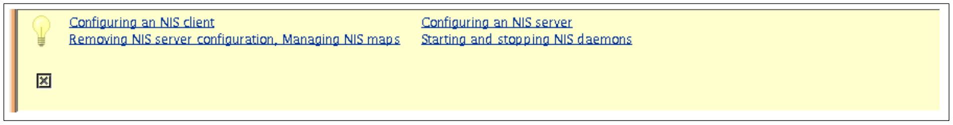

Configuring an NIS client
Removing NIS server configuration, Managing NIS maps
Configuring an NIS server
Starting and stopping NIS daemons
☒

Figure 49. Tips bar example

4.12.2.5 Preferences

In the AIX 5L release of the Web-based System Manager, it is possible to have a customized environment for any user in any machine for the Web-based System Manager. This can be done through the new control for preferences.

When the Web-based System Manager is started, the session uses the stored preferences. This includes such preferences as the console window format and the machines being managed. By default the preference file is saved to $HOME/WebSM.pref, which is the user's home directory on the managing machine.

To save the state of the console without closing a session, use the menu option **Console**, and then **Save**. A user is always prompted to save the console state when closing Web-based System Manager.

Table 13 shows which components are saved in the preferences file.

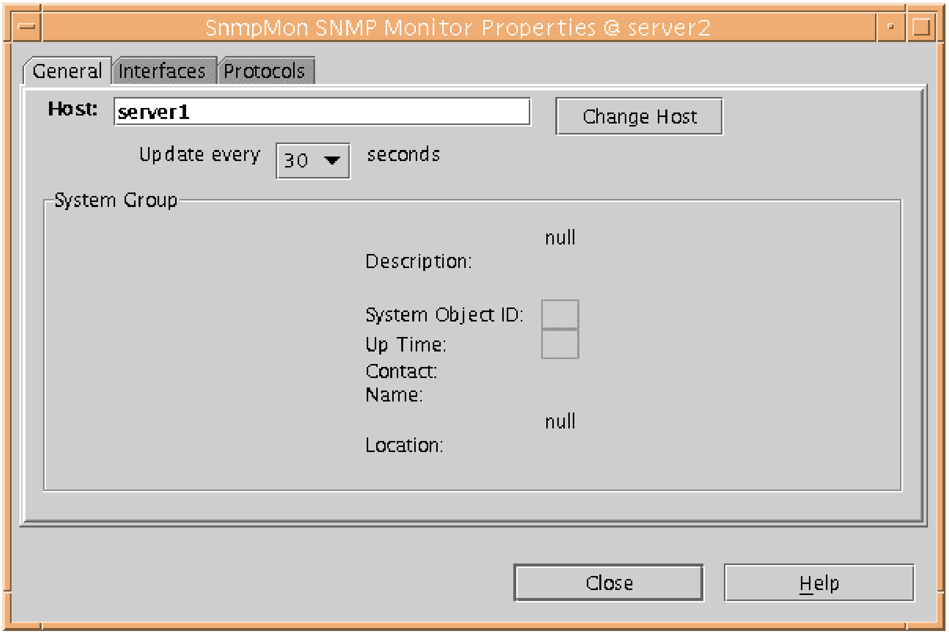
Table 13. Components that are saved in the preferences file

Component	Status saved in preferences file?
Navigation Area	No
Tool Bar	Yes
Tips Bar	Yes
Description Bar	Yes
Status Bar	Yes

4.12.2.6 SNMP integration

AIX 5L provides the SNMP interface for the Web-based System Manager framework for use by applications that need to do monitoring; it also provides overview query enhancements to Network applications.

Figure 50 on page 157 shows the panel for the SNMP monitor configuration.



Host: server1

Update every 30 seconds

Description: null

System Object ID:

Up Time:

Contact:

Name:

Location: null

Figure 50. SNMP Monitor configuration though Web-based System Manager

4.12.2.7 Enterprise management framework integration

In AIX 5L, there is a new way to launch the Web-based System Manager: it can be context launchable from the tool pallet and tool menu from Tivoli NetView NT and AIX.

In environments that already have Tivoli Netview server running, AIX 5L servers can be easily integrated and remotely managed through any Tivoli Netview servers launching the Web-based System Manager.

4.12.3 Accessibility for Web-based System Manager

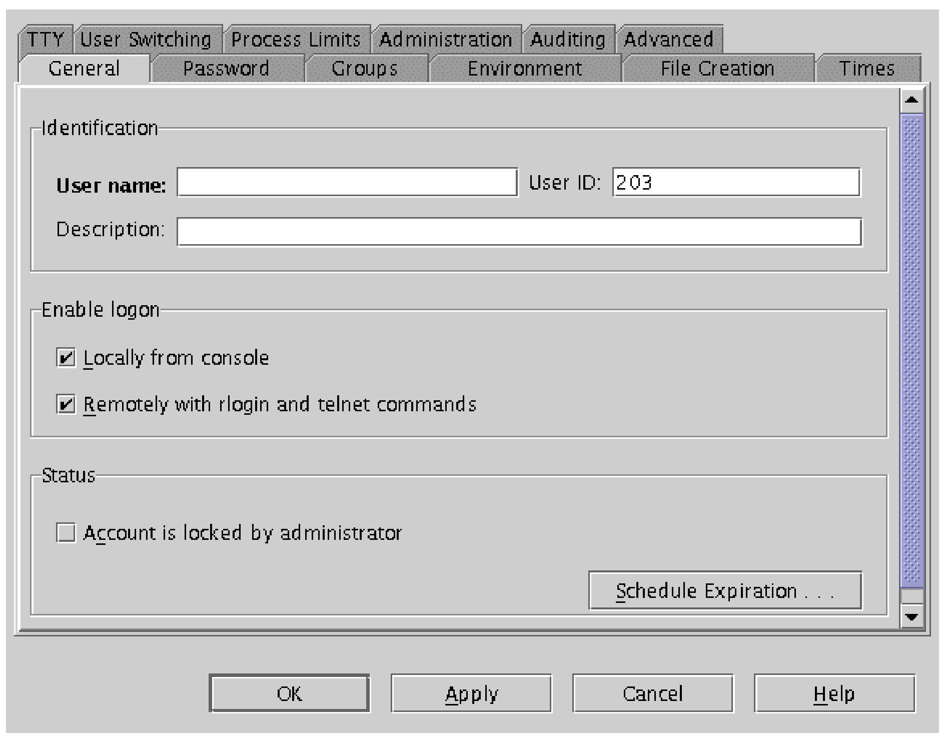
Because the Web-based System Manager in AIX 5L is using Java 2 Standard Edition 1.3, or more specifically the Java Foundation Classes, which are a default part of this version, you can now operate most of the panels, menus, screen controls, and dialogs without using a mouse or other pointing device.

Both limited mobility users and *power* users will welcome this function.

Two accessibility features are provided by default: mnemonics and accelerators. Mnemonics allow you to execute a certain action on a visible dialog without pressing the space bar or Enter key by simultaneously holding down the Alt key and the underlined letter designated in the label belonging to the desired action. Accelerators, on the other hand, are always available, even if the dialog or menu panel with the accompanying action is not visible. These accelerators or shortcuts are usually a combination of the Ctrl, Alt, or Shift key, or a combination of these with a regular letter key or special keys (such as Tab or function keys).

A Keys Help provides a complete list of navigation and windowing keys, and the mnemonics and accelerators for menus are shown in the user interface.

Figure 51 shows an example for the mnemonic key. In this example, pressing Alt-R selects the entry Remotely with `rlogin` and `telnet` commands in the Enable login group, regardless where the cursor is currently located. The Ctrl-Q key shortcut exits the Web-based System Manager, independent of which dialog is currently active.



User name:

User ID: 203

Description:

☑ Locally from console

☑ Remotely with rlogin and telnet commands

□ Account is locked by administrator

Figure 51. Accessibility example

4.13 User and group integration

In previous AIX releases, DCE and NIS were supported as alternate authentication mechanisms. AIX Version 4.3.3 added LDAP support and the initial support for specifying a loadable module as an argument for the user/group managing commands, such as `mkuser`, `lsuser`, `rmuser`, and so forth. But this was only generally documented in the /usr/lpp/bos/README file. AIX 5L is now offering a general mechanism to separate the identification and authentication of users and groups, and defines an application programming interface (API) that specifies what function entry points a module has to make available to be able to work as an identification or authentication method. This allows for more sophisticated customized login methods beyond what is provided by the standard ones based on /etc/passwd or DCE.

At the time of writing, DCE, LDAP, and Kerberos are not supported on Itanium-based systems.

4.13.1 Existing authentication methods

The standard AIX authentication method is a variant of the regular UNIX shadow password based implementation, meaning that the information about groups and their members is stored in the /etc/group file, information about users is stored in the /etc/passwd file (with the exception of the encrypted passwords), and related information, which is stored in /etc/security/passwd. This standard method is only implicitly defined and is therefore referred to by the name files when you have to distinguish it from other methods. Other authentication methods have to be explicitly defined in configuration files, as explained in the following section.

The information stored in the /etc/group and /etc/passwd files is called the basic attributes, while the information in the files in the /etc/security directory is called the extended attributes. The files in the /etc/security directory are AIX specific files, such as the /etc/security/user.roles that defines which roles a user can take. All the regular AIX commands that create groups or users, change their settings, or remove them are working with this set of files.

DCE, for instance, is an identification and authentication mechanism (in addition to the standard file method supported in AIX). This allows DCE users to be locally authenticated on an AIX system by specifying their DCE identity and password. For user and group management, you have to use the DCE specific commands; you cannot use the `mkuser` command, for example, to create a DCE user.

The setup for using this alternate authentication involves several steps. DCE uses a loadable binary module named /usr/lib/security/DCE. This module belongs to the dce.client.core.rte.security fileset. It handles the communication between user, local AIX commands, and the DCE servers. You can specify the full path to this module as a stanza with a freely chosen name as the value for the program attribute in the /usr/lib/security/methods.cfg file. If you choose the name *DCE*, the stanza appears as follows:

```
DCE:
        program = /usr/lib/security/DCE
```

Because there was no clear separation between user identification and authentication before AIX 5L, the name of this stanza is used for two different purposes.

- First, as a value for the registry attribute in the /etc/security/user file for either single specific users or in the default stanza. This informs AIX that this user is not locally managed, but managed by a remote mechanism.
- Second, to enable authentication through DCE. The primary authentication method is specified as the value of the auth1 attribute in the /etc/security/user file and has the default value SYSTEM. It is also possible to have a secondary authentication method specified with the auth2 attribute, but this is rarely used. The default value for SYSTEM is “compat,” which is an abbreviation for the combination of the standard AIX mechanism (files) and NIS. To enable authentication using DCE, override the value of the SYSTEM attribute, for example, with the following statement:

  ```
  SYSTEM = "DCE OR DCE[UNAVAIL] AND compat"
  ```

When a user tries to login to an AIX system with this setting for a user ID, the user ID and password are automatically handed over to the loadable module specified as the value of the program attribute of the DCE stanza in /usr/lib/security/methods.cfg. This module checks with the DCE servers, if the user ID and password combination is valid. If it is, the user is authenticated locally in the AIX system and obtains DCE credentials. If this fails due to the unavailability of DCE, not because of a wrong password, the next step is to check if this user ID and password combination is a locally valid one. If it is, the user is authenticated locally, but has no DCE credentials. If it fails, the user receives the message that either a wrong user ID or wrong password was used. There is a defined grammar which describes to specify the order of authentication modules to try, and what actions to take if one of them fails or is unavailable.

If you set the registry attribute to DCE, to indicate that the DCE loadable module is responsible for managing the user IDs, and use the `lsuser` command to see the attributes for a specific user, you will miss some of the attributes, such as unsuccessful_login_count or roles. Some attributes are not even listed and some of them are listed but without their values. If you want to see or reset the value for the unsuccessful_login_count of a user, you have to temporarily switch the registry attribute back to files. Starting with AIX Version 4.3.3, several user and group managing commands now support an optional -R flag, which specifies the loadable module used for accessing the user and group attributes. The commands supporting the -R flag are:

- `chfn`
- `chgroup`
- `chgrpmem`
- `chsh`
- `chuser`
- `lsgroup`
- `lsuser`
- `mkgroup`
- `mkuser`
- `passwd`
- `rmgroup`
- `rmuser`

4.13.2 Identification and authentication architecture

In AIX 5L, support for loadable identification and authentication modules is now fully documented and enhanced, in comparison to the feature already available with AIX Version 4.3.3. The tasks of user identification and user authentication are now clearly separated and can be executed by two different loadable modules.

User identification comprises all the necessary information about what user IDs exist and what the attributes for these user IDs are. This information must be consistent, so some kind of database must be used. This database can be flat file based, such as the regular /etc/passwd mechanism, or it can be a relational database, such as DB2, as in the case of IBMs LDAP implementation.

User authentication, on the other hand, is a transitory process where a user claims to have a certain identity and the system has to check if this is true or not. For this process, the system requires a unique piece of information about this user (usually a password). When the user authenticates, the system challenges them by requesting that they type in their password. The user's response is then compared to the stored unique piece of information and, depending on the outcome of this comparison, the request is accepted or denied. This information, which uniquely identifies a user, must also be stored permanently, but it does not necessarily have to be in the same database where the user identification is stored. With this separation of identification and authentication, and the definition of an API, the architecture in AIX exists to support authentication methods that are far more sophisticated than the usual password-based mechanism.

AIX 5L now supports loadable modules that are either responsible for identification, for authentication, or both, (as already supported in the past). For a fully supported login process, you need both identification and authentication as well. You can use either one loadable module, which supports both (as in the past), or you can specify one loadable module, one of which is responsible for the identification part and one of which is responsible for authentication. Such a combination of two modules is called a compound module.

To support this new feature, the stanzas in the /usr/lib/security/methods.cfg file now accept the attributes domain and option in addition to the already supported program and program_64 attributes. With the optional domain attribute, you can specify an arbitrary text string that is passed as is to the loadable module. The module can use this string for whatever purposes it likes, but usually it is used to distinguish between several supported domains. The options attribute also takes an arbitrary text string, consisting of comma separated values or name/value pairs, which is then passed to the loadable module as is. There are some predefined values which are interpreted by the AIX system itself. You can specify either authonly or dbonly to indicate that this module is only responsible for the authentication or the identification part. To connect a single purpose module with a specific module for the complementary part of the identification and authentication process, you can use the db=<module> or auth=<module> options.

For example: suppose you want to configure a system to use LDAP for user identification and DCE for user authentication. You have to create, at minimum, two stanzas in the /usr/lib/security/methods.cfg file which specify these two programs. The following provides an example:

```
DCE:
        program = /usr/lib/security/DCE
        options = authonly

LDAP:
        program = /usr/lib/security/LDAP
        options = auth=DCE
```

With this setting you can, for example, specify LDAP as the value for the registry attribute. For identification purposes, the LDAP load module would be used and as soon as authentication is needed, the module specified in the DCE stanza would be used. You can create the same effect with the following three stanzas:

```
DCE:
        program = /usr/lib/security/DCE
        options = authonly

LDAP:
        program = /usr/lib/security/LDAP

LDAPDCE:
        options = auth=DCE,db=LDAP
```

In this case, you would specify LDAPDCE as the value of the registry attribute. This would allow for other possible authentication modules to be used in conjunction with LDAP identification. Stanza names can only be used in other stanzas if they have been previously defined.

In AIX 5L, programming interfaces have been documented which describe what function calls a loadable module has to support if it wants to handle the identification part or the authentication part. There are also a couple of support and administrative function calls that handle the internal table that tracks pointers to all available authentication and identification modules that must be opened and closed.

If you are using user or group accounting commands, such as `lsuser` without using the -R flag, information from all defined identification load modules is displayed. Therefore, a user ID may be listed twice if it is defined for two modules. The displayed attributes can also be different, because not all attributes have to be supported by all modules. Values for attributes defined for more than one module are shown as set for the first loaded module (this is often the implicitly defined standard files module). To avoid confusion, it is recommended to always supply a name for a specific load module using the -R flag.

4.13.3 Native Kerberos Version 5 support

AIX 5L includes native Kerberos Version 5 support which can be used as an authentication loadable module, as described in the Section 4.13.2, “Identification and authentication architecture” on page 161. If you use the Kerberos Version 5 authentication method as the default login method, a user will automatically acquire appropriate credentials after a successful login. This support has to be installed separately and is provided in the following filesets:

```
# lslpp -L "krb5*"
  Fileset                      Level  State  Description
  ----------------------------------------------------------------------------
  krb5.client.rte            1.1.0.0    C    Network Authentication Service
                                             Client
  krb5.client.samples        1.1.0.0    C    Network Authentication Service
                                             Samples
  krb5.doc.en_US.html        1.1.0.0    C    Network Auth Service HTML
                                             Documentation - U.S. English
  krb5.doc.en_US.pdf         1.1.0.0    C    Network Auth Service PDF
                                             Documentation - U.S. English
  krb5.msg.en_US.client.rte  1.1.0.0    C    Network Auth Service Client Msgs
                                             - U.S. English
  krb5.server.rte            1.1.0.0    C    Network Authentication Service
                                             Server
  krb5.toolkit.adt           1.1.0.0    C    Network Authentication Service
                                             App. Dev. Toolkit
```

The executables and documentation are installed in the /usr/krb5 directory; configuration files, logs, and other changing files are in the /etc/krb5 and /var/krb5 directories. This avoids any mix-up with an already existing Kerberos installation (for example, from DCE).

The only exception are the files and links put into /usr/sbin, as shown in the following partial directory listing:

```
# ls -l /usr/sbin/*krb*
lrwxrwxrwx   1 root     security         26 Sep 13 08:45 /usr/sbin/config.krb5 ->
/usr/krb5/sbin/config.krb5
-r-x------   1 root     security       8119 Aug 23 12:33 /usr/sbin/mkkrb5clnt
-r-x------   1 root     security       8648 Aug 23 12:33 /usr/sbin/mkkrb5srv
-r-x------   1 root     security      13864 Aug 24 22:41 /usr/sbin/mkseckrb5
lrwxrwxrwx   1 root     security         25 Sep 13 08:45 /usr/sbin/start.krb5 ->
/usr/krb5/sbin/start.krb5
lrwxrwxrwx   1 root     security         24 Sep 13 08:45 /usr/sbin/stop.krb5 ->
/usr/krb5/sbin/stop.krb5
lrwxrwxrwx   1 root     security         28 Sep 13 08:45 /usr/sbin/unconfig.krb5 ->
/usr/krb5/sbin/unconfig.krb5
```

The configure, unconfigure, start, and stop scripts are only here for convenience, so you do not have to type the complete path to these commands. The `mkkrb5srv` command sets up an Kerberos V5 server and the `mkkrb5clnt` command sets up a Kerberos V5 client. Finally, the `mkseckrb5` command migrates existing users from the default authentication method to the Kerberos V5 method.

To make this setup work, the `hostname` command should provide a full, qualified host name, as shown in the following line:

```
# hostname
server1.itsc.austin.ibm.com
```

Note

If your `hostname` command only outputs a short name without the domain name the setup will not work, because only a principal for the short name will be created. The request from the client, where a user wants to login with the Kerberos method, coming over the network will always be the conjunction of the short hostname and the domain name and no principal exists for this situation.

The first step in this setup is to create a Kerberos server. To accomplish this task, use the `mkkrb5srv` command, specifying the flags as shown in the following example:

```
# mkkrb5srv -r DG.itsc.austin.ibm.com -s server1.itsc.austin.ibm.com -d
itsc.austin.ibm.com -a admin/admin
```

The flags used specify a realm with the -r flag (which is a free form string), the server name with the -s flag, and a domain with the -d flag. If you do not specify an admin principal with the -a flag, the default is admin/admin. These commands create the /etc/krb5/krb5.conf file and some other configuration files in the /var/krb5/krb5kdc directory. If these configuration files already exist, they are not modified by this command. Several default principals that manage the Kerberos environment will also be created. The command will also add two entries to the /etc/inittab file as shown in the following example output:

```
krb5kdc:2:once:/usr/krb5/sbin/krb5kdc
kadm:2:once:/usr/krb5/sbin/kadmind
```

These two daemons are also started by the `mkkrb5srv` command. The kadmind daemon is the administration daemon and the krb5kdc is the actual Key Distribution Center (KDC) daemon, which is responsible for the creation of the secret keys. During the setup process, you are prompted to provide passwords for various principals. You should make note of them, because they are needed in further steps of this setup.

On any machine where you want to use the Kerberos authentication method, you have to run the `mkkrb5clnt` command with several flags. An example is shown in the following line:

```
# mkkrb5clnt -r DG.itsc.austin.ibm.com -c server1.itsc.austin.ibm.com -s
server1.itsc.austin.ibm.com -d itsc.austin.ibm.com -a admin/admin -A -i
files -K -T
```

The meanings of the -r, -d, and -a flags are the same as described previously for the `mkkrb5srv` command. The -c and -s flag specify the host where the kadmind and the KDC daemon are running. The -i flag with the files argument specifies the integrated login, and the -K flag makes Kerberos the default authentication method. The -A flag makes root an administrator for Kerberos on this machine. Finally, the -T flag requests a Ticket-Granting Ticket (TGT) from the server. This creates a keytab file in the /var/krb5/security/keytab directory and the /etc/krb5/krb5.conf configuration file. The last step is omitted if you create the client on the same machine you created the server on, because this file already exists in this case. The command also creates the following two entries in the /usr/lib/security/methods.cfg file:

```
KRB5:
        program = /usr/lib/security/KRB5

KRB5files:
        options = db=BUILTIN,auth=KRB5
```

The last entry is used to modify the SYSTEM attribute of the default stanza in the /etc/security/user file to read:

```
default:
        SYSTEM = "KRB5files OR compat"
```

With this setting, Kerberos is tried, as a first step, as the authentication method; if this fails, the regular AIX method is tried.

After being authenticated with the `/usr/krb5/bin/kinit` command, root can create users residing in the KRB5files domain. The following example commands can be used to create a user krb5user and to set an initial password (it is recommended that you use a more secure password):

```
# mkuser -R KRB5files krb5user
# passwd -R KRB5files krb5user
```

The output of the `lsuser` command shows all the Kerberos attributes, beginning with krb5_, defined for this user in addition to the regular AIX user attributes:

```
# lsuser -R KRB5files krb5user
krb5user id=202 pgrp=staff groups=staff home=/home/krb5user
shell=/usr/bin/ksh login=true su=true rlogin=true daemon=true admin=false
sugroups=ALL admgroups= tpath=nosak ttys=ALL expires=0 auth1=SYSTEM
auth2=NONE umask=22 registry=KRB5files SYSTEM=KRB5files or compat
```

```
logintimes= loginretries=0 pwdwarntime=0 account_locked=false minage=0
maxage=0 maxexpired=-1 minalpha=0 minother=0 mindiff=0 maxrepeats=8
minlen=0 histexpire=0 histsize=0 pwdchecks= dictionlist= fsize=2097151
cpu=-1 data=262144 stack=65536 core=2097151 rss=65536 nofiles=2000
time_last_login=0 time_last_unsuccessful_login=0 tty_last_login=/dev/pts/4
host_last_login=server1.itsc.austin.ibm.com unsuccessful_login_count=0
roles= krb5_principal=krb5user@DG.itsc.austin.ibm.com
krb5_principal_name=krb5user@DG.itsc.austin.ibm.com
krb5_realm=DG.itsc.austin.ibm.com maxage=0 expires=0
krb5_last_pwd_change=968878232 admchk=false
krb5_attributes=requires_preauth
krb5_mod_name=krb5user@DG.itsc.austin.ibm.com krb5_mod_date=968878232
krb5_kvno=4 krb5_mkvno=0 krb5_max_renewable_life=604800 time_last_login=0
time_last_unsuccessful_login=0 unsuccessful_login_count=0
krb5_names=krb5user:server1.itsc.austin.ibm.com
```

The new user can `telnet` to the client machine and login with the password just set up. After a successful login, the user environment has the following settings:

```
AUTHSTATE=KRB5files
KRB5CCNAME=FILE:/var/krb5/security/creds/krb5cc_krb5user@DG.itsc.austin.ib
m.com_202
```

These settings show that the user is authenticated using the KRB5files method and the path to the credentials file.

With the help of the `mkseckrb5` command, you can migrate a user existing in the files domain to the KRB5files domain. The following lines show an example session for a user krb5eins:

```
# mkseckrb5 krb5eins
Please enter the admin principal name: admin/admin
Enter password:
Importing krb5eins
Enter password for principal "krb5eins@DG.itsc.austin.ibm.com":
Re-enter password for principal "krb5eins@DG.itsc.austin.ibm.com":
```

If you do not want to enter the password twice for the migrated user, you can use the -r flag, which creates a random password for you. You can then use the `passwd` command to set a password for this user.

4.14 IBM SecureWay Directory Version 3.2

Version 3.2 of the IBM SecureWay Directory implements the Lightweight Directory Access Protocol (LDAP) Version 3.2 and is offered with the AIX operating system product at no additional charge.

At the time of writing, this feature is only available on the POWER platform.

The IBM SecureWay Directory Version 3.2 consists of the following components:

- slapd: the server executable
- Command line import/export utilities
- A server administration tool with a web-browser based interface for configuration and administration of the directory
- A Java-based directory content management tool and online user guide
- Online Administration Helps
- Online LDAP Programming References (C, Server Plug-ins, and Java/JNDI)
- SecureWay Directory Client Software Development Kit (SDK) that includes C runtime libraries and Java classes

The product includes a Lightweight Directory Access Protocol (LDAP) Version 3 server that supports IETF LDAPv3 (RFC 2251) protocol, schema, RootDSE, UTF-8, referrals, Simple Authentication and Security Layer (SASL) authentication mechanism and related specifications. In addition, it includes support for Secure Socket Layer (SSL), replication, access control, client certificate authentication, CRAM MD5 authentication, change log, password encryption, server plug-ins, enhanced search capability for compound Relative Distinguish Name (RDN), web-based Server Administration, LDAP V3 schema definitions, IBM common schema definitions, schema migration and performance improvements.

With over 18 major product enhancements, Version 3.2 of the IBM SecureWay Directory represents one of the most significant updates of the product to date. Some of the more significant enhancements and new functions and features include:

- Fine grain access control - attribute level ACLs

  The IBM SecureWay Directory now allows the management of access down to the individual attribute level. A directory administrator may now control who may see individual attributes for each entry within the directory. This allows access to be managed on individual attribute level which gives a much finer control. Fine grain access control is often used when specific attributes need to be managed by an entry owner and other entry attributes are managed by the directory administrator.

- Unlimited Connections - improved server threading model

  The IBM SecureWay Directory has proven to be a performance leader. To sustain and further enhance the striking performance of the product the threading model for the directory has been improved. The IBM SecureWay Directory will now utilize thread pools, thus reducing the number of threads utilized when many clients connect to the server concurrently. This change will allow a much larger number of clients to connect to a server, which in turn reduces the number of servers required in a given LDAP environment.

- Support for Kerberos V5 (server and client, including C and JNDI) - GSSAPI

  The IBM SecureWay Directory now supports authentication utilizing Kerberos V5. Kerberos V5 has become an important authentication method. Supporting Kerberos V5 authentication methods improves the ability of the directory to provide a single authentication method across the enterprise.

The SecureWay Directory Client SDK includes a Java-based Directory Management Tool, APIs to locate LDAP servers that are published in DNS, client-side caching for the Java-based JNDI interface, as well as other JNDI enhancements.

LDAP is a new technology that is rapidly evolving. IBM is committed to deliver the latest LDAP technology achievements in the robust high-performance LDAP server implementation of the IBM SecureWay Directory product. Version 3.2 of the IBM SecureWay Directory not only keeps pace with the industry but provides many industry-leading innovations as documented by the list of improvements given below:

- Performance improvements through Table Reduction (for Fast Server Startup)
- Componentization of Install
- Integrated install for selection of prerequisite software, separate server versus client install
- WebAdmin and DMT GUI
- Separation of Configuration versus Data Management Tasks

- Enhancements to Directory Management functions supported by DMT
- Improved panel helps, messages, error logging and reporting
- Exploitation of Java 1.2
- Replication Enhancements
- Event Notification (Server and Client support)
- Security Auditing
- Limited Transaction Support
- Automatic LDAP Server Selection for C and JNDI client
- Support for latest DB/2 releases - UDB 6.1 and UDB 7.1
- GSKit 4.0 exploitation
- Backup/Restore Support
- Sample Java Beans illustrating JNDI usage

On AIX, the new IBM SecureWay Directory version translates messages for Group 1 national languages including Brazilian Portuguese, French, German, Italian, Spanish, Japanese, Korean, Simplified Chinese, Traditional Chinese, Czech, Polish, Hungarian, Russian, Catalan, and Slovakian.

The directory provides scalability by storing information in the IBM DB/2 Universal Database (UDB). DB/2 is packaged with the directory product, but you may only use the DB2 component in association with your licensed use of the SecureWay Directory.

IBM SecureWay Directory is designed from the ground up to be a standards-based, reliable, secure, high-performing enterprise directory that can scale as your directory usage grows. For further information on the IBM SecureWay Directory please refer to the URL:
`http://www-4.ibm.com/software/network/directory/`

4.15 LDAP name resolution enhancement

The Lightweight Directory Access Protocol (LDAP) is an open industry standard that defines a method for accessing and updating information in a directory.

Prior to AIX 5L, the name resolver routines only resolve names using the Domain Name System (DNS) hierarchical naming function, through the Network Information Services (NIS and NIS+), or by the use of the local /etc/hosts file.

AIX 5L enhances the name resolver routines to optionally utilize the information stored in an LDAP server hosts database to accomplish name resolution.

At the time of writing, this feature is only available on the POWER platform.

In order to implement LDAP name resolution support in AIX 5L some extensions to the LDAP server schema are indispensable. The relevant new object class and the related attributes are described in Section 4.14, “IBM SecureWay Directory Version 3.2” on page 168. A new AIX command helps to migrate existing local /etc/hosts information to the LDAP server hosts database. More information about this command and the related LDAP Data Interchange Format file is given in Section 4.15.2, “LDIF file for LDAP host database” on page 173. Section 4.15.3, “LDAP configuration file for local resolver subroutines” on page 174 explains the integration of the LDAP name resolution support with the other, more traditional sources for name resolution in the AIX network subsystem environment. For a quick start and for experienced administrators a brief outline of the procedures necessary to configure an LDAP based name resolution is provided in Section 4.15.4, “LDAP based name resolution configuration” on page 176. Finally Section 4.15.5, “Performance and limitations” on page 177 covers performance aspects and limitations of the LDAP based name resolution.

4.15.1 IBM SecureWay Directory schema for LDAP name resolution

A LDAP directory entry describes some object. An object class is a general description, sometimes called a template, of an object as opposed to the description of a particular object. For instance, the object class person has a surname attribute, whereas the object describing John Smith has a surname attribute with the value Smith. The object classes that a directory server can store and the attributes they contain are described by schema. Schema define what object classes are allowed where in the directory, what attributes they must contain, what attributes are optional, and the syntax of each attribute. More generically one can say that an LDAP schema defines the rules for ordering data within the directory structure.

In order to support LDAP name resolution the new object class ibm-HostTable was introduced to the IBM SecureWay Directory schema. IBM SecureWay Directory designates IBM's implementation of the LDAP server and client functionality and is included in the AIX operating system product at no additional charge. The new ibm-HostTable object class can be used to store the name-to-Internet-address mapping information for every host on a given network.

The ibm-HostTable object class is defined as follows:

```
Object Class name:     ibm-HostTable
Description:          Host Table entry which has a collection of hostname
to
                       IP address mappings.
OID:                   TBD
RDN:                   ipAddress
Superior object class: top
Required Attributes:   host, ipAddress
Optional Attributes:   ibm-hostAlias, ipAddressType, description
```

The attribute definitions follow:

```
Attribute Name: ipAddress
Description:    IP Address of the hostname in the Host Table
OID:            TBD
Syntax:         caseIgnoreString
Length:         256
Single Valued:  Yes
Attribute Name: ibm-hostAlias
Description:    Alias of the hostname in the Host Table
OID:            TBD
Syntax:         caseIgnoreString
Length:         256
Single Valued:  Multi-valued
Attribute Name: ipAddressType
Description:    Address Family of the IP Address (1=IPv4, 2=IPv6)
OID:            TBD
Syntax:         Integer
Length:         11
Single Valued:  Yes
Attribute Name: host
Description:    The hostname of a computer system.
OID:            1.13.18.0.2.4.486
Syntax:         caseIgnoreString
Length:         256
Single Valued:  Multi-valued
Attribute Name: description
Description:    Comments that provide a description of a directory object
entry.
OID:            2.5.4.13
Syntax:         caseIgnoreString
Length:         1024
Single Valued:  Multi-valued
```

Please note that only the three attributes ipAddress, ibm-hostAlias, and ipAddressType are new to the IBM SecureWay Directory LDAP implementation. The attributes host and description were previously part of the IBM SecureWay Directory schema.

4.15.2 LDIF file for LDAP host database

When an LDAP directory is loaded for the first time or when many entries have to be changed at once, it is not very convenient to change every single entry on a one-by-one basis. For this purpose, LDAP supports the LDAP Data Interchange Format (LDIF) that can be seen as a convenient, yet necessary, data management mechanism.

The LDIF format is used to convey directory information or a description of a set of changes made to directory entries. An LDIF file consists of a series of records separated by line separators. A record consists of a sequence of lines describing a directory entry or a sequence of lines describing a set of changes to a single directory entry. An LDIF file specifies a set of directory entries or a set of changes to be applied to directory entries, but not both at the same time.

To support the implementation and configuration of LDAP based name resolution, AIX 5L offers the new `hostsldif` command. The `hosts2ldif` command resides in the /usr/bin directory and creates an LDIF file from /etc/hosts or another file that has the same format. With no options, the /etc/hosts file is used to create the /tmp/hosts.ldif LDIF file using cn=hosts as the base Distinguished Name (baseDN). The baseDN specifies the starting point for the name resolution database within the Directory Information Tree (DIT) structure of the LDAP server. The LDIF file can be used during the configuration process for the LDAP server to load any existing name resolution information which are stored in /etc/hosts files.

The listing below shows a sample LDAP Data Interchange Format (LDIF) file that needs to be generated by the `hostsldif` command:

```
dn: cn=hosts
objectclass: top
objectclass: container
cn: hosts
dn: ipAddress=127.0.0.1, cn=hosts
host: loopback
ipAddress: 127.0.0.1
objectclass: ibm-HostTable
ipAddressType: 1
ibm-hostAlias: localhost
description: loopback (lo0) name/address

dn: ipAddress=1.1.1.1, cn=hosts
```

```
host: testaix5l
ipAddress: 1.1.1.1
objectclass: ibm-HostTable
ipAddressType: 1
ibm-hostAlias: e-testaix5l
ibm-hostAlias: testaix5l.austin.ibm.com
description: first ethernet interface

dn: ipAddress=fe80::dead, cn=hosts
host: testaix5l
ipAddress: fe80::dead
objectclass: ibm-HostTable
ipAddressType: 2
ibm-hostAlias: test-ll
ibm-hostAlias: test-ll.austin.ibm.com
description: v6 link level interface
```

The numbers in the value of the ipAddressType attribute are defined in RFC 1700 where ipAddressType 1 refers to IP Version 4 and ipAddressType 2 designates the IP Version 6 protocol.

4.15.3 LDAP configuration file for local resolver subroutines

The process of obtaining an Internet address from a host name is known as name resolution and is done by the gethostbyname subroutine. The process of translating an Internet address into a host name is known as reverse name resolution and is done by the gethostbyaddr subroutine. These routines are essentially accessors into a library of name translation routines known as resolvers.

Resolver routines on hosts running TCP/IP normally attempt to resolve names using the following sources:

- BIND/DNS (named)
- Network Information Service (NIS and NIS+)
- Local /etc/hosts file

Traditionally, the ordering of name resolution services can be specified in the /etc/netsvc.conf file, the /etc/irs.conf file, or the NSORDER environment variable. The settings in the /etc/netsvc.conf configuration file override the settings in the /etc/irs.conf file. The NSORDER environment variable overrides the settings in the /etc/irs.conf and the /etc/netsvc.conf files.

Beginning with AIX 5L the name resolver routines can optionally utilize the information of an LDAP server database to accomplish name resolution.

An entry in the /etc/irs.conf file is of the following format: map mechanism [option]. If the system administrator specifies hosts as the value for the map parameter the given entry defines the mechanism for mapping host names to their IP addresses. AIX 5L allows you to configure ldap as a new value for the mechanism parameter. The ldap parameter value prompts the resolver routines to query an LDAP server. For example, to use an LDAP server to resolve a host name that cannot be found in the /etc/hosts file you would have to enter the following lines in the /etc/irs.conf file:

```
# Use LDAP server to resolve host names that cannot be found in the
# /etc/hosts file
hosts local continue
hosts ldap
```

The necessary information about the related LDAP server is supplied by the /etc/resolv.ldap file that must be configured for this mechanism to work.

The /etc/netsvc.conf configuration file format was similarly expanded to add support for LDAP based name resolution. Within the /etc/netsvc.conf file the ordering of the name resolution mechanism is specified by an entry of the following format: hosts = value [, value]. Beginning with AIX 5L the keyword hosts accepts in addition to the previously known values like bind, local, and nis the new value ldap. In analogy to the /etc/irs.conf file entries, the ldap value causes the network subsystem to use LDAP services for resolving names and the necessary information about the related LDAP server is supplied by the /etc/resolv.ldap file that must be configured to activate this mechanism. For example, to use the LDAP server for resolving names, indicate that it is authoritative, and to also use the BIND service as alternative, you would have to enter the following lines in the /etc/netsvc.conf file:

```
# Use LDAP server authoritative for resolving names, and use the BIND
# service if the resolver cannot contact the LDAP
hosts = ldap = auth , bind
```

Finally, the NSORDER environment variable accepts also a new keyword ldap to refer to the LDAP based name resolution. For example, if one likes to supplement the default name services ordering (bind, nis, the local /etc/hosts file) by the additional support of an LDAP server, the NSORDER environment variable has to be defined as follows:

```
# export NSORDER=bind,nis,local,ldap.
```

What ever way is chosen to enable the network subsystem to benefit from an LDAP based name resolution, the related /etc/resolv.ldap configuration file has to be present and appropriately configured. The /etc/resolv.ldap file defines the LDAP server information for local resolver subroutines. If the /etc/resolv.ldap file is not present the system will rely on the default or alternative name resolution mechanisms defined by the /etc/netsvc.conf file, the /etc/irs.conf files, or the NSORDER environment variable.

The resolv.ldap file contains one ldapserver entry, which is required, and one searchbase entry, which is optional. The ldapserver entry specifies the Internet address of the LDAP server to the resolver subroutines. The entry must take the following format:

```
ldapserver address [ port ]
```

The address parameter specifies the dotted decimal address of the LDAP server. The port parameter is optional; it specifies the port number that the LDAP server is listening on. If you do not specify the port parameter, then it defaults to 389.

The searchbase optional entry specifies the base Distinguished Name (baseDN) of the name resolution database on the LDAP server. This entry must take the following format:

```
searchbase baseDN
```

The baseDN parameter specifies the starting point for the name resolution database on the LDAP server. If you do not define this entry, then the searchbase entry defaults to cn=hosts.

For example, to define an LDAP server with an IP address 192.9.201.1, that listens on the port 636, and with a searchbase cn=hosttab, enter the following lines in the /etc/resolv.ldap file:

```
# LDAP server information for local resolver subroutines
ldapserver 192.9.201.1 636
searchbase cn=hosttab
```

4.15.4 LDAP based name resolution configuration

Use the following procedure to configure the LDAP server to store name-to-Internet-address mapping host information:

1. Add a suffix on the LDAP server. The suffix is the starting point of the hosts database. For example, "cn=hosts". This can be done using the web-based IBM SecureWay Directory Server Administration tool.
2. Create an LDAP Data Interchange Format (LDIF) file. This can be done manually or with the `hosts2ldif` command, which creates an LDIF file from the /etc/hosts file. Refer to the `hosts2ldif` command manual page in the AIX documentation library for more information.

3. Import the hosts directory data from the LDIF file on the LDAP server. This can be done with the `ldif2db` command or through the web-based IBM SecureWay Directory Server Administration tool.

To configure the client to access the hosts database on the LDAP server, use the following procedure:

1. Create the /etc/resolv.ldap file. Refer to the section resolv.ldap File Format for TCP/IP in the AIX documentation library for more information and a detailed example of a resolv.ldap file.
2. Change the default name resolution through the NSORDER environment variable, the /etc/netsvc.conf file, or the /etc/irs.conf file. Refer to the section netsvc.conf File Format for TCP/IP or the section irs.conf File Format for TCP/IP in the AIX documentation for more information.

4.15.5 Performance and limitations

The AIX 5L enhancements of the resolver routines are designed and capable to support LDAP based name resolution for either Version 2 or Version 3 of the Lightweight Directory Access Protocol. But in order to enable LDAP base name resolution with an LDAP server that uses the protocol Version 2, it is necessary to manually create extensions to the LDAP schema. Refer to Section 4.15.1, “IBM SecureWay Directory schema for LDAP name resolution” on page 171 for more detailed information about the new and indispensable object class ibm-HostTable and the related attributes which were used to extend the LDAP schema of the IBM SecureWay Directory LDAP version 3 implementation.

Since the resolver can possibly search through additional maps and the time-out for the LDAP search is 30 seconds, there could be some performance degradation in the amount of time it takes to resolve a name. However, if the LDAP server environment is properly designed and implemented to support LDAP base name resolution and if on the client side the appropriate configurations of the /etc/netsvc.conf file, the /etc/irs.conf file or the NSORDER environment variable are established, the performance will be of the same order as for the DNS mechanism.

4.16 Documentation search-engine enhancement

The Documentation Library Service in AIX 5L uses a new search engine. The Text Search Engine (TSE) is replacing the NetQuestion version 1.2.3 (IMNSearch) that was presented in AIX Version 4.3.3.

At the time of writing, this feature is only available on the POWER platform.

Some of the enhancements of Text Search Engine over NetQuestion includes:

- Use of a single search engine for both single byte or double byte character sets, instead of one engine for each type of character.
- The Text Search Engine does not need an writable index file, so you can have the Documentation CD-ROM mounted and do all the searches through the mounted CD-ROM without file write permission problems.
- The new Text Search Engine supports Russian Language through the ISO-8859-5 Russian codeset.
- The Text Search Engine is installed by default with the AIX base installation unless Minimal install is used.

The Text Search Engine provides binary compatibility, and can read all NetQuestion search indexes. From a migration path point of view, AIX Version 4.3 machines will be able to upgrade to this new version without problems. However, re-building old user-created documents using the new engine will significantly improve search performance.

4.17 Tivoli readiness

AIX 5L for the POWER architecture is compliant with the specifications that the *Tivoli Ready* mark requires for operating systems.

At the time of writing, this feature is only available on the POWER platform.

The difference from the previous AIX Version 4.3 version is that the Tivoli Management Agent (TMA) is now part of the base CDs, and it is installed automatically with a normal AIX installation.

The following lines are the list of filesets installed for Tivoli Readiness:

```
# lslpp -L "Tivoli*"
  Fileset                      Level  State  Description
  ----------------------------------------------------------------------------
  Tivoli_Management_Agent.client.rte
                             3.2.0.0    C    Management Agent runtime"
```

4.18 CATIA Welcome Center

When ordering a new workstation machine with software pre-loaded, a Netscape browser will start automatically right after AIX initializes. This browser will show you the new CATIA Welcome Center for AIX 5L.

This feature is only available on the POWER platform.

Figure 52 shows the CATIA Welcome Center main screen.

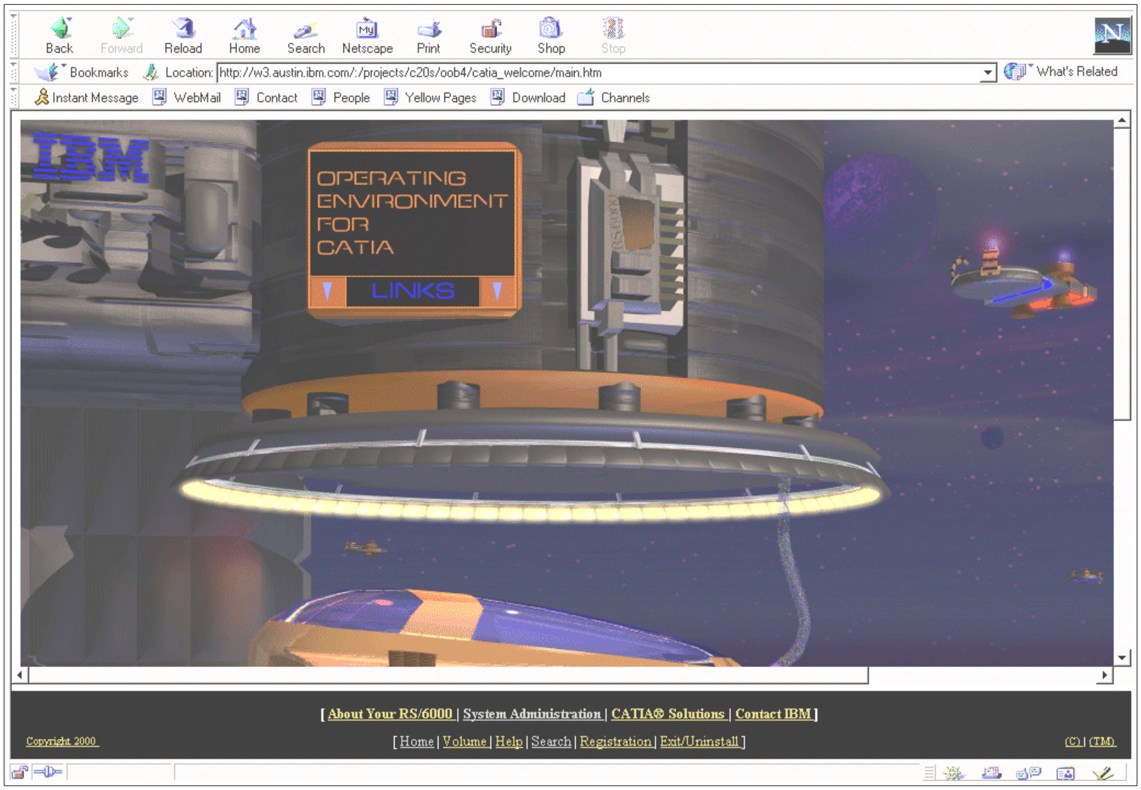
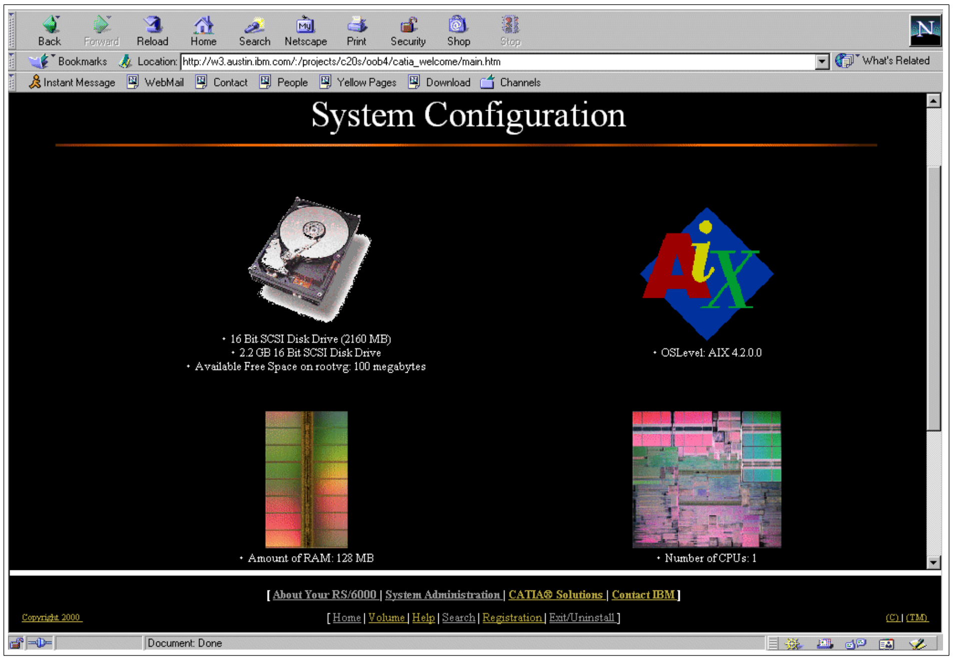

Back
Forward
Reload
Home
Search
Netscape
Print
Security
Shop
Stop
Bookmarks
Location: http://w3.austin.ibm.com/:/projects/c20s/oob4/catia_welcome/main.htm
What's Related
Instant Message
WebMail
Contact
People
Yellow Pages
Download
Channels
IBM
OPERATING ENVIRONMENT FOR CATIA
LINKS
[About Your RS/6000 | System Administration | CATIA® Solutions | Contact IBM]
Copyright 2000
[Home | Volume | Help | Search | Registration | Exit/Uninstall]
(C) | (TM)

Figure 52. CATIA Welcome Center main screen

You can navigate through the Welcome Center options by either using the hyperlinks provided at the bottom of the page, or by pressing the *Links* word on the page.

The following options are available through the main screen:

- About your RS/6000: This link provides access to the contents of CATIA CD-ROM, RS/6000 hardware technical details and offerings, AIX links, and access to the Online Documentation Library.
- System Administration: This link provides access to the System Configuration (see Figure 53) which gives you details on the current hardware and software environment of the local system; Configuration Assistant, a Web-based System Manager and System Expert links are also available.



System Configuration

- 16 Bit SCSI Disk Drive (2160 MB)
- 2.2 GB 16 Bit SCSI Disk Drive
- Available Free Space on rootvg: 100 megabytes

- OSLevel: AIX 4.2.0.0

- Amount of RAM: 128 MB

- Number of CPUs: 1

[About Your RS/6000 | System Administration | CATIA® Solutions | Contact IBM]

Copyright 2000 [Home | Volume | Help | Search | Registration | Exit/Uninstall] (C) | (TM)

Document: Done

Figure 53. System Configuration details in the CATIA Welcome Center

- CATIA Solutions: Gives you access to various links related to CATIA V5, CATIA V4 and Enovia CATWeb Solutions.
- Contact IBM: Provides links to RS/6000, AIX and CATIA Services and Support Homepages, as well as the Education and Training Homepage.

Chapter 5. Networking enhancements

AIX 5L provides many enhancements in the networking area. They are described in this chapter.

5.1 Quality of service support

A new method for regulating network traffic flows called Quality of Service (QoS) was introduced in AIX Version 4.3.3. The demand for QoS arises from applications such as digital audio/video or real-time applications and the need to manage bandwidth resources for arbitrary administratively-defined traffic classes.

AIX 5L further enhances the QoS implementation to support overlapping policies in the QoS manager. Directly related to this feature is the new and additional capability to specify a priority for a given policy. To improve the manageability of a QoS configuration, AIX 5L also offers four new commands to add, delete, modify, and list QoS policies.

5.1.1 QoS manager overlapping policies

The QoS Implementation in AIX 5L offers, among other features, a policy-based network traffic categorization and conditioning for the Differentiated Services (DS) and Integrated Services (IS) QoS model. In order for network equipment to provide QoS features from various vendors that interoperate correctly, it is necessary to standardize the underlying policy scheme for QoS. The AIX policy schema is based on the Internet Draft <draft-rajan-policy-qosschema-01.txt> of the Internet Engineering Task Force (IETF).

A policy condition is characteristic of an IP packet, and a policy action is an action the packet receives when it meets a policy condition. A policy condition is defined by five characteristics of a packet. They are source IP address, source port number, destination IP address, destination port, and protocol type (TCP or UDP). A policy action includes token bucket parameters and a TOS byte value defining in-profile traffic.

From an administrator’s point of view, a policy is essentially a collection of configuration parameters to regulate certain types of traffic flow.

There are two core components of the QoS subsystem which are of relevance for the policy-based networking function:

- QoS kernel extension (/usr/lib/drivers/qos)

  The QoS kernel extension resides in /usr/lib/drivers/qos and is loaded and unloaded using the cfgqos and ucfgqos configuration methods. This kernel extension enables QoS support and provides the QoS manager functionality.

- Policy agent (/usr/sbin/policyd)

  The policy agent is a user-level daemon that resides in /usr/sbin/policyd. It provides support for policy management and interfaces with the QoS kernel extension (QoS manager) to install, modify, and delete policy rules. Policy rules may be defined in the local configuration file (/etc/policyd.conf), retrieved from a central network policy server using LDAP, or both. AIX 5L also offers a command line interface to manage and administer policy rules.

Each policy definition requires a ServicePolicyRules and a ServiceCategories object within the /usr/sbin/policyd.conf file. The ServicePolicyRules object establishes the policy condition and the ServiceCategories object determines the policy action. The structure for the ServicePolicyRules object is shown in the following example:

```
Used conventions:
i   : integer value
s   : a character string
a   : IP address format B.B.B.B
(R) : Required parameter
(O) : Optional parameter

 ServicePolicyRules   s
{
    SelectorTag             s       # Required tag for LDAP Search
    ProtocolNumber          i       # Transport protocol id for the policy rule
    SourceAddressRange      a1-a2
    DestinationAddressRange a1-a2
    SourcePortRange         i1-i2
    DestinationPortRange    i1-i2
    PolicyRulePriority      i       # Highest value is enforced first
    ServiceReference        s       # Service category name for this policy rule
}

where
s                        (R): is the name of this policy rule
SelectorTag              (R): required only for LDAP to Search object class
ProtocolNumber           (R): default is 0 which causes no match, must explicitly specify
SourceAddressRange       (O): from a1 to a2 where a2 >= a1, default is 0, any source
address
SourcePortRange          (O): from i1 to i2 where i2 >= i1, default is 0, any source port
DestinationAddressRange (O): same as SourceAddressRange
DestinationPortRange     (O): same as SourcePortRange
PolicyRulePriority       (O): Important to specify when overlapping policies exist
ServiceReference         (R): service category this rule uses
```

Note that the newly introduced attribute PolicyRulesPriority and each ServicePolicyRules object is associated with a unique instance of the ServiceCategory referred to by the ServiceReference attribute.

During the start of the QoS subsystem the policy agent installs the defined policies to be used by the QoS manager. Previous AIX releases took a conservative approach toward overlapping policies by completely disallowing them. This had implications for deployment and actual usage, where the system administrator may want to specify or assume a given ordering between the potentially overlapping policies. In AIX releases prior to AIX 5L the QoS manager effectively searched for a matching policy in a way that did not allow a priority among the policies.

One example to illustrate the issues related to overlapping policies is as follows. A customer desires to configure simultaneous policies for application audio (AppA) and applicationvideo (AppV). The first application (AppA) may select a valid port number for the source port and a wild card for the destination, while the second application (AppV) selects a wild card for the source port and a valid port number for the destination. The five attributes of the related ServicePolicyRules objects (source IP address, source port number, destination IP address, destination port, and either TCP or UDP) that are used by the QoS Manager to identify specific policy rules may have all fields identical with the exception of source and destination port for the two applications. When installing the policy definitions for both applications under AIX Version 4.3.3, the second policy in the installation sequence was found to be overlapping, an error was flagged and the policy was not installed. While the policies were overlapping, in practice, if the system allowed the installation of both policies the two applications would not have assigned conflicting ports. The policies would not have overlapped since for the application (AppA) that uses the source port it would not have assigned a destination port overlapping with the second application (AppV) and vice versa.

With different applications in other scenarios it may happen that even though the policies are allowed to install in practice they may overlap so then order of policy installation becomes important.

In order to allow the installation of overlapping policies the order in which the policies are input to the QoS Manager needs to be preserved. The highest priority policy in the overlapping case will be input to the QoS manager from the policy agent last and that order is maintained for proper policy enforcement. The last policy installed from the policy agent that matches will be enforced over previously installed policies in the overlapping case.

The policy agent's capability was extended to allow system administrators to set priorities for policies so that they get installed in a desired order onto the QoS kernel extension. In order to do this, an attribute called PolicyRulePriority was added to the ServicePolicyRules structure. The ServicePolicyRules objects are defined in the /etc/policyd.conf configuration file. The PolicyRulePriority attribute can be set to any positive integer. If no value is specified, the default is set to 0. The absolute value of this attribute has no meaning and only the relative values are important. The policies are installed onto the AIX 5L kernel in the order of the highest priority first. Every time a new policy is added to the policy agent, it is inserted into the policies list based on its priority and finally the whole list is installed onto the QoS manager stack.

The priority for any specific policy can be specified by manually editing the ServicePolicyRules stanzas in the /etc/policyd.conf policy agent configuration file. Alternatively you can use the new command line interface as described in the Section 5.1.2, "QoS manager command line support" on page 184.

QoS is an optionally installable feature and packaged with the bos.net.tcp.server fileset.

5.1.2 QoS manager command line support

Beginning with AIX 5L four new command line programs will be available to add, modify, delete, or list Quality of Service policies. These AIX commands operate on the /etc/policyd.conf policy agent configuration file so the use of a text editor is not required any more to manage policies. Once an `add`, `modify` or `remove` command is executed, the change takes effect immediately and the local configuration file of the policy agent gets updated to permanently keep the change. The `list` command will prompt the policy agent to query its internal indexed list to provide the information about ServiceCategories and ServicePolicyRules which define the active policies. Also, a flag will be available for the command line programs to allow prioritization of policies so the correct order of enforcement can be determined in the event of a policy overlap. The Policy Agent must input the policies to the QoS Manager in the order of lowest priority first.

The QoS command line interface consists of the commands provided in the following sections with their given syntax and usage.

5.1.2.1 The qosadd command

The `qosadd` command adds the specified Service Category or Policy Rule entry in the policyd.conf file and installs the changes in the QoS Manager.

To add a service category or a policy rule:

```
#qosadd
usage: qosadd  -s ServiceCategory   [-t OutgoingTOS] [-b MaxTokenBucket]
              [-f Flow ServiceType] [-m MaxRate] service
usage: qosadd  -s ServiceCategory   -r ServicePolicyRules
              [-l PolicyRulePriority] [-n ProtocolNumber] [-A SrcAddrRange]
             [-a DestAddrRange] [-P SrcPortRange] [-p DestPortRange] policy
```

5.1.2.2 The qosmod command

The `qosmod` command modifies the specified Service Category or Policy Rule entry in the policyd.conf file and installs the changes in the QoS Manager.

To modify an existing service category or policy rule:

```
# qosmod
usage: qosmod  -s ServiceCategory   [-t OutgoingTOS] [-b MaxTokenBucket]
              [-f Flow ServiceType] [-m MaxRate] service
usage: qosmod  -s ServiceCategory   -r ServicePolicyRules
              [-l PolicyRulePriority] [-n ProtocolNumber] [-A SrcAddrRange]
             [-a DestAddrRange] [-P SrcPortRange] [-p DestPortRange] policy
```

5.1.2.3 The qoslist command

The `qoslist` command lists the specified Service Category or Policy Rule. The qoslist command lists all Service Categories or Policy Rules if no specific name is given.

```
#qoslist
usage: qoslist [ServiceCategory][Policy Rule] <policy or service>
```

5.1.2.4 The qosremove command

The `qosremove` command removes the specified Service Category or Policy Rule entry in the policyd.conf file and the associated policy or service in the QoS Manager.

```
#qosremove
usage: qosremove <ServicePolicyRule or ServiceCategory> <policy or service>
```

Refer for further details on the meaning of the flags, parameters and arguments listed above to the appropriate manual pages of the standard AIX documentation library.

5.2 TCP/IP routing subsystem enhancements

AIX 5L offers multipath routing and Dead Gateway Detection (DGD) as new features of the TCP/IP routing subsystem. They are intended to enable administrators to configure their systems for load balancing and failover.

Multipath routing provides the function necessary to configure a system with more than one route to the same destination. This is useful for load balancing by routing IP traffic over different network segments, or to specify backup routes to use with Dead Gateway Detection. Section Section 5.2.1, “Multipath routing” on page 186 covers the details on this new routing feature.

Dead Gateway Detection enables a system to discover if one of its gateways is down and use an alternate gateway. DGD offers an active and a passive mode of operation to account for different kinds of customer requirements in respect to performance and availability. Section Section 5.2.2, “Dead gateway detection” on page 192 provides more in depth information about this enhancement to the TCP/IP routing subsystem.

Both new routing features are implemented for IP Version 4 (IPV4) and IP Version 6 (IPV6).

5.2.1 Multipath routing

Prior to AIX 5L, a new route could be added to the routing table only if it was different from the existing routes. The new route would have to be different by either destination, netmask, or group ID. The sample output of the netstat command, depicted in the following, shows two routing table entries that have the same netmask. However, the route for the token ring interface differs from the route associated with the Ethernet interface by the destination:

```
# netstat -rn
Routing tables
Destination    Gateway    Flags  Refs   Use    If  PMTU   Exp  Groups

Route tree for Protocol Family 2 (Internet):
9.3.21/24      9.3.21.22  U      106    17412  tr1 -      .
9.3.22/24      9.3.22.37  U      0      266344 en0 -      .
```

The following netstat command output was taken from a system where two routes for two different gateways are defined with the same destination but for different netmasks.

```
# netstat -rn
Routing tables
Destination Gateway Flags Refs Use If PMTU Exp  Groups

Route tree for Protocol Family 2 (Internet):
10/24 9.3.21.22 UGc 0 0 tr1 - -   =>
10/23 9.3.22.37 UGc 0 0 en0
```

In the case where the destination address is the same, but the netmask is different, the most specific route that matches will be used. In the previous example, packets sent to 10.0.0.1 - 10.0.0.255 would use the 10/24 route, since it is more specific, while packets sent to 10.0.1.1 - 10.0.1.255 would use the 10/23 route, since they do not match the 10/24 route but do match the 10/23 route.

The third possible differentiator for a unique route definition is given by the group ID list. The groups associated with a route are listed in the column of the `netstat -r` output, which is labeled with the keyword Groups. These groups are comprised of AIX group IDs, and they determine which users have permission to access the route. This feature is used by system administrators to enforce security policies or to provide different classes of service to different users.

With the new multipath routing feature in AIX 5L, routes no longer need to have a different destination, netmask, or group ID list. If there are several routes that equally qualify as a route to a destination, AIX will use a cyclic multiplexing mechanism (round-robin) to choose between them. The benefit of this feature is twofold:

- Enablement of load balancing between two or more gateways.
- Feasibility of load balancing between two or more interfaces on the same network can be realized. (The administrator would simply add several routes to the local network, one through each interface.)

In order to implement multipath routing, AIX 5L allows you to define a user-configurable cost attribute for each route and offers the option to associate a particular interface with a given route. These enhancements are configurable by the parameters -hopcount and -if of the `route` command. The syntax and usage for the `route` command is documented in the AIX command reference. In the following, you find an excerpt of the manual page for the `route` command. Note the new -active_dgd parameter that turns on the active DGD for a given route, which will be described later on in Section 5.2.2.3, “Active Dead Gateway Detection” on page 198:

```
route [ -n ] [ -q ] [ -v ] Command [ Family ] [ [ -net | -host ] Destination [-prefixlen
n ] [-netmask] [ Address ] ] Gateway ] [ Arguments ]

Flags:

-n             Displays host and network names numerically, rather than
               symbolically, when reporting results of a flush or of any
               action in verbose mode.
-q             Specifies quiet mode and suppresses all output.
-v             Specifies verbose mode and prints additional details.
```

```
-net           Indicates that the Destination parameter should be
               interpreted as a network.
-netmask       Specifies the network mask to the destination address. Make
               sure this option follows the Destination parameter.
-host          Indicates that the Destination parameter should be
               interpreted as a host.
-prefixlen n   Specifies the length of a destination prefix (the number of
               bits in the netmask).

Parameters:

Arguments    Specifies one or more of the following arguments. Where n is
             specified as a variable to an argument, the value of the n
             variable is a positive integer.

      -active_dgd   Enables Active Dead Gateway Detection on the route.
      ...
      -hopcount n   Specifies maximum number of gateways in the route.
      ...
      -if ifname    Specifies the interface (en0, tr0 ...) to associate
                    with this route so that packets will be sent using this
                    interface when this route is chosen.
Commands     Specifies one of six possibilities: add, flush, delete,
             change, monitor, or get.
Family       Specifies the address family (inet, inet6, or xns).
Destination  Identifies the host or network to which you are directing the
             route.
Gateway      Identifies the gateway to which packets are addressed.
```

5.2.1.1 User-configurable cost attribute of routes

The user-configurable cost of a route is specified as a positive integer value for the variable associated with the -hopcount parameter. The integer can be any number between 0 and the maximum possible value of MAX_RT_COST, which is defined in the /usr/include/net/route.h header file to be INT_MAX. The value of INT_MAX is defined in /usr/include/sys/limits.h to be 2147483647. The header files will be on your system after you install the bos.adt.include fileset. The -hopcount parameter existed in the past, and the assigned integer value was supposed to reflect the number of gateways in the route. However, in previous AIX releases, the parameter value given during the configuration of the route had no effect on how the route was used.

Even so, the -hopcount parameter in AIX 5L refers historically to the number of gateways in the route; the number configurable by the system administrator can be totally unrelated to the actual presence or absence of any real gateways in the network environment. The user-configurable cost attribute's sole purpose is to establish a metric, which is used to create a priority hierarchy among the entries in the routing table.

If the routing table offers several alternative routes to the desired destination, the operating system will always choose the route with the lowest distance metric as indicated by the lowest value for the current cost. In the case where multiple matching routes have equal current cost, a lookup mechanism chooses the most specific route. When both criteria are equal for multiple routes, AIX 5L will round-robin between them. Higher-cost routes ordinarily will never be used; they are only there as backups. If the lower-cost routes are deleted or their costs are raised, the backup routes will be used. This provides a mechanism for marking bad routes when a gateway failure is detected; indeed, the DGD feature, as described in Section 5.2.2, "Dead gateway detection" on page 192, exploits this particular feature.

The kernel resident routing table is initialized when interface addresses are set by making entries for all directly connected interfaces. The routing entry structure rtentry is defined in the route.h header file that will be located in the usr/include/net/ directory after you optionally install the bos.adt.include fileset.

The route entry structure is expanded by the new field rt_cost_config which holds the integer value of the user-configured cost. The rtentry structure definition is also modified to allow an extra field that will store routes with the same current cost and netmask as a linked list using the field called rt_duplist. The existing field rn_dupedkey will be used to maintain a sorted list of these lists. This makes the route entries look like they are stored in a two dimensional linked list. They are first grouped in a list with routes with the same cost and netmask. Then a list of these groups is sorted by netmask and current cost. The primary sort criteria is the netmask and the secondary is the current cost. (See also the radix_node structure definition in the /usr/include/net/radix.h header file.)

Since routes are alternated (round-robin) between routes with the same cost and netmask, a field is required to remember which route was used last. A pointer to the last route used is stored in the first rtentry structure of a group. A new field called rt_lu was added to the rtentry structure and this field will serve to store the reference to the last route used.

The behavior of the code to select routes has only changed when duplicate routes exist. For nodes with multiple routes, the rn_dupedkey list is followed until a route which matches is found. If there are other entries with the same cost and netmask (rt_duplist and rt_lu are not NULL), the route that was last used (rt_lu) is skipped and the next one chosen.

The costs on all routes can be displayed using the new -C flag on the netstat command as indicated by the following example.

With the -C flag set, the netstat command shows the routing tables, including the user-configured and current costs of each route. The user-configured cost is set using the -hopcount flag of the route command. The current cost may be different than the user-configured cost if, for example, the Dead Gateway Detection has changed the cost of the route. For further details on DGD, refer to Section 5.2.2, “Dead gateway detection” on page 192.

```
# netstat -Cn
Routing tables
Destination       Gateway        Flags      Refs   Use     If     Cost   Config_Cost

Route tree for Protocol Family 2 (Internet):
9.3.149.96/28     9.3.149.100    U          5      23      en1    0      0
9.3.149.160/28    9.3.149.163    U          1      5       tr0    0      0
9.53.150/23       9.3.149.160    UGc        0      0       tr0    0      0 =>
9.53.150/23       9.3.149.97     UGc        0      0       en1    1      1
127/8             27.0.0.1       U          1      130425  lo0    0      0

Route tree for Protocol Family 24 (Internet v6):
::1               ::1            UH         0      0       lo0    0      0
```

5.2.1.2 Interface specific routes

The implementation of TCP/IP routing in previous AIX releases did not provide any mechanism to associate a specific interface with a route. When there were multiple interfaces on the same network, the same outgoing interface for all destinations accessible through that network was always chosen. In order to configure a system for network traffic load balancing, it is desirable to have multiple routes so that the network subsystem routes network traffic to the same network segment by using different interfaces. AIX 5L introduces the new -if argument to the route command, which can be used to associate a particular interface with a specific route.

The -if parameter argument must not be mistaken for the -interface parameter argument of the route command. The -interface argument specifies that the route being added is an interface route, which means it is a direct route to the local network and does not go through a gateway.

The following example shows the usage of the route command to establish an interface specific host route from a given computer on one network to its counterpart on a different network:

```
route add 192.100.201.7 192.100.13.7 -if tr0
```

The 192.100.201.7 address is that of the receiving computer (destination parameter) and the 192.100.13.7 address is that of the routing computer (gateway parameter). The -if argument assigns the static host route to the token ring interface tr0.

5.2.1.3 Deletion and modification of routes

The route command, used in conjunction with the delete qualifier command, examines the entries in the kernel route table and deletes only the specified route in the routing table if a unique route has been successfully identified. In previous AIX releases, this command could only fail if no route entry matched the specified command line parameters. Since AIX 5L offers the option to specify multiple routes to the same destination, but with different gateways or interfaces the route delete command may fail, because more than one route matches the criteria for deletion. So if the attempt to delete a route fails, an error message is returned (as always), but this message explicitly mentions that there are now two possible error conditions which have to be considered. The following example shows the error message returned by the route delete command on a system with more than one defined default route:

```
# route delete default
0821-279 writing to routing socket: The process does not exist.
default net default: route: not in table or multiple matches
```

In order to account for the possible existence of multiple routes to the same destination but with different gateways or interfaces in AIX 5L, similar modifications were implemented for the command to change a route. This means that the route change command will return an error message whenever no unique route could be identified regardless of the absence of a given route or the existence of multiple routes to the same destination. Note that only the user-configurable cost, gateway, and interface of a route can be changed.

5.2.1.4 Limitations for multipath routing

You must completely understand the limitations when using Multipath routing in conjunction with the path maximum transfer unit (PMTU) discovery feature of AIX. A full read of this section is recommended to better understand the relationship between these two facilities.

When the destination of a connection is on a remote network, the operating system's TCP defaults to advertise a maximum segment size (MSS) of 512 bytes. This conservative value is based on a requirement that all IP routers support an MTU of at least 576 bytes.

The optimal MSS for remote networks is based on the smallest MTU of the intervening networks in the route between source and destination. In general, this is a dynamic quantity and could only be ascertained by some form of path MTU discovery.

The AIX 5L operating system supports a path MTU discovery algorithm as described in RFC 1191. Path MTU discovery can be enabled for TCP and UDP applications by modifying the tcp_pmtu_discover and udp_pmtu_discover options of the `no` command. When enabled for TCP, path MTU discovery will automatically force the size of all packets transmitted by TCP applications to not exceed the discovered path MTU size. Since UDP applications themselves determine the size of their transmitted packets, UDP applications must be specifically written to utilize path MTU information by using the IP_FINDPMTU socket option, even if the udp_pmtu_discover network option is enabled. By default, the tcp_pmtu_discover and udp_pmtu_discover options are disabled on Version 4.2.1 through Version 4.3.1, and enabled on Version 4.3.2 and later.

When the path MTU has been discovered for a network route, a separate host route is cloned for the path. These cloned host routes, as well as the path MTU value for the route, can be displayed using the `netstat -r` command. Accumulation of cloned routes can be avoided by allowing unused routes to expire and be deleted. Route expiration is controlled by the route_expire option of the `no` command. Route expiration is disabled by default on Version 4.2.1 through Version 4.3.1, and set to one minute on Version 4.3.2 and later.

Beginning with AIX 5L, you may have several equal-cost routes to a given network but with different associated gateways on a system for which PMTU discovery is enabled. When traffic is sent to a host on that specific network, a host route will be cloned from whichever network route was chosen by the cyclic multiplexing code of the multipath routing algorithm. Because the cloned host route is always more specific than the original network route of which the clone was derived, all traffic to that host will use the same gateway as long as the cloned route exists and, consequently, no cyclic multiplexing among the different gateways associated with the equal-cost route to the specific network will take place.

Since PMTU discovery is enabled by default in AIX 5L, system administrators may consider disabling the network options tcp_pmtu_discover or udp_pmtu_discover to turn off route cloning (in order to take full advantage of the new multipath routing feature). This measure will prevent the creation of the cloned host routes and will instead allow cyclic multiplexing between equal-cost routes to the same network.

5.2.2 Dead gateway detection

The new Dead Gateway Detection (DGD) feature in AIX 5L implements a mechanism for hosts to detect a dysfunctional gateway, adjust its routing table accordingly, and re-route network traffic to an alternate backup route if available. DGD is generally most useful for hosts that use static rather than dynamic routing.

5.2.2.1 Overview

AIX releases prior to AIX 5L did not permit you to configure multiple routes to the same destination. If a route's first hop gateway failed to provide the required routing function, AIX continued to try to use the broken route and address the dysfunctional gateway. Even if there was another path to the destination which would have offered an alternative route, AIX did not have any means to identify and switch to the alternate route unless a change to the kernel routing table was explicitly initiated, either manually or by running a routing protocol program such as `gated` or `routed`. Gateways on a network run routing protocols and communicate with one another. So if one gateway goes down, the other gateways will detect it, and adjust their routing tables to use alternate routes. Only the hosts continue to try to use the dead gateway.

The new DGD feature in AIX 5L enables host systems to sense and isolate a dysfunctional gateway and adjust the routing table to make use of an alternate gateway without the aid of a running routing protocol program.

AIX 5L implements DGD based on the requirements given in RFC 1122 Sections 3.3.1.4 and 3.3.1.5, and RFC 816. These RFCs contain a number of suggestions on mechanisms for doing DGD, but currently no completely satisfactory algorithm has been identified. In particular, the RFCs require that pinging to discover the state of a gateway be avoided or extremely limited, and they recommend that the IP layer receive *hints* that a gateway is up or down from transport and other layers that may have some knowledge of whether a data transmission succeeded. However, in one of the two possible modes (active mode) for the AIX 5L DGD feature, status information of a gateway is collected with the help of pinging, and hence the AIX 5L DGD implementation is not fully compliant with the RFCs mentioned above.

DGD utilizes the functions of AIX 5L multipath routing. The multipath routing feature allows for multiple routes to the same destination which can be used for load balancing and failover. Refer to Section 5.2.1, “Multipath routing” on page 186 for further details.

The DGD implementation in AIX 5L offers the flexibility to address two distinct sets of customer requirements:

- Requirement for minimal impact on network and system environment in respect to compatibility and performance. The detection of a dysfunctional gateway and the switch from the associated route over to an alternate gateway route must be accomplished without any significant overhead.

- Requirement for maximum availability of network services and connections. If a gateway goes down, a host must always discover that fact within a few seconds and switch to a working gateway.

Since both sets of requirements cannot be satisfied by a single mechanism, AIX 5L DGD offers a passive and an active mode of operation.

The passive Dead Gateway Detection addresses the need for minimal overhead, while the active Dead Gateway Detection ensures maximum availability while imposing some additional workload onto network segments and connected systems. Passive DGD is disabled system wide by default, but active DGD is defined as an attribute for a particular route, and therefore requires to be enabled on a route to route basis.

5.2.2.2 Passive Dead Gateway Detection

One of the two modes for Dead Gateway Detection will work without actively pinging the gateways known to a given system; therefore, this mode is referred to as passive DGD.

Passive DGD will take action to use a backup route if a dysfunctional gateway has been detected. The backup route can have a higher current cost than the route associated with the dysfunctional gateway which allows you to configure primary (lower cost) gateways and secondary (higher cost) backup gateways. As such, DGD expands the TCP inherent failover between alternate equal cost routes, as introduced by the new AIX 5L multipath routing feature.

The passive DGD mechanism depends on protocols which provide information about the state of the relevant gateways. If the protocols in use are unable to give feedback about the state of a gateway, a host will never know that a gateway is down and no action will be taken.

The Transmission Control Protocol (TCP), in conjunction with the Address Resolution Protocol (ARP), is able to give the necessary feedback about the state of a specific gateway. It is important to note that these two protocols give different types of feedback, and that you have to use both protocols to receive the full benefit of the passive DGD feature.

TCP identifies round-trip traffic that is not getting through. It will correctly detect that the gateway in question is down if it is indeed no longer forwarding traffic. However, it may incorrectly report that the gateway is down if there is a temporary routing problem elsewhere in the network that the first-hop gateway will soon detect and adjust to, or if the address it is sending to is unreachable or nonexistent.

On the other hand, ARP still perceives a gateway to be up even if it is no longer forwarding traffic. The only thing ARP can detect with certainty is whether the first-hop gateway can be reached, but it does not sense whether the network traffic is forwarded and reaches its final destination. So transitory problems elsewhere in the network cannot cause ARP to mistake a functional for a dysfunctional gateway.

Because TCP cannot detect if the destination for the network traffic is supposed to be reachable, the decisions about a gateway's state cannot be based only on TCP. Instead, TCP is used to prompt Dead Gateway Detection under certain conditions to determine the state of a gateway based on feedback from ARP.

Note

For IPv6, it is not necessary to do passive dead gateway detection. The Neighbor Discovery Protocol (NDP) contains all the functions that passive DGD will add for IPv4.

Multipath routing in AIX 5L allows you to specify a distance metric or cost associated with a route. Routes to the same destination with equal cost will be selected by a cyclic multiplexing algorithm. Routes with a higher cost will not be used unless there is a problem with the lower-cost routes. Passive DGD exploits the multipath routing feature to provide failover for dysfunctional gateways.

If feedback is received from ARP that a gateway might be down, the current costs of all routes using that gateway will be increased to the maximum value MAX_RT_COST (refer to Section 5.2.1.1, “User-configurable cost attribute of routes” on page 188 for further details). The user-configurable cost will not be changed, but eventually will be used in the future to restore the current cost to the original value if the gateway comes up again. If alternative routes to the same destination with a cost equal to the original cost of the deprecated route are defined, the TCP/IP subsystem will use those exclusively, and the route whose current cost was increased is no longer addressed. If there were no other routes to the destination, the original route is still the lowest-cost route, and the system will continue to try to use it.

When the current cost of a route is increased as described previously, a timer will be set for a configurable period of time. This will be specified by a new network option called dgd_retry_time. The default value for this network option is set to five minutes, since that is about the amount of time it will take a gateway that has crashed to reboot. Use the `no -o` command to display or change the dgd_retry_timer network option. The `no` command output in the following example shows the value for the dgd_retry_timer on a system where this specific network option is set to the default of 5:

```
# no -o dgd_retry_time
dgd_retry_time = 5
```

Note that the network options set by the `no` command are only in effect until the next reboot. If you like to use the customized settings for the network options permanently, you will have to include the appropriate `no` commands in the network startup script /etc/rc.net. This script is executed during the boot process and will establish the network options to have the customized values of your choice.

When the timer expires, the cost will be restored to its original user-configured value. If the gateway did not come up in the meantime, the next attempt to send traffic will raise the current cost for the routes in question again to the maximum value and the timer is reset for another five minutes wait. If the gateway is back up, that route will continue to be used. The only exception to this is when active DGD is in use, as described in Section 5.2.2.3, “Active Dead Gateway Detection” on page 198. In this case, a flag on the route will indicate that active detection is in use, and passive detection should not restore the cost to its original value.

ARP requests are only sent out if the ARP cached entry has expired. By default, ARP entries expire after 20 minutes. So if a gateway goes down, it may take quite a long time (relative to transaction events that require responsive networks) before DGD senses any problem with a given gateway through ARP protocol. For this reason, the DGD mechanism monitors to see if TCP retransmits packets too many times, and in the case where it suspects that a gateway is down, it deletes the ARP entry for that gateway. The next time any traffic is sent along the given route, an ARP request is initiated, which provides the necessary information about the state of the gateway to DGD.

TCP is not supposed to initiate a change of the cost associated with a route since it does not know whether the gateway is actually down or if the destination is just unreachable. For this reason, TCP indirectly initiates an ARP request by deleting the ARP cache entry for the gateway in question. On the other hand, TCP is aware of any particular failing connection. So, TCP explores (independently of the feedback of the initiated ARP requests) if there is any other route to its destination with a cost equal to the one it is currently using. If TCP identifies alternate routes, it tries to use them. This way, the connection in question will still recover right away, if the gateway really was down.

It is important to carefully choose the criteria for deciding that a gateway is down. A failover to a backup gateway just because a single packet was lost in the network must be avoided, but in the case of an actual gateway failure, network availability must be restored with as little delay as possible. The number of lost packets needed before a gateway will be suspected or considered as dysfunctional is user-configurable by the new network option named dgd_packets_lost. The network option dgd_packets_lost can be displayed and changed by the `no -o` command and is set to 3 by default. The `no` command output in the following example shows the value for the dgd_packets_lost on a system where this specific network option is set to the default of 3:

```
# no -o dgd_packets_lost
dgd_packets_lost = 3
```

The same restrictions which were mentioned before in respect to the dgd_retry_timer network option apply for the dgd_packets_lost network option.

If TCP retransmits the same packet as many number of times as defined by dgd_packets_lost and gets no response, it deletes the ARP entry for the gateway route it was using and tries to use an alternative route. When the next attempt is made to send a packet along the gateway route, no ARP cache entry is found, and ARP sends out a request to collect the missing information. The value for dgd_packets_lost also determines how often no response of an ARP request is tolerated before a gateway finally will be considered to be down and the costs of all routes using it will be increased to the maximum possible value.

The control flow for DGD as described implies that DGD will work even when non-TCP traffic occurs. Under this condition, DGD depends on the ARP protocol feedback only, and the related relatively long lifetime values for ARP cache entries will slow down the detection of dysfunctional gateways. However, DGD will still allow you to configure primary (lower cost) and secondary (higher cost) gateways, and it handles the failover from a dysfunctional primary gateway to the secondary backup gateway.

One important aspect in respect to passive DGD must be considered in security sensitive environments. There are many cases where TCP could mistake a functional gateway to be dysfunctional. The destination that TCP is trying to reach may be turned off, has crashed, be unreachable, or be non-existent. Also, packets may be filtered by a firewall or other security mechanism on their way to the destination and there are numerous other possibilities too, for a number of other situations. In these cases, the ARP entry for the gateway in use will be deleted in order to force Dead Gateway Detection to be initiated and to find out if the gateway is actually down. This will cause extra overhead and traffic on the network for the ARP packets to be sent, and also for other connections to wait for an ARP response. In general, this extra overhead will be fairly minimal. It does not happen very often that a connection will be attempted to an unreachable address, and the overhead associated with an ARP is quite small. However, the possibility exists that malicious users could continually try to connect to addresses they knew to be unreachable to purposely degrade performance for other users on the system and generate extra traffic on the network.

To protect systems and users against these types of attacks, a new network option named passive_dgd was introduced with the implementation of DGD in AIX 5L. The passive_dgd default value is 0, indicating that passive DGD will be off by default. The network option passive_dgd can be displayed and changed by the `no -o` command. The `no` command output in the following example shows the value for the `passive_dgd` on a system where this specific network option is set to the default of 0:

```
# no -o passive_dgd
passive_dgd = 0
```

If you want to permanently enable passive DGD, you will have to include the following command line in the network startup script /etc/rc.net:

```
no -o passive_dgd=1
```

5.2.2.3 Active Dead Gateway Detection

Passive Dead Gateway Detection has low overhead and is recommended for use on any network that has redundant gateways. However, passive DGD is done on a best-effort basis only. Some protocols, such as UDP, do not provide any feedback to the host if a data transmission is failing, and in this case, no action can be taken by passive DGD. Passive DGD detects that a gateway is down only if it does not respond to ARP requests.

Under the circumstance that no TCP traffic is being sent through a gateway, passive DGD will not sense a dysfunctional state of the particular gateway. The host has no mechanism to detect such a situation until TCP traffic is sent or the gateway's ARP entry times out, which may take up to 20 minutes. But this situation does not modify route costs. In other words, a gateway not forwarding packets is not considered dead.

This behavior is unacceptable in information technology environments with very strict availability requirements. AIX 5L offers a second DGD mechanism, specifically for these environments, called active Dead Gateway Detection. Active DGD will ping gateways periodically, and if a gateway is found to be down, the routing table is changed to use alternate routes to bypass the dysfunctional gateway.

A new network option called dgd_ping_time will allow the system administrator to configure the time interval between the periodic ICMP echo request/reply exchanges (ping) in units of seconds. The network option dgd_ping_time can be displayed and changed by the `no -o` command and is set to 5 seconds by default. The `no` command output in the following example shows the value for dgd_ping_time on a system where this specific network option is set to the default of 5:

```
# no -o dgd_ping_time
dgd_ping_time = 5
```

You should include an appropriate `no` command line in the /etc/rc.net file to ensure that a value for this network option which deviates from the default stays in effect across reboots of your system.

Active dead gateway detection will be off by default and it is recommended to be used only on machines that provide critical services and have high availability requirements. Since active DGD imposes some extra network traffic, network sizing and performance issues have to receive careful consideration. This applies especially to environments with a large number of machines connected to a single network.

Active DGD operates on a per-route basis, and it is turned on by the new parameter argument -active_dgd of the `route` command. The following example shows how the `route` command is used to add a new default route through the 9.3.240.58 gateway with a user-configurable cost of 2 and which is under the surveillance of active DGD:

```
# route add default 9.3.240.58 -active_dgd -hopcount 2
```

The `netstat -C` command list the routes defined to the system including their current and user-configurable cost. The new flag A, as listed for the default route through the 9.3.240.58 gateway, indicates that the active DGD for this particular route is turned on.

```
# netstat -C
Routing tables
Destination      Gateway           Flags   Refs     Use  If   Cost Config_Cost

Route Tree for Protocol Family 2 (Internet):
default          9.3.240.59        UG         3   104671  tr1     2         2 =>
default          9.3.240.58        UGA        0        0  tr1     2         2
9.3.240/24       server2           U         32    67772  tr1     0         0
127/8            loopback          U          6     1562  lo0     0         0
```

```
Route Tree for Protocol Family 24 (Internet v6):
::1             ::1              UH        0        0  lo0     0         0
```

The kernel will keep a list of all the gateways that are subject to active DGD. Each time dgd_ping_time seconds passes, all the gateways on the list will be pinged. A pseudo-random number is used to slightly randomize the ping times. If several hosts on the same network use active DGD, the randomized ping times ensure that not all of the hosts ping at exactly the same time. If any gateways fail to respond, they will be pinged several times repeatedly with a 1 second pause between pings. The total number of times they are pinged will be determined by the dgd_packets_lost network option. This network option was already introduced in Section 5.2.2.2, “Passive Dead Gateway Detection” on page 194 above, but note that this option has a slightly different meaning for passive DGD compared to active DGD.

The network option dgd_packets_lost in passive DGD refers to the number of TCP packets lost (if any) in the course of data transmission, whereas for active DGD, the option is specifically related to the packets used in an ICMP echo request/reply exchange (ping) to sense the state of the gateways that are under the surveillance of active DGD.

If the gateway does not respond to any of these pings, it will be considered to be down, and the costs of all routes using that gateway will be increased to the maximum value, that is defined to be MAX_RT_COST. MAX_RT_COST in turn is equal to INT_MAX=2147483647, the highest possible value for an integer. These definitions can be examined in the /usr/include/net/route.h and the /usr/include/sys/limits.h header files, which are optionally installed on your system as part of the bos.adt.include fileset.

The gateway will remain on the list of gateways to be pinged, and if it responds at any point in the future, the costs on all routes using that gateway will be restored to their user-configured values.

Passive DGD does not decrease the cost on any route for which active detection is being done, as active detection has its own mechanism for recovery when a gateway comes back up. However, passive DGD is allowed to increase the cost on a route for which active detection is in use, as it is quite likely that passive detection will discover the outage first when TCP traffic is being sent.

5.2.2.4 DGD network options and command changes

Four new network options are defined for Dead Gateway Detection and all of them are runtime attributes that can be changed at any time. Table 14 gives details of the attributes of these options:

Table 14. Network options for Dead Gateway Detection

Network Option	Default	Description
dgd_packets_lost	3	Specifies how many consecutive packets must be lost before Dead Gateway Detection decides that a gateway is down.
dgd_ping_time	5	Specifies how many seconds should pass between pings of a gateway by active Dead Gateway Detection.
dgd_retry_time	5	Specifies how many minutes a route's cost should remain raised when it has been raised by Passive Dead Gateway Detection. After this many minutes pass, the route's cost is restored to its user-configured value.
passive_dgd	0	Specifies whether Passive Dead Gateway Detection is enabled. A value of 0 turns it off, and a value of 1 enables it for all gateways in use.

If the customized DGD network attributes are intended to be permanent, the system administrator must include the appropriate `no` command in /etc/rc.net. Otherwise, the customized network options will be reset to their default during a system boot. For example, if you like to turn on passive DGD permanently, you have to include the following line in /etc/rc.net:

```
# The following no command enables passive Dead Gateway Detection
# after each system boot
if [ -f /usr/sbin/no ] ; then
        /usr/sbin/no -o passive_dgd=1
fi
```

5.2.2.5 DGD sample configuration

Figure 54 on page 202 depicts a basic system environment that will be used throughout this section to give an example for active Dead Gateway Detection. Server1 attached to the Token Ring network 9.3.240.0 (netmask 255.255.255.0) has two default routes to the Client1 computer in the Ethernet segment 10.47.0.0 (netmask 255.255.0.0). One route goes through the Gateway1, that has a token ring interface tr0 with the IP address 9.3.240.58 and an Ethernet interface en0 with the IP address 10.47.1.1. The second route uses Gateway2, which is configured to have a token ring interface tr0 with the IP address 9.3.240.59 and an Ethernet interface en0 with the IP address 10.47.1.2. The `no -o ipforwarding=1` command was used on both gateway systems to enable the gateway function. The Ethernet interface of Client1 has the IP address of 10.47.1.3. Server1 and Client1 run AIX 5L and on both systems the `no -o tcp_pmtu_discover=0` and the `no -o udp_pmtu_discover=0` command were used to disable dynamic PMTU discovery interference with multipath routing. Also, on both computers, the passive_dgd network option was set to 1 by the `no -o passive_dgd=1` command to enable passive DGD. It is not required to have passive DGD enabled in order to use the active DGD function, but for TCP-based network traffic, passive DGD may initiate the failover to the backup gateway earlier than active DGD normally would. If the network traffic is not TCP-based, then the active pinging of the gateways by active DGD will get the information about the state of the gateway faster than passive DGD potentially could get it through the expiration of the ARP cache entry.

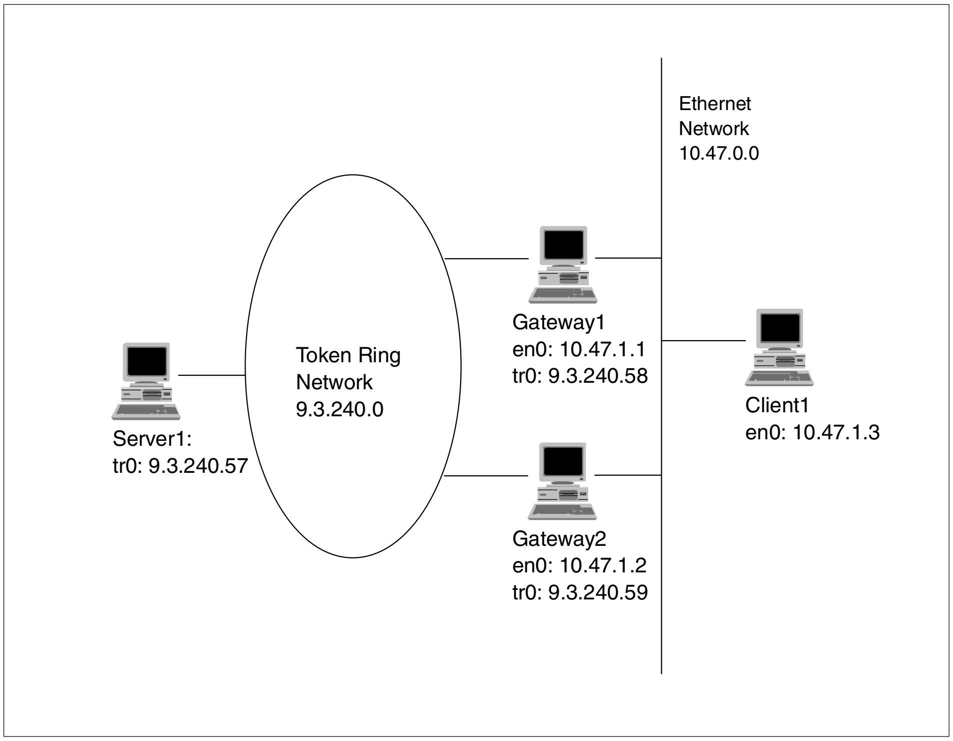


Figure 54. DGD sample configuration

For Server1 and Client1, the default routes were configured through the SMIT menu Add Static Route, which you can access directly by the fastpath mkroute using the command `smit mkroute`. The default routes were defined to have the same user-configurable cost, but to use different gateways. The underlying SMIT script, which is associated with the Add Static Route SMIT task, uses the `chdev` command for the inet0 device to permanently define routes. The `route` command affects only the current kernel routing table and all additions and changes applied to the routing table will be lost after a system boot.

The `netstat -Cn` command output, shown in the following lines, reflects the routing table entries which were made. The reference count for both gateway routes is 2, because after the setup of the routing environment, four telnet sessions to Client1 were initiated from Server1. Multipath routing ensured (through cyclic multiplexing) that the sessions are divided evenly among the two default routes. The flag A in the Flags column indicate that active DGD is set for both default routes:

```
# netstat -Cn
Routing tables
Destination      Gateway           Flags   Refs     Use  If   Cost Config_Cost

Route Tree for Protocol Family 2 (Internet):
default          9.3.240.58        UGA       2      154  tr1     2         2 =>
default          9.3.240.59        UGA       2      177  tr1     2         2
9.3.240/24       9.3.240.57        U         4      160  tr1     0         0
127/8            127.0.0.1         U         4      190  lo0     0         0

Route Tree for Protocol Family 24 (Internet v6):
::1              ::1               UH        0        0  lo0     0         0
```

To test the active DGD feature, the `ifconfig tr0 down` command was used to disable the gateway function of Gateway1. After the takeover has been completed, `netstat -Cn` returns the following output:

```
# netstat -Cn
Routing tables
Destination      Gateway           Flags   Refs     Use  If   Cost Config_Cost

Route Tree for Protocol Family 2 (Internet):
default          9.3.240.59        UGA       4      604  tr1     2         2 =>
default          9.3.240.58        UGA       0      245  tr1   MAX         2
9.3.240/24       9.3.240.57        U         5      479  tr1     0         0
127/8            127.0.0.1         U         0      190  lo0     0         0

Route Tree for Protocol Family 24 (Internet v6):
::1              ::1               UH        0        0  lo0     0         0
```

The reference count for the route through Gateway1 has dropped from 2 to 0 and both associated connections are now handled by the backup route through Gateway2. In order to mark the dysfunctional gateway as unusable, the current cost of that route was set to the maximum possible value, as indicated by the keyword MAX.

5.2.3 User interface for multipath routing and DGD

System management tasks that are related to the new multipath routing and DGD features are supported on the command line interface level by new parameters and flags to the `route` and `netstat` command.

Two parameters were added to the `route` command in order to support the multipath routing feature. The -hopcount argument of the route parameters requires a positive integer as variable value. The variable value refers to the user-configurable cost for a given route and supposedly relates to the maximum number of gateways in the route. However, the ultimate objective in introducing the user-configurable costs for a route is to implement a priority hierarchy among the defined routes. The new -if argument must be supplemented by a variable which takes a defined network interface as the variable value. The -if argument specifies the interface to associate with a route, so that packets will be sent using this interface when the given route is chosen.

In addition to the two new parameters which support multipath routing, one parameter was specifically added to the `route` command to implement active DGD. The name of this parameter is active_dgd, and whenever this parameter is given during the definition of a route, active DGD will be enabled for the particular route.

Note that the `route` command only changes the kernel routing table but does not permanently change the attributes of the inet0 device.

To preserve route definitions across system boot processes, you have to change the attributes of the inet0 device either by `chdev` command or with the aid of the Add Static Route SMIT menu.

Table 15 provides an overview of the new parameters added to the `route` command that support the new routing features in AIX 5L:

Table 15. route command parameters for multipath routing and DGD

Parameter argument	Argument variable	Description
-active_dgd	NA	Enables active DGD on given route.
-hopcount	`n`	Specifies relative cost of a given route, if the `n` variable is a positive integer.
-if	`ifname`	Specifies the interface `ifname` (en0, tr0, ...) to associate with this route, so that packets will be sent using this interface when this route is chosen.

The new -C flag (as shown in Table 16) was added to the `netstat` command to provide additional routing table information. The `netstat -C` command displays the routing tables, including the user-configured and current costs of each route.

The current cost is either dynamically determined during the route definition process and reflects the number of gateways in the route or it is equal to the user-configured cost. The user-configurable costs can be set just for the routes in the current kernel routing table using the `route` command with the -hopcount parameter, or they are permanently defined by the appropriate `chdev` command as attributes of the inet0 device. The current cost may be different than the user-configured cost if Dead Gateway Detection has changed the cost of the route.

Table 16. New netstat command flags

Command	Description
netstat -C	Shows the routing tables, including the user-configured and current costs of each route. The user-configured cost is set using the -hopcount flag of the `route` command. The current cost may be different than the user-configured cost if Dead Gateway Detection has changed the cost of the route.

More details about the command line interfaces for multipath routing and DGD are given in Section 5.2.2.2, “Passive Dead Gateway Detection” on page 194 and Section 5.2.2.3, “Active Dead Gateway Detection” on page 198 and in the standard AIX documentation library.

In addition to the command line interface for configuration and administration of the multipath routing and DGD feature AIX 5L provides graphical user interface support for the relevant systems management tasks through SMIT and the Web-based System Manager tool.

The menus of the System Management Interface Tool (SMIT), which assists the addition of a static route for IP Version 4 (IPV4) and for IP Version 6 (IPV6), were changed to accommodate the new user-configurable metric (cost) option, to account for the added flexibility to associate a particular interface with a specific route, and to support Dead Gateway Detection.

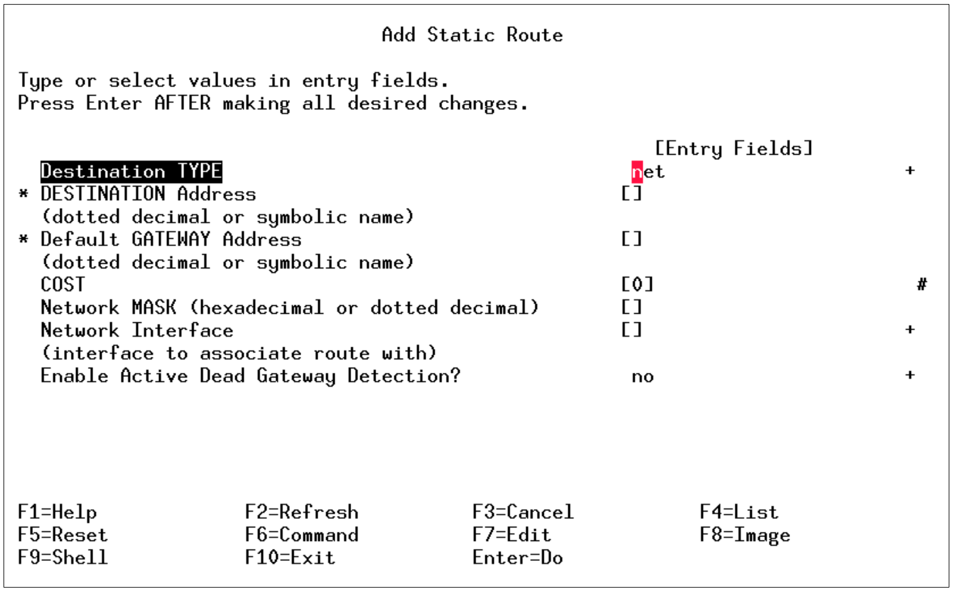
In the SMIT menus Add a Static Route and Add an IPV6 Static Route, three new fields were added to take input for the underlying SMIT script, that in turn uses the `chdev` command to set the route attribute for the inet0 internet network extension. Refer to Table 17 for further details about the field definition:

Table 17. Static Route and Add an IPV6 Static Route SMIT menu new fields

Field	Description
Network Interface (interface to associate route with)	Specifies the interface (en0, tr0 ...) to associate with this route so that packets will be sent using this interface when this route is chosen.
COST	User-configurable distance metric for route.
Enable Active Gateway Detection	Enables Active DGD on the route.

In order to add an alternate default route to your system, you will have to use the keyword default as destination address in the SMIT input panel.

The SMIT fastpaths mkroute and mkroute6 bring you directly to the SMIT menus for IPV4 and IPV6 (that are related to the systems management task) to add a static route. Figure 55 depicts the SMIT menu Add Static Route which supports the IPV4 specific task.

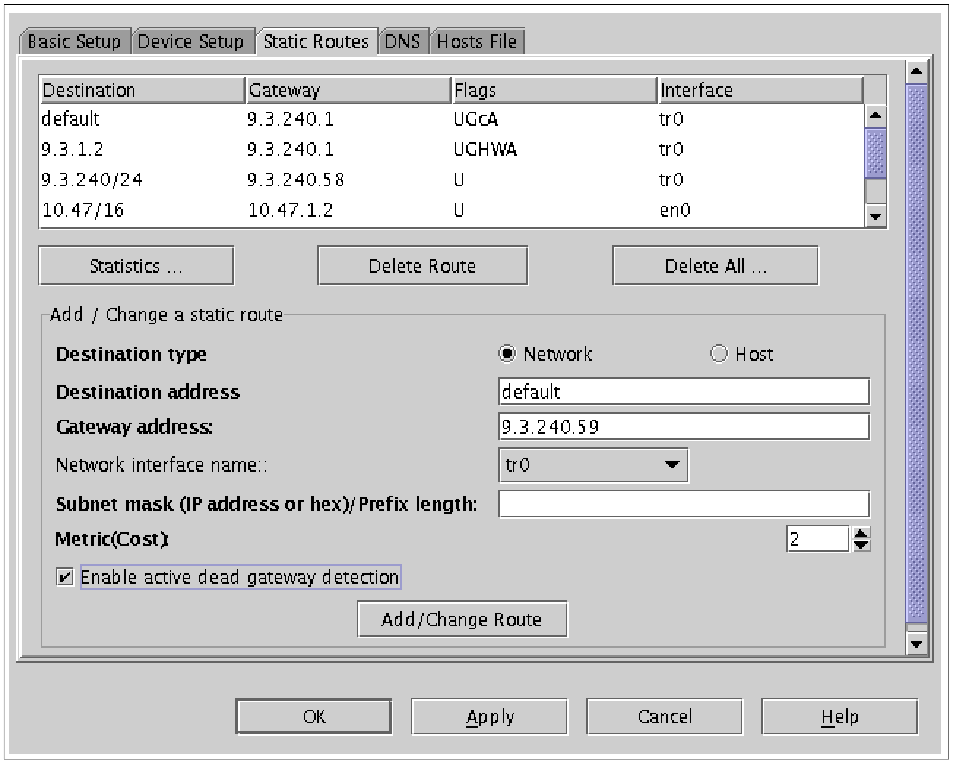


```
                              Add Static Route

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

                                                       [Entry Fields]
  Destination TYPE                                   net                     +
* DESTINATION Address                               []
  (dotted decimal or symbolic name)
* Default GATEWAY Address                           []
  (dotted decimal or symbolic name)
  COST                                              [0]                       #
  Network MASK (hexadecimal or dotted decimal)      []
  Network Interface                                 []                       +
  (interface to associate route with)
  Enable Active Dead Gateway Detection?              no                      +




F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 55. Add Static Route SMIT menu

The Web-based System Manager environment for multipath routing and DGD is accessible through the following sequence of menu selections on the Web-based System Manager console:

Select **Network** --> **TCPIP (IPv4 and IPv6)** --> **Protocol Configuration** --> **TCP/IP**. Select --> **Configure TCP/IP** --> **Advanced Methods**. Click **Static Routes**. Complete the following in the Add/Change a static route menu: Destination Type, Gateway address, Network interface name (drop-down menu), Subnet mask, Metric (Cost), and the Enable active dead gateway detection check box. Click **Add/Change Route**. Figure 56 shows the Web-based System Manager menu for static route management related tasks.



Destination	Gateway	Flags	Interface
default	9.3.240.1	UGcA	tr0
9.3.1.2	9.3.240.1	UGHWA	tr0
9.3.240/24	9.3.240.58	U	tr0
10.47/16	10.47.1.2	U	en0

Add / Change a static route

Destination type ● Network ○ Host

Destination address default

Gateway address: 9.3.240.59

Network interface name:: tr0

Subnet mask (IP address or hex)/Prefix length:

Metric(Cost): 2

☑ Enable active dead gateway detection

Figure 56. Web-based System Manager menu for static route management

5.3 Virtual IP address support

In previous AIX releases an application had to bind to a real network interface in order to get access to a network or network services. If the network became inaccessible or the network interface failed, the application's TCP/IP session was lost, and consequently, the application was no longer available.

To overcome application availability problems as described, AIX 5L offers support for virtual IP addresses (VIPA) for IPv4 and IPv6. The VIPA related code is part of the bos.net.tcp.client fileset, which belongs to the BOS.autoi and MIN_BOS.autoi system bundles, and therefore will always be installed on your AIX system.

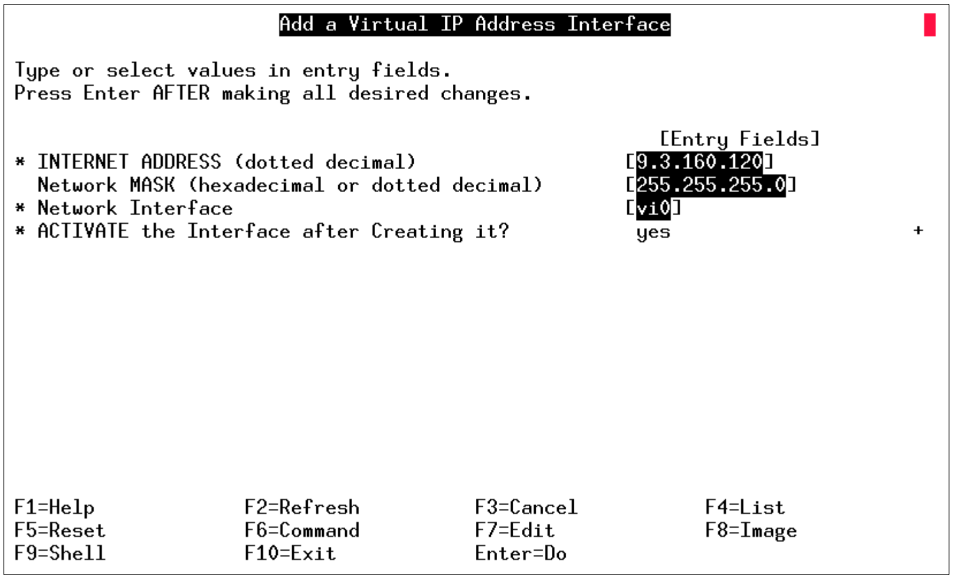
With VIPA, the application is bound to a virtual IP address, not a real network interface that can fail. When a network or network interface failure is detected (using routing protocols or other schemes), a different network interface can be used by modifying the routing table. If the re-routing occurs fast enough, then TCP/IP sessions will not be lost.

A traditional IP address is associated with a specific network adapter. Virtual IP address are supported by a network interface that is not associated with any particular network adapter. The operating system will interact with a virtual interface through the interface specific device special file. The device special file will be located in the /dev directory and the device name consists of the two letter abbreviation for virtual interface (vi) and an appended interface number. The VIPA system management tasks are supported by the appropriate changes and additions to the interface related high level operating system commands `mkdev`, `chdev`, `rmdev`, `lsdev`, `lsattr`, `ifconfig`, and `netstat`. Also, all VIPA management tasks are covered by SMIT and the Web-based System Manager tool.

The following example shows how to configure a virtual interface (vi0) for the internet address 9.3.160.120 with the netmask of 255.255.255.0, using the high level command `mkdev`. The virtual interface belongs to the device class *`if`*, the subclass VI and is of the device type vi.

```
# mkdev -c if -s VI -t vi -a netaddr='9.3.160.120' -a netmask='255.255.255.0' -w 'vi0' -a
state='up'
```

If the support of SMIT to configure a virtual interface is wanted, you can use the SMIT fastpath mkinetvi (`smit mkinetvi` command) to get access to the relevant SMIT menu, as shown in Figure 57 on page 209.



```
                       Add a Virtual IP Address Interface

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

                                                      [Entry Fields]
* INTERNET ADDRESS (dotted decimal)                 [9.3.160.120]
  Network MASK (hexadecimal or dotted decimal)      [255.255.255.0]
* Network Interface                                 [vi0]
* ACTIVATE the Interface after Creating it?          yes                    +

F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 57. Add a Virtual IP Address Interface SMIT menu

The `lsdev` command will list the virtual network interface and the traditional network interfaces as members of the interface class `if`:

```
# lsdev -HCc if -F 'name class subclass type status description'
name class subclass type status    description

en0  if    EN       en   Available Standard Ethernet Network Interface
en1  if    EN       en   Defined   Standard Ethernet Network Interface
et0  if    EN       ie3  Defined   IEEE 802.3 Ethernet Network Interface
et1  if    EN       ie3  Defined   IEEE 802.3 Ethernet Network Interface
lo0  if    LO       lo   Available Loopback Network Interface
tr0  if    TR       tr   Available Token Ring Network Interface
vi0  if    VI       vi   Available Virtual IP Address Network Interface
```

Also, the `netstat` command reports the existence of the newly defined interface:

```
# netstat -in
Name  Mtu   Network     Address            Ipkts Ierrs    Opkts Oerrs  Coll
lo0   16896 link#1                          191957     0   191961     0     0
lo0   16896 127         127.0.0.1           191957     0   191961     0     0
lo0   16896 ::1                             191957     0   191961     0     0
en0   1500  link#2      0.6.29.c5.1d.68      28048     0     2580     0     0
en0   1500  10.47       10.47.1.2            28048     0     2580     0     0
tr0   1492  link#3      0.6.29.be.d2.a2     155075     0    42520     0     0
tr0   1492  9.3.240     9.3.240.58          155075     0    42520     0     0
vi0   0     link#4                               0     0        0     0     0
vi0   0     9.3.160     9.3.160.120              0     0        0     0     0
```

System administrators can use the `lsattr` command to examine the device attributes for virtual network interfaces, and the `ifconfig` command is enabled to handle the new network interface type:

```
# lsattr -El vi0
netaddr   9.3.160.120   N/A                                         True
state     up            Standard Ethernet Network Interface         True
netmask   255.255.255.0 Maximum IP Packet Size for This Device      True
netaddr6                Maximum IP Packet Size for REMOTE Networks True
alias6                  Internet Address                            True
prefixlen               Current Interface Status                    True
alias4                  TRAILER Link-Level Encapsulation            True

# ifconfig vi0
vi0: flags=84000041<UP,RUNNING,64BIT>
        inet 9.3.160.120 netmask 0xffffff00
```

As indicated by the example, virtual network interfaces are similar to traditional network interfaces in most ways. A virtual interface is apparently configured and customized using the same system management commands as for real network interfaces. A system administrator has the option to define multiple virtual interfaces and can choose to associate aliases with them.

One of the main advantages of choosing a virtual device, as opposed to defining aliases to real network interfaces, is that a virtual device can be brought up or down separately without having any effect on the real interfaces of a system. Furthermore, it is not possible to change the address of an alias (aliases can only be added and deleted), but the address of a virtual interface can be changed.

For applications and processes, the difference between a real and a virtual IP address is completely transparent, and therefore they can bind to a virtual interface just like to any other network interface.

However, a virtual address takes precedence over other interface addresses in source address selection if an application locally binds to a wildcard address. (Telnet would be an example for an application having this binding characteristic.) This enables applications to make use of VIPA without any changes. In situations where there are multiple virtual addresses, the address of the first virtual interface on the list of interfaces will be chosen.

Since a virtual interface does not have a device associated with it, no route pointing to this interface will be added at configuration time. It is not possible to add routes on your local system that point to a virtual interface.

The gated process, which provides the gateway routing function in AIX, does not add a route for any virtual interface; also, gated will not send advertisements over the virtual interface, like it does for the other interfaces. However, gated does include the virtual interface in its advertisement to its neighboring routers, which enable these routers to add a host route for the virtual address.

Because the virtual interface does not relate to any real network interface, packets will never go in or out of the interface, and, consequently, the packet count for the virtual interface will always be zero. For the same reason, the virtual network interface will not respond to ARP requests.

Considering all the information given in the paragraphs above, you can complete the description of the data and control flow for network traffic through a virtual interface as follows:

When an application locally bound to a wildcard address connects to a remote host, a VIPA is selected as its source address. The interface the outgoing packet actually uses is determined by the route table based solely on the destination address. The remote host receives the packet and then tries to send a response to the host using the virtual address. The remote host and all routers along the way must have a route that will send the packet with the virtual address to one of the network interfaces of the host with the virtual address.

Either gated running on the host with VIPA will send information, which enables the adjacent routers and the remote host to add a host route for the virtual address, or the intermediate routes have to be configured manually along the route.

5.4 Network Buffer Cache dynamic data support

The Network Buffer Cache (NBC) was introduced in AIX Version 4.3.2. to improve the performance of network file servers, such as the Web server, FTP server, and SMB server. In AIX Version 4.3.3, the NBC design was improved to allow the use of 256 MB private memory segments for caching additional data. This design was chosen to eliminate the need to use pinned kernel heap and the network memory pools that had size restrictions. The use of private segments allows a system limit, set by the `no` option nbc_pseg, of 2**20 segments. A setting should not exceed 2**19 because file systems, processes, and other applications also require segments. Therefore, the total amount of data can be 256*2**19 or the limit set by the nbc_pseg_limit option. Only as much physical memory is consumed as data exists in a segment.

With the same AIX release, a second key for the cache access mechanism was introduced to support the HTTP GET kernel extension in conjunction with the Fast Response Cache Architecture (FRCA).

AIX 5L further enhances the Network Buffer Cache kernel extension to facilitate a dynamic data buffer cache and to support an expiration time per cache object. Also, internal memory usage code optimizations were applied to expand the caching capacity of NBC.

Within the scope of the kernel address space, NBC uses network memory for caching data which is accessed frequently through networks. For example, by enabling and using the NBC, the IBM HTTP Server can cache frequently referenced Web pages to eliminate the repetitive costs of moving data among the file buffers, user buffers, and the networking buffers. NBC, as a kernel component, provides kernel services for its users to take advantage of the network buffer cache. In the NBC context, the term users refer to other kernel components or kernel extensions. Application level users have to go through APIs provided by those kernel components or kernel extensions to interact with the NBC.

There are two ways for an application to exploit the NBC feature:

- Using the send_file() system call.
- Using the Fast Response Cache Architecture (FRCA) API.

The new AIX 5L NBC enhancements are only accessible for applications through the FRCA API.

5.4.1 Dynamic data buffer cache

In previous AIX releases, there is only one type of cache object that is cached in the NBC. Each cache object held copies of original data already existing in the file subsystem and, therefore, the related cache object type was named NBC_NAMED_FILE. Since the NBC was designed to improve the performance of typical network file servers, this single cache type was sufficient to improve the performance of Web Servers in static Web page access scenarios. However, more and more Web pages consist of dynamically generated data and contents. These Web pages are not necessarily saved in files, and they are much more volatile than static file pages. For these reasons, NBC's capability was expanded to accommodate dynamically generated data (for example, dynamic pages or page fragments) generated by user level applications.

Beginning with AIX 5L, NBC offers support for caching data buffers created and given by kernel users. The most prominent kernel user which depends on NBC is the FRCA kernel extension. FRCA utilizes the NBC and provides a platform independent API for Web servers to add and delete dynamic data buffer caches on AIX systems. FRCA also accesses the NBC cache whenever an HTTP GET request can be satisfied by the cache in the system interrupt context. The new NBC features provides adequate kernel services for FRCA to improve the overall IBM HTTP Web Server performance.

To the NBC, the dynamic data buffer cache is a group of buffers that were allocated and given by other kernel extensions or kernel components. These buffers are in the mbuf chain format for keeping and accessing from the NBC. The buffers are pinned in memory, and the cache object creators have the responsibility to keep these memory pinned for the lifetime of the cache. These buffers can be allocated from regular mbuf pool (m_get(), net_malloc(), etc), from kernel heap (xmalloc()), or from private segments. When the buffers are given to the NBC for caching, it is the responsibility of the kernel extension or kernel component using NBC to build up an mbuf chain and set up the mbuf headers correctly for the corresponding buffers. The private segments do not have to be mapped by users at the time of adding, but they have to be pinned all the time.

The buffer cache is subject to the previously existing NBC flushing control. All caches are on the LRU (least recently used) list in the NBC. When the total cache size reaches the NBC system limits (multiple configured network options), any buffer cache may get removed from the NBC just like other caches.

A new cache type, NBC_FRCA_BUF, will be the cache type for the dynamic buffer cache associated with the FRCA. A primary key for type NBC_FRCA_BUF is generated and controlled by FRCA to uniquely identify each piece of cache within the NBC_FRCA_BUF type in the NBC.

Three new statistics were added for keeping track of the cache objects of the new cache type in the NBC.

1. Current total NBC_FRCA_BUF entries: Number of cache entries with NBC_FRCA_BUF type which currently exist in the cache.
2. Maximum total NBC_FRCA_BUF entries: Highest number of cache entries with NBC_FRCA_BUF type that has ever been created in cache.
3. Current total user buffer size: Byte count of the total buffer size currently in the NBC that is not accounted in either the mbuf pool memory or the private segments.

Use the `netstat -c` command to display the NBC statistics which are related to the new cache type, as in the following example:

```
# netstat -c

Network Buffer Cache Statistics:
-------------------------------
Current total cache buffer size: 256
```

```
Maximum total cache buffer size: 256
Current total cache data size: 0
Maximum total cache data size: 0
Current number of cache: 1
Maximum number of cache: 1
Number of cache with data: 1
Number of searches in cache: 1
Number of cache hit: 0
Number of cache miss: 1
Number of cache newly added: 1
Number of cache updated: 0
Number of cache removed: 0
Number of successful cache accesses: 0
Number of unsuccessful cache accesses: 0
Number of cache validation: 0
Current total cache data size in private segments: 0
Maximum total cache data size in private segments: 0
Current total number of private segments: 0
Maximum total number of private segments: 0
Current number of free private segments: 0
Current total NBC_NAMED_FILE entries: 0
Maximum total NBC_NAMED_FILE entries: 0
Current total NBC_FRCA_BUF entries: 1
Maximum total NBC_FRCA_BUF entries: 1
Current total user buffer size: 131072
```

5.4.2 Cache object-specific expiration time

In previous AIX releases the NBC provides cache invalidation based on a time limit specified by the cache access client, not the creator. In other words, once the cache is loaded, it is assumed to be good; the frequency of invalidation checking or updating is up to the client's tolerance. This is acceptable with a cache object that is expected to be reasonably static. For dynamic data, however, it is necessary to support an expiration time per cache object.

In AIX 5L, the NBC will invalidate the buffer cache according to a time-to-live value specified by the creator. Each buffer cache object has a live-time limit specified when it is first added to the NBC. When the cache is accessed, and if the age of the cache object exceeds the live-time limit, the NBC will remove this particular piece of cache and return NULL to the client. The client can also specify a time to make sure that the cache object is not older than expected. If the cache is older than the client's time limit, the NBC will return a NULL, the cache object, however, is still considered valid. The resolution for both time limit values is in units of seconds.

5.5 HTTP GET kernel extension enhancements

Starting with AIX Version 4.3.2, the Fast Response Cache Architecture (FRCA) with the HTTP GET kernel extension was introduced to AIX.

AIX 5L improves the FRCA HTTP GET kernel extension to support HTTP 1.1 persistent connections. Other enhancements to the HTTP GET kernel extension include an external 64-bit ready API (to give every user space program access to the existing function of the HTTP GET kernel extension) and additional support for a new cache type based on memory buffers.

The FRCA utilizes the AIX Network Buffer Cache (NBC) to greatly improve the Web server response time for HTTP GET requests. Figure 58 illustrates the FRCA data flow for an incoming request, which refers to a Web page located on a given Web server. The HTTP GET requests are intercepted and the response is sent directly from the AIX NBC on the input interrupt. No data is copied between kernel and user space, and no user context switch is necessary. If the HTTP GET request can be serviced by the engine, the user space Web server is not contacted and never sees the request. GET requests that cannot be serviced by the kernel engine are passed to the user space Web server.

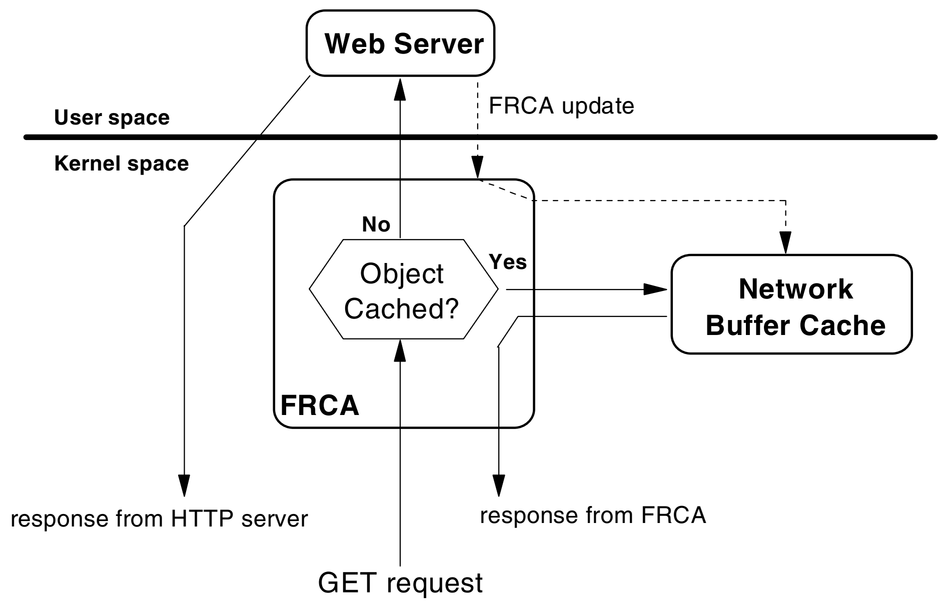


Figure 58. FRCA GET Data Flow

5.5.1 HTTP 1.1 persistent connections support

When AIX Version 4.3.2 was released, the predominant protocol in use was HTTP Version 1.0, with a major part of all requests referring to static content. Since then, a shift towards HTTP Version 1.1 has taken place. One of the major differences between the two versions of HTTP is the newer version's well-defined ability to handle multiple requests per connection while the previous version almost always closes a connection after a single request. Keeping a connection established for several requests allows the underlying transport layer protocol (TCP) to make better use of the available bandwidth by adapting to it over time.

The implementation of the HTTP GET kernel extension prior to AIX 5L either transparently redirected the pending request to a user space Web server, or it closed the connection after serving a single request.

With HTTP 1.1, a well-defined way of imposing entity boundaries on the exchanged HTTP data has been introduced, which will rapidly result in wide-spread use of persistent connections. For that reason, AIX 5L adds support for HTTP 1.1 persistent connections to the FRCA feature.

The support for persistent connections was added in a way that the HTTP GET kernel extension parses an incoming packet like before, but with only a little addition to the previously used code path: as the packet may contain multiple requests, it loops over the data and marks down the number of bytes from the input buffer that belong to the current request, the request's protocol version, and the absence of a connection header that includes the connection-token *close*.

On a per request basis, the kernel extension then acts according to the following rules:

- If the protocol version of the current request is not HTTP 1.1, then in case of a cache hit, it adds the response to the response buffer, sends the buffer and closes the connection, or in case of a cache miss, it sends the buffer and reconnects the connection to the user space Web server.
- If the protocol version of the current request is HTTP 1.1 and the *close* token has been detected, then in case of a cache hit, it adds the response to the response buffer, sends the buffer and closes the connection, or in case of a cache miss, it sends the buffer and reconnects the connection to the user space Web server.
- If the protocol version of the current request is HTTP 1.1 and the *close* token has not been detected, then in case of a cache hit, it adds the response to the response buffer, sends the buffer and keeps the connection in kernel space, or in case of a cache miss, it sends the buffer and reconnects the connection to the user space Web server.

5.5.2 External 64-bit FRCA API

Beginning with AIX 5L, an external 64-bit FRCA API is offered to allow more user space applications to exploit the existing function of the HTTP GET kernel extension.

The external API largely follows the structure of the internal API that consists of a set of functions to create and control an FRCA instance and another set of functions to create and fill a cache for a given FRCA instance. It is implemented as a layer on top of the internal API, which results in no changes to the previously existing HTTP GET kernel extension itself. The API will cover only the major part of the existing function of the HTTP GET kernel extension, but not all of it. Functions specific to the AIX platform, such as control over the amount of time that the HTTP GET kernel extension may spend on interrupt, will not be covered by the external API, and is left to the existing frcactrl program. The `frcactrl` command controls and configures the FRCA kernel extension and is documented in the AIX documentation library.

As the internal API continues to exist unchanged, all currently existing code developed against the internal API continues to work without a single change required.

AIX 5L provides a 64-bit version of the external API library to accommodate 64-bit applications. The following services that compose the external API are defined in /usr/include/net/frca.h. They are made available to user space applications through the libfrca.a library:

FrcaCtrlCreate	Creates a FRCA control instance.
FrcaCtrlDelete	Deletes a FRCA control instance.
FrcaCtrlStart	Starts the interception of TCP data connections for a previously configured FRCA instance.
FrcaCtrlStop	Stops the interception of TCP data connections for a FRCA instance.
FrcaCtrlLog	Modifies the behavior of the logging subsystem.
FrcaCacheCreate	Creates a cache instance within the scope of a FRCA instance.
FrcaCacheDelete	Deletes a cache instance within the scope of a FRCA instance.
FrcaCacheLoadFile	Loads a file into a cache associated with a FRCA instance.
FrcaCacheUnloadFile	Removes a cache entry from a cache that is associated with a FRCA instance.

5.5.3 Memory based HTTP entities caching

AIX 5L adds new services to the internal FRCA API to support caching of HTTP entities that are based on memory buffers and have no association with a file. The underlying NBC data cache provides the related NBC cache object type NBC_FRCA_BUF. The NBC_FRCA_BUF type in NBC refers the new dynamic data buffer cache, which is introduced with AIX 5L in order to expand the NBC caching capabilities to allow for Web pages with dynamically generated data and contents. For further details about the new NBC cache object type, refer to Section 5.4, “Network Buffer Cache dynamic data support” on page 211.

The previous implementation of the HTTP GET kernel extension only handled cache objects with content data that is tightly coupled to files in the local file system. This works fine in the case of static HTML pages that are stored in the local file system but it does not handle semi-dynamic content very well. The term “semi-dynamic” refers to content that is static to a certain degree (for example, a dynamically rendered HTML page that changes only once a minute, but has a reasonably higher access rate, such as once a second).

Although the semi-dynamic content could be written to a file, which in turn could be loaded into the HTTP kernel extension using the existing API, this involves some overhead, especially when the code that renders the content is executed on a different machine.

AIX 5L introduces a new service to the internal API to support caching of memory-based HTTP cache objects, which allows FRCA to handle caching of HTTP data that is not represented in the file system. One of the main purposes of the service is to accommodate application level cache managers residing on remote systems.

5.6 Packet capture library

Previous AIX operating system releases and AIX 5L offer the Berkeley Packet Filter (BPF) as a packet capture system. AIX 5L introduces, in addition to that, a Packet Capture Library (libpcap.a), which provides a high-level user interface to the BPF packet capture facility. The AIX 5L Packet Capture Library is implemented as part of the libpcap library, Version 0.4 from LBNL (Lawrence Berkeley National Laboratory).

The Packet Capture Library user-level subroutines interface with the existing BPF kernel extensions to allow users access for reading unprocessed network traffic. By using the new 24 subroutines of this library, users can write their own network-monitoring tools.

To accomplish packet capture, you have to follow the following procedure:

1. Decide which network device will be the packet capture device. Use the pcap_lookupdev subroutine to do this.
2. Obtain a packet capture descriptor by using the pcap_open_live subroutine.
3. Choose a packet filter. The filter expression identifies which packets you are interested in capturing.
4. Compile the packet filter into a filter program using the pcap_compile subroutine. The packet filter expression is specified in an ASCII string. Refer to Packet Capture Library Filter Expressions for more information.
5. After a BPF filter program is compiled, notify the packet capture device of the filter using the pcap_setfilter subroutine. If the packet capture data is to be saved to a file for processing later, open the previously saved packet capture data file, known as the savefile, using the pcap_dump_open subroutine.
6. Use the pcap_dispatch or pcap_loop subroutine to read in the captured packets and call the subroutine to process them. This processing subroutine can be the pcap_dump subroutine, if the packets are to be written to a savefile, or some other subroutine you provide.
7. Call the pcap_close subroutine to cleanup the open files and deallocate the resources used by the packet capture descriptor.

The current implementation of the libpcap library applies to IP Version 4 and only the reading of packets is supported. Applications using the Packet Capture Library subroutines must be run as root user. The files generated by libpcap applications can be read by `tcpdump` and vice-versa. However, the `tcpdump` command in AIX 5L does not use the libpcap library.

The Packet Capture Library libpcap.a is located in the /usr/lib directory after you have optionally installed the bos.net.tcp.server fileset. The bos.net.tcp.server fileset also provides the BPF kernel extension (/usr/lib/drivers/bpf), which is used by the libpcap subroutines. The library related header file pcap.h can be examined in the /usr/include/ directory, if you choose to install the bos.net.tcp.adt fileset. The libpcap sample code, which is also part of the bos.net.tcp.adt fileset, can be found in /usr/samples/tcpip/libpcap.

Further information about BPF can be found in *UNIX Network Programming, Volume 1: Networking APIs: Sockets and XTI*, Second Edition by W. Richard Stevens, 1998.

5.7 Firewall hooks enhancements

The AIX TCP/IP stack provides a way for other kernel extensions to insert themselves into the stack at specific points using hooks.

AIX 5L introduces two new firewall hooks which expand the functional spectrum of the already existing hooks for IP filtering and offers additional potential to improve the performance of firewalls. The new hooks will be part of the existing netinet kernel extension, which is packaged in bos.net.tcp.client.

The firewall hook routines provide kernel-level hooks for IP packet filtering enabling IP packets to be selectively accepted, rejected, or modified during reception, transmission, and decapsulation. These hooks are initially NULL, but are exported by the netinet kernel extension and will be invoked if assigned non-NULL values.

The following routines are included in AIX 5L as hooks for IP packet filtering:

- ip_fltr_in_hook
- ip_fltr_out_hook
- ipsec_decap_hook
- inbound_fw (new in AIX 5L)
- outbound_fw (new in AIX 5L)

The ip_fltr_in_hook routine is used to filter incoming IP packets, the ip_fltr_out_hook routine filters outgoing IP packets, and the ipsec_decap_hook routine filters incoming encapsulated IP packets.

The new AIX 5L inbound_fw and outbound_fw firewall hooks allow kernel extensions to get control of packets at the place where IP receives them. The outbound_fw hook was added exactly at the point where IP is entered when transmitting packets and the inbound_fw hook at the point where IP is called to process receive packets. The two new firewall hooks in AIX 5L are supplemented by additional methods to call the main IP code and to save firewall hook arguments in order to inject the filtered packets into the network at a later time. Also, some changes to existing routines were made alongside with the implementation of the new firewall hooks.

The code of following existing functions had been changed:

ipintr_noqueue2

The ipintr_noqueue2 hook itself and all references to ipintr_noqueue2 are removed. The function of ipintr_noqueue2 is provided by passing a null NDD parameter to ipintr_noqueue.

ipintr_noqueue
Most of ipintr_noqueue's code was moved to ipintr_noqueue_post_fw.

ip_output
Most of ip_output's code was moved to ip_output_post_fw.

The following new functions were added in AIX 5L to support the new firewall hooks:

ipintr_noqueue_post_fw
The ipintr_noqueue_post_fw hook contains the code that used to be in ipintr_noqueue and may be called from either ipintr_noqueue or from the firewall hook routine pointed at by inbound_fw.

inbound_fw_save_args
The inbound_fw_save_args hook gives a firewall hook routine, called through the inbound_fw variable, the ability to save a copy of the inbound_fw_args_t *args. This copy can be used to call ipintr_noqueue_post_fw at a later time.

inbound_fw_free_args
The inbound_fw_free_args hook frees a inbound_fw_args_t created by inbound_fw_save_args.

ip_output_post_fw
The ip_output_post_fw hook contains largely the code that used to be in ip_output.

outbound_fw_save_args
The outbound_fw_save_args hook creates a copy of outbound_fw_args_t *args. In doing so, it also makes sure all the things pointed at by *args remain valid indefinitely, either by copying or making references.

outbound_fw_free_args
The outbound_fw_free_args hook frees a outbound_fw_args_t created by outbound_fw_save_args. It also frees and removes references from anything pointed at by outbound_fw_args_t *args.

If inbound_fw is set, ipintr_noqueue, the IP input routine, calls inbound_fw and then exits. If not, ipintr_noqueue calls ipintr_noqueue_post_fw and then exits. If the inbound_fw hook routine wishes to pass the packet into IP, it can call ipintr_noqueue_post_fw. The inbound_fw hook may copy its args parameter by calling inbound_fw_save_args, and may free its copy of its args parameter by calling inbound_fw_free_args.

Similarly, ip_output calls outbound_fw if it is set, and calls ip_output_post_fw if not. The outbound_fw hook can call ip_output_post_fw if it wants to send a packet. The outbound_fw hook may copy its args parameter by calling outbound_fw_save_args, and later free its copy of its args parameter by calling outbound_fw_free_args.

5.8 Fast Connect enhancements

IBM AIX Fast Connect provides support for the SMB (Server Message Block) protocol to deliver File and Print Serving to PC clients. In AIX 5L, there are several improvements that will be discussed in this section.

At the time of writing, this feature is only available on the POWER platform.

5.8.1 Locking enhancements

Some applications require shared files between AIX server-based applications and PC client applications. The file server requires lock mechanisms to protect these files against multiple modifications at the same time. Because of this, Fast Connect implements UNIX locking, in addition to internal locking, to allow exclusions based on file locks taken by PC clients. . AIX 5L implements the following lock enhancements:

- Opportunistic locks take exclusive lock on the file when the exclusive opportunistic lock is granted and the file will be unlocked when the opportunistic lock is broken.
- SMB share modes are implemented with a UNIX lock consistent with the granted open mode and share mode.

5.8.2 Per share options

Several advanced features of AIX Fast Connect are available as per-share options. These options are encoded as bit fields within the sh_options parameter of each share definition. These options must be defined when the share is created with the `net share /add` command, or set through the SMIT file share panel.

Per-share options currently allowed by `net share /add` are shown in the Table 18.

Table 18. Per-share value options

Parameter	Values	Default	Description
sh_oplockfiles	(0,1)	1	If oplocks=1, enables opportunistic lock on this share
sh_searchcache	(0,1)	0	If searchcache=1, enables search caching on this share
sh_sendfile	(0,1)	0	If sendfile=1, enables sendfile API on this share
mode	(0,1)	1	Mode=1, enables read/write access mode=0, enables read only access

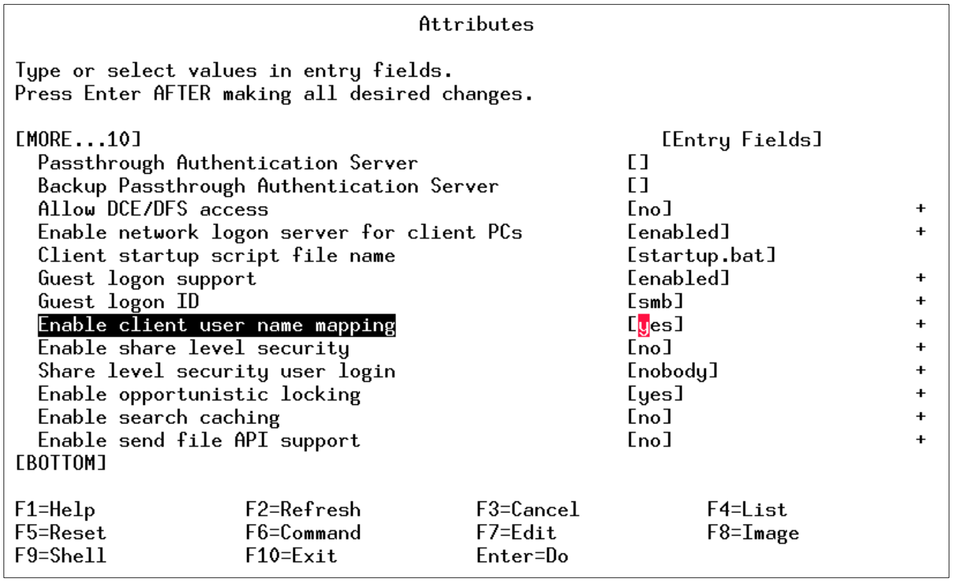
5.8.3 PC user name to AIX user name mapping

When a client tries to access resources on the server, it needs to establish an SMB/CIFS session. SMB/CIFS session setup can use either user level security or share level security.

In case of user level security, clients must present their user names. In previous Fast Connect releases, it was required that the user name match the one on AIX exactly. In many situations, this one-to-one mapping of user names is not possible.

AIX Fast Connect on AIX 5L allows the server administrators to configure the mapping of PC user names to AIX user names. When enabled, AIX Fast Connect tries to map every incoming client user name to a server user name, and then uses that server user name for further user authentication and AIX credentials.

Figure 59 shows the SMIT panel with the user name mapping option highlighted.



```
                                Attributes

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

[MORE...10]                                            [Entry Fields]
  Passthrough Authentication Server                 []
  Backup Passthrough Authentication Server          []
  Allow DCE/DFS access                              [no]                    +
  Enable network logon server for client PCs        [enabled]               +
  Client startup script file name                   [startup.bat]
  Guest logon support                               [enabled]               +
  Guest logon ID                                    [smb]                   +
  Enable client user name mapping                   [yes]                   +
  Enable share level security                       [no]                    +
  Share level security user login                   [nobody]                +
  Enable opportunistic locking                      [yes]                   +
  Enable search caching                             [no]                    +
  Enable send file API support                      [no]                    +
[BOTTOM]

F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 59. SMIT panel with user name mapping option highlighted

5.8.4 Windows Terminal Server support

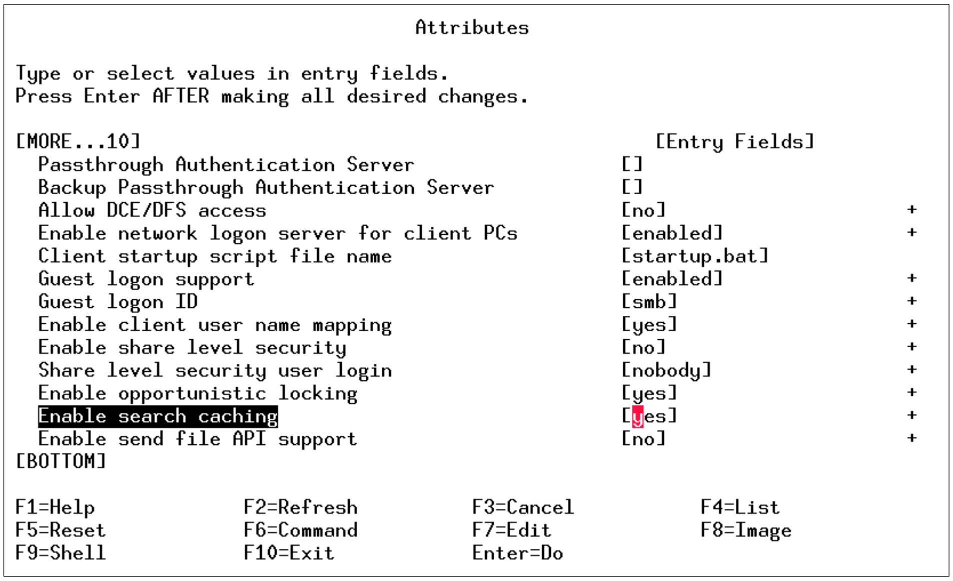
Windows Terminal Server from Microsoft and other similar products allow support of multiple users on one Windows NT machine. When a multiuser NT machine connects to a Fast Connect server for File and Print Services, it can use multiple SMB sessions over one transport session. In AIX 5L, Fast Connect allows multiple SMB sessions over one transport session. In previous releases, Fast Connect was limited to one SMB session per transport connection.

5.8.5 Search caching

Generally, file search operation requests from a PC client take large amounts of resources, and performance issues may arise if a large number of clients do file search operations at the same time.

In AIX 5L, Fast Connect allows you to enable search caching. If enabled, all the cached structures will compare their time stamps to the original files to check for modifications periodically. This feature improves file searching significantly.

Figure 60 shows the SMIT panel with the Enable search caching option highlighted. Search caching must be enabled for the share by enabling the per share option in addition to the global parameter shown.



```
                                    Attributes

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

[MORE...10]                                             [Entry Fields]
  Passthrough Authentication Server                  []
  Backup Passthrough Authentication Server           []
  Allow DCE/DFS access                               [no]                     +
  Enable network logon server for client PCs         [enabled]                +
  Client startup script file name                    [startup.bat]
  Guest logon support                                [enabled]                +
  Guest logon ID                                     [smb]                    +
  Enable client user name mapping                    [yes]                    +
  Enable share level security                        [no]                     +
  Share level security user login                    [nobody]                 +
  Enable opportunistic locking                       [yes]                    +
  Enable search caching                              [yes]                    +
  Enable send file API support                       [no]                     +
[BOTTOM]

F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 60. SMIT panel with Enable search caching option highlighted

Appendix A. AIX 5L POWER and Itanium-based fileset differences

The following list provides a reference of all of the packages and filesets that are currently part of the AIX 5L for the POWER platform distribution that are *not* included on the AIX 5L for Itanium-based systems distribution. This list provides a broad look at what components are not currently supported on the Itanium-based platform.

If a package or component is listed, such as X11.vsm or OpenGL, it indicates that all filesets that are part of that package or component are also not part of the Itanium-based system distribution. For example, you would not find the X11.vsm.rte fileset or the OpenGL.GL32 package on the Itanium-based distribution.

Filesets for unsupported and discontinued devices, such as the GXT100 Graphics Accelerator, or other POWER platform filesets, such as CHRP- or ISA-related devices, have been excluded from this list.

- Java.adt (note that Java130 is shipped on both platforms, with the exceptions listed)
- Java.rmi_iiop
- Java.rte
- Java.samples
- Java.security
- Java.swing
- Java130.ext.plugin
- Java130.ext.svk
- OpenGL
- PEX_PHIGS
- X11.Dt.xdt2cde
- X11.apps.custom
- X11.apps.msmit
- X11.apps.pm
- X11.compat.lib
- X11.vsm
- X11.x_st_mgr
- bos.64bit

- bos.INed
- bos.adt.debug
- bos.adt.prt_tools
- bos.atm
- bos.cns
- bos.compat
- bos.dlc.qllc
- bos.dlc.sdlc
- bos.loc.pc_compat
- bos.pmapi
- bos.rcs
- bos.rte.jfscomp
- bos.rte.printers
- bos.som
- bos.sysmgt.quota
- bos.twintail
- bos.up
- devices.mca
- ipfx.rte
- sysback

Appendix B. Special notices

This publication is intended to help AIX system administrators, developers, and support professionals understand the key technical differences between AIX 5L Version 5.0 and AIX Version 4.3. The information in this publication is not intended as the specifications for any programming interfaces that are provided by AIX 5LVersion 5.0. See the PUBLICATIONS section of the IBM Programming Announcement for AIX 5L for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter 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 the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM ("vendor") products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

This document contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples contain 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.

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

AIX
Current
IBM
Micro Channel
Redbooks
RS/6000
System/390
CT
DB2
Lotus
Netfinity
Redbooks Logo
SecureWay
SP

The following terms are trademarks of other companies:

Tivoli, Manage. Anything. Anywhere.,The Power To Manage., Anything. Anywhere.,TME, NetView, Cross-Site, Tivoli Ready, Tivoli Certified, Planet Tivoli, and Tivoli Enterprise are trademarks or registered trademarks of Tivoli Systems Inc., an IBM company, in the United States, other countries, or both. In Denmark, Tivoli is a trademark licensed from Kjøbenhavns Sommer - Tivoli A/S.

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

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

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

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

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

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

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

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

Appendix C. Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

C.1 IBM Redbooks

For information on ordering these publications see "How to get IBM Redbooks" on page 235.

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

C.2 IBM Redbooks collections

Redbooks are also available on the following CD-ROMs. Click the CD-ROMs button at ibm.com/redbooks for information about all the CD-ROMs offered, updates and formats.

CD-ROM Title	**Collection Kit Number**
IBM System/390 Redbooks Collection	SK2T-2177
IBM Networking Redbooks Collection	SK2T-6022
IBM Transaction Processing and Data Management Redbooks Collection	SK2T-8038
IBM Lotus Redbooks Collection	SK2T-8039
Tivoli Redbooks Collection	SK2T-8044
IBM AS/400 Redbooks Collection	SK2T-2849
IBM Netfinity Hardware and Software Redbooks Collection	SK2T-8046
IBM RS/6000 Redbooks Collection	SK2T-8043
IBM Application Development Redbooks Collection	SK2T-8037
IBM Enterprise Storage and Systems Management Solutions	SK3T-3694

C.3 Other resources

These publications are also relevant as further information sources:

- *UNIX Network Programming, Volume 1: Networking APIs: Sockets and XTI*, Second Edition by W. Richard Stevens, 1998.

You can access all of the AIX documentation through the Internet at the following URL:

- `www.ibm.com/servers/aix/library`

The following types of documentation are located on the documentation CD that ships with the AIX operating system:

- User guides
- System management guides
- Application programmer guides
- All commands reference volumes
- Files reference
- Technical reference volumes used by application programmers

C.4 Referenced Web sites

These Web sites are also relevant as further information sources:

- `http://www.kornshell.com` - Home page of the KornShell command and execution language
- `http://www.ptools.org/` - Home page of the light weight core file browser
- `http://www.ibm.com/java/jdk/aix/index.html` - AIX Java reference

How to get IBM Redbooks

This section explains how both customers and IBM employees can find out about IBM Redbooks, redpieces, and CD-ROMs. A form for ordering books and CD-ROMs by fax or e-mail is also provided.

- **Redbooks Web Site** **ibm.com**/redbooks

  Search for, view, download, or order hardcopy/CD-ROM Redbooks from the Redbooks Web site. Also read redpieces and download additional materials (code samples or diskette/CD-ROM images) from this Redbooks site.

  Redpieces are Redbooks in progress; not all Redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

- **E-mail Orders**

  Send orders by e-mail including information from the IBM Redbooks fax order form to:

	e-mail address
In United States or Canada	pubscan@us.ibm.com
Outside North America	Contact information is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Telephone Orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	Country coordinator phone number is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Fax Orders**

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	Fax phone number is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

This information was current at the time of publication, but is continually subject to change. The latest information may be found at the Redbooks Web site.

IBM Intranet for Employees

IBM employees may register for information on workshops, residencies, and Redbooks by accessing the IBM Intranet Web site at http://w3.itso.ibm.com/ and clicking the ITSO Mailing List button. Look in the Materials repository for workshops, presentations, papers, and Web pages developed and written by the ITSO technical professionals; click the Additional Materials button. Employees may access MyNews at http://w3.ibm.com/ for redbook, residency, and workshop announcements.

IBM Redbooks fax order form

Please send me the following:

Title	**Order Number**	**Quantity**

First name Last name

Company

Address

City Postal code Country

Telephone number Telefax number VAT number

□ Invoice to customer number

□ Credit card number

Credit card expiration date Card issued to Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

Abbreviations and acronyms

ABI	Application Binary Interface
ACL	Access Control List
AFPA	Adaptive Fast Path Architecture
AH	Authentication Header
ANSI	American National Standards Institute
API	Application Programming Interface
ARP	Address Resolution Protocol
ASR	Address Space Register
ATM	Asynchronous Transfer Mode
AuditRM	Audit Log Resource Manager
AUI	Attached Unit Interface
AWT	Abstract Window Toolkit
BIND	Berkeley Internet Name Daemon
BOS	Base Operating System
BLOB	Binary Large Object
BSC	Binary Synchronous Communications
CDE	Common Desktop Environment
CDLI	Common Data Link Interface
CD-R	CD Recordable
CE	Customer Engineer
CEC	Central Electronics Complex
CGE	Common Graphics Environment
CHRP	Common Hardware Reference Platform
CISPR	International Special Committee on Radio Interference
CLVM	Concurrent LVM
CMOS	Complimentary Metal-Oxide Semiconductor
COFF	Common Object File Format
CORBA	Common Object Request Broker
CSID	Character Set ID
DAD	Duplicate Address Detection
DASD	Direct Access Storage Device
DBE	Double Buffer Extension
DBCS	Double Byte Character Set
DCE	Distributed Computing Environment
DES	Data Encryption Standard
DHCP	Dynamic Host Configuration Protocol
DIT	Directory Information Tree
DMA	Direct Memory Access
DN	Distinguished Name
DNS	Domain Naming System
DS	Differentiated Service

DSA	Dynamic Segment Allocation
DSE	Diagnostic System Exerciser
DSMIT	Distributed SMIT
DTE	Data Terminating Equipment
EA	Effective Address
ECC	Error Checking and Correcting
EIA	Electronic Industries Association
EMU	European Monetary Union
EOF	End of File
ERRM	Event Response Resource Manager
ESID	Effective Segment ID
ESP	Encapsulating Security Payload
FCAL	Fibre Channel Arbitrated Loop
FCC	Federal Communication Commission
FDDI	Fiber Distributed Data Interface
FDPR	Feedback Directed Program Restructuring
FIFO	First In/First Out
FLASH EPROM	Flash Erasable Programmable Read-Only Memory
FLIH	First Level Interrupt Handler
FRCA	Fast Response Cache Architecture
FSRM	File System Resource Manager
GAI	Graphic Adapter Interface
GPR	General Purpose Register
GUI	Graphical User Interface
HACMP	High Availability Cluster Multi-Processing
HCON	IBM AIX Host Connection Program/6000
HFT	High Function Terminal
HostRM	Host Resource Manager
IAR	Instruction Address Register
ICCCM	Inter-Client Communications Conventions Manual
ICE	Inter-Client Exchange
ICElib	Inter-Client Exchange library
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IHV	Independent Hardware Vendor
IIOP	Internet Inter-ORB Protocol
IJG	Independent JPEG Group
IKE	Internet Key Exchange
ILS	International Language Support
IM	Input Method
INRIA	Institut National de Recherche en Informatique et en Automatique

IPL	Initial Program Load
IPSec	IP Security
IS	Integrated Service
ISA	Industry Standard Architecture
ISAKMP/Oakley	Internet Security Association Management Protocol
ISNO	Interface Specific Network Options
ISO	International Organization for Standardization
ISV	Independent Software Vendor
ITSO	International Technical Support Organization
I/O	Input/Output
JDBC	Java Database Connectivity
JFC	Java Foundation Classes
JFS	Journaled File System
KDC	Key Distribution Center
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LDIF	LDAP Directory Interchange Format
LFT	Low Function Terminal
LID	Load ID
LP	Logical Partition
LPI	Lines Per Inch
LPP	Licensed Program Products
LPR/LPD	Line Printer/Line Printer Daemon
LP64	Long-Pointer 64
LRU	Least Recently Used
LTG	Logical Track Group
LV	Logical Volume
LVCB	Logical Volume Control Block
LVD	Low Voltage Differential
LVM	Logical Volume Manager
L2	Level 2
MBCS	Multi-Byte Character Support
MCA	Micro Channel Architecture
MDI	Media Dependent Interface
MII	Media Independent Interface
MODS	Memory Overlay Detection Subsystem
MP	Multiple Processor
MPOA	Multiprotocol Over ATM
MST	Machine State
MWCC	Mirror Write Consistency Check
NBC	Network Buffer Cache
ND	Neighbor Discovery
NDP	Neighbor Discovery Protocol
NFS	Network File System
NHRP	Next Hop Resolution Protocol
NIM	Network Installation Management
NIS	Network Information System
NL	National Language
NLS	National Language Support

NTF	No Trouble Found
NVRAM	Non-Volatile Random Access Memory
OACK	Option Acknowledgment
ODBC	Open DataBase Connectivity
ODM	Object Data Manager
OEM	Original Equipment Manufacturer
OLTP	Online Transaction Processing
ONC+	Open Network Computing
OOUI	Object-Oriented User Interface
OSF	Open Software Foundation, Inc.
PCI	Peripheral Component Interconnect
PDT	Paging Device Table
PEX	PHIGS Extension to X
PFS	Perfect Forward Security
PHB	Processor Host Bridges
PHY	Physical Layer
PID	Process ID
PII	Program Integrated Information
PMTU	Path MTU
PPC	PowerPC
PSE	Portable Streams Environment
PTF	Program Temporary Fix
PV	Physical Volume
QoS	Quality of Service
RAID	Redundant Array of Independent Disks
RAN	Remote Asynchronous Node
RAS	Reliability Availability Serviceability
RDB	Relational DataBase
RDISC	ICMP Router Discovery
RDN	Relative Distinguished Name
RDP	Router Discovery Protocol
RFC	Request for Comments
RIO	Remote I/O
RIP	Routing Information Protocol
RMC	Resource Monitoring and Control
RPA	RS/6000 Platform Architecture
RPC	Remote Procedure Call
RPL	Remote Program Loader
RSCT	Reliable Scalable Cluster Technology
RSVP	Resource Reservation Protocol
SA	Secure Association
SACK	Selective Acknowledgments
SBCS	Single-Byte Character Support
SCB	Segment Control Block
SCSI	Small Computer System Interface
SCSI-SE	SCSI-Single Ended
SDRAM	Synchronous DRAM
SE	Single Ended
SGID	Set Group ID

SHLAP	Shared Library Assistant Process
SID	Segment ID
SIT	Simple Internet Transition
SKIP	Simple Key Management for IP
SLB	Segment Lookaside Buffer
SLIH	Second Level Interrupt Handler
SM	Session Management
SMIT	System Management Interface Tool
SMB	Server Message Block
SMP	Symmetric Multiprocessor
SNG	Secured Network Gateway
SP	Service Processor
SPCN	System Power Control Network
SPI	Security Parameter Index
SPM	System Performance Measurement
SPOT	Shared Product Object Tree
SRC	System Resource Controller
SRN	Service Request Number
SSA	Serial Storage Architecture
SSL	Secure Socket Layer
STP	Shielded Twisted Pair
SUID	Set User ID
SVC	Supervisor or System Call
SYNC	Synchronization
TCE	Translate Control Entry
TCP/IP	Transmission Control Protocol/Internet Protocol
TGT	Ticket Granting Ticket
TOS	Type Of Service
TTL	Time To Live
TSE	Text Search Engine
UCS	Universal Coded Character Set
UIL	User Interface Language
ULS	Universal Language Support
UP	Uni-Processor
USLA	User-Space Loader Assistant
UTF	UCS Transformation Format
UTM	Uniform Transfer Model
UTP	Unshielded Twisted Pair
VFB	Virtual Frame Buffer
VG	Volume Group
VGDA	Volume Group Descriptor Area
VGSA	Volume Group Status Area
VHDCI	Very High Density Cable Interconnect
VMM	Virtual Memory Manager
VP	Virtual Processor
VPD	Vital Product Data
VPN	Virtual Private Network
VSM	Visual System Manager

WLM	Workload Manager
XCOFF	Extended Common Object File Format
XIE	X Image Extension
XIM	X Input Method
XKB	X Keyboard Extension
XOM	X Output Method
XPM	X Pixmap
XVFB	X Virtual Frame Buffer

Index

IBM Redbooks review

Your feedback is valued by the Redbook authors. In particular we are interested in situations where a Redbook "made the difference" in a task or problem you encountered. Using one of the following methods, **please review the Redbook, addressing value, subject matter, structure, depth and quality as appropriate.**

- Use the online **Contact us** review redbook form found at **ibm.com**/redbooks
- Fax this form to: USA International Access Code + 1 845 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Document Number **Redbook Title**	SG24-5765-00 AIX 5L Differences Guide Version 5.0 Edition
Review	
What other subjects would you like to see IBM Redbooks address?	
Please rate your overall satisfaction:	O Very Good O Good O Average O Poor
Please identify yourself as belonging to one of the following groups:	O Customer O Business Partner O Solution Developer O IBM, Lotus or Tivoli Employee O None of the above
Your email address: The data you provide here may be used to provide you with information from IBM or our business partners about our products, services or activities.	 O Please do not use the information collected here for future marketing or promotional contacts or other communications beyond the scope of this transaction.
Questions about IBM's privacy policy?	The following link explains how we protect your personal information. **ibm.com**/privacy/yourprivacy/

IBM

Redbooks

AIX 5L Differences Guide Version 5.0 Edition

AIX 5L Differences Guide Version 5.0 Edition

AIX - The Industry's #1 industrial strength UNIX platform

An inside look at AIX 5L Version 5.0 enhancements

An excellent way to leverage your UNIX business

This redbook focuses on the latest enhancements introduced in AIX 5L Version 5.0. It is intended to help system administrators, developers, and users understand these enhancements and evaluate potential benefits in their own environments.

AIX 5L is available for POWER and Itanium-based systems. The initial offering of AIX 5L for POWER is available as a no-charge i-listed PRPQ. AIX 5L for Itanium-based systems is covered under the beta program. Both platforms were developed from the same common code base.

AIX 5L introduces many new features, including virtual IP, quality of service enhancements, enhanced error logging, dynamic paging space reduction, hot-spare disk management, advanced Workload Manager, JFS2, and others. The availability of an improved Web-based System Manager continues AIX's move towards a standard, unified interface for system tools. There are many other enhancements available with AIX 5L, and you can explore them all in this redbook.

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

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

SG24-5765-00 ISBN 0738421286