IBM RISC System/6000
Scalable POWERparallel Systems
Administration Guide

Document Number GC23-3897-01

Note!

Before using this information and the product it supports, be sure to read the general information under “Notices” on page xv.

First Edition (August 1995)

This edition applies to IBM Parallel System Support Programs for AIX Version 2 Release 1, which runs on IBM RISC System/6000 Scalable POWERparallel Systems, and to all subsequent releases and modifications until otherwise indicated in new editions or Technical Newsletters.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

IBM welcomes your comments. A form for your comments appears at the back of this publication. If the form has been removed, address your comments to:

IBM Corporation, Department 55JA, Mail Station P384
522 South Road
Poughkeepsie, NY 12601-5400
United States of America

FAX: (International Access Code)+1+914+432-9405

IBMLink (United States customers only): KGNRCF at KGNVMC
IBM Mail Exchange: USIB27BQ at IBMMAIL
Internet: kgnrcf@vnet.ibm.com

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

Copyright International Business Machines Corporation 1995. All rights reserved. Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Notices xv
Trademarks xv
Publicly Available Software xvi

About This Book xvii
Who Should Use This Book xviii
How This Book Is Organized xviii
Typographic Conventions xx
Interface Instructions xxi
Related Publications xxi
SP Hardware Publications xxi
SP Software Publications xxi
Recoverable Virtual Shared Disk Publications xxii
LoadLeveler Publications xxii
Parallel Environment for AIX Publications xxii
PVMe for AIX Publications xxii
Client Input Output/Sockets (CLIO/S) Publications xxii
Parallel I/O File System Publications xxiii
Other IBM Publications xxiii
AIX and RISC System/6000 Publications xxiii
Non-IBM Publications xxiv
Tcl xxiv
Ordering Additional Copies of This Book xxiv
Manual Pages for Public Code xxiv

Chapter 1. Introduction 1
New Features in the IBM Parallel System Support Programs for AIX Version 2 Release 1 1

Chapter 2. RISC System/6000 Scalable POWERparallel Systems Overview 5
Hardware Overview 5
Software Overview 7

Security on the SP System 15

Chapter 3. Security Features of the SP System 17
Understanding Authentication 17
Using SP Security Services 26
Administering SP Security Services 32
Using AFS Authentication Services 47

Chapter 4. Sysctl 51
Sysctl Components 51
Relationship to the SP Authentication Services 53
Overview 53
Installation and Configuration 56
Using Sysctl 62
Developing a Sysctl Application 64
Sysctl Scripts 68

© Copyright IBM Corp. 1995

Sysctl Reference Information 70

Managing the SP System 75

Chapter 5. Managing a High Availability Control Workstation Configuration 77
Understanding HACWS Function 78
Managing SP Subsystems for High Availability Control Workstation Function . 86
Customizing HACMP/6000 91
Verifying Frame Supervisor Cables 93

Chapter 6. Managing System Partitions 95
Understanding System Partitioning 95
Partitioning the SP System 100
Displaying System Partition Configuration Information 113
Verifying System Partition Configuration 114

Chapter 7. Using the SP System Monitor 117
Understanding the Hardware Monitor 118
Configuring the SP System Monitor 120
Customizing SP System Monitor GUI Default Resources 121
Using the Command Line Interfaces 121
Using the SP System Monitor GUI 121
Starting the SP System Monitor GUI 123
Displaying Frame, Node, and Switch Layouts 127
Monitoring and Controlling the High Performance Switch 128
Operating Frame Environment Controls 130
Operating the Front Panel Controls 132
Using Global Controls 139
Determining If Nodes Are Operating: All Node Summary 141
Determining the State of Network Adapters within the SP System 144
Setting Up the Logging Daemon on Separate Workstations 147
Setting Up the Hardware Monitor Clients on Separate Workstations 147

Chapter 8. Starting Up and Shutting Down the SP System 149
Overview of System Startup and Shutdown 149
Controlling the Sequence of Startup and Shutdown Operations 149
Starting the SP System 150
Shutting Down the SP System 152
Checking the Status of Startup and Shutdown Requests 155
Defining Startup and Shutdown Sequence Files 155
Creating a Special Subsystem 159

Chapter 9. The High Performance Switch 161
Switch Communication Modes 161
System Partitioning and the High Performance Switch 162
Selecting Appropriate Switch Clocks 163
Managing the Switch Topology File 164
Starting the High Performance Switch 165
Isolating Nodes from the High Performance Switch 166

Chapter 10. Parallel Management Commands 169
Using dsh to Execute Parallel Management Commands 169

The hostlist Command . . . 171
List of Parallel Management Commands . . . 172

Chapter 11. Managing SP Resources . . . 179
Understanding the Resource Manager . . . 179
Installing the Resource Manager . . . 180
Defining the Resource Manager Configuration . . . 181
Starting the Resource Manager . . . 184
Reconfiguring the Resource Manager . . . 189
Stopping the Resource Manager . . . 190
Locating the Resource Manager . . . 191
Querying the Resource Manager . . . 191
Resource Manager Recovery . . . 192

Chapter 12. Managing User Accounts . . . 193
Understanding SP User Management . . . 193
Adding User Accounts . . . 196
Changing User Accounts . . . 198
Removing User Accounts . . . 199
Listing a User Account . . . 201
Changing Passwords . . . 201
Controlling User Login . . . 203

Chapter 13. Managing NTP . . . 211
NTP Options . . . 211

Chapter 14. Managing File Collections . . . 213
Understanding File Collections . . . 214
Reporting File Collection Information . . . 221
Verifying File Collections Using scan . . . 223
Updating Files in a File Collection . . . 224
Adding and Deleting Files in a File Collection . . . 226
Building a File Collection . . . 228
Installing a File Collection . . . 231
Refusing Files in a File Collection . . . 233
Modifying the File Collection Hierarchy . . . 233
Removing a File Collection . . . 235
supper list File Syntax, Keywords, and Operands . . . 236

Chapter 15. Managing Amd . . . 239
Understanding the NFS and Amd Implementation . . . 239
Establishing an Amd Map for Users' Directories . . . 241
Using Amd for Other File Systems . . . 242

Chapter 16. Managing Print Service . . . 245
Print Management Security Modes . . . 245
Print Management Command Support . . . 246
SP Print Management System Requirements . . . 248
The Print Management System and Kerberos . . . 248
Integrating SP Print Management . . . 249
Installing Print Management on a Control Workstation . . . 251
Using a Non-AIX Workstation as Print Host . . . 251

Chapter 17. Managing Mail Service . . . 253

Chapter 18. Accounting . . . 267
Accounting by Node or by Class . . . 267
Setting Up SP Accounting . . . 269
Accounting Files . . . 276

Chapter 19. Maintaining the crontabs File . . . 281

Chapter 20. Managing Error Logs . . . 283
Installing and Configuring Error Log Management . . . 284
Managing the AIX Error Log Facility . . . 285
Managing the BSD syslog Facility . . . 289
Viewing Error Logs . . . 292
Archiving Error Logs . . . 293
Collecting System Data for IBM Service . . . 294

Chapter 21. Tuning the SP System . . . 297
Scheduling Administrative Tasks . . . 297
System Topology Considerations . . . 297
Tunables and Tunable Values . . . 298
Changing Network Options on the Nodes . . . 299
Tuning Router Nodes . . . 299
Tuning the SP Ethernet . . . 301
Tuning the SP Network for Specific Workloads . . . 301
Tuning ESCON Connections for IP to MVS . . . 304
Tuning For Single Parent–Multiple Child Applications . . . 304
PCA Card and CLIO/S Tuning on the SP System . . . 306

IBM Virtual Shared Disk . . . 309

Chapter 22. Defining and Managing IBM Virtual Shared Disks . . . 311
Overview . . . 312
What's New in This Release . . . 312
Planning to Install the IBM Virtual Shared Disk . . . 313
Installation Requirements . . . 313
Migrating to the IBM Virtual Shared Disk Upgraded Software . . . 314
Installing the IBM Virtual Shared Disk Software . . . 314
Usage Notes . . . 316
IBM Virtual Shared Disk Configuration Data . . . 317
Defining IBM Virtual Shared Disk Information in the System Data Repository 317
An Example of Defining an IBM Virtual Shared Disk . . . 321
Displaying IBM Virtual Shared Disk Information Stored in the SDR . . . 324
Deleting IBM Virtual Shared Disk Information from the SDR . . . 326
Managing IBM Virtual Shared Disks . . . 327
Testing to Verify an IBM Virtual Shared Disk (the Verification Test Suite) . . 333
Getting IBM Virtual Shared Disk Information with C Interfaces . . . 337
Tuning IBM Virtual Shared Disk Performance . . . 337

Chapter 23. Defining and Managing IBM Virtual Shared Disk Data Striping Devices (HSDs) . . . 343
Overview . . . 343
HSD and LVM Striping . . . 344
What's New in This Release . . . 344
The HSD Software . . . 344

When to Use the HSD . . . 345
Planning to Use the HSD . . . 345
Planning to Install the Data Striping Device . . . 345
Installation Using SMIT . . . 346
Managing Data Striping Devices . . . 350
Verifying Data Striping Device Installation . . . 352
Management and Usage Notes . . . 353
Getting HSD Information with C Interfaces . . . 354
Tuning Data Striping Device Performance . . . 355

Distributed Tape Management System Enablement . . . 357

Chapter 24. Distributed Tape Management System Enablement . . . 359
Overview . . . 359
Magnetic Tape Extension (MTE) Pseudo Device Driver . . . 360
Physical Tape Device Driver with the Tape Management Enhancement (TMSDD) . . . 368

Appendixes . . . 391

Appendix A. Using an Alternate Boot System Image . . . 393
Enabling Alternate Boot System Image Function . . . 393
Using the Alternate Boot System Image Function . . . 394
Setting the Boot List Manually . . . 394
Using Alternate System Image on the Control Workstation . . . 395

Appendix B. SP Daemons . . . 397

Appendix C. Sample Files . . . 399
afsclient.cust . . . 399
block_usr_sample . . . 402
bootptab.info File . . . 404
jmd_config . . . 405
script.cust File . . . 407
tuning.cust File . . . 412
/usr/lib/X11/app-defaults/SPmon . . . 414

Appendix D. System Monitor Variables . . . 415
Frame Hardware Variables Common to All Frame Supervisors . . . 415
Frame Hardware Variables Unique to Card Type 17 and 20 Frame Supervisor 416
Frame Hardware Variables Unique to SEPBU (Type 18) and HACWS SEPBU (Type 22) Frame Supervisors . . . 417
Frame Hardware Variables Unique to HACWS (Type 20) and SEPBU HACWS (Type 22) Frame Supervisors . . . 418
Frame Hardware Variables Unique to Card Type 17 and SEPBU (Type 18) Frame Superviors . . . 419
Node Hardware Variables Common to All Node Supervisors . . . 419
Node-Specific Hardware Variables for Card Types 33 and 65 Node Supervisor 422
Node-Specific Hardware Variables for Card Type 97 Node Supervisors . . . 423
Node-Specific Hardware Variables for Card Types 81 and 83 Node Supervisors . . . 424
Switch Variables . . . 426

Appendix E. The System Data Repository 429
Understanding the System Data Repository 429
Accessing the System Data Repository 430
Updating the host_responds Class (the hrd Daemon) 437
Backing Up the SDR 438
SDR Shadow Files 438
System Data Repository Objects and Attributes 438

Glossary of Terms and Abbreviations 455

Index 463

Figures

1. Basic SP Configuration . . . 5
2. An Overview of Kerberos Authentication . . . 21
3. Sysctl Elements . . . 52
4. An HACWS Configuration . . . 79
5. Addressing in an HACWS Configuration . . . 82
6. Frame Environment Layout Showing Active Control Workstation . . . 88
7. HACWS Pre- and Post-Events . . . 92
8. System Partition Configuration SMIT Panel . . . 102
9. Archive System Data Repository SMIT Panel . . . 103
10. System Partition Configuration List SMIT Panel . . . 104
11. Select System Partition Configuration SMIT Panel . . . 105
12. Display Information for Given Partition or Layout . . . 106
13. Display Information for Given Partition or Layout Output . . . 106
14. Select System Partition Layout SMIT Panel . . . 107
15. Select System Partition Layout Output SMIT Panel . . . 108
16. Partition Directory SMIT Panel . . . 109
17. Enter Customization Arguments for this System Partition SMIT Panel . . . 109
18. Verify System Partition Configuration SMIT Panel . . . 111
19. Apply System Partition Configuration SMIT panel . . . 112
20. List System Partition Database Information Output . . . 114
21. Verify SSP System Partition Configuration SMIT Panel . . . 115
22. The SP System Monitor . . . 118
23. SP System Monitor GUI Main Window . . . 125
24. View Pulldown Menu . . . 126
25. SP Pulldown Menu . . . 126
26. SP System Monitor GUI Display Layouts Selection Window . . . 127
27. SP System Monitor GUI Switch Front Panel Layout Window . . . 129
28. Frame Environment Layout for Type 18 Frame Supervisor . . . 131
29. Frame Environment Layout for Type 22 Frame Supervisor . . . 132
30. Node Front Panel Display . . . 133
31. Power Off Menu . . . 135
32. Power On Menu . . . 135
33. Reset Menu . . . 136
34. Shutdown Menu . . . 137
35. Fence Menu . . . 138
36. Global Controls Display . . . 140
37. Node Variable Selection Window . . . 142
38. 3-Digit Display . . . 143
39. LAN Topology Display . . . 145
40. LAN Status Display . . . 146
41. The Change/Show Switch Clock Source Settings SMIT menu. . . . 163
42. The Set Switch Run Phase Duration SMIT menu. . . . 168
43. Example Resource Manager Configuration . . . 183
44. SMIT Panel for Site Environment Information . . . 194
45. SMIT Panel for Adding an SP User . . . 197
46. Example of Adding an SP User . . . 198
47. SMIT Panel for Changing SP User Information . . . 199
48. SMIT Panel for Removing an SP User . . . 200
49. SP Default File Collection Hierarchy . . . 217
50. /var/sysman/sup Files and Directories . . . 218

© Copyright IBM Corp. 1995

51. sup.admin Master Files 219
52. SP Hierarchy of an Offline Boot/install Server 234
53. Sample /etc/amd/amd-maps/amd.u File 242
54. Option 1: Using DNS 255
55. Option 2: Using NIS 260
56. Option 3: Using /etc/hosts 263
57. Accounting Fields on the Site Environment SMIT Panel 271
58. SP Accounting Information SMIT Panel 273
59. Accounting Template for crontabs 276
60. Log Management SMIT Panel 284
61. AIX Error Log SMIT Panel 286
62. Syslog SMIT Panel 290
63. General Log Viewing SMIT Panel 292
64. Archive Logs SMIT Panel 293
65. Collect Logs for Service SMIT Panel 295
66. Simplified View of a Single Parent–Multiple Child Application 305
67. An IBM Virtual Shared Disk IP Network Implementation 312
68. An Example of an IBM Virtual Shared Disk Configuration 321
69. Our Example of an IBM Virtual Shared Disk Configuration So Far . . . 322
70. Our Defined IBM Virtual Shared Disk Example. 324
71. IBM Virtual Shared Disk States and Associated Commands 328
72. SMIT Main Menu 365
73. SMIT Devices Menu 365
74. SMIT Tape Drive Menu 366
75. Tape Drive Type Menu 366
76. Add a Tape Management Pseudo Control Point Menu 367
77. TMPCP COMMAND STATUS Window 367
78. Configure a TMPCP Selection List 367
79. Configure a Defined TMPCP COMMAND STATUS Window 368
80. SMIT Main Menu 374
81. SMIT Devices Menu 375
82. SMIT Tape Drive Menu 375
83. Tape Drive Type Menu 376
84. Parent Control Unit Menu 376
85. Add a 3480 Tape Drive Menu 377
86. Add a 3490 Tape Drive COMMAND STATUS Window 378
87. Configure a Defined Tape Drive Selection List 378
88. Configure a Defined Tape Drive COMMAND STATUS Window 378
89. SMIT Main Menu 379
90. SMIT Devices Menu 380
91. Tape Drive Type Menu 380
92. Tape Drive Submenu 381
93. Change / Show Characteristics of a 3480 Tape Drive Menu 381
94. Device Features List Submenu 382
95. Media Type List Submenu 383
96. Command Status Menu 384
97. SMIT Main Menu 385
98. SMIT Devices Menu 386
99. Tape Drive Type Menu 386
100. Tape Drive Submenu 387
101. Change / Show Characteristics of a 3490E Tape Drive Menu 387
102. Device Features List Submenu 388
103. Media Type List Submenu 389
104. Command Status Menu 390

105. Comparing Two SDR Classes 429
106. SP System Management SMIT Panel 432
107. List 9076 SP Configuration Information 434
108. List Node File System Information Panel 435

Tables

1. Authorization Variables . . . 67
2. Built-in Tcl Commands and Their Default Authorization Callbacks . . . 71
3. Built-in Extended Tcl Commands and Their Default Authorization Callbacks . . . 71
4. ACL Processing Commands . . . 72
5. Service Commands . . . 72
6. Other Commands . . . 73
7. High Availability Control Workstation Configuration and Task Summary . 85
8. Frame, Node, and Switch Layouts . . . 128
9. Default Variable Colors . . . 144
10. Graphical Element Colors . . . 146
11. Specifying Nodes to Start with cstartup . . . 151
12. Specifying Nodes to Shut Down with cshutdown. . . . 152
13. Time Service Choices . . . 211
14. Initial Values for AIX 4.1 . . . 302
15. Initial Values for AIX 3.2.5 . . . 302
16. IBM Virtual Shared Disk Information . . . 322
17. IBM Virtual Shared Disk Node Information . . . 323
18. IBM Virtual Shared Disk Volume Group Information . . . 323
19. IBM Virtual Shared Disk Definition Information . . . 323
20. LED Translation Values . . . 422
21. SDR Authorizations . . . 430
22. Commands Invoked by SMIT Panels . . . 434

© Copyright IBM Corp. 1995

Notices

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 product, program, or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the user's responsibility.

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:

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, NY 10594
USA

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
Mail Station P131
522 South Road
Poughkeepsie, NY 12601-5400
USA
Attention: Information Request

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

AIX
ESCON
IBM
LoadLeveler
Micro Channel
RISC System/6000
RISC System/6000 Scalable POWERparallel Systems

Windows is a trademark of Microsoft Corporation.

© Copyright IBM Corp. 1995

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other company, product, and service names, which may be denoted by a double asterisk (**), may be trademarks or service marks of others.

Publicly Available Software

This product includes software that is publicly available:

Amd Berkeley Software Distribution automount daemon

expect Programmed dialogue with interactive programs

Kerberos Provides authentication of the execution of remote commands

NTP Network Time Protocol

Perl Practical Extraction and Report Language

SUP Software Update Protocol

Tcl Tool Command Language

TclX Tool Command Language Extended

Tk Tcl-based Tool Kit for X-windows

This book only discusses the use of these products as they apply specifically to the SP system. The distribution for these products includes the source code and associated documentation. (Kerberos does not ship source code.) **/usr/lpp/ssp/public** contains the compressed **tar** files of the publicly available software. (IBM has made minor modifications to the versions of Tcl and Tk used in the SP system to improve their security characteristics. Therefore, the IBM-supplied versions do not match exactly the versions you may build from the compressed **tar** files.) All copyright notices in the documentation must be respected. You can find version and distribution information for each of these products that are part of your selected install options in the **/usr/lpp/ssp/README/ssp.public.README** file.

About This Book

The *IBM RISC System/6000 Scalable POWERparallel Systems: Administration Guide* provides information to help you understand and perform SP system administration. It includes concepts and instructions pertaining to:

- Understanding IBM RISC System/6000 Scalable POWERparallel Systems (SP)
- Making the system secure
- Managing the system
- Tuning the system
- Defining and managing IBM Virtual Shared Disks
- Installing, defining and configuring Distributed Tape Management System Enablement

Additional task information for System Administrators is covered in separate manuals.

Other books that help you administer and use the SP system include:

- The *IBM RISC System/6000 Scalable POWERparallel Systems: System Planning* book helps you plan for the SP system. It tells you how to prepare for software installation and provides considerations for integrating it into your existing computing network.
- The *IBM RISC System/6000 Scalable POWERparallel Systems: Installation Guide* helps you understand and perform SP installation, configuration, and maintenance. It includes concepts and instructions for:
  - Installing and configuring the IBM Parallel System Support Programs (PSSP) for AIX
  - Migrating to PSSP Version 2 Release 1 from PSSP Version 1 Release 2
  - Reconfiguring the system
  - Performing system maintenance
- The *IBM RISC System/6000 Scalable POWERparallel Systems: Diagnosis and Messages* book provides information to help you diagnose SP system problems and describes error messages, their meaning, and appropriate actions to take to recover from problems.
- The *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference* provides detailed syntax and parameter information for all commands you can use to install, customize, and maintain the SP system.

Documentation Note

The information in this book supports the use of IBM Parallel System Support Programs for AIX Version 2 Release 1, which runs on AIX 4.1.3. If you are running IBM AIX Parallel System Support Programs, Version 1 Release 2, which runs on AIX 3.2.5, in a system partition, you may also need to refer to the documentation that supports that product level.

© Copyright IBM Corp. 1995

Who Should Use This Book

This book is intended for system administrators responsible for setting up and maintaining the SP system. It assumes the administrators have a working knowledge of AIX or UNIX and experience with network systems.

The System Administrators Guide of USENIX (SAGE), has developed a classification for skills required for system administration as published in "Job Descriptions for System Administrators." Administrators of small SP systems (one or two frames in a relatively simple environment) are expected to have level II skills (Junior System Administrator). Administrators of larger SP systems are expected to have level III skills (Intermediate/Advanced System Administrator).

Parts of this book can also be used by system operators or others that need to monitor the status of the hardware and software or interact with the power and reset controls.

How This Book Is Organized

SP system administration is a complex topic encompassing many different kinds of tasks. This book is divided into sections that cover tasks related to providing system security, monitoring and operating the system, and managing the day-to-day running of the SP system. Separate sections address managing the optional IBM Virtual Shared Disk and Distributed Tape Management System Enablement.

- Chapter 1, "Introduction" on page 1 briefly describes the new and improved features of the SP System.
- Chapter 2, "RISC System/6000 Scalable POWERparallel Systems Overview" on page 5 describes the hardware and software components of the SP system.

Security on the SP System

- Chapter 3, "Security Features of the SP System" on page 17 describes how to set up, maintain, and use authentication services.
- Chapter 4, "Sysctl" on page 51 describes how to install, configure, and use the authenticated client/server system for running commands remotely and in parallel.

Managing the SP System

- Chapter 5, "Managing a High Availability Control Workstation Configuration" on page 77 provides information for understanding High Availabiltity Control Workstation (HACWS) function, performing tasks to ensure that subsystems fail over to the backup control workstation in the event the primary control workstation fails, and customizing AIX High Availability Cluster Multi-Processing/6000 (HACMP/6000).
- Chapter 6, "Managing System Partitions" on page 95 describes the tasks necessary to partition (and repartition) your SP system.
- Chapter 7, "Using the SP System Monitor" on page 117 describes how to use the SP System Monitor with both the command interface and the SP System Monitor Graphical User Interface (GUI) to monitor and control the hardware.
- Chapter 8, "Starting Up and Shutting Down the SP System" on page 149 shows how to use commands to bring up the SP system and shut it down.

- Chapter 9, “The High Performance Switch” on page 161 discusses switch operations.
- Chapter 10, “Parallel Management Commands” on page 169 describes commands useful for controlling several nodes in parallel.
- Chapter 11, “Managing SP Resources” on page 179 describes how to use the Resource Manager.
- Chapter 12, “Managing User Accounts” on page 193 provides information on passwords, user access, and instructions on how to give your users access to the system.
- Chapter 13, “Managing NTP” on page 211 discusses how time service is used in the SP system.
- Chapter 14, “Managing File Collections” on page 213 describes procedures for installing, updating, and removing file collections.
- Chapter 15, “Managing Amd” on page 239 discusses how to use the Automount daemon (Amd).
- Chapter 16, “Managing Print Service” on page 245 describes how print management is handled in the SP system.
- Chapter 17, “Managing Mail Service” on page 253 shows you how to integrate your mail system.
- Chapter 18, “Accounting” on page 267 describes how to run SP Accounting in your environment.
- Chapter 19, “Maintaining the crontabs File” on page 281 contains a few hints on running **cron** jobs.
- Chapter 20, “Managing Error Logs” on page 283 describes the functions provided for managing the AIX Error Log facility, the BSD syslog, and SP specific log files.
- Chapter 21, “Tuning the SP System” on page 297 contains recommendations for fine-tuning your environment for better performance.

IBM Virtual Shared Disk

- Chapter 22, “Defining and Managing IBM Virtual Shared Disks” on page 311 describes how IBM Virtual Shared Disk works, shows you how to get IBM Virtual Shared Disk up on your SP system, and describes how to perform IBM Virtual Shared Disk tasks.
- Chapter 23, “Defining and Managing IBM Virtual Shared Disk Data Striping Devices (HSDs)” on page 343 describes how to install the data striping device and how to use and manage disks defined as striped devices.

Distributed Tape Management System Enablement

- Chapter 24, “Distributed Tape Management System Enablement” on page 359 describes how to define, install, and configure Distributed Tape Management System Enablement.

Appendixes

- This book also includes appendixes containing supplementary information as well as a glossary and index.

Typographic Conventions

This book uses the following typographic conventions:

Typographic	Usage
Bold	• **Bold** words or characters represent system elements that you must use literally, such as commands, flags, and path names. • **Bold** words also indicate the first use of a term included in the glossary.
Italic	• *Italic* words or characters represent variable values that you must supply. • *Italics* are also used for book titles and for general emphasis in text.
`Constant width`	Examples and information that the system displays appear in `constant width` typeface.
[ ]	Brackets enclose optional items in format and syntax descriptions.
{ }	Braces enclose a list from which you must choose an item in format and syntax descriptions.
|	A vertical bar separates items in a list of choices. (In other words, it means “or.”)
< >	Angle brackets (less-than and greater-than) enclose the name of a key on the keyboard.
...	An ellipsis indicates that you can repeat the preceding item one or more times.
<**Ctrl-***x*>	The notation <**Ctrl-***x*> indicates a control character sequence. For example, <**Ctrl-c**> means that you hold down the control key while pressing <**c**>.
<**Enter**>	The notation <**Enter**> refers to the key on your terminal or workstation that is labeled with the word Enter (or sometimes with a left arrow).

Interface Instructions

Some sections of this book give step-by-step instructions for performing tasks with a graphical user interface. The instructions use a format that distinguishes between the user action and the system response.

> User actions appear in uppercase bold type.
>
> **PRESS** Cancel
>
> Selections from a menu bar are indicated with an →.
>
> **SELECT** SP → Topology
>
> The system response to an action follows the ▲ symbol.
>
> ▲ The diagnostic display appears.

Related Publications

SP Hardware Publications

- *IBM RISC System/6000 Scalable POWERparallel Systems: Technical Brochure*, GH23-2485
- *IBM RISC System/6000 Scalable POWERparallel Systems: Commercial Brochure*, GA23-2475
- *IBM RISC System/6000 Scalable POWERparallel Systems: Site Planning*, GC23-3905
- *IBM RISC System/6000 Scalable POWERparallel Systems: Maintenance Information, Volume 1, Installation and CE Operations*, GC23-3903
- *IBM RISC System/6000 Scalable POWERparallel Systems: Maintenance Information, Volume 2, Maintenance Analysis Procedures and Parts Catalog*, GC23-3904
- *IBM RISC System/6000 Scalable POWERparallel Systems: Library Guide*, GC23-3868
- *IBM RISC System/6000 Scalable POWERparallel Systems: Installation Aid Licensed Program Specification*, GC23-3838

SP Software Publications

- *IBM RISC System/6000 Scalable POWERparallel Systems: System Planning*, GC23-3902
- *IBM RISC System/6000 Scalable POWERparallel Systems: Installation Guide*, GC23-3898
- *IBM RISC System/6000 Scalable POWERparallel Systems: Diagnosis and Messages Guide*, GC23-3899
- *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference*, GC23-3900

- *IBM RISC System/6000 Scalable POWERparallel Systems: Parallel System Support Programs Licensed Program Specification*, GC23-3901

As an alternative to ordering the individual books, you can use GBOF-8587 to order the entire SP software library.

Recoverable Virtual Shared Disk Publications

- *IBM Recoverable Virtual Shared Disk: User's Guide and Reference*, GC23-3849
- *IBM Recoverable Virtual Shared Disk: Licensed Program Specification*, GC23-3848

LoadLeveler Publications

- *IBM LoadLeveler: Administration and Installation Guide*, SH26-7220
- *IBM LoadLeveler: User's Guide*, SH26-7226
- *IBM LoadLeveler: Licensed Program Specification*, GH23-0040

Parallel Environment for AIX Publications

- *IBM AIX Parallel Environment: General Information*, GC23-3906
- *IBM Parallel Environment for AIX: Hitchhiker's Guide*, GC23-3895
- *IBM Parallel Environment for AIX: Installation, Administration, and Diagnosis*, GC23-3892
- *IBM Parallel Environment for AIX: Operation and Use*, GC23-3891
- *IBM Parallel Environment for AIX: MPL Programming and Subroutine Reference*, GC23-3893
- *IBM Parallel Environment for AIX: MPI Programming and Subroutine Reference*, GC23-3894
- *IBM Parallel Environment for AIX: Licensed Program Specification*, GC23-3896

As an alternative to ordering the individual books, you can use GBOF-8588 to order the entire Parallel Environment for AIX library.

PVMe for AIX Publications

- *IBM PVMe for AIX: Fact Sheet*, GC23-3907
- *IBM PVMe for AIX: User's Guide and Reference*, GC23-3884
- *IBM PVMe for AIX: Licensed Program Specification*, GC23-3885

Client Input Output/Sockets (CLIO/S) Publications

- *IBM CLIO/S General Information*, GC23-3879
- *IBM CLIO/S Licensed Program Specification*, GC23-3789
- *IBM CLIO/S User's Guide and Reference*, available from your IBM representative

Parallel I/O File System Publications

- *IBM AIX Parallel I/O File System: Fact Sheet*, G325-0649
- *IBM AIX Parallel I/O File System: Installation, Administration, and Use*, SH34-6065
- *IBM AIX Parallel I/O File System: Licensed Program Specification*, GC34-6066

Other IBM Publications

AIX and RISC System/6000 Publications

- *IBM AIX for RISC System/6000 System Overview*, GC23-2406
- *IBM Getting Started: Using IBM RISC System/6000*, GC23-2377
- *IBM Getting Started: Managing RISC System/6000*, GC23-2378
- *IBM General Concepts and Procedures for RISC System/6000*, GC23-2202
- *IBM AIX Communication Concepts and Procedures*, GBOF-1524
- *IBM AIX Version 3.2 for RISC System/6000 Files Reference*, GC23-2200
- *IBM AIX Version 4.1 Files Reference*, SC23-2512
- *IBM AIX Version 3.2 System Management Guide*, SC23-2457
- *IBM AIX Version 3.2 System Management Guide: Communications and Networks*, GC23-2487
- *IBM AIX Version 4.1 System Management Guide: Communications and Networks*, SC23-2526
- *IBM AIX Version 3.2 Installation Guide*, SC23-2341
- *IBM AIX Version 4.1 Installation Guide*, SC23-2550
- *IBM AIX for RISC System/6000 Commands Reference*, GBOF-1802
- *IBM AIX Version 4.1 Commands Reference*, SBOF-1851
- *IBM AIX Versions 3.2 and 4.1 Performance Tuning Guide*, SC23-2365
- *IBM RISC System/6000 Problem Solving Guide*, SC23-2204
- *IBM AIX Version 4.1 Problem Solving Guide and Reference*, SC23-2606
- *IBM AIX Version 4.1 Messages Guide and Reference*, SC23-2641
- *IBM AIX Version 4.1 Network Installation Management Guide and Reference*, SC23-2627
- *IBM AIX Version 4.1 System Management Guide: Operating System and Devices*, SC23-2525
- *IBM AIX Version 4.1 General Programming Concepts: Writing and Debugging Programs*, SC23-2533
- *IBM AIX Version 4.1 Communications Programming Concepts* SC23-2610
- *Diskless Workstation Management Guide*, SC23-2433
- *C++ for AIX/6000: Language Reference*, SC09-1606
- *C++ for AIX/6000: Standard Class Library Reference* SC09-1604

- *C++ for AIX/6000: User's Guide*, SC09-1605

Non-IBM Publications

Tcl

- Ousterhout, John K., *Tcl and the Tk Toolkit*, Addison-Wesley, Reading, MA, 1994, ISBN 0-201-63337-X.

Ordering Additional Copies of This Book

To order additional copies of this book, request form number GC23-3897.

Manual Pages for Public Code

The following manual pages for public code are available in this product:

SUP /usr/lpp/ssp/man/man1/sup.1

Amd /usr/lpp/ssp/man/man8/amd.8

/usr/lpp/ssp/man/man8/amq.8

NTP /usr/lpp/ssp/man/man8/xntpd.8

/usr/lpp/ssp/man/man8/xntpdc.8

Perl /usr/lpp/ssp/perl/man/perl.man

/usr/lpp/ssp/perl/man/h2ph.man

/usr/lpp/ssp/perl/man/s2p.man

/usr/lpp/ssp/perl/man/a2p.man

Manual pages and other documentation for Tcl, TclX, Tk, and expect can be found in the compressed **tar** files located in /**usr**/**lpp**/**ssp**/**public**.

Chapter 1. Introduction

RISC System/6000 Scalable POWERparallel Systems (SP) offer an information processing solution that provides:

- Scalability

  Standard system configurations of two to 128 processor nodes are supported. You can expand your system over time to meet your changing computing requirements.

- Cost effectiveness

  Because the SP system is based on AIX and RISC System/6000 technology, thousands of existing RISC System/6000 applications can be run on the SP, eliminating the need for costly reprogramming.

  The scalability of the SP system ensures that hardware and software invested in now will continue to serve you over time.

- Parallel computing

  The SP system provides a strong platform for parallel application development and execution. System management applications provided in the IBM Parallel System Support Programs for AIX, the High Performance Switch, and related IBM products such as IBM Parallel Environment for AIX enhance the parallel computing capabilities of the SP system.

- Support of open system standards

  The SP system accomodates a wide number and variety of I/O attachments and disk storage.

  Enablers are provided that permit a number of commercial applications to exploit the parallel capabilities of the SP system.

- Flexibility

  The IBM Parallel System Support Programs for AIX software allows you to run various combinations of serial, parallel, batch and interactive jobs concurrently.

- Simplified operation

  The control workstation serves as a single point of control for installing, managing, and monitoring the SP system.

These features and capabilities make the SP an outstanding system for technical computing in a variety of market segments including Engineering Analysis, Chemistry, and Seismic/Petroleum. At the same time, the SP system is an excellent choice for the parallel commercial market, supporting database query, business management, decision support, online transaction processing, and batch business applications.

New Features in the IBM Parallel System Support Programs for AIX Version 2 Release 1

Version 2 Release 1 of the IBM Parallel System Support Programs for AIX provides support for AIX 4.1.3 on the SP system. Functional enhancements made in this release include:

© Copyright IBM Corp. 1995

Installation Support

- Automated configuration of the Network Installation Management (NIM) environment is supplied. The NIM environment provides for installation of AIX 4.1 on nodes. NIM masters are defined on the control workstation and on boot/install servers.
- Installation over Ethernet type DIX is supported.

Although new function has been added to installation support, to maintain ease-of-use, changes to the installation procedure are minimal.

System Partitioning

System Partitioning is a method for organizing the SP system into groups of nodes for various purposes such as testing new software and creating multiple non-interfering production environments. Each system partition appears to users and subsystems as a logical SP system within one administrative domain.

High Availability Control Workstation (HACWS) Function

High Availability Control Workstation Connectivity (HACWS) function provides for a backup control workstation for your SP system. Automatic fail over and reintegration to the backup control workstation are provided should the primary control workstation become unavailable due to scheduled hardware and software maintenance or unscheduled outages. (This feature will be available October 13, 1995.)

Node Isolation

The Node Isolation function provides for isolating nodes that are part of the High Performance Switch network from that network. Isolated nodes can be repaired, replaced, or rebooted without creating a High Performance Switch fault that would affect other users or applications.

Error Log Management

Error Log Management function provides for performing the following tasks:

- Managing the AIX Error Log
- Generating reports and trimming Syslog log files
- Performing general log archiving and viewing
- Collecting log and system data for IBM service

IBM Virtual Shared Disk Performance Improvement

The throughput of the IBM Virtual Shared Disk for remote I/O operations is improved.

Security

- The SP system authentication services provide for an increased maximum ticket lifetime of 30 days.
- Sysctl and remote commands functions can address any host by any of the host's names.

Heartbeat Dynamic Configuration

Heartbeat Dynamic Configuration function allows the introduction of a new node to an existing system without having to reboot the system.

Chapter 2. RISC System/6000 Scalable POWERparallel Systems Overview

This chapter provides an overview of of the hardware and software that compose RISC System/6000 Scalable POWERparallel Systems (SP).

Hardware Overview

The basic hardware components of an SP system are:

- Processor nodes
- Frames
- Optional High Performance Switch
- A control workstation
- Network connectivity adapters

These components connect to your existing computer network through a Local Area Network (LAN), making the SP system accessible from any network-attached workstation.

Figure 1 illustrates a basic SP suitable for parallel and serial batch technical computing in a departmental setting.

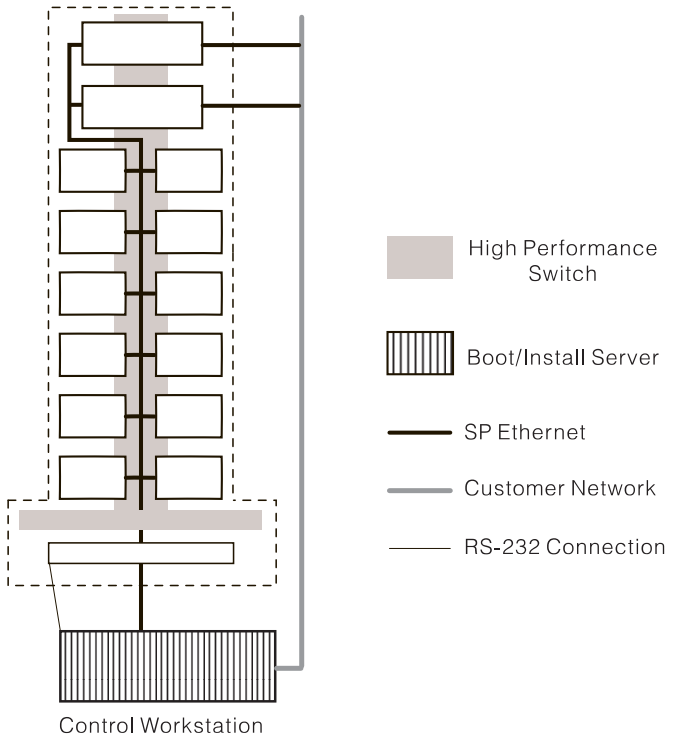


Figure 1. Basic SP Configuration

© Copyright IBM Corp. 1995

Processor Nodes

The SP system is scalable from two to 128 processor nodes that can be contained in multiple SP frames. A single frame can contain from two to 16 processor nodes.

The SP nodes are available in two types: thin nodes and wide nodes. Thin nodes are typically configured as compute nodes, while wide nodes are more often used as servers to provide high-bandwidth data access. However, no rigid rule governs the logical configuration of a node. You can configure a physical node type for the logical functions that best serve your computing requirements.

Each processor node includes a minimum of 64 megabytes of memory, one gigabyte of direct access storage devices (DASD), and an Ethernet adapter.

Thin Nodes

The SP thin node is functionally equivalent to a RISC System/6000 desktop system. Half the width of the wide node, it can be packaged 16 to a frame and can contain either a RISC System/6000 POWER or POWER2 processor. The POWER processor operates at 62.5 megahertz, and the POWER2 processor operates at 66.7 megahertz with 64 kilobytes of data cache, and 32 kilobytes of instruction cache. The thin processor node has four Micro Channel adapter (MCA) slots. It can have up to two SCSI disks packaged internally, and can have access to external storage devices attached through SCSI-2 adapter features. The Ethernet adapter for the SP Ethernet is integrated and does not use an MCA slot.

Wide Nodes

The SP wide node is functionally equivalent to a RISC System/6000 deskside system. It is a RISC System/6000 POWER2 processor that operates at 66.7 megahertz with 256 kilobytes of data cache, and 32 kilobytes of instruction cache. The wide node occupies twice the space of the thin node and can be packaged up to eight per frame. In addition to a maximum of four SCSI disks packaged internally, the wide node can have external storage devices attached through SCSI-2 adapters. It has seven MCA slots and also supports the High Performance Parallel Interface (HiPPI) adapter. The Ethernet adapter for the SP Ethernet uses one MCA slot.

Frames

An SP frame can house up to eight wide nodes or 16 thin nodes. Wide and thin nodes can be mixed together in a single frame.

The SP frame contains redundant power supplies; if one power supply fails, another takes over. The frame is also designed for concurrent maintenance; each processor node can be removed and repaired without interrupting operations on the other nodes.

High Performance Switch

The High Performance Switch provides low latency, high-bandwidth communication between nodes, supplying a minimum of four paths between any pair of nodes. The High Performance Switch dramatically speeds up TCP/IP, file transfers, remote procedure calls, and relational database functions. It consists of a switch assembly and the internal cables to support connection to 16 processor nodes in a system (one switch per frame). High Performance Switch capabilities include:

- Interframe connectivity and communication

- Scalability up to 128 node connections, including intermediate switch frames
- Constant bandwidth and latency between node pairs
- Support for Internet Protocol (IP) communication between nodes
- IP Address Resolution Protocol (ARP) support
- Support for dedicated or multi-user environments
- Error detection and retry

The powerful High Performance Switch Adapter-2 connects each SP node to the High Performance Switch subsystem. See Chapter 9, “The High Performance Switch” on page 161 for more information.

Control Workstation

The SP system requires a customer-supplied RISC System/6000 workstation with a color monitor. The control workstation serves as a point-of-control for managing and maintaining the SP processor nodes using the IBM Parallel System Support Programs for AIX. The workstation connects to each frame via an RS-232 line to provide hardware control functions. A system administrator can log in to the control workstation from any other workstation on the network to perform system management, monitoring, and control tasks. (See the section on control workstation planning in the *IBM RISC System/6000 Scalable POWERparallel Systems: Installation Guide* to ensure your workstation meets minimum requirements.)

The control workstation also acts as a boot/install server for other servers in the SP system. In addition, the control workstation can be set up as an authentication server. It can be the primary server, with the master database and administration service as well as the ticket-granting service. Or it can be set up as a secondary server, with a backup database and just the ticket-granting service. (For more information, see Chapter 3, “Security Features of the SP System” on page 17.)

The High Availability Control Workstation option provides for automatic failover and reintegration to a backup control workstation should the primary control workstation become unavailable.

Network Connectivity Adapters

Network connectivity for the SP system is supplied by various adapters, some optional, that can provide connection to I/O devices, networks of workstations, and mainframe networks. Ethernet, FDDI, token-ring, HiPPI, SCSI, FCS, and ATM are examples of adapters that can be used as part of an SP system.

The SP Ethernet is a network that connects all nodes and the control workstation in the SP. A 50-foot Ethernet cable is provided with each frame to use in the wiring of this network. Additional optional adapters such as Ethernet, FDDI, and token-ring are automatically configured on each node. Other optional supported adapters can be individually configured on each node.

Software Overview

The SP system software infrastructure includes:

- AIX, the base operating system
- The IBM Parallel System Support Programs for AIX
- Other IBM system and application software products

- Independent software vendor products

AIX

AIX provides operating system functions such as the AIXwindows user interface, extended real-time support, network installation management, advanced file system support, physical disk space management, and a platform for application development and execution. Some of the AIX capabilities that can be exploited for system management on an SP system include:

- The Network Installation Management (NIM) environment provides the ability to install an AIX **mksysb** image over the network
- The Logical Volume Manager (LVM), which improves data management productivity, enables files to span multiple disk drives, and provides, through disk mirroring, a high availability option for critical data
- The System Management Interface Tool (SMIT), which provides a single consistent, expandable interface to system management commands
- The System Resource Controller (SRC), which simplifies the management of subsystems and daemons
- Security features, which include a shadow password file and comprehensive auditing facilities
- AIX device support, which allows devices to be added and deleted at any time without disrupting the system

AIX provides UNIX functionality and conformance to industry standards for open systems.

IBM Parallel System Support Programs for AIX (PSSP)

The PSSP software provides a single point of control for administrative tasks and helps increase productivity by letting administrators view, monitor, and control system operation. The IBM Virtual Shared Disk device driver, which makes disks globally accessible; Distributed Tape Management System (DTMS) enablement; and High Availability Control Workstation Connectivity (HACWS) function are also available with the PSSP software.

Highlights

The PSSP software provides a comprehensive suite of applications for the installation, operation, management, and administration of the IBM RISC System/6000 Scalable POWERparallel System:

Installation: Installation of AIX 4.1.3 on nodes is provided for by the Network Installation Management (NIM) environment.

Configuration Management: Node configuration data is entered into the System Data Repository (SDR) using SMIT or line commands. The SDR provides storage and retrieval of system data across the control workstation, file servers, and nodes.

Operation:

- The Centralized Management Interface provides a SMIT-based interface for system management commands, as well as command line equivalents.
- The SP System Monitor and a command line interface provide for monitoring and controlling frame and node hardware.

- Tools are provided for orderly shutdown and restart of the system.

Communication Subsystem Support: The Communication Subsystem supports the reliability and performance of the High Performance Switch.

User Management:

- Commands are supplied to add, change, and delete users and passwords.
- Users have the same account, home directory, and environment across all nodes in the system.
- The **user.admin** file collection is provided for propagating user administration files to the nodes on the system.

Security: Authentication services, based on Kerberos IV, are provided for:

- The Sysctl remote command execution facility
- The SP System Monitor
- Versions of the **rsh** and **rcp** commands

File Management: Groups of files can be defined as a file collection and changes to any files in that collection will be propagated to the appropriate nodes.

Job Management: The Resource Manager server runs on one of the nodes and manages the execution of parallel jobs.

Print Management: Tools are supplied to shift printing functions from the nodes to a print server, eliminating the need to maintain and support print queues on a large number of nodes.

Accounting: Accounting support provides accounting record consolidation and parallel job accounting.

Change Management:

- System partitions can be used to apply and test new levels of software.
- Parallel management commands can be used to apply maintenance directly to nodes.
- File collections can be used to propagate changed files to nodes that have the file collection installed.
- For extensive service a new **mksysb** can be created and the nodes reinstalled using NIM.

Problem Management:

- A single point of control for configuring, archiving, and maintaining various system logs on individual nodes is provided.
- Additional logs may be specified for use by applications.

Packaging

The IBM Parallel System Support Programs for AIX (Program Number 5765-529) are packaged as the following **installp** images:

- **pssp**
- **spimg**
- **ssp.csd**
- **ssp.mte**
- **ssp.hacws**

pssp

The **installp** image for **pssp** contains the code to install, operate and manage your SP system. The options in this package include:

ssp.basic: The following support functions are included in this option:

- Installation and configuration
  (Refer to the *IBM RISC System/6000 Scalable POWERparallel Systems: Site Planning* and the *IBM RISC System/6000 Scalable POWERparallel Systems: System Planning* guides to help you prepare for installing and configuring an SP system.)
- System operation and monitoring
  (See Chapter 7, “Using the SP System Monitor” on page 117.)
- Distributed shell
  (See Chapter 10, “Parallel Management Commands” on page 169.)
- Login control
  (See Chapter 12, “Managing User Accounts” on page 193.)
- System Data Repository (SDR)
  (See Appendix E, “The System Data Repository” on page 429.)

ssp.gui: This option contains the graphical user interface for the System Monitor. Refer to Chapter 7, “Using the SP System Monitor” on page 117 for more information.

ssp.sysman: This option contains the following functions:

- Time services
  (See Chapter 13, “Managing NTP” on page 211.)
- User management
  (See Chapter 12, “Managing User Accounts” on page 193.)
- Print support
  (See Chapter 16, “Managing Print Service” on page 245.)
- File collections
  (See Chapter 14, “Managing File Collections” on page 213.)
- Accounting
  (See Chapter 18, “Accounting” on page 267.)
- Parallel management commands
  (See Chapter 10, “Parallel Management Commands” on page 169.)
- Error Log Management
  (See Chapter 20, “Managing Error Logs” on page 283.)

ssp.jm: This option contains the *Resource Manager* facility for allocating nodes to run parallel jobs, and for tracking parallel jobs across the SP system. It allows space sharing and time sharing of nodes as well as interactive and batch execution of parallel jobs. See Chapter 11, “Managing SP Resources” on page 179 for more information.

ssp.css: **Communications Subsystem Support** performs High Performance Switch initialization during system bring-up, and automatic switch reconfiguration following switch detected errors. CSS also provides software error detection codes for all messages, and concurrent servicing of SP nodes and frames through

message retry following a switch fault. Chapter 9, “The High Performance Switch” on page 161 contains more details.

ssp.top: This option contains directories and files for supported system partitioning configurations. Directories and files include:

- Subdirectories for SP system sizes
- Subdirectories for partition configurations for SP system sizes
- Layout directories for partition configurations
- A layout text file that describes the configuration
- Topology and nodelist files for the configuration

Refer to the *IBM RISC System/6000 Scalable POWERparallel Systems: Installation Guide* and Chapter 6, “Managing System Partitions” on page 95 in this guide for more information.

ssp.authent: This option contains the server (daemon) code that provides Kerberos ticket-granting services, and primary and secondary database administration. It also contains a number of privileged utility commands for manipulating the local Kerberos database. This option can be installed on the control workstation or on other RISC System/6000 workstations that are configured as Kerberos server systems. It is not installed on the SP nodes. Refer to the *IBM RISC System/6000 Scalable POWERparallel Systems: Installation Guide* and Chapter 3, “Security Features of the SP System” on page 17 in this guide for more information.

ssp.sysctl: *Sysctl* is an authenticated client/server system for running commands remotely and in parallel. Sysctl consists of a server daemon, commands, configuration files, and a client shell program. Refer to Chapter 4, “Sysctl” on page 51 for a full description.

ssp.clients: This option provides the nonprivileged (user) Kerberos commands, as well as client support for authenticated services, namely the SP System Monitor that runs on an SP control workstation and the **kshell** service of the SP authenticated remote commands (Kerberos-authenticated **rsh** and **rcp**). This option is required on the control workstation and is automatically installed on every SP node. It can also be installed on any other RISC/System 6000 workstation on which you desire these capabilities. Refer to the *IBM RISC System/6000 Scalable POWERparallel Systems: Installation Guide* and Chapter 3, “Security Features of the SP System” on page 17 in this guide for more information.

ssp.public: This option contains tar files of full public distributions of Amd (Berkeley Automount daemon), perl (Practical Extraction and Report Language), NTP (Network Time Protocol), SUP (System Update Protocol), Tcl (Tool Command Language), TclX (Tool Command Language Extended), Tk (Tcl-based Tool Kit for X-windows), and Expect (Programmed dialogue with interactive programs).

ssp.docs: This option contains the files for man pages and PostScript files and hypertext versions (accessible with the **InfoExplorer** tool) of this manual, as well as the *IBM RISC System/6000 Scalable POWERparallel Systems: Installation Guide*, the *IBM RISC System/6000 Scalable POWERparallel Systems: Diagnosis and Messages Guide*, the *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference*, the *IBM RISC System/6000 Scalable POWERparallel Systems: Site Planning* and the *IBM RISC System/6000 Scalable POWERparallel Systems: System Planning*.

spimg

This **installp** image contains a single file containing a **mksysb** image of a *minimal* AIX 4.1.3 system. This minimal system is composed of the basic operating system, extensions, networking and basic utilities. It is provided as a base for getting your SP system up and running quickly. You should customize this image to meet the needs of your production environment. Instructions for customizing the minimal image are in the *IBM RISC System/6000 Scalable POWERparallel Systems: Installation Guide*. See the IBM Parallel System Support Programs for AIX (PSSP) *Memo to Licensees* for a description of the contents.

ssp.csd

This image provides the IBM Virtual Shared Disk and the IBM Virtual Shared Disk Data Striping Device (HSD) function for the SP system. The IBM Virtual Shared Disk function allows each node in the cluster to access a disk as if it were local to the node, even though the disk is not physically attached to all SP nodes. The HSD is a device driver that lets application programs stripe data across the physical disks in multiple IBM Virtual Shared Disks. The **csd** image is split into three install options:

ssp.csd.vsd IBM Virtual Shared Disk Device Driver, Methods and Commands

ssp.csd.cmi IBM Virtual Shared Disk Configuration Management Support which includes the SMIT panels and the commands to interface with the System Data Repository (SDR).

ssp.csd.hsd IBM Virtual Shared Disk Data Striping Device

The **ssp.csd.vsd** and **ssp.csd.cmi** options must be installed on all SP nodes that are to be used as either IBM Virtual Shared Disk server or client nodes. The **ssp.csd.hsd** option must be installed on all nodes with IBM Virtual Shared Disks, if data striping will be used.

See Chapter 22, “Defining and Managing IBM Virtual Shared Disks” on page 311 and Chapter 23, “Defining and Managing IBM Virtual Shared Disk Data Striping Devices (HSDs)” on page 343 for details.

ssp.mte

The magnetic tape extension (**ssp.mte**) pseudo device driver provides enabling facilities that can be used to support a distributed, multinode tape management system. The **ssp.mte** pseudo device driver needs to be installed on every node that requires remote access. It also needs to be installed on every node that has a physical tape drive attached to it that will be used by a Distributed Tape Management System (DTMS). Refer to Chapter 24, “Distributed Tape Management System Enablement” on page 359 for more details.

ssp.hacws

The **ssp.hacws** install option supports High Availability Control Workstation (HACWS) function. HACWS provides automatic fail over and reintegration to a backup control workstation should the primary control workstation become unavailable.

ssp.hacws contains:

- Scripts to create a backup control workstation
- Error notification object samples
- Error log templates

- Object Database Manager updates
- Verification programs
- HACMP pre- and post-scripts
- Scripts to synchronize the backup and primary control workstations

Note

This is an optional separately priced feature that will be available October, 1995.

Other IBM Products

A large number of system and application software products developed by IBM can be run on the SP system. For example:

- IBM LoadLeveler
  - Distributed, network-wide job management
- IBM Parallel Environment for AIX
  - Parallel application development and execution
  - Message passing parallel task communications
- IBM PVMe for AIX
  - Message passing for parallel programs on distributed memory parallel computers
- IBM Parallel I/O File System
  - Data file system for parallel access to large files
- ADSM/6000
  - System and user data management
- PTX/6000
  - Performance monitoring
- NetView for AIX
  - Enterprise network management
- Trouble Ticket for AIX
  - Problem management
- PSF/6000
  - Printing support
- Job Scheduler for AIX
  - Schedules production batch workload in a distributed AIX environment

Independent Software Vendor Products

There is an aggressive program in place to enable and encourage independent software vendors to enable and port their applications to the SP system.

Security on the SP System

© Copyright IBM Corp. 1995

Chapter 3. Security Features of the SP System

This chapter includes information on the security services provided with the IBM RISC System/6000 Scalable POWERparallel Systems (SP). The security services are part of the IBM Parallel System Support Programs for AIX (PSSP) software that runs on the SP.

- "Understanding Authentication" provides an introduction to general security concepts and terminology, as well as an overview of the way authentication services are provided.
- "Using SP Security Services" on page 26 describes the tasks involved in running authenticated client-server applications supplied with the SP system.
- "Administering SP Security Services" on page 32 explains the administration of the security services, including the tasks of setting up authentication services on an SP system and on other workstations, adding principals to the authentication database, changing principals' passwords and other attributes, managing server key files, backing up the authentication database, and using secondary authentication servers.
- "Using AFS Authentication Services" on page 47 provides an overview of how AFS can provide an alternative to using PSSP authentication servers.

For more information about installing and configuring authentication services, refer to the *IBM RISC System/6000 Scalable POWERparallel Systems: Installation Guide*. For information about diagnosing authentication problems, refer to the *IBM RISC System/6000 Scalable POWERparallel Systems: Diagnosis and Messages Guide*.

Understanding Authentication

Administration of an SP system from a single point-of-control requires the use of administrative applications that are based upon the client-server programming model. Typically, a client-side command-line or GUI interface is used to perform an administrative task on one or more network-attached nodes or workstations, in parallel or serially. Where privileged applications such as these communicate across networks, security is compromised unless a reliable means is used to restrict access to authorized users. Accomplishing this authorization first requires authentication, which is the validation of the identities of client and server. Authentication is also the foundation for providing other security features such as accountability, discretionary access controls, and least privilege.

An SP system implementation, for example, can consist of SP nodes with Internet connections, a control workstation, and connections to other clustered workstations. Any of these system elements can be vulnerable to a security compromise. All networked services, as well as clients who may need to provide sensitive data to services, must be able to restrict access to those services according to a meaningful security policy. Such access restrictions can be successful only by the reliable authentication of the identities of clients and servers.

The PSSP software includes two types of administrative programs that benefit from these security features:

- Applications that manage the hardware resources of the SP system

© Copyright IBM Corp. 1995

- Applications that provide secure remote execution of SP administration commands. These include support for delegation of root user tasks to other users without distributing root's password.

SP authentication is based on Version 4 of the Kerberos authentication service developed at MIT. It was named after the mythical three-headed dog that guards the entrance to the underworld. The significance of the name is that a so-called “trusted third party”, namely a Kerberos authentication server, provides credentials that allow the other two parties (client and server) to reliably determine each other's identity.

Terminology

To understand the SP security services and how they are used and managed, you should be familiar with the following terms:

ACL Access Control List. A list that defines who has permission to access certain services; that is, for whom a server may perform certain tasks.

AFS A distributed file system that uses Kerberos authentication and includes authentication services that can be used by an SP system.

Authentication The process of validating the identity of either a user of a service or the service itself.

Authentication Database
A set of files containing the names and authentication information of all principals within a realm. An authentication realm has one primary database and may have multiple secondary databases. Secondary databases are backup copies of the primary database and may be provided to improve performance or availability.

Authenticator An authentication protocol string created each time authentication occurs and sent with the ticket to the server. It contains a time-stamp encrypted in the session key that can reliably show that the authentication request actually came from the client identified in the ticket.

Authorization The process of obtaining permission to access resources or perform tasks. In SP security services, authorization is based on the principal identifier.

Cell An independently administered collection of file server and client machines running AFS. An AFS cell is equivalent to a Kerberos realm.

Client The user of the shared services of a server. This term may refer to a program or to the person who invoked it. For authentication purposes it is a principal identifier registered in the authentication database.

Credentials A protocol message, or part thereof, containing a ticket and an authenticator supplied by a client and used by a server to verify the client's identity. The message can contain additional information used by the server to verify its identity to the client.

Data Encryption Standard (DES)
The secret-key (also known as “private-key”) encryption algorithm that is used by Kerberos.

Domain A collection of systems over which an administrator exercises control.

Instance A qualifier for a principal name. For services, an instance represents a particular occurrence of the server. For users, an instance allows a single user to assume additional (or alternate) roles with different authority.

Kerberos A service for authenticating users in a network environment.

Key An 8-byte value derived by encrypting a password or generating a random number, used to encrypt protocol strings used for authentication. Principal's private keys are stored in the authentication database. Session keys are contained (encrypted) in tickets and other protocol strings.

Kerberos Master Key
The key derived from the Kerberos Master Password supplied initially by the administrator when the primary SP authentication server is created. This key is saved in the **/.k** file, from whence the Kerberos daemons read it (in lieu of prompting for a password). It can also be read by certain database utility commands for the same purpose.

Kerberos Master Password
The password the administrator supplies when initializing the primary authentication server.

Principal An entity whose identifier and key are maintained in the authentication database. A principal can represent a user or a service. The format of a principal identifier is *name.instance@realm*, where *name* is either a user's login-name or the name by which a server identifies itself to Kerberos.

Realm A domain which shares an authentication database and servers. There is a single name-space for principal name/instance pairs within a realm. A realm is also a logical collection of clients and servers registered in the database.

Server A functional unit (usually a daemon program and child processes) running on a particular host that provides shared services to users over a network.

Service The name of a principal whose identity is assumed by a server for purposes of authentication. Multiple servers can use the same service name.

Server Key File (Also referred to as a **srvtab**) A file containing the names and private keys of the local instances of services. It is accessible only to processes that run under the UID of the user owning the server daemons. On an SP system all services run as root.

Session Key A temporary key, a random 64-bit value, supplied by an authentication server to clients and servers, that is used to encrypt parts of authentication protocol messages. Its lifetime is the same as the ticket with which it is created.

Ticket An encrypted protocol message used to pass the identity of a user from a client to a server. Tickets are created by the Kerberos authentication server and cached in disk files on the client's system.

Ticket-granting-ticket
The initial ticket obtained by a user. It is used by client programs to obtain additional tickets for authentication with application services.

Kerberos Authentication

SP authentication services use the authentication database and the server to:

- Initially authenticate the identity of the user
- Provide the information through which a server can authenticate the identity of a client (user) in a distributed environment

The authentication services provided in the SP System are based on MIT Kerberos version 4. Kerberos servers function as a trusted third party to authenticate the identities of users and distribute encryption keys to clients and servers, known as *principals*. Every user and every service instance on a particular host is a principal in the Kerberos database. Each Kerberos principal has an identifier, a password, an expiration date, and a maximum ticket lifetime registered in the authentication database. Kerberos does not store the clear-text password, but a 64-bit binary value derived from it using a one-way hash. Database information is also encrypted in the Kerberos master key, known only to the database administrator. Passwords are never sent across the network in clear text-form, either for user authentication or during database administration.

Overview of Kerberos Interaction with Client and Server

On the authentication server system there are two daemons that access the database. The **kadmind** daemon provides network access for database administration via the **kpasswd** and **kadmin** commands. The **kerberos** daemon is the actual authentication server. It is sometimes referred to as the “Key Distribution Center (KDC)” because of its role in distributing session keys. The KDC actually consists of two separate functions: the Authentication Service, which generates ticket-granting-tickets based on authentication using the client user's private key; and the Ticket Granting Service (TGS), which generates service tickets, based on authentication using a ticket-granting-ticket.

The underlying mechanism for accomplishing the authentication is a ticket scheme, consisting of the following steps, illustrated in Figure 2 on page 21.

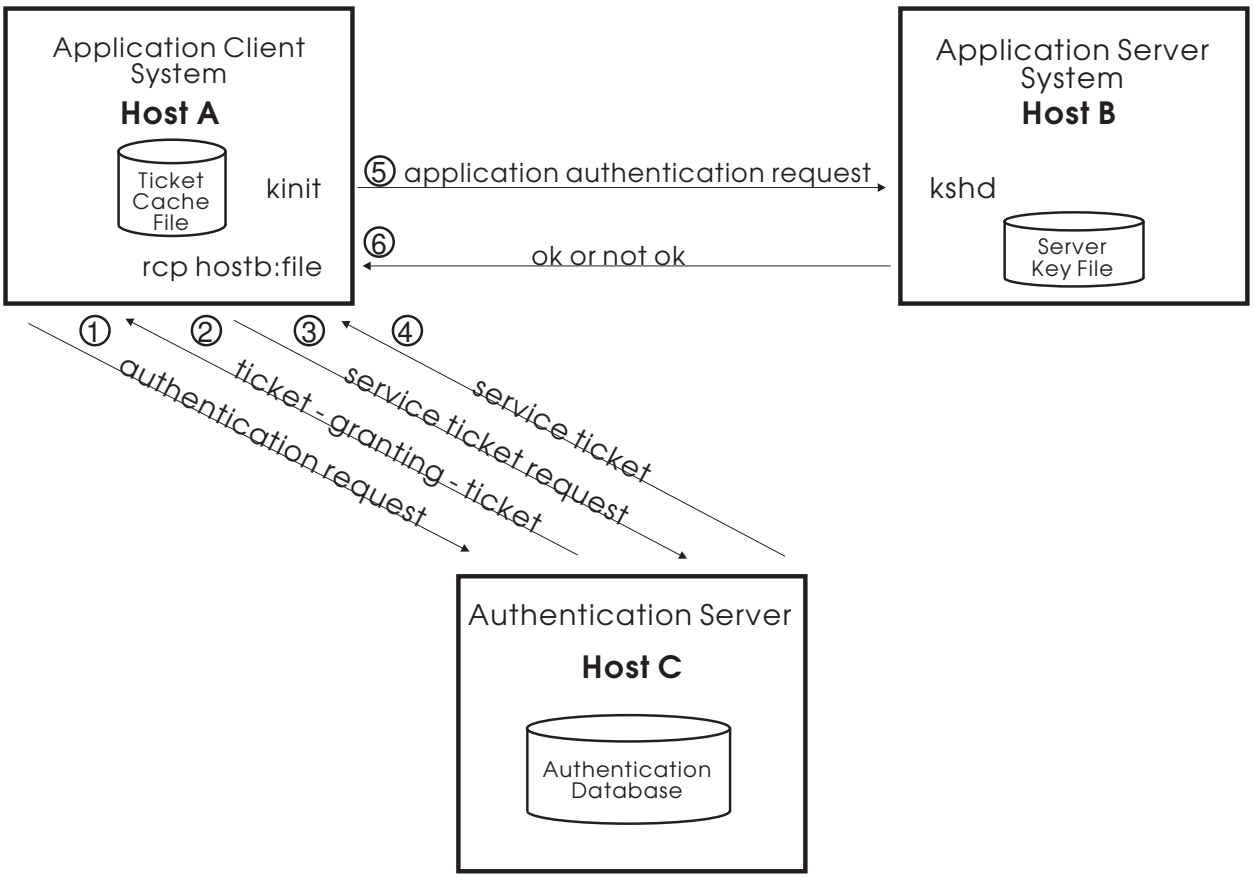


Figure 2. An Overview of Kerberos Authentication

1. The client process, usually as the result of a user invoking the **kinit** command, sends a message to the Authentication Service containing its principal identifier and the identifier of the Ticket Granting Service (a predefined Kerberos principal). This message is not encrypted. The Authentication Service generates a random session key to be shared by the client and the TGS and creates a ticket for the client to use with the Ticket Granting Service. It contains the client identifier and the session key, as well as fields defining the ticket's valid lifetime. The ticket is encrypted in the private key of the Ticket Granting Service. A return message is built, containing the key, the client's identifier, and the ticket-granting-ticket. The Authentication Service encrypts the message in the private key of the user, a one-way hash of the password.
2. The client program reads the user's password and applies the one-way hash necessary to determine the private key. If the password is correct, the key can be used to successfuly decrypt the returned message block containing the session key, the client identifier, and the ticket. When successful, the ticket and the session key are stored in the current ticket cache file.
3. The user invokes an application's client interface, which needs to perform authentication on its first network communication with its server. It creates an authenticator, containing the client identifier and a current time-stamp, and encrypts the authenticator using the sesion key returned with the ticket-granting-ticket. It sends to the Ticket Granting Service a message consisting of the authenticator, the ticket-granting-ticket, and the principal identifier of the service instance which it wants to use. The TGS reads its private key from the authentication database and uses it to decrypt the ticket,

obtaining a copy of the sesssion key and the identifier of the client to whom the ticket-granting-ticket was issued. Using the session key, it decrypts the authenticator. The TGS verifies that the client's identity is the same, and that the authenticator's time stamp indicates that it was very recently created.

4. Like the Authentication Service, the TGS generates a session key, this one to be shared by the client and the application server. It also generates another ticket, called a service ticket, encrypted in the private key of the application service principal. The ticket is returned to the client in a message encrypted with the old session key (shared by the client and the TGS).
5. The client program uses the original session key to decrypt the message. It extracts the new session key and ticket, and saves them in the ticket cache file. When the client program sends its first application protocol message to the application server, it includes the service ticket and a new authenticator, encrypted in the new session key. This session key was also placed in the ticket by the TGS, so the server can retrieve it when it decrypts the service ticket. In order to do so, the application server obtains its private key from the server key file, **/etc/krb-srvtab** on the server system.
6. The authentication by the application server is performed in the same way as by the TGS. The message returned to the client simply indicates success or failure, and an error message number if failure.

The scenario outlined above assumes no unexpired service ticket was found in the ticket cache file. The ticket-granting-ticket can be used to get as many different tickets as are needed, without requiring the user to reenter the password.

A client's service ticket can be used to re-authenticate as often as the client and server require. Typically, clients using TCP/IP authenticate once on connection to the server. Users of connectionless protocol should authenticate each message exchange, though less rigorous schemes can be used following an initial authentication, by using the shared session key to encrypt a small piece of known data.

Kerberos and DES

The SP authentication services use the Data Encryption Standard (DES) algorithm. The DES algorithm is used only as an internal mechanism for authenticating the identities of clients and servers for SP services. In order to comply with U.S. export regulations on the DES algorithm, SP authentication services provide a subset of the full MIT Kerberos version 4 services. No interface is provided for using the encryption routines directly. No interface is provided to allow the encryption of any portion of service messages that are not directly connected with the authentication of the client and server identities.

For additional information, refer to the **Kerberos** man page in the *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference*.

Types of Principals

Kerberos principals are either users who use authentication services to run the Kerberos-authenticated applications supplied with the SP system or the individual instances of the servers that run on SP nodes, the control workstation, and on other IBM RISC System/6000 workstations that have network connections to the SP system.

User Principals for SP System Management

An implementation of the SP system must have at least one user principal defined. This user is the authentication database administrator, who must be defined first so that other principals can be added later. When AFS authentication servers are being used, the AFS administrator ID already exists when the SP authentication services are initialized. When PSSP authentication servers are being used, one of the steps included in setting up the authentication services is the creation of a principal whose identifier includes the **admin** instance. It is recommended, but not essential, that the first authentication administrator also be the root user. Various installation tasks performed by root, or other users with UID 0, require the Kerberos authority to add service principals to the authentication database. At sometime prior to invoking the **setup_server** command to complete setup of the control workstation and boot server nodes, a user with UID 0 must be defined as a Kerberos database administrator with authority to add other principals. This is because the service principals used by Kerberos-authenticated applications are all defined automatically by **setup_server** during installation and customization. Apart from this installation requirement, you may use other user principal identities to perform system administration (and authentication administration) tasks.

Service Principals Used by PSSP Components

Two service names are used by the Kerberos-authenticated applications in an SP system:

1. **hardmon**, used by the System Monitor daemon on the control workstation and by logging daemons
2. **rcmd**, used by **sysctl** and the authenticated **rsh** and **rcp** commands

Kerberos version 4 conventionally uses the short form of the hostname where a server daemon runs as the instance in the server's principal identifier. Different principal identifiers thus indicate the target network address of the different providers of each service. The authentication support in the SP system expands the Kerberos implementation to support workstations with multiple network interfaces as servers. This provides instances for all network interfaces rather than just the standard hostname. Kerberos works with the resolved network names, rather than any aliases you may use locally on your hosts.

The **hardmon** daemon runs only on the control workstation. The SP logging daemon, **splogd**, can run on other IBM RISC System/6000 workstations. Therefore, for each (short) network interface name on these workstations, a service principal is created with the name **rcmd** and the network name as the instance.

The remote commands can be run from, or to, any IBM RISC System/6000 host on which the SP system authenticated client services (**ssp.clients**) are installed. Therefore, for each (short) network interface name on all SP nodes, the control workstation, and other client systems, a service principal is created with the name **hardmon** and the network name as the instance.

Authentication Realms

The authentication realm is the set of systems that is served by a single logical authentication database. A realm can include any combination of one or more SP systems, IBM RISC System/6000 workstations, and other systems/workstations. The authentication database for a realm is administered as a single entity: principals are defined across the systems within a realm and principal identifiers must be unique within a realm.

There are options for defining authentication realms within a network of systems:

- Each system may be its own realm
- A single realm may include all of the systems in a network
- Systems may be grouped into realms with multiple realms occurring within a network

Generally, systems that are administered together, or by the same set of administrators, should be grouped into a single authentication realm. For example, one or more SP systems and their associated workstations could form a single authentication realm. Within an SP system, all SP nodes are automatically added to the control workstation's local realm during installation.

If multiple realms are configured within a network, the realm names should be unique. During setup of the SP authentication services, the name of the realm defaults to the local system's network domain name converted to uppercase. If a different realm name is going to be used, the local system's realm name must be supplied by the administrator in the /**etc/krb.conf** file. For information on the format of this file, refer to the *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference*.

If you are using AFS authentication servers, the entire SP system must be in the same cell. The authentication realm known to the SP authentication services is the local AFS cell. The realm name is set to the AFS cell name converted to uppercase.

Although a realm is served by a single logical authentication database, multiple copies of the database may exist on different workstations within a realm. One must be the primary database in a realm; others are secondary servers.

Understanding Tickets

Kerberos tickets are encrypted byte strings returned to clients by an authentication server. Tickets contain the client principal's identifier, a time-stamp and ticket lifetime used to determine the expiration time, the session key, and other information, all encrypted in the private key of the service instance with whom the client wishes to authenticate. In the case of a ticket-granting-ticket, such as that created by the **kinit** command, that service is the Ticket Granting-Service. Other tickets are application service tickets, encrypted in the private key of one particular instance of an application service. The client cannot understand the key; it simply forwards it to the server with an authenticator in an internet protocol packet, using either UDP or TCP socket communications.

Ticket Cache Files

The caching of tickets takes place automatically in disk files on the client system that is the workstation or SP node on which the client user is logged in. These files are created with UNIX permissions set so they can be accessed only by the owner, for they contain sensitive information. Any process using the tickets can authenticate as the owner of the file.

A user can have tickets cached in any number of files simultaneously, and may have reason to do so. If you have multiple logins, run background processes, or use a shared UID, you must do so to avoid destruction of tickets still in use by other processes. This is accomplished by setting the KRBTKFILE environment variable to the path name of a unique ticket cache file prior to using authentication

services for each login session or background process. If you do not set KRBTKFILE, then your tickets will be cached in your default file: /**tmp**/**tkt***uid*.

The first ticket stored in the file is always the ticket-granting-ticket. The file is created by the **kinit**, **ksrvtgt**, or **rcmdtgt** command; and when the TGT is stored, all previously issued tickets in the file are deleted. The file contains, in addition to the TGT, any number of other tickets for application services. There will be one for each service instance with which you have communicated using that particular current ticket cache file. They are obtained by the client commands for the various Kerberos-authenticated services, such as **rcp** and **rsh** (and indirectly by SP system administration tools that invoke them).

Kerberos does not automatically delete tickets after they expire. Users must either explicitly destroy their ticket cache files or replace them by re-authenticating (by obtaining a fresh TGT). Tickets are never removed individually from the ticket cache file. The **kdestroy** command eliminates the entire current ticket cache file. Ticket cache files other than the current file are untouched, however.

Ticket-lifetimes

The tickets that you obtain when you issue the **kinit** command are valid only for a limited time. The same limit applies also to service tickets obtained by client programs that you execute. In most circumstances, you probably will not be concerned about how this limit is determined and applied. However, when you run **kinit**, you have the option of requesting a ticket with a specific lifetime. The maximum lifetime you may specify, which is also the default value, was assigned when your principal was registered in the authentication database, or it was changed subsequently by the database administrator. The minimum ticket lifetime is five minutes; the maximum lifetime can be set to one of several discrete values from five minutes to thirty days. The ticket lifetimes of service tickets are always the same as the lifetime of the ticket-granting-ticket used to obtain them. It is the TGT held in the current ticket cache file owned by the user who invokes the application client command.

The external representation of the ticket lifetime is rather inconsistent among the various user and administrator commands. Some commands represent the one-byte ticket lifetime value as a decimal integer between 0 and 255. This is the representation used by the administrator when creating or modifying a principal using the **kdb_edit** command. It is also the format of the ticket life field displayed by the **get** subcommand of **kadmin**. The output of the **klist** command shows the actual ticket creation and expiration as standard date and time stamps. If you specify **-l** to request a a specific ticket lifetime when you run **kinit**, you are prompted to enter the value as a number of minutes. The value you enter is then rounded up to the next higher discrete lifetime interval or the maximum ticket lifetime allowed for your principal, whichever is less, and then mapped to a one-byte value between 1 and 191.

The following table shows the ticket lifetimes that correspond to the encoding used internally by Kerberos. The values on the left of each column are the one-byte encodings; on the right, the lifetime interval in days, hours, and minutes.

Ticket lifetimes from 0 to 128 represent multiples of 5 minutes, for example:

```
0-1     00:05    12     01:00    24     02:00    36     03:00    48     04:00
 60     05:00    72     06:00    84     07:00    96     08:00   108     09:00
120     10:00   125     10:25   126     10:30   127     10:35   128     10:40
```

Lifetimes from 129 to 255 represent intervals of from 11:24 to 30 days:

```
129     11:24   130     12:11   131     13:02   132     13:56   133     14:54
134     15:55   135     17:01   136     18:12   137     19:28   138     20:48
139     22:15   140     23:47   141  1d+01:26   142  1d+03:11   143  1d+05:04
144  1d+07:05   145  1d+09:14   146  1d+11:32   147  1d+13:59   148  1d+16:37
149  1d+19:25   150  1d+22:26   151  2d+01:38   152  2d+05:04   153  2d+08:44
154  2d+12:40   155  2d+16:51   156  2d+21:21   157  3d+02:08   158  3d+07:16
159  3d+12:45   160  3d+18:36   161  4d+00:52   162  4d+07:34   163  4d+14:44
164  4d+22:23   165  5d+06:35   166  5d+15:20   167  6d+00:41   168  6d+10:42
169  6d+21:23   170  7d+08:50   171  7d+21:03   172  8d+10:08   173  9d+00:06
174  9d+15:03   175 10d+07:01   176 11d+00:06   177 11d+18:22   178 12d+13:53
179 13d+10:46   180 14d+09:05   181 15d+08:56   182 16d+10:27   183 17d+13:44
184 18d+18:53   185 20d+02:04   186 21d+11:24   187 22d+23:02   188 24d+13:08
189 26d+05:52   190 28d+01:26   191+ 30d
```

Using SP Security Services

Managing Your Authentication Credentials

Users who are going to use the Kerberos-authenticated applications that are provided with the SP system, as system administrators or ordinary users, need to understand how to perform a few simple tasks to manage their authentication environment. These tasks involve:

- Obtaining Kerberos credentials
- Examining Kerberos credentials
- Deleting credentials that are no longer needed
- Changing their Kerberos passwords

When your realm is set up to use PSSP authentication servers, the PATH variable for these users should include **/usr/kerberos/bin**. When AFS authentication servers are used, the PATH variable should include **/usr/afsws/bin:/usr/kerberos/bin**.

Obtaining Authentication Credentials

The first step in using the SP authentication services is to login to the authentication system. Logging in to the SP authentication services is how you obtain a ticket-granting ticket. The ticket-granting-ticket permits the client programs you invoke, such as **sysctl**, to obtain service tickets they can use to certify your identity to the servers with which they communicate.

Logging in to SP authentication services is a separate step from logging in to your AIX system; the Kerberos principal name space is separate from the AIX user name space. The administrator may assign principal names and passwords that are the same as AIX login names and passwords.

With PSSP Authentication: Use the **kinit** command to login to the SP authentication services. You can either enter the **kinit** command interactively as needed, or you could add a **kinit** command to your login script (**.login** or **.profile** file). If you make infrequent use of the Kerberos-authenticated services, you might prefer the first option.

The easiset way to use the **kinit** command is to specify your Kerberos principal identifier as the only argument with no options.

kinit *principal-identifier*

For the principal identifier, specify your Kerberos user name, followed by a period and the instance, if you have more than one Kerberos identity and the instance you want to use is not null. Append an ”at” sign (@) and a realm name if the realm you are logging into is not the local realm. Since the entire SP system is in a single realm, you will probably never need to specify a realm. Examples of this form of **kinit** are:

```
kinit joeuser
kinit root.admin
kinit ben@XYZ.ABC.COM
```

You might also choose to invoke **kinit** without arguments. It responds by prompting for your Kerberos name, but only for the name portion of the fully qualified identifier.

```
kinit
Kerberos name: ben
```

A common error is to omit the identifier on the command line and try to enter both name and instance in response to the prompt for the name. The result is:

```
kinit
Kerberos name: root.admin
2502-003 Bad Kerberos name format
```

Instead, you can enter the instance separately by specifying the **-i** flag on the command line and responding to the prompt:

```
Kerberos instance:
```

Similarly, the realm qualifier can only be entered as part of the command-line identifier unless you use the optional **-r** flag.

The other option that you might want to use, **-l**, allows you to request a ticket with a lifetime shorter than the maximum allowed for your principal by the database administrator. If you specify this flag, **kinit** prompts you:

```
Kerberos ticket lifetime (minutes):
```

After you enter the **kinit** command and respond to these optional prompts, the command contacts the authentication server to obtain a ticket. If no authentication server can be contacted, you will be notified of a time-out and the command will fail. Reasons for failure could be a missing or invalid configuration file or a networking problem.

When a ticket is obtained, **kinit** prompts you for your Kerberos password. If you enter it correctly, **kinit** saves the ticket-granting-ticket in the ticket file specified by your KRBTKFILE environment variable. If this variable is undefined, your ticket is stored in the file /**tmp**/**tkt***uid*, where *uid* specifies your AIX user identification number.

When the ticket-granting-ticket expires, subsequent attempts to be authenticated to use a Kerberos-supported service will fail. If the ticket-granting-ticket has expired, you must use **kinit** again to obtain a new ticket. There is no automatic renewal of tickets upon their expiration.

When you use **kinit**, it is valid for all processes running with your AIX user ID. It is not necessary to do a **kinit** for each window you open, as long as you have no background processes that require authentication. Note, however, that if you run authenticated applications as root in background processes or login under a shared user ID, you can potentially destroy the tickets of someone who logged in prior to you using the shared ID. In those cases you should use the KRBTKFILE environment variable to specify multiple ticket cache files. If you are uncertain of the current ticket cache file, use the **klist** command to display the name and content of the file.

Refer to the **kinit** and **klist** commands in the *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference* for more information.

With AFS Authentication: In most situations users of your SP system will be able to use the same **kinit** command described above without regard to which type of authentication servers are being used. **kinit** can always be used whenever a user is logged into an SP node or the control workstation. When the user is logged into another IBM RISC System/6000 workstation which has been set up as an authenticated services client system, **kinit** can be used if the user has the SP_NAME environment variable set to the network name of the control workstation in the local realm.

Users who do not have the SP_NAME variable set, or who wish to use more familiar AFS commands, should use the **klog.krb** command. The following example shows the format of this command:

```
klog.krb -pr root.admin -li 48:00
```

When the **admin** instance is included in the principal identifier, **klog.krb** responds with a warning message:

```
Non-null instance (admin) may cause strange behavior.
```

You can ignore this message, since these principals are not used with the AFS file system.

As with **kinit**, a lifetime can be specified, but in a more reasonable *hh:mm* format.

Displaying Your Credentials

With PSSP Authentication: Use the **klist** command to display a list of currently held authentication tickets and their expiration time. The output would appear like the following, for example, if the user **ben** enters the **klist** command after using the **kinit** command followed by any of the remote command services on four other hosts:

```
Ticket file:    /tmp/tkt12763
Principal:  ben@XYZ.ABC.COM

  Issued           Expires          Principal
Aug 12 16:26:11  Sep 11 16:26:11  krbtgt.XYZ.ABC.COM@XYZ.ABC.COM
Aug 12 16:45:15  Sep 11 16:45:15  rcmd.spcwkst@XYZ.ABC.COM
Aug 15 09:02:42  Sep 14 09:02:42  rcmd.spnode3@XYZ.ABC.COM
Aug 15 09:02:43  Sep 14 09:02:43  rcmd.spnode5@XYZ.ABC.COM
Aug 15 09:02:43  Sep 14 09:02:43  rcmd.spnode6@XYZ.ABC.COM
```

Refer to the **klist** command in the *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference* for more information.

With AFS Authentication: You may also use the **klist** command to display tickets obtained from an AFS authentication server. The format of the output is the same as shown above, except that a service ticket for the "afs" service will appear, if you used the AFS **klog.krb** command to identify yourself. This additional service ticket is not used by SP authentication services. As an AFS user, you may prefer to use the **tokens.krb** command to display your tickets. Its output will appear like:

```
Tokens held by the Cache Manager:

User's (AFS ID 12763) tokens for afs@afs.abc.com [Expires May 31 03:24]
User ben's tokens for krbtgt.AFS.ABC.COM@afs.abc.com [Expires Jun 25 19:49]
User ben's tokens for rcmd.wksta2@afs.abc.com [Expires Jun 25 19:50]
   --End of list--
```

Notice that **tokens.klog** shows the AFS cell name in place of the Kerberos realm name.

Deleting Your Credentials

With PSSP Authentication: Use the **kdestroy** command to destroy Kerberos tickets in the current ticket cache file. Depending on your local security policy, you may want to destroy your tickets whenever you end your AIX login session. **kdestroy** overwrites the ticket cache file with nulls before removing the file, so storage containing credentials cannot be reused. When you enter the **kdestroy** command, the system will respond:

```
Tickets destroyed.
```

If the ticket cache file does not exist, you will get the response:

```
No tickets destroyed.
```

Refer to the **kdestroy** command in the *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference* for more information.

With AFS Authentication: You may also use the **kdestroy** command to delete tickets obtained from an AFS authentication server and stored in a ticket cache file. The responses are the same as shown above. **kdestroy** will not remove the token held by the AFS Cache Manager, however, that was also obtained if you used **klog.krb** instead of **kinit** to obtain your initial ticket.

The AFS **unlog** command will remove your tokens held by the AFS Cache Manager, but will not remove tickets from the Kerberos ticket cache file.

Changing Your Authentication Password

With PSSP Authentication: To change your password, use the **kpasswd** command:

1. Enter the **kpasswd** command:
2. At the prompt, enter your old password
3. At the prompt, enter your new password
4. At the prompt, reenter your new password

For example:

```
kpasswd
Old password for ben:
New Password for ben:
Verifying, please re-enter New Password for ben:
Password changed.
```

With AFS Authentication: To change a password for a principal in AFS:

1. Enter the AFS **kpasswd** command. Be sure you have the directory for the AFS commands, /**usr**/**afsws**/**bin** ahead of the directory path for the Kerberos command, /**usr**/**kerberos**/**bin** in your search order, or specify the full pathname:
2. At the prompt, enter your old AFS password
3. At the prompt, enter your new password
4. At the prompt, reenter your new password to verify

For example:

```
kpasswd
Changing password for 'ben' in cell 'abc.com'
Old password:
New password (RETURN to abort):
Retype new password:
Password changed.
```

SP Authenticated Services

The SP System provides authentication services for:

- The SP **rsh** and **rcp** commands
- The Sysctl remote command execution facility
- The SP System Monitor

Separate chapters describe the configuration and use of the System Monitor and the Sysctl remote command execution facility.

Authenticated Remote Commands on the SP System

SP authentication services provide authenticated versions of **rsh** and **rcp**. The use of the authenticated versions of **rsh** and **rcp** removes the requirement for a root level **.rhosts** file.

Kerberos version 4 authentication services must be installed on both the source host issuing and the target host servicing the authenticated remote commands. Refer to “Setting Up PSSP Authentication” on page 33 for more information.

To use the **rsh** and **rcp** authenticated services, a user or client must be included in the authentication database and must first issue the **kinit** command.

If you cannot be authenticated or if authentication services fail, messages are issued and the SP authenticated remote commands pass the arguments to the corresponding AIX remote commands.

Remote Command Authorization

The root user's **.klogin** file contains a list of principals that are authorized to invoke processes as the root user with the SP authenticated remote commands (**rsh**, **rcp**). A **.klogin** file is created in the root user's home directory rather than a root **.rhosts** file. No root **.rhosts** file is used by the SP system management installation and configuration programs. The **.klogin** file contains service principal entries to allow the system management installation and configuration programs to authenticate without requiring login as an administrative user principal. The service tickets allow these programs running on the SP nodes to communicate with their boot/install servers and the control workstation without forcing the root user to log in at the nodes and perform a **kinit** as administrative user during the installation process and/or administrative operations.

Although the **.klogin** file may contain more entries, the following are required for the installation and operation of the SP system authenticated remote commands and **sysctl** to perform properly. These entries are created during installation on the control workstation and each SP node:

- An entry for the administrative user
- An entry for each host in the SP System

in the form

`rcmd.`*`hostname`*`@`*`realm`*

where *hostname* is the short hostname.

A principal name for the root ID is not automatically added to the **.klogin** file or the authentication database. Refer to **.klogin** file in the *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference* for more information.

Adding Principals and Distributing the root .klogin File: The root **.klogin** file on the control workstation is distributed to an SP node when it is rebooted. Note that you should only add principals to the **.klogin** file on the control workstation; don't delete the principals that already exist in the file.

If you wish to add principals to the root **.klogin** file and distribute the new file without rebooting, you have several options:

- **dsh**

  You can use **dsh** to copy the **.klogin** file from the control workstation to each node in the SP System.

- **pcp**

  You can use the parallel copy command, **pcp**, to copy the **.klogin** file from the control workstation to each node in parallel.

- File collections

  You can use file collections to manage the **.klogin** file. The appropriate collection to place the file in would be **node.admin**.

File collection information can be found in Chapter 14, “Managing File Collections” on page 213. Refer to the *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference* for information on **dsh** and **pcp**.

Administering SP Security Services

Protecting Your Authentication Database

An important consideration when you install and configure your SP system is the security of the workstations on which you will be running Kerberos authentication servers. Because it is the repository of all principals' secret keys, you need to consider steps to protect it from break-in or compromise of the master password. The following are steps you should consider:

- Authentication server systems should be located in physically secure areas, with access limited to administrators authorized to perform tasks related to their maintenance.
- Ordinary users, including SP system administrators who are not authorized as security administrators, should not have user IDs on these systems. Remote access to them via **telnet**, **rlogin**, and **ftp** should be disabled.
- AIX system auditing should be enabled on these systems to record security events.
- An appropriate password selection and aging policy should be implemented for the master password, and it should be changed regularly.
- Because compromise of the master password would expose all private keys stored in the database, a recovery plan should be available that would provide for changing all passwords, replacement of all server key files, and destruction of all outstanding tickets.

The root User and Authentication Administration Roles

When you set up your authentication realm and install and configure your SP system, the roles of system installer and administrator and security administrator become intertwined. This is necessitated by the fact that the PSSP installation and customization scripts are designed to automatically recognize the service principals that have to be created for the **hardmon** and **rcmd** services. The **setup_server** script creates service principals whenever it discovers a new network interface defined on the control workstation for which it has no Kerberos principal defined. At these times, the script must be running in an environment where the invoking user has authentication credentials as a database administrator. If the script needs to add a principal but finds that you have not got a ticket as an **admin** principal, it will fail. If this happens, simply get the required ticket and issue the command or invoke it from SMIT again.

Other installation and customization tasks invoke the **setup_server** script internally. Any of these tasks which involve adding or renaming of adapters or hostnames affect the Kerberos principals, and will also require a ticket as a database administrator. Examples of installation scripts used to perform these tasks are **spadaptrs**, **sphostnam**, **spbootins**, and **spethernt**.

Other administration tasks that simply use the remote command facilities of **dsh**, **rsh**, or **rcp** require normal user principal's credentials, since they do not involve changes to the authenication database.

Setting Up PSSP Authentication

To provide authentication services on your SP System, you must choose from the following three authentication implementations.

1. The authentication services provided with the SP System, based on MIT Kerberos version 4. SP authentication services are designed to provide an easily-installable and configurable Kerberos for the SP System.
2. An existing implementation of MIT Kerberos version 4, provided it is compatible with the SP Kerberos-authenticated services.
3. AFS version 3.3 or 3.3a.

Once you have installed and initialized authentication on your SP system, you might want to add other IBM RISC System/6000 workstations to your authentication realm. Two reasons are:

1. If you are not using AFS authentication servers, you may want the added reliability of one or more secondary SP authentication servers. Providing secondary servers is particularly important if your SP system has to provide high availability with no single points of failure.
2. You or other administrators may also want to install the SP authenticated services on office workstations. This will allow system administrators to manage the SP system, without having to login to the control workstation or the SP nodes directly.

When you use the SP authentication server support in the **ssp.authent** option, you may choose to have more than one server system. If you configure your local authentication realm with more than one server, you must designate one as the primary server. All others are secondary servers. Only the primary server has the administrative daemon, **kadmind**, that manages the content of the authentication database. The databases used by the secondary authentication servers are copies of the primary database and are updated periodically from the primary authentication server.

Additional servers can provide for greater reliability, since the authentication service will try all servers listed in the configuration file before failing a service request. If there are network problems, or if your primary server system fails, authentication requests can be handled as long as one of the configured servers is active and accessible. There may also be performance reasons to configure a secondary server. Only when there is an authentication server running on the control workstation can the High Performance Switch be used for authentication protocol traffic to and from the SP nodes. In configurations where you are integrating the SP system into an existing authentication realm, where the primary authentication server is already on another workstation, you can make this possible by setting up a secondary server on the control workstation. In this way, for example, you could have multiple SP systems in the same authentication realm, and give all SP nodes access to an authentication server across the switch.

The addition of principals and changing of passwords takes place in the primary database, and is propagated to secondary databases through these periodic updates only. Secondary databases are maintained by a special daemon called **kpropd**, that runs only on secondary server workstations and receives the database content in encrypted form from a program called **kprop** that runs on the primary server workstation. The SP authentication services include a script that

you can schedule for execution in the root **crontab** file to keep your secondary authentication databases up-to-date.

Configuration Files

There are several configuration files in the **/etc** directory that may be used in setting up your SP system's authentication realm:

- **/etc/inetd.conf**
- **/etc/services**
- **/etc/krb.conf**
- **/etc/krb.realms**

inetd.conf: The **inetd.conf** file contains information used by the internet routing daemon **inetd** to route incoming requests for service to one of a large number of dynamically started daemons, named in the file. When the SP athentication services are installed on your systems, the file is updated to route "kshell" service requests to the Kerberos-authenticated remote command daemon named **kshd**. You should not modify this information, but could possibly have to resolve conflicts with locally installed services with conflicting names.

/etc/services: The **/etc/services** file maps the names of network services to the well-known ports that they use to receive requests from their clients. This file may already contain entries relating to authentication services, since the file shipped with AIX 4.1 contains entries used by DCE. In addition to the DCE entries, many other reserved port names were added in this version of AIX, including an entry for the Kerberos 5 port, 88. The service name for this port is given as "kerberos", which is also the name used by the standard MIT Kerberos 4 service. The port number usually assigned to the Kerberos 4 service is 750. In order to be consistent with and interoperate with AIX 3.2.5 systems running PSSP 1.2 with authentication services based on Kerberos 4, it was necessary to use the name "kerberos4". It is not necessary for you to create an entry in the file for "kerberos4", because the default port of 750/udp will be used if no "kerberos4" entry is found. You should only have to make modifications if some other service requires the traditionally used (but not formally reserved) ports by Kerberos version 4. They Kerberos version 4 ports are:

Service name: kerberos4 Port: 750/udp

Service name: kerberos_admin Port: 751/tcp

Service name: krb_prop Port: 754/tcp

/etc/krb.conf: The **/etc/krb.conf** file contains the name of the local realm for your SP system and identifies the hostnames of all of the authentication servers for all Kerberos realms known to the local realm. For more information on the content of the file, refer to **krb.conf** in the *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference*.

If you configure your realm to use AFS authentication servers, this file is built for you automatically from the corresponding AFS configuration file **CellServDB**. When you use PSSP authentication servers, you create this file or one is generated by default by the **setup_authent** script that must run on each workstation after you install the PSSP files. You must provide the file if the workstation is a client workstation or a secondary authentication server. If it is the primary authentication

server, you need to provide the file only if the system's hostname contains no domain portion, following conventional network naming rules.

By convention, Kerberos uses the domain portion of a network interface name, converted to upper case, as the default realm to which the host belongs. If you want to define your own local realm name that does not follow convention, or if the host name of your primary server host has no domain portion, you must supply a file with your choice of local realm name in the first line. When using AFS authentication, the local AFS cell name, contained in the **ThisCell** file, converted to uppercase, becomes the local Kerberos realm name.

/etc/krb.realms: The **/etc/krb.realms** file is a file that maps network interface (host) names to realms. When Kerberos needs to determine what realm a network interface is assigned to, it looks in this file. If there is no entry for the hostname or for the domain part of the hostname, the default is a realm name equal to the domain portion of the hostname, converted to upper case. Any network interface names on systems using SP authenticated services which have a domain portion which is not the same as the local realm name, except for case, must have an entry in this file.

Entries for all network interfaces on the SP nodes and on the control workstation that require this mapping in **/etc/krb.realms** are added automatically by either **setup_authent** initially or by the **setup_server** script. The file is kept identical on the control workstation and the nodes by the node installation customization process. There may be cases involving other work stations, however, where you will have to add entries to the file on one or more systems.

When you set up a secondary authentication server or set up a system as an authentication client system, you are instructed to copy the configuration files from the primary server. If the new system requires mapping in **krb.realms**, the **setup_authent** script will create the entry in the local copy of the file. You will have to manually add the entry for the workstation you just configured into the control workstation's file and into the files on any other client and server workstations already installed.

Setting Up the Primary Authentication Server

The very first system that you set up, when you install your PSSP software, is your primary authentication server. That may be your control workstation; or it could be some other IBM RISC System/6000 workstation. You may want to have your servers entirely off of the SP system itself, to provide greater physical security for them. You may want to allow logins to the control workstation that are not appropriate for the authentication servers.

The **setup_authent** script is used to configure your authentication services. If you have installed the **ssp.authent** fileset on the workstation prior to running the script, it assumes that the system will be a PSSP authentication server. If you provide your own **krb.conf** file, the script will look for an entry for the local hostname. If it cannot find one, it allows you to configure the system without any server (as an authentication client system).

setup_authent creates the service principals for local service instances and creates the server key file for them. It updates the **krb.realms** file as needed, and creates the root user's **.klogin** file to authorize the initial administrator principal to use remote commands.

The daemons which provide the authentication server (the KDC), and the database administration server, are started by adding entries to **/etc/inittab**.

The procedure you must follow to set up the primary authentication server is as follows:

1. Install **ssp.clients** and **ssp.authent**.
2. Create a **/etc/krb.conf** file if you want or if your hostnames do not follow Kerberos convention.
3. Execute the **/usr/lpp/ssp/bin/setup_authent** program and follow the instructions in the prompts.

Setting Up Secondary Authentication Servers

The procedure you must follow to set up a secondary authentication server is as follows:

1. Install **ssp.clients** and **ssp.authent**.
2. Copy the **/etc/krb.conf** file from the primary authentication server to this secondary server system.
3. Add a line to the **/etc/krb.conf** file listing this system as a secondary server for the local realm.
4. Copy the **/etc/krb.realms** file from the primary server to the secondary server system.
5. Execute the **/usr/lpp/ssp/bin/setup_authent** program and follow the instructions when prompted.
6. If **setup_authent** created an entry for the local **/etc/krb.realms** file, copy the file to the other systems.
7. Add an entry for the new secondary server to the **/etc/krb.conf** file on the systems on which you had previously initialized authentication.
8. On the primary server, if this is the first secondary server, you should create a root **cron** entry that invokes the script **/usr/kerberos/etc/push-kprop** to periodically propagate database changes.

These steps are the same as those you would have followed to set up your SP control workstation as a secondary authentication server. For a more detailed description of how to do this, see *IBM RISC System/6000 Scalable POWERparallel Systems: Installation Guide*.

Setting Up Authentication Client Systems

The procedure you must follow to set up an authentication client system is as follows:

1. Install **ssp.clients**.
2. Copy the **/etc/krb.conf** file from the primary authentication server to this system.
3. Copy the **/etc/krb.realms** file from the primary server to this system.

   Note: If the new workstation is outside the realm of the primary server, the new workstation needs to be added to the primary server's **/etc/krb.realms** file first, before copying the **/etc/krb.realms** over to the new workstation.

Otherwise, the next step, which runs the **setup_authent** program, will fail on the new client workstation.

4. Execute the /**usr**/**lpp**/**ssp**/**bin**/**setup_authent** program and follow the instructions when prompted.
5. If **setup_authent** created an entry for the local /**etc**/**krb.realms** file, copy the file to the other systems.

These steps are the same as those you would have followed to set up your SP control workstation as an authentication client system. For a more detailed description of how to do this, see *IBM RISC System/6000 Scalable POWERparallel Systems: Installation Guide*.

Directory Pathnames for SP Authentication Services

The following are some of the directory paths that are used with authentication services.

/usr/lpp/ssp/bin
: Contains **setup_authent**

/usr/lpp/ssp/kerberos/bin
: Contains commands for users of authentication services and authentication database administrators.

/usr/kerberos/bin
: Contains a symbolic link for **/usr/lpp/ssp/kerberos/bin**

/usr/lpp/ssp/kerberos/etc
: Contains authentication daemons and commands used by root to maintain a local SP authentication database.

/usr/kerberos/etc
: Contains a symbolic link for **/usr/lpp/ssp/kerberos/etc**

/var/kerberos/database
: Contains the SP authentication database on the server

/var/adm/SPlogs/kerberos
: Contains error logs for the authentication servers

/etc Contains the files **krb.conf**, **krb.realms**, and **krb-srvtab**

/usr/lpp/ssp/rcmd/bin
: Contains the Kerberos-authenticated versions of **rsh** and **rcp**.

/usr/vice/etc Contains AFS cell information and utilities

/usr/afsws/etc Contains AFS executables

Managing the Authentication Database

Backing up the Authentication Database

An important task that the administrator of a PSSP authentication database should perform is to keep a current back-up of the database contents. The database utility commands that are available to the root user on the primary authentication database system include functions you can use for this purpose. If you have secondary authentication servers and follow the steps outlined for setting them up, you will automatically create a backup of the database every time the **cron** file entry runs that propogates the database to the secondary servers.

The **push-kprop** script, invoked daily or at some other interval from the root user's **cron** file, creates a backup file in ASCII format called **slavesave**. This file is created in the database directory, where it is accessible only to the root user. If you want to create a backup file without using **push-kprop** (for example, if you do not have secondary servers and want to avoid the mail messages **push-kprop** sends to root), just add a **cron** entry to invoke the following command instead:

```
kdb_util dump /var/kerberos/database/slavesave
```

The result will be a backup file with the same pathname that can be used with the **kdb_util load** command to recover from a lost or corrupted database, or viewed or printed to examine the database contents.

Changing the Kerberos Master Password

Changing the master password can enhance database security somewhat, but should not be substituted for adequate physical security. Changing the master password re-encrypts the database in the new key. It does not, however, change any of the Kerberos principals' passwords (keys) themselves. Therefore, it offers limited protection unless all (possibly compromised) passwords are changed also.

Changing the master key is one of the capabilities of the **kdb_util** command. See the command description in the reference manual. Note that the various **kdb_*** commands may only be issued by the root user, on a system configured as a PSSP authentication server. An idiosyncrasy of these commands is that they prompt only once for the new password (unlike normal password-changing protocol).

After you change the password, be sure to issue the **kstash** command to store the new key in the /**.k** file. You'll also have to kill and respawn the Kerberos daemons on the server (**kerberos** and **kadmind**). If you have secondary servers, you will have to force re-propagation of the database and respawning of the backup servers also.

The sequence that should be executed on the primary server is:

1. Login to Kerberos as the admin principal used to set up authentication. For example:

   ```
   /usr/kerberos/bin/kinit root.admin
   ```

2. Change the password (you will be prompted for old and new).

   ```
   /usr/kerberos/etc/kdb_util new_master_key /var/kerberos/database/newdb.$$
   /usr/kerberos/etc/kdb_util load /var/kerberos/database/newdb.$$
   ```

3. Replace the /**.k** file (another prompt for the new password).

   ```
   /usr/kerberos/etc/kstash
   ```

4. Kill and respawn the primary authentication server daemons.

   ```
   kill 'ps -e | egrep 'kerberos|kadmind' | cut -d'-' -f1'
   ```

5. If secondary servers exist, propagate the new database, copy the new master key cache to the secondary servers, and kill and respawn the secondary server daemons.

```
if [[-s /var/kerberos/database/slavelist]]
then
  /usr/kerberos/etc/push-kprop
 cat /var/kerberos/database/slavelist | while read slave
 do
    /usr/lpp/ssp/rcmd/bin/rcp /.k $slave:/
   /usr/lpp/ssp/rcmd/bin/rsh $slave \
        kill \ ps -e \| grep kerberos \| cut -d'-' -f1\
  done
fi
```

Managing Authentication Principals

Displaying Information about Principals

With PSSP Authentication: There are two ways for a PSSP authentication database administrator to examine information about principals. The **kadmin** command offers any administrator listed in the **admin_acl.get** file access from any network-connected system running SP authentication services. The root user on the primary authentication database system can also directly examine the database content by viewing the database backup file in /**var**/**kerberos**/**database**/**slavesave**.

Using **kadmin**, for example:

```
kadmin
Welcome to the Kerberos Administration Program, version 2
Type "help" if you need it.
admin: get ben
Admin password:
Info in Database for ben.:
   Max Life: 255   Exp Date: Fri Dec 31 23:59:00 1999

   Attribs:  00  key: 0 0
admin: quit
Cleaning up and exiting
```

Using the database backup file:

```
grep "^rcmd node3sw" /var/kerberos/database/slavesave
rcmd node3sw 255 1 2 0 a49bf286 d45c6560 200001010459 199503301502 root admin
```

The fields in each entry are:

- The principal's name
- The principal's instance
- The maximum ticket lifetime
- The key version of the master key used to encrypt the entry.
- The key version of the principal's secret key.
- The two halves of the encrypted private key.
- The expiration date (in Universal Coordinated Time) as YYYYMMDDHHMM
- The date the entry was created or modified
- The name and instance of the administrator who made the last update

With AFS Authentication: To display the database information for a user of AFS, an AFS administrator uses the **kas examine** command. For example:

```
kas ex -na ben -admin_user afsadm
Administrator's (afsadm) Password:

User data for ben
  key (0) cksum is 1004957312, last cpw: Mon May 22 10:12:50 1995
  password will never expire.
  An unlimited number of unsuccessful authentications is permitted.
  entry never expires.  Max ticket lifetime 25.00 hours.
  last mod on Mon May 22 10:12:50 1995 by afsadm
```

Adding Principals and Assigning Intitial Passwords

With PSSP Authentication: To add a principal and set the initial password in the authentication database, a database administrator with an **admin** principal that is listed in the **admin_acl.add** file, can use **kadmin** or **add_principal**.

To use **kadmin**:

1. Enter the **kadmin** command:

   ```
   kadmin
   ```

   A welcome message and explanation of how to ask for help are displayed.

2. Enter the **add_new_key** or **ank** subcommand:

   ank *name[.instance]*

   The only required argument for the subcommand is the principal's name.

3. At the prompt, enter your **admin** password
4. At the prompt, enter the principal's new password
5. At the prompt, reenter the principal's new password

For more information, refer to **kadmin** in the *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference*.

add_principal operates like **kadmin**, but allows an arbitrarily large number of principals to be added at one time. Whereas **kadmin** always prompts for your administrative password, **add_principal** will use tickets you have already obtained as *login-name*.**admin**. This makes **add_principal** suitable for use in scripts or background processes where a password cannot be provided interactively.

1. Create an ASCII file containing a list of principals and their initial passwords. Make sure it is readable only by you.
2. Enter the **add_principal** command.

   add_principal *filename*

3. Immediately remove the file containing the initial passwords.

The following script fragment shows one way to add a user to the database without interactively editing a file, and keeping the initial password on disk for a minimal amount of time:

```
#( OMASK=$(umask); umask 077; INITIALPW=changeme$RANDOM; \
   print "johndoe $INITIALPW">/tmp/Addp$$; \
   /usr/kerberos/bin/add_principal /tmp/Addp$$; \
   /bin/rm /tmp/Addp$$; umask $OMASK; \
   print "johndoe's initial password is $INITIALPW"; )
johndoe's initial password is changeme28745
#
```

On the system which is the primary authentication server, the root user can use the **kdb_edit** command to add principals. The steps are:

1. Enter the command, which operates interactively, prompting for requests.

   ```
   kdb_edit -n
   ```

2. Reply to the prompt for a principal name by entering the user name of the new principal and then the instance.

   ```
   Principal name: joeuser
   Instance:
   ```

3. Reply **y** to confirm that you want to add the pricipal.
4. Enter the initial password, when prompted.
5. Reenter the password, when prompted.
6. Accept the default expiration date by pressing Enter or type an alternate date in the format shown by the prompt.
7. Accept the default maximum ticket lifetime value or enter another.
8. Press enter when prompted for attributes.
9. The command will respond Edit OK and begin prompting for another principal to add. Repeat the same sequence or press enter to terminate the command.

For a complete example and more information, refer to **kdb_edit** in the *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference*.

With AFS Authentication: To add a principal and set a password in an AFS authentication database:

1. Enter the **kas** command:

   ```
   kas create -name user [-cell cellname]
   ```

2. At the prompt, enter your AFS password
3. At the prompt, enter the principal's new password
4. At the prompt, reenter the principal's new password to verify

Adding an Authentication Administrator

With PSSP Authentication: To perform the tasks involved in administering SP authentication services, the system administrator must be authorized to do so. The administrator must be a user with an **admin** instance whose name appears on one of the authentication administration access control lists (ACLs).

For example, to perform authentication administration, the administrator **gail** must have a principal **gail.admin** in the authentication database. In addition, **gail.admin** must be in the access control list file for the task she wants to perform. There are three ACL files:

/var/kerberos/database/admin_acl.add
: The list of principals authorized to add entries to the authentication database

/var/kerberos/database/admin_acl.get
: The list of principals authorized to retrieve entries from the authentication database

/var/kerberos/database/admin_acl.mod
The list of principals authorized to modify entries in the authentication database

Adding an authentication database administrator requires two steps:

1. Adding the principal with the **admin** instance, as described above.
2. Adding the principal identifier to one or more of the ACL files, which must be done by the root user on the primary database system.

With AFS Authentication: AFS administration does not use an **admin** instance or ACL files. You, an AFS administrator, can authorize another AFS user to perform administrative tasks by assigning it the **admin** attribute:

1. Follow the steps described above to add the user, if not already defined.
2. Enter the **kas** command:

   ```
   kas setfields -name user -flags admin [-cell cell]
   ```

3. At the prompt, enter your AFS password

Changing a Principal's Password and other Attributes

With PSSP Authentication: To change a password for a principal in the authentication database, a PSSP authentication database administrator can use either the **kpasswd** command or the **kadmin** program's **change_password** subcommand. These commands can be issued from any system running SP authentication services and do not require a prior **kinit**.

To use the **kpasswd** command:

1. Enter the **kpasswd** command with the name of the principal whose password is being changed:

   ```
   kpasswd -n name
   ```

2. At the prompt, enter the old password
3. At the prompt, enter the new password
4. At the prompt, reenter the new password

To use the **kadmin** program:

1. Enter the **kadmin** command:

   ```
   kadmin
   ```

   A welcome message and explanation of how to ask for help are displayed.

2. Enter the **change_password** or **cpw** subcommand with the name of the principal whose password is being changed:

   ```
   cpw name
   ```

   The only required argument for the subcommand is the principal's name.

3. At the prompt, enter your **admin** password
4. At the prompt, enter the principal's new password
5. At the prompt, reenter the principal's new password

For more information, refer to **kpasswd** and **kadmin** in the *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference*.

To change your own **admin** instance password, you can use either the **kpasswd** command or the **kadmin** program's **change_admin_password** subcommand.

To use the **kpasswd** command:

1. Enter the **kpasswd** command with your **admin** instance name:

   ```
   kpasswd -n name.admin
   ```

2. At the prompt, enter your old **admin** password
3. At the prompt, enter your new **admin** password
4. At the prompt, reenter your new **admin** password

To use the **kadmin** program:

1. Enter the **kadmin** command:

   ```
   kadmin
   ```

   A welcome message and explanation of how to ask for help are displayed.

2. Enter the **change_admin_password** or **cap** subcommand:

   ```
   cap
   ```

3. At the prompt, enter your old **admin** password
4. At the prompt, enter your new **admin** password
5. At the prompt, reenter your new **admin** password

For more information, refer to **kpasswd** and **kadmin** in the *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference*.

In addition to changing the password, you may want to change either the expiration date of the principal or its maximum ticket lifetime, though these are not so likely to be necessary. To do so, the root user on the primary authentication database system must use the **kdb_edit** command, just as when adding new principals locally. Instead of not finding the specified principal, the command finds it already exists, and prompts for changes to all its attributes, starting with the password, followed by the expiration date and maximum ticket lifetime.

The following example shows the use of the **kdb_edit** command to change a user principal's maximum ticket lifetime from the default value (30 days) to approximately 7 days:

```
#/usr/kerberos/etc/kdb_edit -n
Opening database...
Previous or default values are in [brackets] ,
enter return to leave the same, or new value.

Principal name: johndoe
Instance:

Principal: johndoe, Instance: , kdc_key_ver: 1
Change password [n] ?
Expiration date (enter yyyy-mm-dd) [ 1996-09-01 ] ?
Max ticket lifetime [ 255 ] ? 171
Attributes [ 0 ] ?
Edit O.K.
Principal name:
#
```

With AFS Authentication: To change the attributes of AFS user principals, the administrator uses the **kas setfields** command. For example, to change the maximum ticket lifetime to two days:

```
kas setfields -name ben -lifetime 48:00 -admin_user afsadm
administrator's (afsadm) Password:
```

Deleting Principals

With PSSP Authentication: Kerberos does not provide any direct way for an administrator to delete a principal from the database. It can be accomplished by the root user on the primary authentication server system, using an editor and the database backup file.

The required steps are:

1. Use a current backup file or create one using **kdb_util dump**
2. Remove the lines for any unwanted principals or principals whose entries need to be re-created. Never use this method to change service principal names. Edit the file or use:

   ```
   grep -v "^rcmd hostname"
   /var/kerberos/database/slavesave > /var/kerberos/database/newdatabase
   ```

3. Re-load the database from the backup file using **kdb_util load**:

   ```
   kdb_util load /var/kerberos/database/slavesave
   ```

With AFS Authentication: An AFS administrator can delete principals directly by using the **kas delete** command. For example:

```
kas delete -name ben -admin_user afsadm
Administrator's (afsadm) Password:
```

Managing Service Keys

Management of principals centers primarily on changing passwords in accordance with local security policy. For user principals this task is straightforward and can be done using simple tools. For service principals, password maintenance requires more than periodic changes to the authentication database. It also requires creation and distribution of files containing encrypted password information to the application server hosts.

Creation of Service Key Files

During setup for a node or control workstation, the keys for service principals are stored in the authentication database (for use by the authentication server) and on the node corresponding to the instance in the file **/etc/krb-srvtab** (for use by the service itself). Therefore, every node in the SP System contains a file, **/etc/krb-srvtab**, that contains the keys for the services that are provided on that node. On the control workstation, the **hardmon** and **rcmd** service principals are in the file. On the nodes, the **rcmd** service principals are in the file.

The local server key files on the control workstation and on other workstations are created by the **setup_authent** command when authentication is first set up on the system.

With PSSP Authentication: The **setup_server** script creates server key files for the SP nodes. When the control workstation is a PSSP authentication server, this occurs for each node when the script is run on the node's boot server.

setup_server creates the files in the **tftpboot** directory, readable only by their owner **nobody**. When network boot occurs during install or customization, the file is transferred to the node using **tftp**. The node uses the file to obtain tickets and invokes authenticated **rsh** to execute a script on the boot server that deletes the file.

With AFS Authentication or No Server on the Control Workstation: The **setup_server** script creates server key files for the SP node, when the script is run on the control workstation for the first time. **setup_server** creates the files in the **tftpboot** directory, readable only by their owner **root**. The file is copied to the node's boot server and ownership is changed to **nobody**, when the script runs on the boot server. When network boot occurs during install or customization, the file is transferred to the node using **tftp**. The node uses the file to obtain tickets and invokes authenticated **rsh** to execute a script on the boot server that deletes the file (gives it back to root, if the boot server is the control workstation). In this case the copy on the control workstation is never deleted.

Examining Service Key Information

There are two programs that you can use to examine the content of a server key file on the local system, **ksrvutil** and **klist**: To view the local system's server key file in /**etc/krb-srvtab**, enter **ksrvutil list**. The output will look like:

```
Version    Principal
   1     rcmd.cwktr@XYZ.ABC.COM
   1     hardmon.cwktr@XYZ.ABC.COM
   1     rcmd.cwkfddi@XYZ.ABC.COM
   1     hardmon.cwkfddi@XYZ.ABC.COM
   1     rcmd.cwken0@XYZ.ABC.COM
   1     hardmon.cwken0@XYZ.ABC.COM
```

The same file, viewed by entering **klist -srvtab** appears as:

```
Server key file:   /etc/krb-srvtab
Service         Instance        Realm      Key Version
-----------------------------------------------------
rcmd            cwktr           XYZ.ABC.COM 1
hardmon         cwktr           XYZ.ABC.COM 1
rcmd            cwkfddi         XYZ.ABC.COM 1
hardmon         cwkfddi         XYZ.ABC.COM 1
rcmd            cwken0          XYZ.ABC.COM 1
hardmon         cwken0          XYZ.ABC.COM 1
```

To examine the server key file in /**tftpboot** directory for a node whose hostname is **sp2node8**, you would enter **ksrvutil list -f /tftpboot/sp2node8-new-srvtab** and the output might look like:

```
   1     rcmd.n08tr0@XYZ.ABC.COM
   1     rcmd.n08tr1@XYZ.ABC.COM
   1     rcmd.n08fddi@XYZ.ABC.COM
   1     rcmd.n08css0@XYZ.ABC.COM
   1     rcmd.n08en0@XYZ.ABC.COM
```

Changing Service Keys

Changing the service keys involves two responsibilities:

- Changing the service keys in accordance with local system policy
- Ensuring that the service keys contained in the authentication database and in the server key files on all workstations and SP nodes always match.

As security administrator, you decide how frequently service keys need to be changed on your system. You may decide to set the service keys at setup time and never change them or you may decide to change them every six months. You should certainly change the service keys if you have reason to believe system security has been compromised.

Use the **ksrvutil** command to change service keys. The **ksrvutil** command changes all of the service keys in the local service key file. The keys for all of the services provided on a single node, or the control workstation, are changed in a single call to **ksrvutil**. The keys are set to randomly generated values. The method of running **ksrvutil** to change the service keys for a node varies according to the authentication configuration of the system:

Primary Authentication Server on the Control Workstation: The **ksrvutil** command should be run on each node where the keys need to be changed. To change all of the service keys for a node to randomly generated values, enter:

```
ksrvutil change
```

The keys are updated in the node's /**etc/krb-srvtab** file and in the authentication database on the control workstation. To issue the command, you must be root. The **dsh** command can be used to issue the **ksrvutil** command on multiple nodes.

The **ksrvutil** command should also be used to change service keys on the control workstation, regardless of the location of the primary authentication database.

All Other Authentication Configurations: If the primary authentication database does not reside on the SP system's control workstation, the **ksrvutil** command should *never* be used on the nodes of the SP System. If the primary authentication database for the local realm resides in some location other than the SP system's control workstation (even if a secondary authentication database is on the control workstation) or if the system is using AFS-based authentication services, the system maintains a copy of each node's /**etc/krb-srvtab** file on the control workstation. These files are in the directory /**tftpboot** with a filename of *nodename***-new-srvtab**. Changes to the **srvtab** file should be made only to the copy on the control workstation in /**tftpboot**. The **ksrvutil** command is used to change the service keys.

To change all of the service keys for a node:

1. Ensure that no users are using authenticated services on the target node
2. On the control workstation, issue:

   ```
   ksrvutil -f nodename-new-srvtab change
   ```

3. Copy the *nodename***-new-srvtab** file from the control workstation to the /**etc/krb-srvtab** file on the target node. To do this, you can use **ftp** or, as an alternative, you can customize the node using the **spbootins** command, followed by a **netboot** on the node.

To change the service keys for a node, you must have read/write permission to the /**tftpboot**/*nodename***-new-srvtab** file and have write access to the /**etc/krb-srvtab** file on the node.

When you change the service keys for a node, or for the control workstation, any service tickets that have been previously issued to users for those service instances are no longer valid. An attempt to use such a ticket to authenticate the user's identity to the server will fail. A message will be issued that the server could not decode the ticket authenticator. In such a case, the user must issue a **kinit** command to obtain a new ticket-granting ticket and then reissue the service request.

Refer to **ksrvutil** in the *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference* for more information. Also, refer to the *IBM RISC System/6000 Scalable POWERparallel Systems: Diagnosis and Messages Guide* for information on recreating server key files.

Distributing Service Key Files to Client Systems

Normal distribution of server key files to SP nodes occurs as described using **tftp**. This entails some risk, but it is reduced by not making the files world-readable, as is usually the case for files transferred in that way. Probably the only secure method is to log into the destination system, use **kinit** to get a ticket, and then invoke Kerberos-authenticated **rcp** to pull the file from the boot server. That program cannot be used to push the file in the other direction because the remote command daemon on the target system needs a valid server key file to perform authentication.

Another means of copying the file that offers some security is the use of the standard AIX **rcp** command, requiring the creation, temporarily, of a root **.rhosts** file.

Less secure yet, and probably unacceptably so, are **ftp**. Unless it is done over a secure (internal) network interface.

Using AFS Authentication Services

AFS is a distributed file system developed by Transarc Corporation. AFS includes an authentication implementation that is based on Kerberos version 4. Therefore, AFS authentication servers can be used in place of SP authentication servers to provide credentials for SP principals.

Although AFS and SP authentication services can work together, there are differences in the way the authentication database is administered (for example, adding principals and changing passwords).

AFS uses a different set of servers, protocols, interfaces, and commands for Kerberos database administration.

Administering AFS Authentication Services

The primary tasks the system administrator performs in administering AFS authentication services are:

- Adding principals (users and services) and setting passwords in the authentication database
- Changing passwords for principals
- Managing the authentication keys for the set of services that use authentication (refer to “Managing Service Keys” on page 44 for more information)

The AFS system has its own administration commands and utilities for the administrator user. These commands and utilities provide for authentication, protection of files, file manipulation, and backup services. The following commands and utilities are relevant for administering authentication services. Refer to the AFS documentation for more information.

afsd Daemon used to connect AFS clients to AFS server

cacheinfo Specifies the directory to place the AFS directory on an AFS client

CellServDB Specifies the name of AFS cell and location of AFS server

kas Administrator interface to work with AFS authentication

klog.krb User interface to obtain a Kerberos ticket in addition to AFS tokens

pts Administrator interface to work with AFS protection services

token.krb Allows the user to list the current tokens and Kerberos tickets

The following SP authenticaton services commands work in an AFS authentication system:

- **kinit** (The SP_NAME environment variable must be set in order to use this command on a client workstation, when using AFS authentication)
- **klist**
- **kdestroy**
- **ksrvutil -afs change**
- **ksrvtgt**
- **rcmdtgt**
- **setup_authent**

The SP authentication services provide only the interfaces from the SP System to work in an AFS authentication system. The SP authentication services do not supply any of the AFS executables.

The AFS system provides multiple daemons that are activated on AFS server machines and support authentication tasks:

bosserver Basic OverSeer server monitors the AFS server processes on the AFS server machine

kaserver The AFS authentication server

ptserver Used to create system entries in the authentication database

buserver Maintains a backup authentication database on the authentication server

The authentication services work primarily with the **kaserver** and the **ptserver**. The **kaserver** allows the administrator to execute various commands to administer the authentication database. The **admin** user can enter **kas help** to view the commands available. The **admin** user needs to provide a password when using the **kas** command.

The **ptserver** protects and maintains system entries that are stored in the authentication database. The **admin** user can enter **pts help** to view the commands available. The **admin** user needs to provide a password when using the **pts** command. The admin user needs to create an AFS token (**klog.krb**) when working with the **pts** command.

Refer to AFS documentation for information on installing and administering the AFS system.

Chapter 4. Sysctl

Sysctl is an authenticated client/server system for running commands remotely and in parallel. It provides:

- **Least privilege capability**

  Root authority can be dynamically provided to non-root users based on their authenticated identity, the task they are trying to perform, access control lists, and any other relevant criteria. The root password need not be given out to as many people, and thus is kept secure.

- **Distributed execution**

  Sysctl applications can be executed on remote hosts with full authentication and authorization. It is essentially a secure, easy-to-program remote command execution mechanism for AIX commands, scripts and programs.

- **Parallel execution**

  Sysctl applications can be efficiently executed in parallel on many hosts.

- **Programmability**

  Sysctl applications can be coded as scripts for ease of implementation.

For example, consider a systems management script that contains commands that are only executable by root. If implemented as a Sysctl application, its security, usability, and performance are enhanced. Least privilege capability lets one of a select group of administrators run the script on each node in the SP without having access to the root password on any node in the system. With distributed execution, the administrator does not have to log in to each node to run the script. With parallel execution, the script can be run on all the nodes simultaneously with one command invocation. With programmability, the task can be automated or simplified with scripting.

Sysctl Components

Sysctl contains the following components:

- A server daemon (**sysctld**) that runs on all nodes and the control workstation.

  This server daemon has root privileges. It executes commands as root on behalf of authorized clients.

- Commands built into the server.

  These do the actual system administration work. Each command is paired with an authorization-checking script (callback) that has access to authenticated information about the client attempting to run the command. These commands include IBM-provided commands and typically also include installation-written ones.

- Configuration files that control aspects of the server operation as well as extend the command set available on a given node.

- Access control lists that list authorized users.

- A client program (**sysctl**) that offers command-line and interactive interfaces for communicating with **sysctld** servers.

© Copyright IBM Corp. 1995

Figure 3 on page 52 illustrates the way in which the Sysctl elements work together.

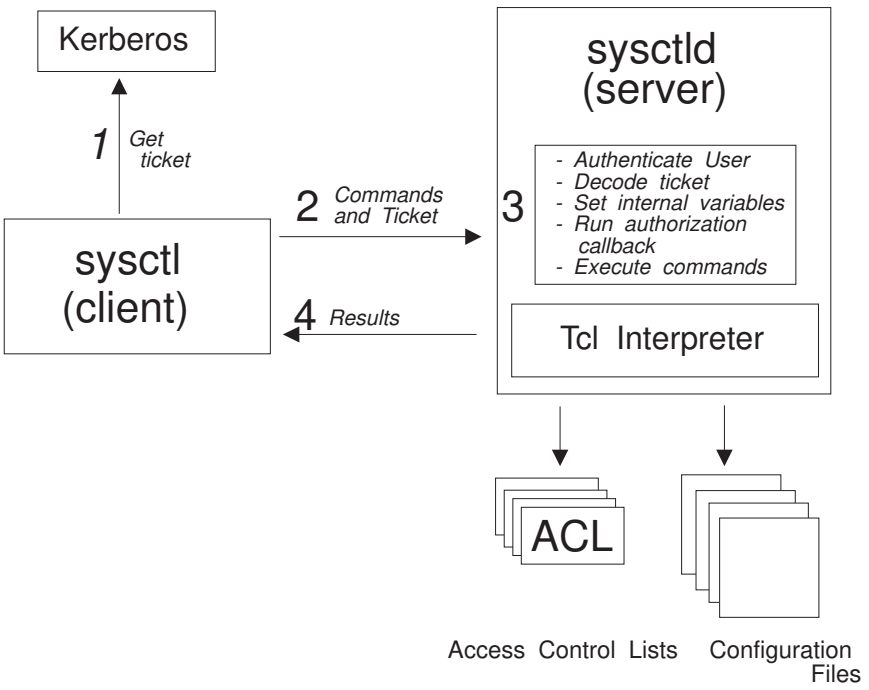


Figure 3. Sysctl Elements

The interpretation of a Sysctl command starts when the user issues a Sysctl client command. This can be embedded in a script or issued directly from the command line. It is of the following form:

```
sysctl target_nodes_specification command args
```

Note that more than one node can be targeted. The command is the name of a Sysctl server command to be run as root on each of the targeted servers, if the client user is authorized.

1. The Sysctl client code obtains authentication information about the command issuer from SP Authentication Services.
2. The Sysctl client code sends the authentication information, along with the command string, to each of the specified Sysctl servers in parallel.
3. Each server then performs the following steps:
   a. The authentication information is decoded and the authenticated information about the client is obtained.
   b. Internal variables, such as the username and host of the client are set. These variables are available to the server routines that will check the authorization of the client (the authorization callbacks) as well as the Sysctl commands to be run.
   c. The authorization is checked via an authorization callback. This can be a supplied callback, or an installation-written one. This callback may check an ACL. If the client passes the authentication check, the next step is done.
   d. The server commands requested are executed as root, on behalf of the client.

4. **stdout** and **stderr** are sent back to the client. Output from each server is displayed with labeling indicating its source.

Relationship to the SP Authentication Services

Sysctl uses the SP Authentication Services. Client identity is authenticated across the network to the server using these services. Users of Sysctl typically need to identify themselves to the SP Authentication Services via the **kinit** command.

See Chapter 3, “Security Features of the SP System” on page 17 for more information on using Authentication Services.

Overview

Architecture

A Sysctl server provides remote access to a set of commands for authenticated and authorized clients. Since the Sysctl server is running with UID 0, these commands are executed with root privilege. These commands are embedded in an interpreter within the server. Sysctl uses the embeddable Tool Command Language (Tcl) as the foundation for its built-in command interpreter. The commands are of the following types:

- Base Tcl commands.

  These are the commands making up the Tcl language. They include control commands (such as **if**), input/output commands (such as **puts**), arithmetic commands (such as **expr**) and commands allowing the execution of other commands, including external AIX commands and scripts (**exec**).

- Built-in Sysctl commands.

  These are provided for you. They include commands to configure and manage Sysctl, as well as some systems management applications. See “Built-in Sysctl Commands” on page 72.

- Your applications.

The Tcl interpreter that Sysctl uses is built from base Tcl and Extended Tcl. Refer to the **ssp.public.README** file for more information on Tcl as supplied with the SP.

Using an embedded language, particularly Tcl, provides the following advantages:

- The command language interface is ideal for administrative tasks.
- The Sysctl command set is extended by writing Tcl scripts (instead of C code). These new Tcl scripts are embedded in the server's command set.
- Users can write simple Tcl wrappers for Perl and shell scripts and then add the scripts to the server command set.
- The Sysctl authentication mechanism gives the administrator complete control over the set of commands available to a given set of users. (see Chapter 3, “Security Features of the SP System” on page 17).

Security

The management tools used to administer a large site need to be secure to prevent outside intrusion and to prevent accidental disasters, such as inadvertently deleting critical data.

Sysctl security has the following components:

- Client authentication
- Authorization callbacks
- Access Control Lists

Client Authentication

The SP Authentication Service provides trusted third-party authentication of users. When a client issues a Sysctl command, a ticket identifying the client is sent along with the command. The server uses the ticket to verify the identity of the client.

Authorization Callbacks

The Sysctl authorization mechanism is based on callbacks. A callback is a Tcl routine paired with a Sysctl command that determines whether the client is authorized to run the command. The server invokes the authorization callback for a command just before executing the command. If the authorization callback returns a Tcl error, the command is not executed. When the callback returns a normal Tcl result, the command is executed.

With the callback mechanism, you can customize the server's authorization model to fit your needs. You can use the built-in authorization callbacks or write new ones, as needed.

Built-In Authorization Callbacks: These authorization callbacks are provided:

NONE No special authorization is required to run a command. Even clients with unauthenticated identity (clients not identified to the SP Authentication Services) can run commands with this callback.

AUTH Any client who has been successfully authenticated by the SP Authentication Service can run commands with this callback.

ACL This callback checks an ACL file for an authenticated client. If the client is listed in the ACL, the callback succeeds. The pathname of an ACL file can be specified as an argument. If no pathname is specified, the client's principal must be listed in the global Sysctl ACL file, /**etc**/**sysctl.acl**.

SYSTEM This level is used for commands that will only be run when authorization callbacks are bypassed (see below). This authorization callback always returns a Tcl error. A client cannot issue commands with this callback from the command line or the Sysctl prompt.

Bypassing Authorization Callbacks: In some situations, the server bypasses the authorization callbacks. Authorization callbacks are bypassed while the server:

- Reads configuration files (that is, when starting up)
- Executes an authorization callback
- Executes the body of a procedure that a user is authorized to run

Access Control Lists (ACLs)

An ACL is a text file that can be used to grant authority to issue particular commands to particular users. Sysctl provides ACL-based authorization. See "Sysctl Files" for more information.

Sysctl Executables

Sysctl Daemon (sysctld)

The Sysctl server daemon, **sysctld**, processes all Sysctl client requests for the node on which it runs. There is a **sysctld** daemon running on each node of the SP system, as well as on the control workstation. **sysctld** runs as **root** and executes Tcl procedures for authenticated and authorized clients who may be local or remote users. See the **sysctld** man page for more information.

Client (sysctl)

The Sysctl client shell program, **sysctl**, offers a command line interface for communicating with **sysctld**. Together, **sysctl** and **sysctld** provide monitoring and execution abilities needed to remotely manage SP nodes. **sysctl** connects to a remote node's **sysctld** using TCP/IP, passes keywords and commands to the server, and writes any output returned to **stdout** and **stderr**. See the **sysctl** man page for more information.

Sysctl Files

Sysctl Configuration File

A configuration file is read in and interpreted each time the Sysctl server is started or restarted. When Sysctl is installed, each SP node will have a provided configuration file. The default pathname for this file is /**etc**/**sysctl.conf**. Alternate configuration files can be used. See the **sysctld** man page for more information.

The configuration file modifies the state of the server by executing one or more configuration commands. The configuration commands create read-only variables, register procedures, include other configuration files, and create classes of commands.

See "Installation and Configuration" on page 56 or the **sysctl.conf** man page for more information on configuring your system for Sysctl.

Access Control Lists (ACLs)

An ACL is a text file that lists authorized clients. (These files should be inaccessible to change by other than trusted administrators.)

There are two type of ACLs:

- The global Sysctl ACL file /**etc**/**sysctl.acl**

  This identifies a server's **trusted** users. Trusted users are authorized to run any Sysctl or Tcl command, except for those with the SYSTEM authorization level. Trusted users have privileges equal to those of a root user. This ACL identifies those clients with the ACL authorization level.

  The supplied /**etc**/**sysctl.acl** file contains commented lines that are samples. Copy the sample lines and insert your own principal identifiers.

- Installation-created ACLs

You can create ACLs for specific reasons. For example, you may designate a set of users that can issue a particular Sysctl command with an ACL for that command. The authorization callback for the command could check if the client was in the ACL.

Each line in an ACL contains either a Kerberos name (principal identifier) or the name of another ACL file. The principals in the included ACL file are added to those listed explicitly in the ACL file in determining the set of authorized clients. For example, an ACL file stored in one system can identify a core set of system administrators, followed by the name of another ACL file that is local to a node. Authorization for each SP node can be customized, while allowing the core administrators access to every machine.

The following is a sample ACL (note that the first line in an ACL file always begins with the magic cookie **#acl#**):

```
#acl#
_PRINCIPAL sarah.admin@TESTCELL.IBM.COM
_PRINCIPAL andrew@TESTCELL.IBM.COM
_ACL_FILE /usr/admin/acls/global_sysctl.acl
```

This ACL identifies the Kerberos names of two trusted users, **sarah.admin** and **andrew**. Both of these users are trusted only if their tickets come from realm **TESTCELL.IBM.COM**. The ACL also names another file, **/usr/admin/acls/global_sysctl.acl**, that contains additional ACL entries.

Note: You may want to use file collections to maintain configuration and ACL files. See Chapter 14, “Managing File Collections” on page 213.

Logfile

/var/adm/SPlogs/sysctl/sysctld.log is a logging file to which the Sysctl daemon (**sysctld**) writes. It logs the invocation of each Sysctl command. This is helpful for debugging. See the **sysctld** man page for information on specifying logfile pathnames.

Installation and Configuration

Installation

Sysctl is contained in the installation option file **ssp.sysctl**. This is installed when you install the SP system management software. See the *IBM RISC System/6000 Scalable POWERparallel Systems: Installation Guide* for more information.

Configuring the System

Listed below is the basic procedure for configuring the system for Sysctl. Most of the steps are performed for you, either by the installation process or when you run **setup_authent** (both processes are described in *IBM RISC System/6000 Scalable POWERparallel Systems: Installation Guide*).

Note: Some of these steps require administrative access to the authentication database. See Chapter 3, “Security Features of the SP System” on page 17 for details.

Configuration Steps Done for You

When you install **ssp.sysctl** the following steps happen automatically:

- The **/etc/sysctl.conf** file, which contains configuration information for the **sysctld** daemon, is installed on each node.
- The following, which describes the IP port and protocol used for that port, is added to **/etc/services**:

  ```
  sysctl    6680/tcp   sysctld
  ```

- **/etc/inittab** is updated to automatically start the **sysctld** daemon. **init**, as root, starts **sysctld** as follows:

  ```
  sysctld:2:once:/usr/bin/startsrc -s sysctld
  ```

Configuration Steps You Need to Perform

Here are the configuration steps that you need to perform:

- Add any trusted users to the global Sysctl ACL (**/etc/sysctl.acl**).

  When the server is operational, you can manipulate Sysctl ACLs through the Sysctl built-in commands described in Table 4 on page 72.

- Define in the **/etc/sysctl.conf** file any new commands you have created and distribute it to each node either manually or via file collections.
- Optionally, create any new ACLs and distribute them either manually or via file collections to each node.
- If you have modified the configuration file, the **sysctld** daemons must be restarted. See "Developing a Sysctl Application" on page 64 for more information.

Debugging Configuration Problems

At this point, your system should be configured for Sysctl. After running **kinit**, you should be able to do Sysctl tasks, such as:

```
sysctl whoami
```

where, if your principal name is **andrew@kingston.ibm.com**, the system response looks like this:

```
andrew@kingston.ibm.com
```

If the authentication principals and server key files are set up properly, the server echoes your authenticated identity. If it returns with **unknown** or another error message, some portion of your configuration is incorrect. See the *IBM RISC System/6000 Scalable POWERparallel Systems: Diagnosis and Messages Guide*.

If a command does not work, run **sysctld** with debugging and security auditing turned on. First, you need to kill the local server and prevent its automatic respawning (such as with **/usr/bin stopsrc -s sysctld**). Then you can start the daemon with:

```
sysctld -sd -l /var/adm/yourdebuglogfilename
```

The **-s** option writes a security audit trace to the log file as the server executes each user request. The **-d** option writes debugging information to the server log

file. It is useful to run **tail -f** (to view the debugging information as it's written to the log file) on the log file.

Once you have issued **kinit** and issued any Sysctl commands, you can issue **klist** to display your current tickets. The **klist** command displays the principal name used with **kinit** and displays the list of service tickets currently active (with an expiration date) for the user.

After issuing **klist**, do the following:

1. Check the principal name. If this is incorrect, you need to issue **kinit** and enter the correct principal name.
2. Check the list of service tickets. The list of service tickets should contain one ticket for each **rcmd** service for which a Sysctl command was issued.
   - If the **rcmd** tickets do not appear, there is a problem with the principal definition set up in the authentication database.
   - If the **rcmd** service tickets appear, but issuing **sysctl whoami** returns `unknown`, it is likely that the SP Authentication Service is improperly configured. See the *IBM RISC System/6000 Scalable POWERparallel Systems: Diagnosis and Messages Guide* for more information.

Sysctl Configuration Commands

Configuration commands create read-only variables, register procedures, include other configuration files, and create classes of procedures. Configuration commands are interpreted when a server is started or restarted, and are contained in configuration files. The configuration commands are:

- **include**
- **create var**
- **create proc**
- **create class**
- **set**
- **load**
- Any other valid Sysctl command

include

The **include** command specifies additional configuration files for the server to read. This makes it easier to manage many external command definitions by letting them be split among a hierarchical set of files. For example:

```
include $buildTop/samples/sysctl/pdfpfck.cmds
```

causes the Sysctl server to read the contents of the specified file at initialization time.

The syntax of **include** is:

```
include filename
```

create var

The **create var** command defines a read-only variable in the Sysctl server. Variables defined in the configuration file using the standard Tcl **set** command exist only while the configuration files are being read.

The syntax of **create var** is:

```
create var variable value [authcallback]
```

where:

variable A read-only variable to which you assign a value.

value The value of the variable

authcallback
An authorization callback for the command. This can be **NONE, AUTH, ACL** or **SYSTEM** (to assign it one of the built-in authorization levels), or the name of another previously-registered procedure.

The following example defines a variable (**STARTTIME**) that contains the time you started up the server:

```
create var STARTTIME [exec /bin/date] NONE
```

create proc

The **create proc** command defines external Sysctl procedures.

The syntax of **create proc** is:

```
create proc name args authcallback body
```

where:

name The name of the Sysctl procedure you are creating.

args A list of argument names to the procedure.

authcallback
An authorization specifier for the command. This can be **NONE, AUTH, ACL** or **SYSTEM** (to assign it one of the built-in authorization levels), or the name of a previously-registered procedure.

body A Tcl expression that forms the body of the new procedure.

See “Developing a Sysctl Application” on page 64 for an example of **create proc**.

create class

The **create class** command defines classes of commands within the server. Command classes provide a way to organize commands into logical groups for clarity.

The syntax of **create class** is:

```
create class name file [authcallback]
```

where

name The name that you assign to the command class.

This name constitutes a **tag** that is prefixed to all commands defined in the command file.

For example, if a class named **test** is created and the class file defines a procedure named **help**, the procedure is created in the interpreters as **test:help**.

file The pathname of the file containing the definition of the class objects.

authcallback
An authorization specifier for the command. This can be **NONE, AUTH, ACL** or **SYSTEM** (to assign it one of the built-in authorization levels), or the name of a previously-registered procedure.

set

The **set** statement sets the value of the server variables ACL, LOG or KEY, or sets values in the Tcl env() array. The default values for the Sysctl server's ACL file, log file and Kerberos key file can be overriden by assigning values to the ACL, LOG and KEY variables in the configuration file. For example, the following line overrides the default value for the log file name:

```
set LOG /var/sysctld.logfile
```

The values assigned to the ACL, LOG and KEY variables are overriden by the optional command line arguments **-a**, **.-l** and **-k** on the **sysctld** command line. Environment variables (such as the default PATH) can also be set within the configuration file by assigning values to the env() array. For example:

```
set env(PATH) /usr/bin:/bin:/usr/etc:/etc
```

sets the PATH environment variable for the Sysctl server. The env() array is assigned an authorization callback of SYSTEM which prevents its modification from outside the Sysctl server by a request sent by a Sysctl client.

load

The **load** command dynamically loads the shared library at **lib_path** into memory. If the **init_proc** parameter is given, it is used as the library's initialization procedure. Otherwise the name of the initialization function is derived from the library name as follows:

```
library name    init function
============    =============
libxxx.sl       xxx_Init()
libxxx.a        xxx_Init()
```

The Sysctl server exports an API that the library uses to define commands, variables, authorization callbacks and interpreter deletion callbacks. See the **load** help page for details.

Other Sysctl Commands

You can include other commands in the Sysctl configuration file. These commands are executed by the server when it is started (or restarted).

Authorization Tasks

Authorization Callbacks

Sysctl provides an extremely flexible authentication mechanism to allow you to restrict the use of registered Sysctl procedures to any easily defined subset of users. Authorization callbacks are the means by which this authentication is implemented. As previously discussed, Sysctl includes four supplied authorization callbacks which provide the following authorization levels:

- NONE
- AUTH
- ACL
- SYSTEM

Adding New Authorization Callbacks

As an administrator you can develop a procedure with an authorization level that does not fit one of the supplied authorization callbacks. To do this, write your own authorization callback. Here is an example of a custom authorization callback:

```
create proc scdate {} SYSTEM {exec /bin/date}


create proc RCRAUTH {cmdName} SYSTEM {
    global SCUSER
    global SCPRINCIPAL
    if {$SCUSER != "rcr"} {
        svclog "Denying \"$cmdName\" access for user $SCPRINCIPAL"
        error "Authorization denied!"
    } else {
        svclog "Authorizing \"$cmdName\" access for user $SCPRINCIPAL"
        return "Authorization OK!"
    }
}

create proc rcrdate {} {RCRAUTH rcrdate} {scdate}
```

This example contains the definition of three procedures:

- **scdate** is a procedure that the SYSTEM authorization callback is attached to. The SYSTEM callback prevents **scdate** from being directly executed from outside of the server (that is, a user cannot successfully issue **scdate** from the Sysctl client). The **scdate** procedure can, however, be successfully executed from within another procedure which has been previously registered with the Server (that is, included in the Sysctl configuration file), since authorization callbacks are bypassed during this execution.
- **RCRAUTH** is a procedure that functions as an authorization callback. It accepts a command name as its single argument (used for logging purposes only). Its function is to return a normal result if the basename of the user is **rcr** and to return an error result in all other cases.
- **rcrdate** is a procedure that invokes the **scdate** procedure. Before control is given to the **scdate** procedure, the RCRAUTH authorization callback is invoked. If the base name of the user is **rcr**, then the callback returns a normal result and the **scdate** command is allowed to be invoked. If the callback returns an error result, then the **rcrdate** procedure is terminated.

ACL Files in Authorization Callbacks

The supplied **ACL** authorization callback is used to check for the remote user's principal name in an ACL file. If this callback is used without any parameters then the ACL file named by the ACL variable (normally /**etc**/**sysctl.acl**) is checked. If a file name is supplied as a variable to the ACL callback, then that file is checked for the user's principal name.

This allows an authorization callback to reference an alternate ACL file in performing the authorization check:

```
create proc myproc {} {ACL /etc/myproc.acl} {
        ....
        ....
        ....
}
```

The above example defines the procedure **myproc**, which takes no parameters, and has an authorization callback of **ACL** /**etc**/**myproc.acl**. This callback returns a normal result if the remote user's principal name appears in the file /**etc**/**myproc.acl** and an error return if it does not.

ACL files can be created and deleted and their contents can be manipulated and checked by using the Sysctl commands **aclcreate**, **acldestroy**, **acladd**, **acldelete**, **acllist**, **aclcheck** and **aclrecreate** (see the Sysctl help information for these commands).

Using Sysctl

Prerequisite Checklist

Before using Sysctl, the following is required:

- You have a principal in the local Kerberos realm and have run the **kinit** command to authenticate your identity to the system.
- The configuration is complete (see “Configuring the System” on page 56).
- The server daemon (**sysctld**) is running on all nodes.

Using Sysctl in Interactive Mode

Sysctl has an interactive mode. Note that only one node can be communicated with at a time in interactive mode.

To enter interactive mode, enter the **sysctl** client command without a command argument. For example, to use Sysctl in interactive mode on the local node, enter:

```
sysctl
```

If you issued the command from **diane.kgn.ibm.com**, you get the following prompt:

```
Sysctl (Version 1.1) on diane.kgn.ibm.com
sysctl>
```

At this point, you can enter Sysctl commands.

You can also specify some of the Sysctl flags when going into interactive mode. For example, to use Sysctl in interactive mode for node **redhook.kgn.ibm.com**, enter:

```
sysctl -h redhook.kgn.ibm.com
```

Sysctl issues this response:

```
Sysctl (Version 1.1) on redhook.kgn.ibm.com
sysctl>
```

In this example, if you enter **svcversion** and **pdf**, the interaction is as follows, where your actions are in **bold**:

```
sysctl> svcversion
1.1
sysctl> pdf
redhook.kgn.ibm.com: ======================================================
Filesystem                Size-KB Used-KB Free-KB %Free  iUsed  iFree %iFree
/                            8192    5164    3028   37%    815   1233    61%
/var                        16384    4176   12208   75%    343   3753    92%
/usr                       552960  467888   85072   16%  23203 116061    84%
/tmp                        49152   23860   25292   52%    235  12053    99%
/home                       65536   63556    1980    4%   5375  11009    68%
sysctl> quit
```

To exit the interactive mode, enter one of the following:

```
exit
quit
<Ctrl-d>
```

Using Sysctl in Noninteractive Mode

Sending Commands to a Local Server

If no target nodes are specified on the Sysctl command, the command is sent to the server on the local node. For example, to send the **whoami** command to the local server, enter:

```
sysctl whoami
```

Directing Commands to Remote Hosts

To direct a command to remote host **sivle**, use **-h** and enter:

```
sysctl -h sivle sys:info
```

The system's response:

```
sivle.kgn.ibm.com power 3.2 rs_aix32
```

To direct a command to remote hosts **sivle**, and **yelserp**, enter:

```
sysctl -h sivle -h yelserp sys:info
```

The system's response:

```
>> sivle
sivle.kgn.ibm.com power 3.2 rs_aix32
<<
>> yelserp
yelserp.kgn.ibm.com power 3.2 rs_aix32
<<
```

Sending Commands to a Collection of Nodes

Multiple nodes can be targeted with a single Sysctl command. The collection of nodes can be specified in a file. Alternatively, the Sysctl command will read node names from **stdin**. In the following example, the file /**tmp**/**node-file** contains node names **r05n13.hpssl.kgn.ibm.com**, **r05n15.hpssl.kgn.ibm.com**, and **r05n09.hpssl.kgn.ibm.com**.

To direct the **pdf** command to the /**tmp**/**node-file** collection, enter:

```
sysctl -c /tmp/node-file pdf
```

The response in this example is the following:

```
>> r05n13
r05n13.hpssl.kgn.ibm------------------------------------------------------------
Filesystem                Size-KB Used-KB Free-KB %Free  iUsed  iFree %iFree
/                            4096    3372     724   18%    792    232    23%
/var                         8192    6320    1872   23%    297   1751    86%
/usr                       335872  331632    4240    2%  17820  66148    79%
/tmp                         8192     520    7672   94%     67   1981    97%
/home                        4096     168    3928   96%     18   1006    99%
<<
>> r05n15
r05n15.hpssl.kgn.ibm------------------------------------------------------------
Filesystem                Size-KB Used-KB Free-KB %Free  iUsed  iFree %iFree
/                            4096    3364     732   18%    796    228    23%
/var                         8192    3028    5164   64%    281   1767    87%
/usr                       335872  331412    4460    2%  17825  66143    79%
/tmp                         8192     464    7728   95%     57   1991    98%
/home                        4096     168    3928   96%     18   1006    99%
<<
>> r05n09
r05n09.hpssl.kgn.ibm------------------------------------------------------------
Filesystem                Size-KB Used-KB Free-KB %Free  iUsed  iFree %iFree
/                            4096    3340     756   19%    790    234    23%
/var                         8192    3336    4856   60%    285   1763    87%
/usr                       331776  331012     764    1%  17742  66226    79%
/tmp                         8192    1212    6980   86%     58   1990    98%
/home                        4096     164    3932   96%     17   1007    99%
<<
```

Developing a Sysctl Application

Creating a Sysctl application consists of the following basic steps:

1. Examine what you are trying to do and divide it among client/server lines. The server side runs as **root**. The client side typically does not require root authority, but requires that the user has run **kinit**.

2. Define the client/server interface. This consists of Sysctl command names and their arguments, return values, and authorizations.
3. Code the server commands. This consists of Tcl code that performs your task, as well as callback authorizations if you do not use one of the supplied ones. Note that the Tcl code may be simple (such as a wrapper around an AIX command or script) or elaborate (a detailed Tcl program).
4. Code the client side. This typically gathers any necessary information and then issues the **sysctl** command to cause the servers to perform the desired action. This can be a GUI, Perl or shell script. Note that client side code may not always be necessary. In some cases, issuing the **sysctl** command directly from the command line is sufficient.
5. If the authorization scheme requires changes to existing ACL files or new ACL files, these changes must be made, and the changed files distributed.
6. Register the new server command with each node's **sysctld** daemon by restarting the daemons with updated configuration files.

Here is an example of how to write a Sysctl application that allows a user to execute a script including some commands requiring root execution privileges on all the nodes in a SP System in parallel. The user is not required to know any of the root passwords, but must be listed in an ACL specifically set up to authorize use of the script. A copy of this ACL resides on each node. The script containing the commands executable only by root is called **/etc/root_script**. Assume that **/etc/root_script** accepts a single argument, and that this script resides on each node.

The server side of the application does the root part of the job, which is to issue the **/etc/root_script** command with the correct arguments on behalf of authorized users. It first needs to determine if the client is authorized based on an authorization policy. As mentioned above, authorization is based on an access control list. There are no return values, but an error message is displayed when the user is not authorized to run the command. **stdout** and **stderr** are returned to the client.

The server command is coded as follows and resides in **/etc/sysctl/apps/adminscript**:

```
create proc adminscript {argument} AUTH {
        global SCUSER
        if [ aclcheck -f /etc/sysctl.adminscript.acl $SCUSER ] {
                exec /etc/root_script $argument
                return
        }
        puts "Not authorized to run adminscript"
}
```

The first line creates a new Sysctl proc called **adminscript**. One argument is accepted, which will be passed unchanged to **/etc/root_script**. This proc uses the provided AUTH authorization callback, which means that only authenticated clients can execute the command.

The second line issues the Tcl **global** command to provide the content of the SCUSER variable to other Sysctl commands. (This is needed because SCUSER is passed as an argument to **aclcheck** later.) The SCUSER variable is the

authenticated username of the client, and is one of the variables set by the server and available to procedures and callbacks. (See “Authorization Variables” on page 67 for a list of all these variables.)

The third line is Tcl code that uses the built-in **aclcheck** command to see if the user is in the ACL (/**etc**/**sysctl.adminscript.acl**) set up for this command. If the user is authorized, the program issues the /**etc**/**root_script** command with the supplied argument and then returns. If the user is not authorized, the program issues an error message. The program returns any output (information from /**etc**/**root_script** command or the error message) to the client user's terminal.

Note that in this example, further authorization checking is done in the body of the procedure after the authorization callback has completed. This is to demonstrate the use of the SCUSER variable provided by the server to procedures. Alternatively, this check could have been made part of the authorization callback itself.

The client side of the application needs to find out the necessary information to provide to the servers. It does not require **root** authority to be issued. The information in this case includes the single argument to be passed through to /**etc**/**root_script**. The client also needs to determine the hostnames of all the nodes in the SP System, so that the command can be sent to and run in parallel on all the nodes.

The client side is coded as follows in /**usr**/**bin**/**doadminscript**:

```
#!/usr/lpp/ssp/perl/bin/perl

# Run /etc/root_script on all the SP nodes in parallel by invoking
# the adminscript sysctl command on all the nodes. Invoke via:
# doadminscript arg_to_root_script

$#ARGV > 1 && die "too many arguments";

exec "/usr/lpp/ssp/bin/hostlist -av |
      /usr/bin/sysctl -c - adminscript $ARGV[0]";
```

The first line checks for extraneous arguments. The **exec** statement uses the SP **hostlist** command to write all the currently responding hostnames in the SP System to **stdout**. The single argument to **adminscript** is passed along to the Sysctl **adminscript** server command.

A new ACL file needs to be put on each node in the system. The file /**etc**/**sysctl.adminscript.acl** looks like this:

```
#acl#
# These are the users that can issue the adminscript command on this
# node:
_PRINCIPAL root@HPSSL.KGN.IBM.COM
_PRINCIPAL newbie@HPSSL.KGN.IBM.COM
_PRINCIPAL guru@HPSSL.KGN.IBM.COM
```

Note that each node can authorize different users to run the new command. In this case, assume that the ACL will be the same on all nodes. To distribute the ACL file to all the responding nodes, an authorized user enters the following (see the **pcp** man page for details):

```
pcp "-av" /etc/sysctl.adminscript.acl
```

In addition, the **/etc/sysctl.conf** file needs to be updated to include the new command. If the server code is to reside in **/etc/sysctl/apps** directory, the following lines can be added to **/etc/sysctl.conf**:

```
# Add the adminscript application
include /etc/sysctl/apps/adminscript
```

The updated **sysctl.conf** file and the actual server code need to be distributed:

```
pcp "-av" /etc/sysctl.conf
pcp "-av" /etc/sysctl/apps/adminscript
```

Now the servers need to be restarted to include the updated **sysctl.conf** file. Note that the following command requires ACL authorization:

```
hostlist -av | sysctl -c - svcrestart
```

Now, any of the users in **/etc/sysctl.adminscript.acl** (**root**, **newbie**, or **guru**) can run **/etc/root_script** with the argument **mikef** on all the nodes simultaneously by entering the following commands:

```
kinit (to get authentication information)

adminscript mikef
```

Authorization Variables

These variables are set by the server based on information on the client provided by SP Authentication Services. Authorization callbacks and Tcl scripts have access to these variables. Table 1 lists the authorization variables.

Table 1. Authorization Variables

Variable	Description
SCACL	Indicates if the client has been authenticated and is a trusted user. Value is **1** if the client has been authenticated and is listed on the server's ACL of trusted users, **0** otherwise.
SCHOST	The name of the host SP node from which the Sysctl request was issued.
SCINSTANCE	The instance of the user or NULL if unauthenticated.
SCLHOST	The host name of the local server.
SCLREALM	The local realm of the server.
SCMODE	The communication mode between the client and server. May be WAIT, NOWAIT, or SOCKET.
SCPRINCIPAL	The authenticated identity of the command issuer, in the form user.instance@realm. If the client has not been authenticated, the value is NULL.
SCREALM	The realm of the user or NULL if unauthenticated.
SCSCRIPT	The commands sent from the client to the server. If a script was passed, the contents of the script is shown.
SCUSER	The base name of the user. NULL if unauthenticated.

Sysctl Scripts

Replaying a Sysctl Script

A Sysctl script consists of a set of Sysctl commands that can be executed with a single Sysctl client invocation. Sending a Sysctl script to a server or set of servers is called *replaying* the script on those servers.

When a script is replayed, each Sysctl command is run sequentially by the servers. Authorization is done on a command by command basis as the script runs. The script stops execution when any command fails to be executed due to an authorization failure.

To run a Sysctl script, send it to the Sysctl server by specifying **-r** (replay) on the **sysctl** command:

```
sysctl -h node -r script-filename
```

The following is a simple script, **/examples/sysctl/script1**:

```
echo I am $SCLHOST
echo You are $SCPRINCIPAL
```

Here is the command that user **sarah** issues to run the script on SP node **r05n11**:

```
sysctl -h r05n11 -r /examples/sysctl/script1
```

sarah receives the following output:

```
I am r05n11.testcell.ibm.com
You are sarah@testcell.ibm.com
```

You can also have Sysctl read a script from **stdin** by typing a dash (-) after the **-r** flag, instead of a file name. For example, to run three commands for node **tappan.kgn.ibm.com**, enter the following sequence of commands:

```
sysctl -h tappan.kgn.ibm.com -r -
echo $SCLHOST
pdf
svcversion
<Ctrl-D>
```

You will not get any responses until you press <**Ctrl-D**>. In this example, the response is the following:

```
tappan.kgn.ibm.com
tappan.kgn.ibm.com: ============================================================
Filesystem                Size-KB Used-KB Free-KB %Free  iUsed  iFree %iFree
/                            8192    5164    3028   37%    815   1233    61%
/var                        16384    4184   12200   75%    343   3753    92%
/usr                       552960  467888   85072   16%  23203 116061    84%
/tmp                        49152   23868   25284   52%    239  12049    99%
/home                       65536   63556    1980    4%   5375  11009    68%
1.1
```

In addition, Sysctl scripts can be written as self-contained programs. Do this by beginning the script with (flags other than **-r** can be specified this way as well):

```
#!/usr/lpp/ssp/bin/sysctl -r
```

Enter the name of the script to run it.

Consider the following example, /**etc**/**fscheck**. This script checks filesystem usage and reports on filesystems that have usage higher than that specified in an argument. The **-L** flag causes output to be formatted as a set of lines, each preceded by the hostname from which it came:

```
$ cat /etc/fscheck
#!/usr/lpp/ssp/bin/sysctl -Lr
#
# Check filesystem usage - high-water pct mark is passed as a parameter
#
foreach fs [listfs] {
     statfs $fs sbuf
     set pct [expr $sbuf(used).0/$sbuf(size).0*100]
if {$pct > $argv(1)} {
          set pct [format "%.2f" $pct]
          echo "$fs ==> ${pct}% Used, $sbuf(free) KB Free"
     }
}
$ cat /tmp/hostlist
harlem
badger
$ fscheck -c /tmp/hostlist 80
harlem: / ==> 84.29% Used, 1538 KB Free
badger: /usr ==> 98.35% Used, 3656 KB Free
$
```

You can override any parameters specified in the script by invoking it explicitly. For example, when you issue:

```
sysctl -r /etc/fscheck
```

Sysctl ignores the **#!** line in the script. Since the **-L** option is not interpreted in this case, the output would be:

```
$ sysctl -c /tmp/hostlist /etc/fscheck 80
<<
>> harlem
/ ==> 84.29% Used, 1538 KB Free
<<
>> badger
/usr ==> 98.35% Used, 3656 KB Free
<<
```

Specifying a Host Target in a Script

Sysctl allows you to target a particular host within a script. The target host is specified in the body of the script via use of the **#host#** keyword. If the script contains a line beginning with **#!**, the **#host#** line must immediately follow it. Otherwise, **#host#** must be the first line in the script.

If you use the **-h** option to specify an SP node name when you invoke **sysctl**, any target host specified in the script is ignored. Here is a script, /**examples**/**sysctl**/**script3**, targeted to host **issc3**:

```
#!/usr/bin/sysctl -r
#host#issc3
echo I am $SCLHOST
```

Using the **-h** option to run the script overrides the host specified within the script. For example, if you specify:

```
sysctl -h r05n11 -r /examples/sysctl/script3
```

the script runs on **r05n11** instead of **issc3**.

Passing Parameters to a Script

When you issue the **sysctl** command with the **-r** parameter, any command line arguments following the script name are treated as parameters to be passed to the script. Here is a sample script, /**examples**/**sysctl**/**script6**:

```
#!/usr/bin/sysctl -r
#
echo Number of parameters is $argc
for {set i 0} {$i < $argc} {incr i 1} {
   echo Parameter #$i is \"$argv($i)\"
}
```

When you invoke the script with parameters **arg1** and **arg2**:

```
/examples/sysctl/script6 arg1 arg2
```

The output is the following:

```
Number of parameters is 3
Parameter #0 is script6
Parameter #1 is "arg1"
Parameter #2 is "arg2"
```

In this example, two parameters were passed to the script. The parameters are stored in the **argv** array, and the number of parameters is stored in variable **argc**, just as in the standard Tcl **exec** function. Because **argv[0]** always contains the name of the script being run, the value of **argc** is one greater than the number of arguments passed to the script. That is why the first line of output says that there are three parameters, when only two were passed to the script.

Sysctl Reference Information

Online Help Facility

To access online help on built-in Sysctl commands, enter:

```
sysctl help  command-name
```

For example, to access help on the **setauth** command, enter:

```
sysctl help setauth
```

Tcl Commands

Table 2 shows the commands available in the Tcl language interpreter and the Sysctl base authorizations required to use them.

Table 2. Built-in Tcl Commands and Their Default Authorization Callbacks

append (AUTH)	array (AUTH)	break (AUTH)	case (AUTH)
catch (AUTH)	cd (ACL)	close (ACL)	concat (AUTH)
continue (AUTH)	eof (ACL)	error (AUTH)	eval (ACL)
exec (ACL)	exit (AUTH)	expr (AUTH)	file (ACL)
flush (ACL)	for (ACL)	foreach (ACL)	format (AUTH)
gets (ACL)	glob (ACL)	global (AUTH)	history (AUTH)
if (AUTH)	incr (AUTH)	info (ACL)	join (AUTH)
lappend (AUTH)	lindex (AUTH)	linsert (AUTH)	list (AUTH)
llength (AUTH)	lrange (AUTH)	lreplace (AUTH)	lsearch (AUTH)
lsort (AUTH)	open (ACL)	pid (ACL)	proc (SYSTEM)
puts (ACL)	pwd (ACL)	read (ACL)	regexp (AUTH)
regsub (AUTH)	rename (SYSTEM)	return (NONE)	scan (AUTH)
seek (ACL)	set (AUTH)	source (SYSTEM)	split (AUTH)
string (AUTH)	switch (AUTH)	tell (ACL)	time (ACL)
trace (ACL)	unset (AUTH)	uplevel (SYSTEM)	upvar (ACL)
while (ACL)			

Extended Tcl Commands

Table 3 shows the commands available in the TclX language interpreter and the Sysctl base authorizations required to use them.

Table 3. Built-in Extended Tcl Commands and Their Default Authorization Callbacks

alarm (ACL)	bsearch (ACL)	catclose (ACL)	catgets (ACL)
catopen (ACL)	cequal (AUTH)	cexpand (AUTH)	chgrp (ACL)
chmod (ACL)	chown (ACL)	chroot (SYSTEM)	cindex (AUTH)
clength (AUTH)	cmdtrace (ACL)	commandloop (ACL)	convertclock (AUTH)
copyfile (ACL)	crange (AUTH)	csubstr (AUTH)	ctoken (AUTH)
ctype (AUTH)	dup (ACL)	echo (AUTH)	execl (ACL)
fcntl (ACL)	flock (ACL)	fmtclock (AUTH)	fork (ACL)
frename (ACL)	fstat (ACL)	funlock (ACL)	getclock (AUTH)
id (ACL)	infox (ACL)	keyldel (AUTH)	keylget (AUTH)
keylkeys (AUTH)	keylset (AUTH)	kill (ACL)	lassign (AUTH)
lempty (ACL)	lgets (ACL)	link (ACL)	lmatch (AUTH)
loop (ACL)	lvarcat (AUTH)	lvarpop (AUTH)	lvarpush (AUTH)
max (ACL)	min (ACL)	mkdir (ACL)	nice (ACL)
pipe (ACL)	profile (ACL)	random (AUTH)	readdir (ACL)
replicate (AUTH)	rmdir (ACL)	scancontext (ACL)	scanfile (ACL)
scanmatch (ACL)	select (ACL)	server_open (ACL)	signal (ACL)
sleep (ACL)	sync (ACL)	system (ACL)	times (ACL)
translit (AUTH)	umask (ACL)	unlink (ACL)	wait (ACL)

Built-in Sysctl Commands

The following Sysctl commands are provided:

Table 4. ACL Processing Commands

ACL Processing Commands	Default Auth	Description
acladd	ACL	Adds entries to an ACL file. If a filename is not specified, the entry is added to the server's ACL file.
aclcheck	ACL	Checks to see if a principal is in an ACL file. Returns **1** if in ACL, **0** if not in ACL. If no filename is specified, Sysctl looks in the server's ACL file.
aclcreate	ACL	Creates a new ACL file, inserting the principals specified. The **-f** parameter must be supplied.
acldelete	ACL	Deletes an ACL from a file.
acldestroy	ACL	Erases an ACL file.
acllist	ACL	Lists the ACLs in a file. If no file is specified, the server's ACL file is listed.
aclrecreate	ACL	Similar to **aclcreate**, but removes the target ACL file if it already exists before creating it again.

Table 5. Service Commands

Service Commands	Default Auth	Description
svcconnect	AUTH	Determines the connection authorization policy for the **sysctld** server.
svcdetach	ACL	Detaches the current process from the calling client; that is, breaks the connection between server and client. Use to start a background daemon.
svclog *string*	ACL	Writes the *string* to the server log file.
svclogevent	SYSTEM	Writes a line to the server log file indicating a user has connected to the server.
svcpid	ACL	Prints the process ID of the **sysctld** server daemon.
svcredirect *handle*	ACL	Redirects the **stdout** and **stderr** for the current session to a new file.
svcrestart	ACL	Restarts the server, causing configuration files to be reread and interpreters to be reinitialized. Use to activate changes to Sysctl servers.
svcversion	NONE	Returns the Sysctl version number.

Table 6. Other Commands

Other Commands	Default Auth	Description
ACL	AUTH	A supplied authorization callback.
AUTH	AUTH	A supplied authorization callback.
NONE	AUTH	A supplied authorization callback.
SYSTEM	AUTH	A supplied authorization callback.
checkauth	ACL	Determines if you are authorized to access an object.
confadd	ACL	Adds items to a server's configuration file.
confdelete	ACL	Removes items from a server's configuration file.
create	SYSTEM	Creates new objects in the server.
getauth	ACL	Displays the authorization callbacks for an object.
include	SYSTEM	Includes additional configuration files.
listfs	ACL	Lists file system names.
load	SYSTEM	Dynamically load a shared library into the server and call an initialization routine within the library.
quit	AUTH	Ends the current session.
safeargs *arg-list*	ACL	Checks arguments for illegal shell characters. Does not execute the command.
safeexec *commands ...*	ACL	Runs the Tcl exec or **system** command, checking arguments for illegal shell characters and preventing execution if any are found.
safesystem	ACL	Runs the Tcl exec or **system** command, checking arguments for illegal shell characters and preventing execution if any are found.
setauth	SYSTEM	Sets the authorization callback for an object.
statfs	ACL	Queries the status of all local file systems.
unload	SYSTEM	Unloads a previously loaded shared library.
whoami	NONE	Echoes the authenticated identity of the issuer or **unknown** if the user is not authenticated.

Managing the SP System

© Copyright IBM Corp. 1995

Chapter 5. Managing a High Availability Control Workstation Configuration

Notice

The High Availability Control Workstation hardware and software functions are separately priced features that will be available October 13, 1995.

This chapter provides overview and task information for the High Availability Control Workstation (HACWS) function. The HACWS function provides for a backup control workstation for your SP system.

We presume that you have planned for, installed, and configured the HACWS function.

For information on planning for HACWS function, refer to *IBM RISC System/6000 Scalable POWERparallel Systems: Site Planning* and *IBM RISC System/6000 Scalable POWERparallel Systems: System Planning*.

We also recommend that you refer to the documentation for AIX High Availability Cluster Multi-Processing/6000 (HACMP/6000) Version 4.1. HACMP/6000 is the foundation for HACWS function.

Information is provided in this chapter for the following tasks:

- Understanding HACWS function
- Modifying typical administrative tasks (maintaining the /**etc**/**passwd** database, setting up mail spools, setting up file collections, for example) so that subsystems will fail over to the backup control workstation in the event the primary control workstation becomes unavailable. We'll cover this information in the order of the related chapters in the *IBM RISC System/6000 Scalable POWERparallel Systems: Administration Guide*:
  - Security Features of the SP System
  - Sysctl
  - Managing System Partitions
  - Using the SP System Monitor
  - Starting Up and Shutting Down the SP System
  - The High Performance Switch
  - Parallel Management Commands
  - Managing SP Resources
  - Managing User Accounts
  - Managing NTP
  - Managing File Collections
  - Managing Amd
  - Managing Print Service
  - Managing Mail Service
  - Accounting
- Customizing HACMP/6000
- Verifying Frame Supervisor Cables

© Copyright IBM Corp. 1995

Understanding HACWS Function

The following definitions will help in understanding HACWS function.

Primary control workstation
The first control workstation that is installed with the SP system. The primary control workstation is a physical machine.

Backup control workstation
The standby backup control workstation that is installed during the HACWS installation. The backup control workstation is a physical machine.

Active control workstation
The control workstation that is currently running the SP control workstation applications (the SDR and **hardmon**, for example). This can be either the primary or backup control workstation.

Inactive control workstation
The control workstation that is not running the SP control workstation applications. This can be either the primary or backup control workstation.

The HACWS function improves the reliability, availability and serviceability of an SP system by providing for a backup control workstation. HACWS provides for extending the SP system configuration to include a second RISC System/6000 to function as a backup to the primary control workstation. Automatic failover and reintegration to the backup control workstation are provided should the primary control workstation fail or in the case of taking the primary control workstation down to perform hardware and software maintenance.

HACWS provides hardware and software features that remove a single point of failure for the SP system. HACWS does not, however, protect against double failures. You should not expect control workstation functions to work during the time that a control workstation is failing over. If a failure on the control workstation occurs while a control workstation function is being performed, the function needs to be restarted.

The following figure illustrates a typical HACWS configuration:

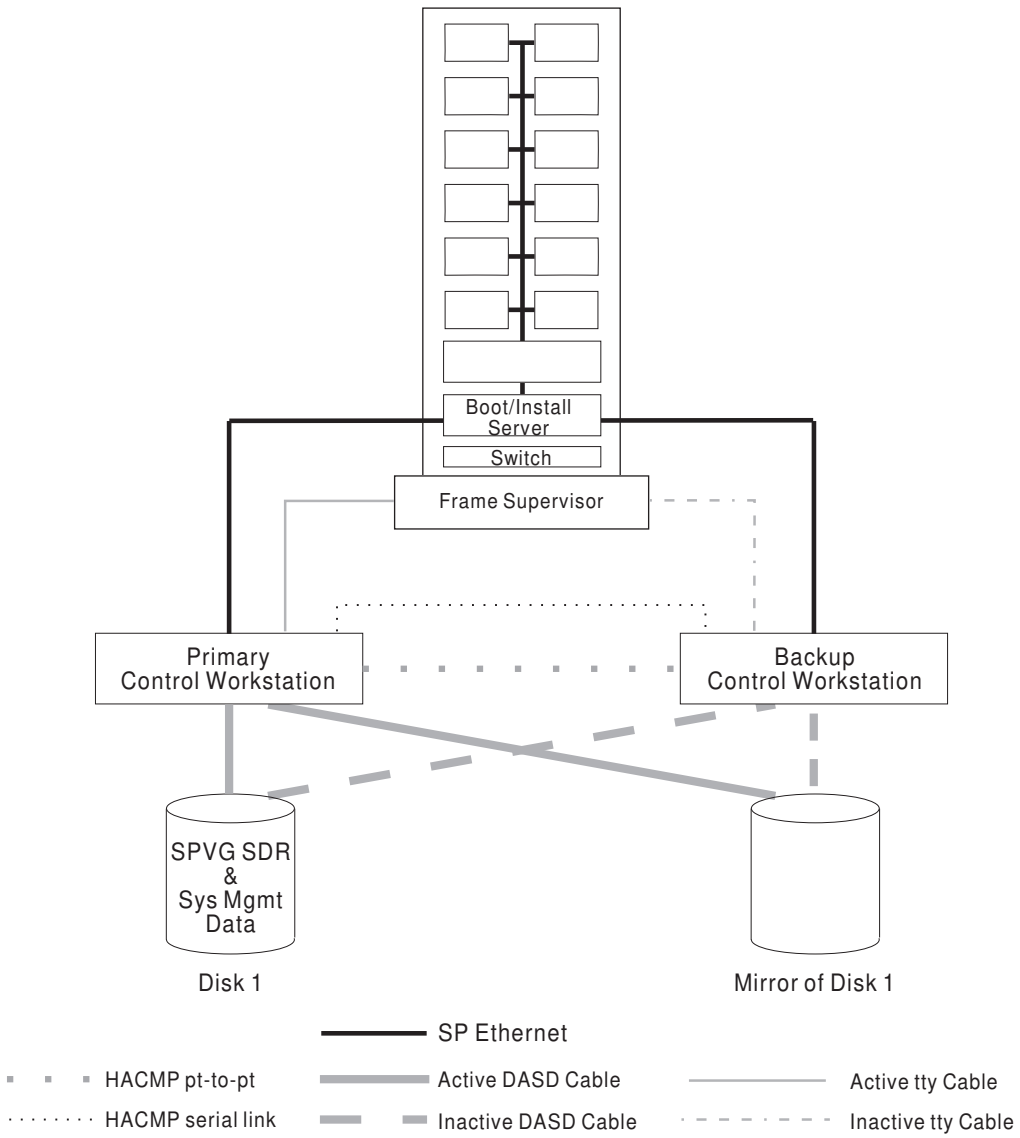


Figure 4. An HACWS Configuration

Hardware

HACWS is a two-node configuration: a primary control workstation with a backup control workstation. (Don't confuse the use of *node* in this context with an SP system node.) We highly recommend that the two RISC System/6000s be configured in exactly the same way with the same model hardware and I/O configurations. This is not required, but simplifies the management of the HACWS configuration. We also highly recommend that the primary and backup control workstations do not use the same power source.

External Fixed Disks and Disk Controllers

External fixed disks that provide only nonconcurrent access are used in an HACWS configuration. We strongly recommend that the external fixed disks be mirrored across two disk controllers. If the external fixed disks are not mirrored, the single point of failure has not been removed from the SP system. The fixed disk becomes the single point of failure: if the disk fails the SP system will be down until the disk can be replaced. We also recommend that the two disk controllers be on different

power sources. Only 9333 serial disks and controllers with multi-tail access and SCSI-2 disks with multi-tail disk controllers are supported. RAID disk controllers and devices are not supported.

Frame Supervisor Cards

HACWS requires a dual RS-232 frame supervisor card per frame with a connection from each control workstation to each SP frame. Hardware feature #1245 provides this new type of frame supervisor card with an RS-232 Y-cable that connects from a frame in the SP system to the control workstation. All frames in the SP system need this new frame supervisor card and the Y-cables in order for the HACWS software to be configured. The control workstation must run AIX 4.1 to support this new frame supervisor card. The frame supervisor connections from each frame must be connected to the exact same tty ports on both control workstations. If not cabled correctly, frame supervisor connections will not be activated by the hardware monitor.

Network Connections

One of the following network connections must exist between the primary and backup control workstations :

- A dedicated TCP/IP network link
- An RS-232 tty link
- The target mode SCSI across the external SCSI fixed disks

Adapters

Each control workstation requires the same number of connections to the SP Ethernet on the same LAN segments. Each SP Ethernet LAN segment must be cabled to the same Ethernet (**en***x*) adapter. Standby network adapters are optional in an HACWS configuration because there may not be enough adapter slots in the control workstation for a standby adapter on each service network. On a small one- or two-frame SP system the normally used control workstation has only two slots and does not have enough slots for standby adapters. In a large multi-frame SP system the number of slots required for frame supervisor and SP Ethernet LAN segments may not allow for standby adapters. However, the presence of a standby adapter may avoid the need to fail over to the inactive control workstation.

Software

The software foundation for HACWS is the separately orderable AIX High Availability Cluster Multi-Processing/6000 (HACMP/6000) Version 4.1. Only the high availability feature (# 5050) is required. The concurrent access feature (# 5051) of HACMP/6000 is not supported (all the SP data on the control workstation is in an AIX Journaled File System and, therefore, does not allow concurrent access).

HACMP/6000 Version 4.1 is required for the control workstation.

Note: In an HACWS configuration, the nodes must run AIX 4.1 and PSSP 2.1. HACMP is not currently supported on the nodes.

The primary and backup control workstation are configured in a two-node rotating configuration. The external fixed disks are configured in nonconcurrent access only. For information on HACMP/6000, refer to *HACMP/6000: Concepts and Facilities*, SC23-2699.

Building on HACMP/6000, the **ssp.hacws** optional installation package that is part of the IBM Parallel System Support Programs for AIX provides:

- Scripts to create a backup control workstation
- Verification programs
- HACMP/6000 pre- and post- event scripts
- Scripts to synchronize the backup and primary control workstations

When a failure of the primary control workstation takes place there is a disruptive failover (that is, the failover is noticeable) of the control workstation to the backup control workstation. This failover:

1. Switches the external fixed disks
2. Reconfigures network adapters from boot addresses to service addresses to perform IP address takeover
3. Performs hardware address takeover
4. Restarts the control workstation applications
5. Remounts file systems
6. Resumes hardware monitoring
7. Allows clients to reconnect to obtain services or update control workstation data

The backup control workstation assumes the IP address (service address) and IP aliases of the primary control workstation, resulting in only one active control workstation at a time and allowing client applications to run without changes. See Figure 5 on page 82 for an illustration of addressing in an HACWS configuration.

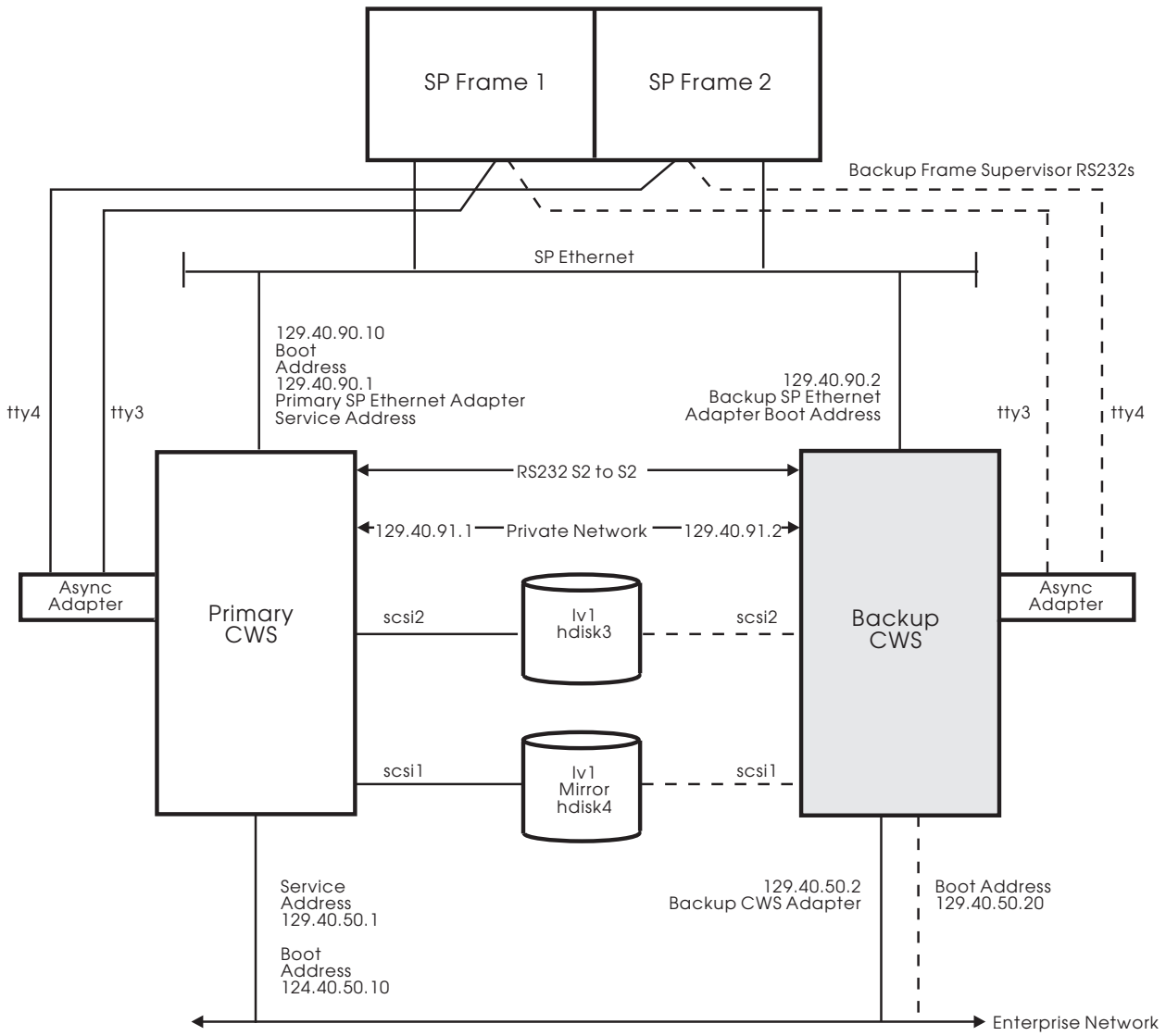


Figure 5. Addressing in an HACWS Configuration

Operating HACMP/6000 and the Control Workstation

This section describes some of the operational characteristics of the control workstation and HACMP/6000.

Booting the Control Workstations

In a two-node rotating HACMP/6000 configuration, if you are booting both the primary and backup control workstations at the same time, the first control workstation to have the cluster manager started will acquire the shared resources. In an HACWS scenario this will cause the control workstation IP addresses (service address) and the /**spdata**/**sys1** file system to be configured on that control workstation. It is important that the RISC System/6000 that is intended to be the active control workstation have the HACMP/6000 cluster manager started on it first. To help ensure that the correct control workstation acquires the shared resources we recommend that the cluster manager be started manually instead of being defined to start at system restart. This allows both control workstations to be booted in any order and any potential race conditions can be avoided during the

boot process. After both control workstations are booted the cluster manager can be started on the intended control workstation.

If the control workstation function is started on the unintended control workstation, you can perform the following sequence of events to make the intended control workstation active:

1. Boot the intended control workstation without starting the cluster manager.
2. Do a graceful shutdown without takeover on the incorrectly active control workstation.

   Using SMIT:

TYPE	**smit clstop**
	▲The Stop Cluster Services menu is displayed.
SELECT	now (for Stop now option)
SELECT	graceful (for Shutdown mode option)
HIT	Enter (to Do)

   Using the Command Line:

   Enter

   ```
   /usr/sbin/cluster/utilities/clstop -y -N -g
   ```

   Note: Never use **kill** to stop the HACMP/6000 cluster manager. Doing so will cause the node to fail.
3. Start the cluster manager on the intended control workstation.

   Using SMIT:

TYPE	**smit clstart**
	▲The Start Cluster Services menu is displayed.
SELECT	now (for Start now option)
SELECT	true (for Startup Cluster Information Daemon? option)
	Note: Don't use the **on system restart** option.
HIT	Enter (to Do)
4. Execute the **clstat** utility to make sure HACMP/6000 is up and running on the correct control workstation.
5. Start the cluster manager on the inactive control workstation.

Avoiding False Adapter Failures on Single-Adapter Networks

If a control workstation is configured without a standby adapter on one of its service networks, and the other control workstation is powered off, the HACMP software may find it difficult to determine service adapter failure. This is because the HACMP cluster manager cannot use a standby adapter to force packet traffic over the service adapter to verify its operation. This shortcoming is less of an exposure if one or more of the following is true:

1. There are network devices which answer broadcast ICMP ECHO requests. This can be verified by pinging the broadcast address and determining the number of different IP addresses that respond.

2. The service adapter is under heavy use. In this instance the inbound packet count will continue to increase over the service adapter without stimulation from the cluster manager.

If neither of these cases is true, then HACMP may report that a service adapter has failed, even though there has not been a failure; it is just that there are not any other adapters to communicate with. (For example, this can occur when all the nodes are powered off.)

An enhancement to **netmon**, the network monitor portion of the HACMP cluster manager, is described below. It can be configured to allow more accurate determination of a service adapter failure. This function can be used in configurations that require a single service adapter per network.

A **netmon** configuration file, **/usr/sbin/cluster/netmon.cf**, will specify additional network addresses to which ICMP ECHO requests can be sent. The configuration file consists of one IP address or IP label per line. The maximum number of addresses used is five. All addresses specified after the fifth will be ignored. No comments are allowed in the file.

The following is an example of a **netmon.cf** configuration file:

```
180.146.181.119
steamer
chowder
180.146.181.121
mussel
```

This file must exist at cluster startup. The cluster software will scan the configuration file during its initialization phase. When **netmon** needs to stimulate the network to verify adapter function, it will send ICMP ECHO requests to each address. After sending the request to every address, **netmon** will check the inbound packet count before determining whether there is an adapter failure.

Conditions that Cause a Failover of the Control Workstation

The following conditions will cause a failover of a control workstation:

1. An intentional failover

   This occurs when the operator stops the cluster manager on a node with a shutdown mode of **graceful with takeover**.

2. A failure of the operating system on the active control workstation

   If the AIX operating system fails, the cluster manager on the inactive control workstation will detect this via missed heartbeats. The cluster manager then declares the crashed control workstation dead and performs a takeover of the shared resources.

3. A failure of an external disk adapter that is not backed up

   If HACWS is configured with only one external disk adapter per control workstation, you need to write an Error Notification Object for the failure of the disk adapter to do a graceful shutdown with takeover. The other control workstation will then take over the shared resources.

4. A service network failure occurs on the active control workstation and the inactive control workstation is available to take over

A failure of a service adapter on the active control workstation, without a standby adapter to take over, will cause some SP node subsystems to perceive that the control workstation has failed and the system will quickly become unstable. To prevent such system instability, a service network failure on the active control workstation is promoted to a node failure, causing a failover to the inactive control workstation.

Failures that Do Not Cause a Failover of the Control Workstation

The following conditions will not cause a failover of a control workstation:

1. A hung control workstation with network adapters that still heartbeat

   In the case where the application software hangs and the network adapters being monitored via HACMP/6000 still respond to heartbeats from the inactive control workstation, HACMP/6000 will not detect the hang and will not fail over.

2. A service network failure occurs on the active control workstation and the inactive control workstation is not available to take over

   If the inactive control workstation is not available to take over, then a service network failure on the active control workstation will not be promoted to a node failure because no failover can occur.

General Guidelines for When to Do Configuration Changes

We recommend that you do configuration changes on the SP system only when the primary control workstation is the active control workstation. Most configuration changes are allowed when the backup control workstation is the active control workstation. However, if changes are made when the backup control workstation is active, some configuration files will need to be updated on the primary control workstation. When changes are made on the primary control workstation, some configuration files will need to be updated on the backup control workstation. If configuration changes are done on the backup control workstation, the updates go the opposite direction (backup to primary instead of primary to backup). It is easier to manage the system when the updates go only in one direction.

You can decide to what extent you do configuration changes on the backup control workstation. Because only two machines are involved, the bidirectional updates may not be a problem.

The following table lists tasks that may be performed in an HACWS configuration and whether or not it is possible to update configuration files on the active backup control workstation. A 'No' under the 'Backup Control Workstation Active' column indicates that the file cannot be updated on the active backup control workstation. In such a case, you will have to update the file on the primary control workstation when it is again active.

Table 7 (Page 1 of 2). High Availability Control Workstation Configuration and Task Summary

Task	Primary Control Workstation Active	Backup Control Workstation Active
Update passwords	Yes	No
Add or change users	Yes	No
Change Kerberos keys	Yes	No

Table 7 (Page 2 of 2). High Availability Control Workstation Configuration and Task Summary

Task	Primary Control Workstation Active	Backup Control Workstation Active
Install a node	Yes	Yes
Change or add system partitions	Yes	Yes
Add nodes to the system	Yes	No
Hardware monitoring	Yes	Yes
Reboot nodes	Yes	Yes
Run diagnostics	Yes	Yes
Shut down and restart system	Yes	Yes
Run parallel jobs	Yes	Yes
Update file collections	Yes	Yes
Accounting	Yes	Yes
Change site environment information	Yes	No

Managing SP Subsystems for High Availability Control Workstation Function

In this section we present information on the administrative tasks you must perform to ensure that the various SP subsystems will fail over to the backup control workstation.

Security Features of the SP System

In an HACWS configuration the primary authentication service is **not** duplicated on the backup control workstation. If the primary control workstation is either a primary or secondary authentication server, then the backup control workstation must be a secondary authentication server. The backup control workstation cannot be a primary authentication server.

When the primary control workstation is down, and it is the primary authentication server, no updates to the authentication database can be made.

Kerberos Configuration Files

The backup control workstation needs to be added to the **.klogin** files on all the nodes and the control workstation. This **.klogin** file also needs to be kept in sync on the backup control workstation.

When control workstation services move back and forth between the two control workstations, the Kerberos service keys need to remain the same. To accomplish this, the following commands were issued on the backup control workstation when HACWS was initially configured:

```
$ /usr/lpp/ssp/rcmd/bin/rcp -p primary_name:/etc/krb-srvtab /etc/krb-srvtab.primary
$ cp -p /etc/krb-srvtab /etc/krb-srvtab.backup
$ cat /etc/krb-srvtab.primary >>/etc/krb-srvtab
```

Repeat this procedure whenever you change Kerberos service keys on either of the two control workstations.

Sysctl

When Sysctl is installed the configuration file **/etc/sysctl.conf** is created and updated on the backup control workstation. If you update this file or use a different configuration file, the files needs to be kept in sync between the primary and backup control workstation.

The Sysctl ACL files must also be kept in sync between the primary and backup control workstations. This file is shipped as **/etc/sysctl.acl**. If you add an ACL record to the **/etc/sysctl.acl** file, the file should be kept in sync between the primary and backup control workstations.

When the backup control workstation is the inactive control workstation, Sysctl commands can be used because Sysctl is installed and running on the inactive control workstation.

Managing System Partitions

System partition configuration changes must be done from the active control workstation. Such changes may be done on either the primary or backup control workstation, depending on which one is active. We recommend that configuration changes be done from the primary control workstation, but it is not required.

If you have an HACWS configuration, you can not have system partitions running AIX 3.2.5 and PSSP 1.2.

When you configure HACWS you are instructed to put any IP address aliases that are required for partitioning or any other reason in a script called **/spdata/sys1/hacws/rc.syspar_aliases**. This script will be invoked by the HACWS event scripts at the correct times to enable system partitioning.

Using the SP System Monitor

If you have installed **ssp.clients** on a RISC System/6000 other than the primary or backup control workstations, you need to give special consideration to monitoring the SP system from other than the active control workstation. If you execute the **spmon** command without logging into the active control workstation and the active control workstation does a failover, this will not be represented on the SP System Monitor until you take another action. Once you take another action, the **spmon** command will exit. You will have to reexecute the **spmon** command and reconnect to the currently active control workstation. This occurs because when you are monitoring a TCP/IP connection and the host fails, the client side of that connection does not know about the failure of the connection until a packet is sent from the client side of the connection. Since the users of the **spmon** command who are off the control workstation should be well known, an Error Notification Object can be set up to send mail or a message to the list of well known users to inform them that the control workstation has performed a failover. Such a notification list for the users that monitor the SP system is a good idea in any event. After a failover of a control workstation (or any other major recovery action) an administrator should be notified. Several options for such notification exist:

- Error Notification Objects
- NetView for AIX control desk alerts
- Trouble Ticket for AIX problem incidents
- HACMP Notify script

Viewing which Control Workstation is Active

You can view which RS-232 tail is active in the Frame Environment Layout display of the SP System Monitor. There is an indicator for activity on both RS-232 connections from the frame and an indicator for which tail of the frame supervisor connection is active.

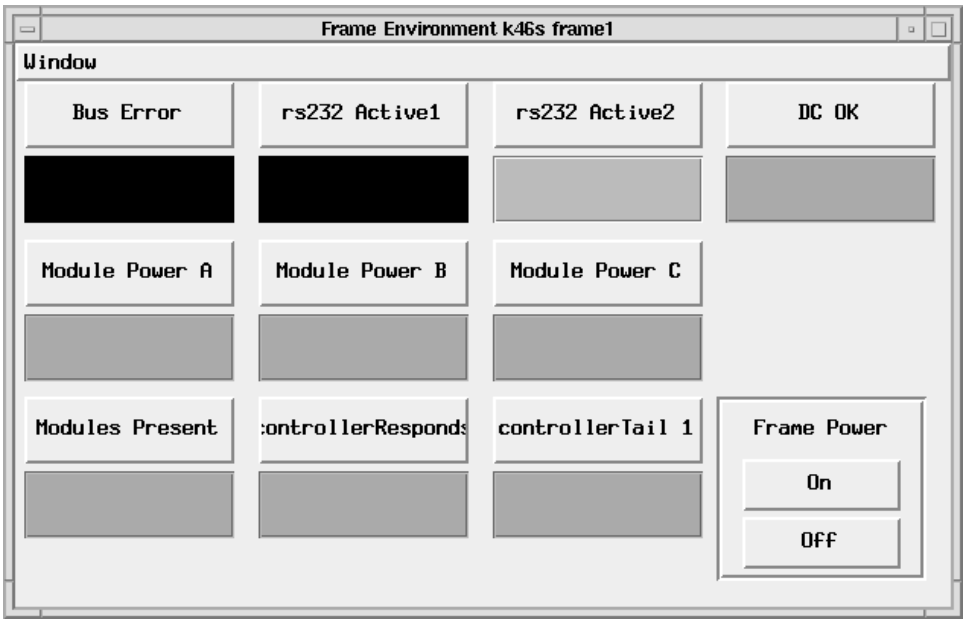


Figure 6. Frame Environment Layout Showing Active Control Workstation

rs232 Active1	Status of the RS-232 Active1 communication with the control workstation. Yellow (RS-232 activity) or black (no RS-232 activity) or gray (RS-232 error).
rs232 Active2	Status of the RS-232 Active2 communication with the control workstation. Yellow (RS-232 activity) or black (no RS-232 activity) or gray (RS-232 error).
controllerTail *X*	The monitor is connected to the frame supervisor via S1 or S2. For whichever controller tail is used, green (active) or red (inactive).

Starting Up and Shutting Down the SP System

The sequence files need to be kept in sync between the primary and backup control workstations in order for the **cshutdown** command to work on the backup control workstation. The following files need to be synchronized:

- **/etc/cstartSeq**
- **/etc/shutSeq**
- **/etc/subsysSeq**

The High Performance Switch

When a High Performance Switch fault occurs during a control workstation failover you should check to see if connectivity across the High Performance Switch is still working. If switch connectivity is not working, you should run **rc.switch** on all the SP nodes. When this double failure occurs (a very low probability) there may be updates to the control workstation for the switch subsystem that are not made. The

switch subsystem will stop working until an **Estart** is invoked from the primary switch node. There are instances when, depending on exactly when the control workstation initially failed and the backup took over as the control workstation, an **Estart** will not start the switch subsystem correctly. In such a case **rc.switch** needs to be run on all the SP nodes. You can try an **Estart** first and if it is not successful, run **rc.switch** on every SP node.

We recommend that you use an Error Notification Object for switch faults and control workstation failures to send mail or a message to alert you about such double failures. You can then determine if the switch subsystem is working correctly. Because the timeframe for this problem is very small, we do not recommend that you run **rc.switch** or **Estart** whenever a failover occurs.

Parallel Management Commands

You can use parallel management commands with the backup control workstation as a target, even when the backup control workstation is inactive, as long as the host name of the backup control workstation is listed in the working collective or host list.

Managing SP Resources

When a failover or reintegration occurs, you need to make a decision about the currently running parallel jobs on the system. You can either kill the running jobs (**suspend** won't work) and restart the Resource Manager or wait until all the running jobs complete and restart the Resource Manager. New parallel jobs cannot start until the Resource Manager is restarted.

Note: This restriction will be removed in a future release of the PSSP.

Managing User Accounts

We highly recommend that HACWS configurations use NIS for user management when large numbers of users need to be managed on the SP system. NIS enforces a master-slave relationship for password databases and is easier to manage than file collections for user management in an HACWS configuration.

Using NIS with HACWS

If you are using NIS in an HACWS configuration and the NIS master database is the primary or backup control workstation, there must be at least one slave database within the NIS domain. For high availability of an SP system, we recommend that all the nodes where users log in be NIS slave database nodes.

Using File Collections with HACWS

If you use file collections for user management, the **/etc/passwd** and associated files should be on the primary control workstation. Once a failover occurs and the backup control workstation is the active control workstation, the actual master **passwd** files are no longer available to the backup control workstation. For this reason changing your passwords, gecos information, and login shell is not allowed when the backup control workstation is active. (This is similar to the restriction with NIS when the NIS master is down.) The backup control workstation, when inactive, will act as a file collection client (similar to an SP node) and update its **passwd** files from the primary control workstation when it is active.

Controlling User Login

If you are using login control on your SP system, the **spacs_data** must be kept in sync between the primary and backup control workstations. The **block_usr_sample** script and any derivatives of it need to be kept in sync between the primary and backup control workstations.

Managing NTP

Both control workstations must use the same source for time (for example, timemaster, Internet, or consensus). NTP should be running on both the inactive and active control workstations.

Managing File Collections

The inactive control workstation acts as a file collection client (**supper** is run via **cron**). This keeps all the file collection files and other default file collections in sync between the primary and backup control workstations. If a configuration or file collection master change is done on the active control workstation, you should perform a **supper update** for the particular file collection. Note that you can do a **supper update** for all the file collections. If the update is not performed after the change and a failover occurs, the updates will not be on the new active control workstation and such updates will not be available to the SP nodes.

Managing Amd

Include the backup control workstation in the distribution of Amd configuration and map files.

Managing Print Service

If the primary control workstation is the print host and a failover occurs, new print jobs will be sent to the active control workstation. Any print jobs in the queue on the previously active control workstation will not be printed until that workstation is booted again.

If the printer is directly attached to the primary control workstation, the backup control workstation also needs to have a printer connected to it. You may choose to send the print jobs from the backup to the primary control workstation via normal **lpd** print queueing. In this case no print jobs will be printed until the primary control workstation is booted again.

Managing Mail Service

If the primary control workstation is the mail hub, you should use NFS to export the **/var/spool/mail** file system to the backup control workstation so that mail files for nodes are put in the same file system. When a failover occurs you should have the **/var/spool/mail** locally mounted on the backup control workstation and the primary control workstation. Use NFS to mount the file on the control workstation when it is the inactive control workstation. The **/var/spool/mail** file system must be an externally mounted file system.

Accounting

If accounting was enabled in the Site Environment Information SMIT panel and the node number **0** was selected as the Accounting Master, you need to make a directory change. Create a symbolic link from both the primary and backup control workstations to put the **cacct** directory in the /**spdata**/**sys1** file system.

Do the following on both the primary and backup control workstations after accounting has been enabled:

1. Enter:

   ```
   ln -s /spdata/sys1/cacct /var/adm/cacct
   ```

   This creates a symbolic link between the **cacct** directory so that the data in the **rootvg** is automatically copied into the external file system when **crunacct** is run via **cron**. This allows accounting reports over a period of time to have no indication of which control workstation was active.

   On an inactive control workstation the **crunacct** and **cmonacct** scripts will fail because the symbolic link for the **cacct** directory on the inactive control workstation will not have a target directory.

2. To keep those scripts from failing, change the invocation in the **crontab** file to run a wrapper shell script that determines if the control workstation is the inactive control workstation. If the control workstation is the inactive control workstation, the script should exit without running the **cmonacct** or **crunacct** scripts. The **lshacws** command should be used for this. Alternatively, you may choose to let the script fail. The following is a sample script:

   ```
   #/bin/ksh
   # This is the state of the control workstation taking
   # the string output of the lshacws command.
   # It takes the name either crunacct or cmonacct
   # as an argument.

   HACWS_STATE=‘lshacws‘

   # Check to see if we are on an INACTIVE control workstation.
   # If we are, exit quietly, if not, run the argument passed.

   if [[ $HACWS_STATE = 1  ||  $HACWS_STATE = 16 ]] ; then
           exit 0
   else

   # Go run the command
   $1
   fi
   exit 0
   ```

Customizing HACMP/6000

To customize HACMP/6000, follow the basic steps for configuring a two-node rotating configuration with nonconcurrent access in the *AIX High Availability Cluster Multi-Processing/6000 Version 4.1 Installation Guide*.

Configuring Resources

When you get to the configuring resources step in the HACMP/6000 installation procedure, you need to define a resource group name. For an HACWS configuration, you **must** name the resource group **hacws_group1**. To define event scripts to handle failovers for HACWS configuration, the name of the resource group needs to be known. The HACWS configuration will not fail over if the **hacws_group1** resource group name is not used.

HACMP/6000 Event Processing

The HACWS software supplies HACMP/6000 pre- and post-event scripts for all HACMP/6000 events. These supplied pre- and post-event scripts are used to customize the HACMP/6000 configuration for HACWS. To permit you to add your own pre- and post-event scripts, HACWS software provides for pre- and post-event scripts in each pre- and post-event script that is supplied by HACWS. The following flow describes when the pre- and post-event scripts are called.

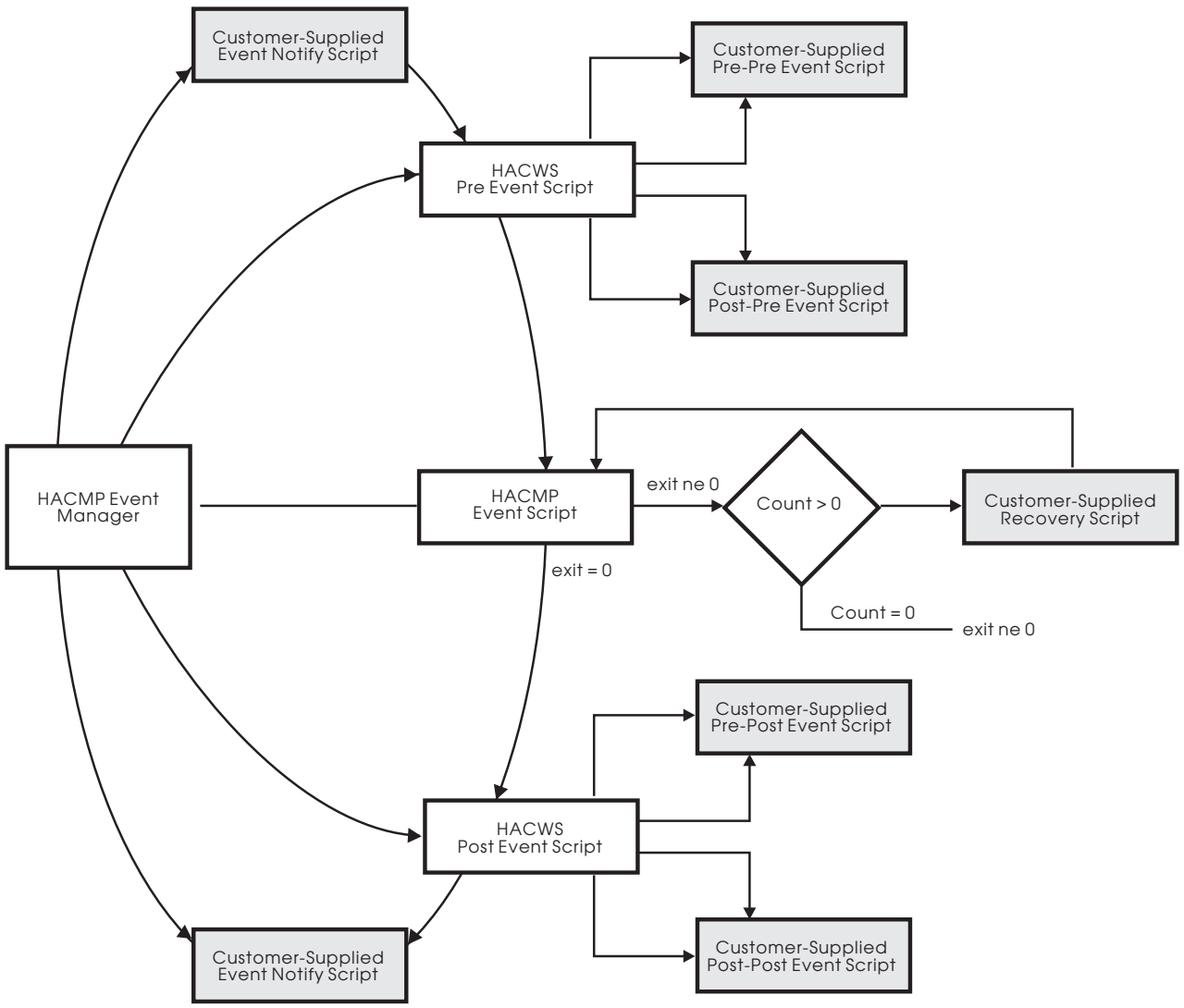


Figure 7. HACWS Pre- and Post-Events

The pre- and post-event scripts supplied with HACWS function are located in the **/usr/sbin/hacws** directory. These scripts are defined to HACMP/6000 with the

spcw_addevents command. Do not modify these scripts. These scripts are generic and will call the scripts supplied with HACWS (found in /**usr**/**sbin**/**hacws**/**events**) on a per event basis. Do not modify the per event scripts.

You may provide additional pre- and post-event scripts to be called by the pre- and post-event scripts supplied with HACWS. The additional scripts you provide must be located in the /**var**/**adm**/**hacws**/**events** directory.

Name the scripts you supply following this naming convention:

event_name.**pre_pre_event**

event_name is the HACMP/6000 event name and **pre_pre_event** is the name for a pre-event script for an HACMP/6000 pre-event script. This naming convention extends to having **post_pre**, **pre_post** and **post_post** event scripts.

To enable the execution of event scripts, place the scripts in the proper directories and turn on the execution mode bits via the **chmod** command.

Verifying Frame Supervisor Cables

In an HACWS configuration the tty assignments for each frame **must** be identical on both the primary and backup control workstations. The new frame supervisor cards (Type 22 and 20) used for HACWS configurations have flash memory on the cards. This flash memory is used to write a unique ID so the frame can be identified. With previous cards the order in which the frames were cabled determined the frame numbers in the configuration. There was no method to check that the frame was actually still the same frame number.

If a frame number on the card does not match the frame number in the SDR, the hardware monitor will not bring the frame online. An error log entry is written to the AIX error log and BSD syslog indicating this. Note that in most cases of miscabling, two frames will not be brought online. An error log entry will be written for each frame. You have to correct the cabling to repair the situation.

When the **hardmon** daemon initializes it verifies that the frame supervisor cables match the information in the SDR. To check that the cables are correct from an inactive control workstation (there must be an active control workstation because the SDR information needs to be compared):

1. Enter **stopsrc -s hardmon** to ensure there is no hardmon daemon running on the inactive control workstation.
2. Issue the **hardmon -B** command from the inactive control workstation.

Any cable problems will be written to the AIX error log.

When an SP system is reconfigured and frames moved around and reassigned, new frame IDs need to be written to existing frames. Note that the system must be completely reinstalled so node numbers and system images are correct. This action should rarely, if ever, be taken.

In the following example we list the steps to change the frame numbers in the flash memory of the card. In this example frame number 4 is now frame number 6 and frame number 6 is now frame number 4. A complete reinstall of the SP system is

also being done. Note that it would be far easier to make the cables match and not swap frames around. Swapping frames should be done only in extreme cases.

1. Determine the tty number of the cable that is connected to the frames.
2. Verify the SDR information about the frame.

   Ensure that the SDR information about the frame and tty number match how the frame is cabled. Use the **SDRGetObjects Frame** command to see which frame number matches with which tty number. Assuming that the SDR information is correct, proceed to the next step.

   If the tty number in the SDR is incorrect, use the **spframe** command to make frame number 6 use the correct tty special device name. For example, if you want to make frame 6 use **/dev/tty6** (assuming no other frame is using **/dev/tty6**), enter **spframe 6 1 /dev/tty6**.
3. Set the frame ID in the flash memory.

   Enter **hcmds -G setid 4,6:0** to set the frame ID to the values in the SDR frame class for frames 4 and 6.

Chapter 6. Managing System Partitions

System partitioning is a method for organizing the SP system into groups of nodes for various purposes such as testing new software and creating multiple production environments. In this chapter we present information on:

- Understanding system partitioning
- Partitioning the system
- Displaying system partition configuration information
- Verifying system partition configuration

You will also find information on system partitioning in the *IBM RISC System/6000 Scalable POWERparallel Systems: System Planning* and the *IBM RISC System/6000 Scalable POWERparallel Systems: Installation Guide*. For information on accounting and system partitioning, refer to "Accounting and System Partitioning" on page 270.

In this chapter we presume that you have installed the IBM Parallel System Support Programs for AIX (PSSP) software. In doing so you have automatically created a single default system partition which consists of all the nodes in your SP system. If you now wish to select an alternate system partition configuration, the information in this chapter will guide you in selecting, customizing, and applying a system partition configuration.

Understanding System Partitioning

Again, system partitioning is a method for organizing an SP system into groups of nodes that can be dedicated to specific purposes. A group of such nodes (not including the control workstation) is a *system partition*. For example, you may wish to configure a partition to use for non-disruptive testing when you migrate to a new release of software. Or you may wish to configure partitions for multiple isolated production environments so that the workload in one environment will not interfere with the workload in another environment.

System partitions appear to most subsystems and for most user tasks as logical SP systems. It is important to understand that from an administrative point-of-view, each system partition is a logical SP system within one administrative domain.

On the control workstation, the administrator is in an environment for *one system partition at a time*. The SP_NAME environment variable identifies the system partition to subsystems. (If this environment variable is not set, the system partition is defined by the `primary:` stanza in the **/etc/SDR_dest_info** file.) Most tasks performed on the control workstation that get information from the SDR will get the information for *that particular system partition*.

In managing multiple system partitions it is helpful to open a window for each system partition. You can set and export the SP_NAME environment variable in each window and set up the window title bar or shell prompt with the system partition name. The following script is an example:

© Copyright IBM Corp. 1995

```
sysparenv:
# !/bin/ksh
  for i in 'splst_syspars'
  do
     syspar='host $i | cut -f 1 -d"."'
     echo "Opening the $syspar partition environment"
     sleep 2
     export SP_NAME=$syspar
     aixterm -T "Work Environment for CWS 'hostname -s' - View: $syspar" -ls -sb &
  done
  exit

.profile addition:
# Added for syspar environment setup
  if [ "'env | grep SP_NAME | cut -d= -f1'" = SP_NAME ]
     then
        PS1="['hostname -s'<p>"$SP_NAME] ['$PWD]> '
     else
        PS1="['hostname -s']["'$PWD]> '
  fi
  export ENV
```

As a user, you can check what system partition you're in with the command:

```
spget_syspar -n
```

Commands and System Partitioning

Important Information

For commands that previously had an option flag to specify *all* nodes in the SP system, that option flag now means *all nodes in the current system partition*.

When you partition your system, you create one or more system partitions which, for most tasks, function as separate and distinct logical SP systems. Most commands function within the boundary of the system partition in which they are executed. A number of commands, however, continue to treat the SP as a single entity and do not respect system partition boundaries. That is, in their normal function they may affect a node or other entity outside of the current system partition. In addition, some commands which normally function only within the current system partition have been given a new parameter which, when used, allows the scope of that command to exceed the boundaries of the current system partition.

The following table summarizes those commands which can exceed the boundary of the current system partition. Unless otherwise stated, commands not listed in this table have as their scope the current system partition.

Command	Effect
arp	Can reference any node (by its host name) in any system partition.
Amd commands	Host names need not be in the current system partition.
crunacct	Merges accounting data from all nodes regardless of system partition boundaries.
cshutdown -G	The new **-G** flag allows specification of target nodes outside of the current system partition.

Command	Effect
cstartup -G	The new **-G** flag allows specification of target nodes outside of the current system partition.
dsh **dsh -w** {*hostname* | -}	Hosts added to the working collective by host name need not be in the current system partition.
dsh -aG	The new **-G** flag modifies the **-a** flag (all nodes in the current system partition) by expanding the scope to all nodes in the entire physical SP system.
Eclock	There is a single switch clock for the SP regardless of the number of system partitions.
Efence -G	The **-G** flag allows specification of nodes outside of the current system partition.
Eunfence -G	The **-G** flag allows specification of nodes outside of the current system partition.
hmcmds -G	The new **-G** flag allows the **hmcmds** commands to be sent to any hardware on the SP system.
hmmon -G	The new **-G** flag allows for the specification of hardware outside of the current system partition.
hostlist **hostlist -f** *filename* **hostlist -w** *hostname*	Host names need not be in the current system partition.
hostlist -aG | **-nG** | **-sG**	The new **-G** flag modifies the **-a**, **-n**, or **-s** flag by expanding the scope to the entire physical SP system.
hsdatalst -G	The **-G** flag causes the display of HSD information to be for all system partitions.
nodecond -G	The new **-G** flag allows specification of a node outside of the current system partition.
psyslrpt -w *hostnames*	The host names supplied with the **-w** flag can be in any system partition (the **-a** flag will select all nodes in the current system partition).
psyslclr -w *hostnames*	The host names supplied with the **-w** flag can be in any system partition (the **-a** flag will select all nodes in the current system partition).
penotify -w *hostnames*	The host names supplied with the **-w** flag can be in any system partition (the **-a** flag will select all nodes in the current system partition).
Parallel commands: • **p_cat** • **pcp** • **pdf** • **pfck** • **pexec** • **pexscr** • **pfind** • **pfps** • **pls** • **pmv** • **ppred** • **pps** • **prm**	Parallel commands can take the following options and will behave accordingly: **-w** Host names specified with **-w** need not be in the current system partition. *noderange* Nodes specified by *noderange* must be in the current system partition. *hostlist_args* Host names specified with **hostlist** options **-w** or **-G** need not be in the current system partition (any other **hostlist** options operate within the current system partition).
SDRArchive, **SDRRestore**	Archives/restores the SDR representing the entire SP.
SDRGetObjects -G	The new **-G** flag allows for retrieval of partitioned class objects from partitions other than the current system partition. Without the **-G**, objects which are in a partitioned class are retrieved from the current system partition only.
SDRMoveObjects	Moves objects from one system partition to another.

Command	Effect
Other SDR commands	SDR commands that create, change or delete values work within the system partition. Note though that System classes (Frame, for example) are shared among all system partitions. Changes to system classes will affect other system partitions.
Security commands: • **ext_srvtab** • **kadmin** • **kdb_destroy** • **kdb_edit** • **kdb_init** • **kdb_util** • **kdestroy** • **kinit** • **klist** • **kpasswd** • **kprop** • **ksrvtgt** • **ksrvutil** • **kstash** • **rcmdtgt** • **rcp** • **rsh** • **setup_authent**	The function of these security commands is unchanged under system partitioning. That is, if they previously affected the entire SP, they continue to do so even if the system has been partitioned. If they previously had the ability to affect a remote node (**rsh**, **rcp**, for example), that function is unchanged in a system partitioned environment.
spapply_config	Applies a system partition configuration to the entire SP.
spbootins	If a boot server outside of the current system partition is specified, that node is prepared appropriately.
spdelfram -G	The new **-G** flag allows the deletion of a frame containing nodes outside of the current system partition.
spframe	Configures data for one or more frames across the entire SP.
splm	The target nodes defined in the input table can include nodes from any system partition.
splstdata -G	The new **-G** flag allows display of information on nodes and adapters outside of the current system partition.
spmon -G	The new **-G** flag allows specification of nodes outside of the current system partition. The **-G** flag is required when performing operations on any frame or switch.
sprestore_config	Restores the entire SP SDR from a previously made archive.
spsitenv	Site environment variables are specified for the SP system as a whole. The specification of **acct_master=** can be any node in the SP regardless of system partition. The specification of **install_image=** may cause boot server nodes outside of the current system partition to refresh the default installation image they will serve to their nodes.
spverify_config	Verifies the configuration of all system partitions in the SP system.
supper	File collections are implemented and managed without respect to system partition boundaries.
sysctl	The Sysctl client can send requests to any node in the SP.
s1term	The new **-G** flag allows specification of a node outside of the current system partition.
vsdatalst -G	The **-G** flag causes the display of IBM Virtual Shared Disk information to be for all system partitions.

Although most commands and tasks are now performed at the system partition level, some SP system elements serve the entire system (that is, they span multiple system partitions):

- The hardware monitor provides the capability to control and monitor the entire system or a system partition
- There is only one Kerberos server for the entire system
- There is only one user name space for the entire system
- There is only one accounting master for the entire system
- The boot/install functions of a node ignore all system partition boundaries, but an AIX 3.2.5 boot/install server is required for AIX 3.2.5 system partitions and an AIX 4.1 boot/install server is required for all AIX 4.1 system partitions.

System partitioning is supported for a fixed set of configurations. (Refer to the *IBM RISC System/6000 Scalable POWERparallel Systems: System Planning* for specific information on supported configurations.) A directory contains the files for all supported system partitioning configurations for a given size SP system. The **ssp.top** option in the **pssp** install package provides the directories and files for system partitioning. This option must be installed on the control workstation if you are partitioning your system into more than one system partition.

Before You Define System Partitions

In the beginning, after you have installed the PSSP software, but before you have configured system partitions, there is one system partition corresponding to the name and IP address of the control workstation returned by the **hostname** command. This is the *default* or *persistent* system partition. It always exists. When you configure more than one system partition, one of them must be defined as the default system partition with this name and IP address.

When configuring more than one system partition, you must define alias IP addresses and names for all but the default system partition. To configure an alias IP address, you use the **ifconfig** command with the alias parameter to establish an additional IP address for the interface which the control workstation's **hostname** maps to.

Note: The alias must be defined on the same subnet as the interface that the control workstation **hostname** maps to.

You should also add alias names that map to these alias IP addresses through your name resolution mechanism. The alias IP addresses should be added to **rc.net** so that they are automatically configured when the control workstation reboots.

The steps for defining alias IP addresses and names are:

1. In the Part II - Traditional Configuration section of the /**etc**/**rc.net** file, a template is provided for changing **inet0** to an alias. Follow the template and edit the file to define alias IP addresses and names. (After editing the /**etc**/**rc.net** file, ensure that you have execute permission on the file or the control workstation will not boot correctly.) For example:

   ```
   /usr/sbin/ifconfig tr0 alias 129.40.127.101 netmask 255.255.255.0 up >>$LOGFILE 2>&1
   ```

2. Execute the /**etc**/**rc.net** script. Alternatively, you could enter the **ifconfig** command directly. For example:

```
ifconfig tr0 alias 129.40.127.101 netmask 255.255.255.0 up
```

3. Validate that the alias is defined correctly (that is, that the same interface is defined for the alias and the control workstation host name). For example, to list the IP addresses and aliases on your system:

```
netstat -in
tr0   129.40.127   129.40.127.43
tr0   129.40.127   129.40.127.101
```

Note: When you add an alias for the control workstation for the purpose of defining a name for a new system partition, you must run the **setup_server** command on the control workstation to properly define the new **rcmd** principal associated with the new host name alias.

Note: To delete the alias we defined in the previous example, the command is:

```
/usr/sbin/ifconfig tr0 delete 129.40.127.101
```

Remember to comment out or delete any entries you made to define aliases in the **/etc/rc.net** file.

In the next section we discuss how you select, customize, and apply a system partition configuration.

Partitioning the SP System

You partition your SP system by choosing a supplied configuration for your size system. The steps involved in partitioning an SP system are:

1. Archive the System Data Repository
2. Select a system partition configuration and layout to apply
3. Customize the system partitions
4. Shut down all nodes in changing system partitions
5. Apply the configuration
6. Reboot all nodes in changed system partitions

We refer to the above steps as *partitioning* the SP system. However, if you wish to change or add system partitions, that is, to *repartition*, the steps are the same. For example, if you wish to add a frame to an existing system partition or create a new partition for a new production environment, you would follow these steps to repartition the system.

To partition or repartition the system, you have the option of using SMIT panels or commands. (For more information, and further examples, of the commands used in this chapter, refer to the *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference*.)

Note

Some of the commands used in system partitioning (**spdisplay_config**, **splstdata -p**, **spapply_config -v**, for example) produce lengthy output. It is helpful to log the output of such commands. For example, to execute the **spapply_config** command while simultaneously logging a copy of the output to the file /**tmp**/**syspar.log**, enter:

```
spapply_config -v config.4_4_8/layout.3 2>&1 | tee /tmp/syspar.log
```

Using SMIT:

TYPE	**smit**
	▲ The System Management menu appears.
SELECT	9076 SP System Management
	▲ The 9076 SP System Management menu appears.
SELECT	9076 SP Configuration Database Management
	▲ The 9076 SP Configuration Database Management menu appears.
SELECT	Enter Database Information
	▲ The Enter Database Information menu appears.
SELECT	System Partition Configuration
	▲ The System Partition Configuration menu appears.

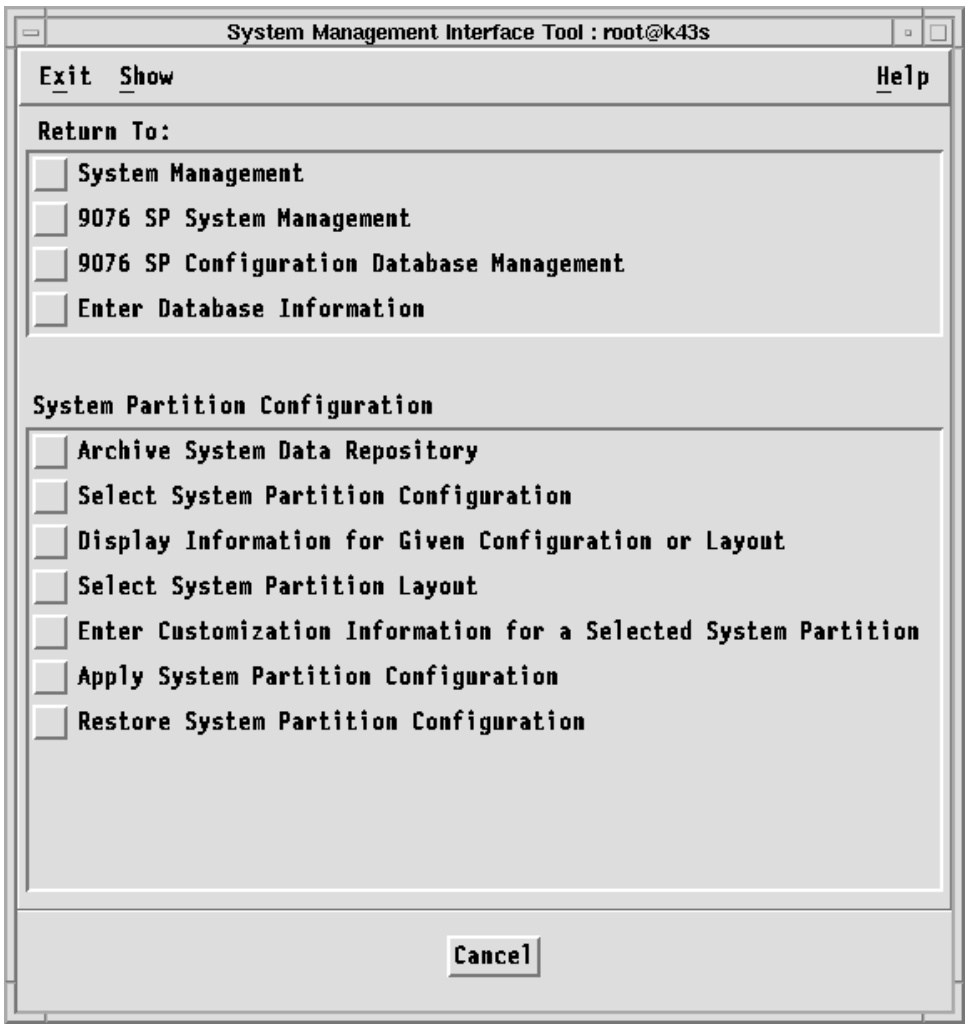


Figure 8. System Partition Configuration SMIT Panel

The fastpath invocation for the System Partition Configuration menu is:

```
smit syspar
```

Step 1: Archive the System Data Repository

Always archive the System Data Repository before partitioning (or repartitioning) your SP system. If you change your mind once you have committed a system partition configuration, or if applying a system partition configuration fails, you should use the archived SDR to recreate the previous system partition configuration. (To do this, select the Restore System Partition Configuration option from the SMIT menu.)

Using SMIT: At the System Partition Configuration menu:

SELECT Archive System Data Repository

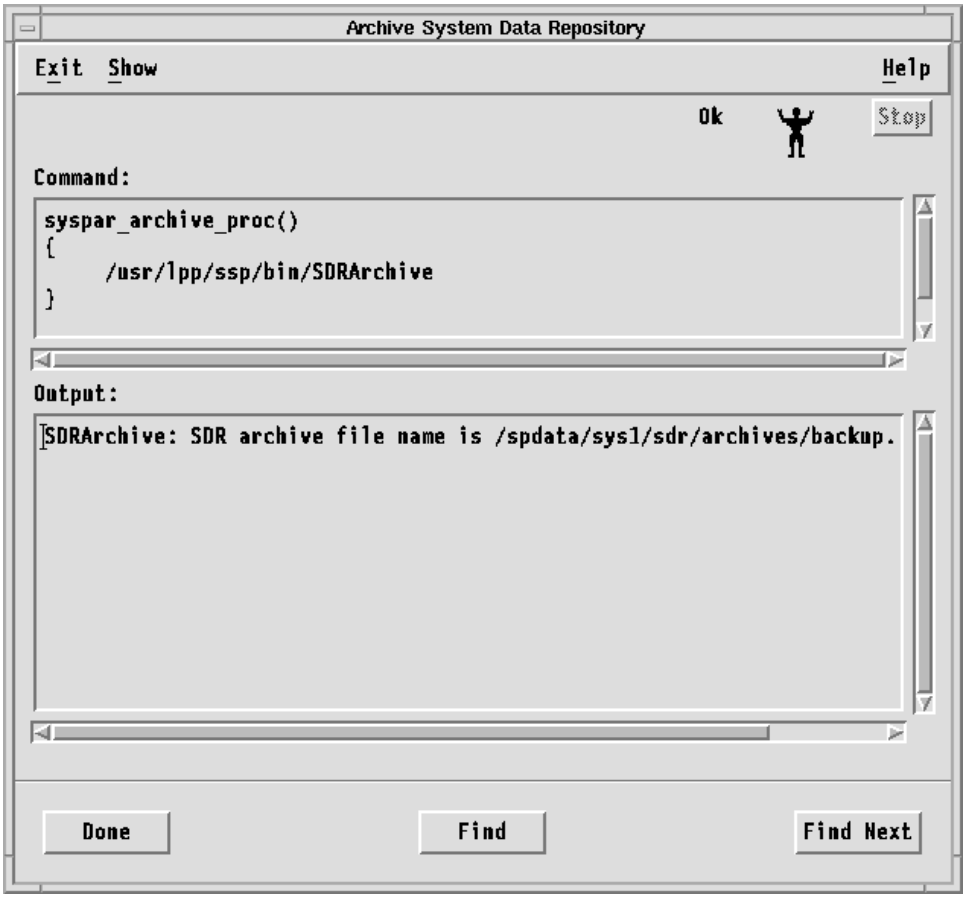


Figure 9. Archive System Data Repository SMIT Panel

Using the command line: Enter:

```
SDRArchive
```

Note: When you archive the SDR from either SMIT or by using the **SDRArchive** command, the archive produced is in **tar** format. Issuing **tar -x** for this archive does not restore the system partition configuration. To restore the system partition configuration, always use the SMIT Restore System Partition Configuration option.

Step 2: Select a System Partition Configuration to Apply

For detailed information on partition schemes for various size SP systems, refer to the chapter “Planning for System Partitions” in *IBM RISC System/6000 Scalable POWERparallel Systems: System Planning*.

To select a system partition configuration to apply:

1. Select a configuration from a list of supported system partition configurations for your SP system
2. Display layout information for the selected system partition configuration
3. Select a system partition layout

Select a System Partition Configuration

To select a supported system partition configuration for your SP system,

Using SMIT: At the System Partition Configuration menu:

SELECT Select System Partition Configuration

▲ A list of system partition configurations appears

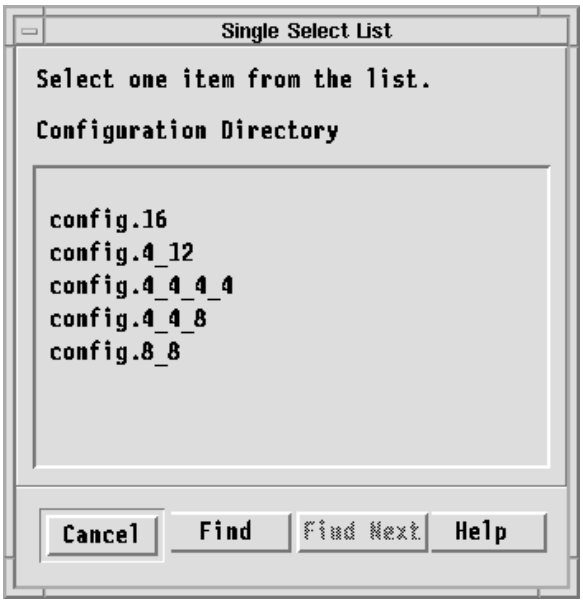


Figure 10. System Partition Configuration List SMIT Panel

SELECT A system partition configuration

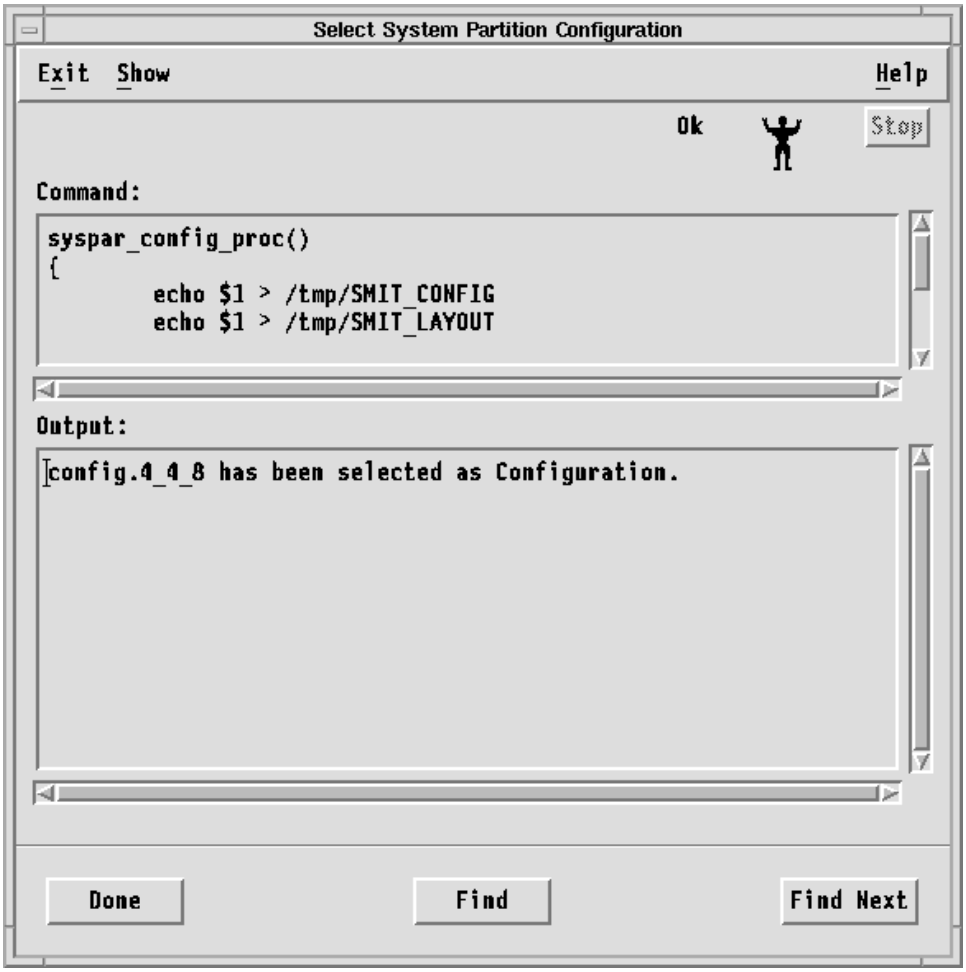


Figure 11. Select System Partition Configuration SMIT Panel

In this example we have selected a system partition configuration for our 16-node system of three system partitions consisting of four nodes, four nodes, and eight nodes respectively.

Using the command line: To display all supported system partition configurations for your SP system, enter:

```
spdisplay_config
```

Display Information for the Selected Configuration

To display information about the supported layouts for the system configuration you have selected,

Using SMIT: At the System Partition Configuration menu:

SELECT Display Information for Given Configuration or Layout

▲ The Display Information for Given Configuration or Layout menu appears

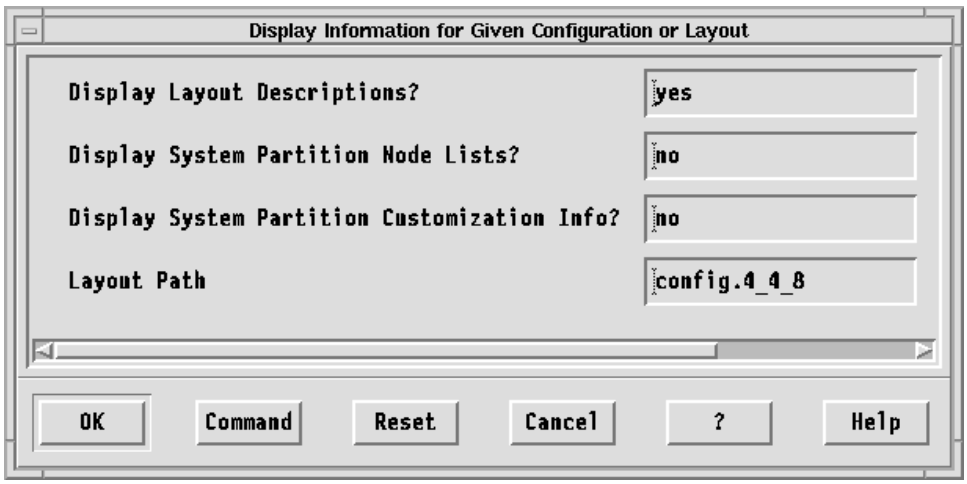


Figure 12. Display Information for Given Partition or Layout

SELECT OK

▲ System partition layout information for the chosen system partition configuration appears.

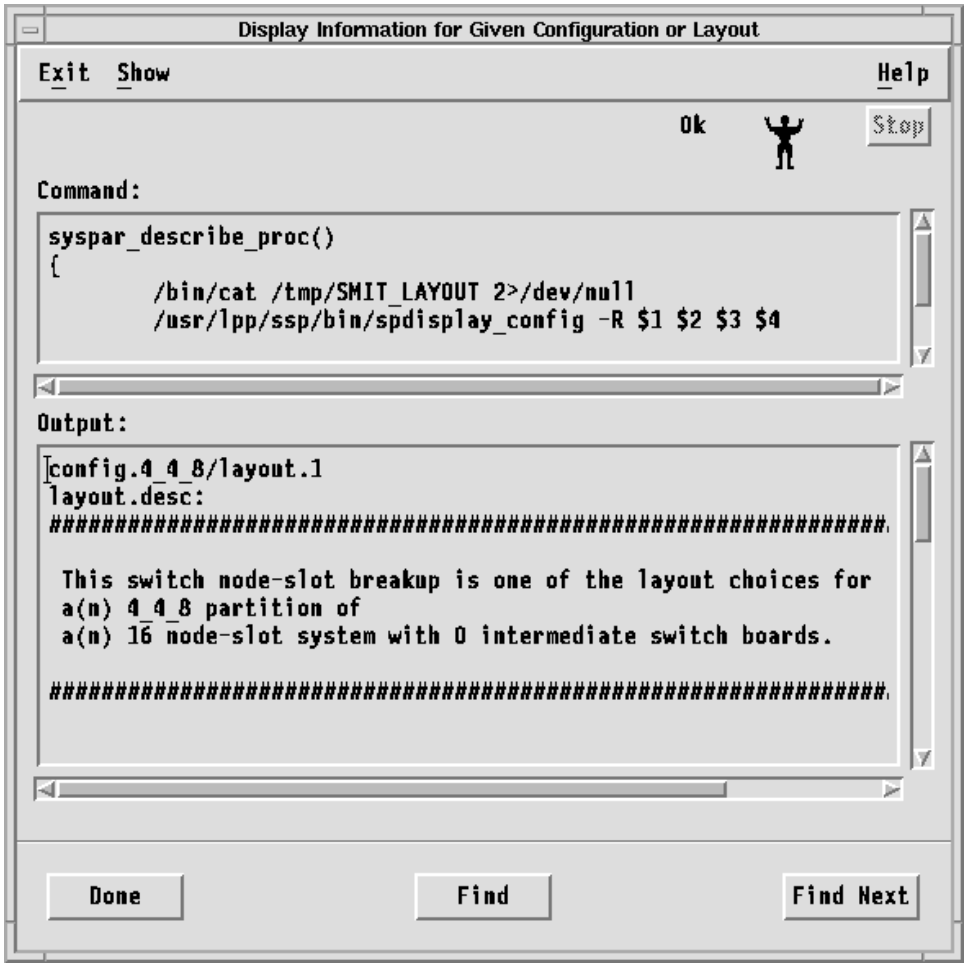


Figure 13. Display Information for Given Partition or Layout Output

(This example shows just the first screen of the layout information displayed for our selected 4-4-8 system partition configuration.)

Using the command line: To display information for the supported layouts for the 4-4-8 system partition configuration we've chosen, enter:

```
spdisplay_config -R -d config.4_4_8
```

Select a System Partition Layout

Based on the layout information you have displayed for the selected system partition configuration, select a layout,

Using SMIT: At the System Partition Configuration menu:

SELECT Select System Partition Layout

▲ A list of system partition layouts for the chosen system partition configuration appears.

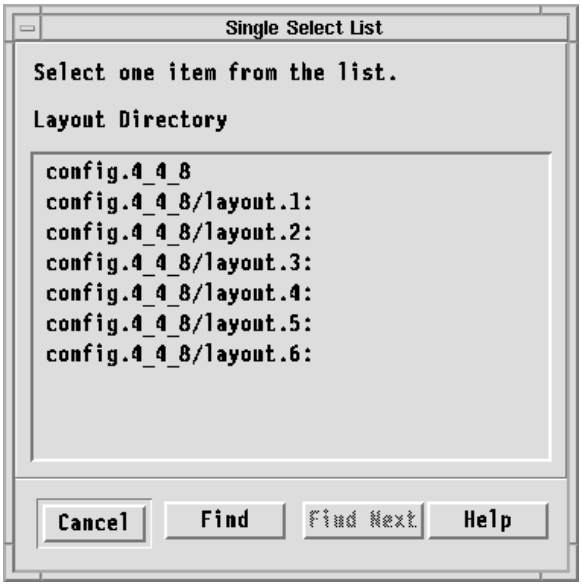


Figure 14. Select System Partition Layout SMIT Panel

SELECT A system partition layout

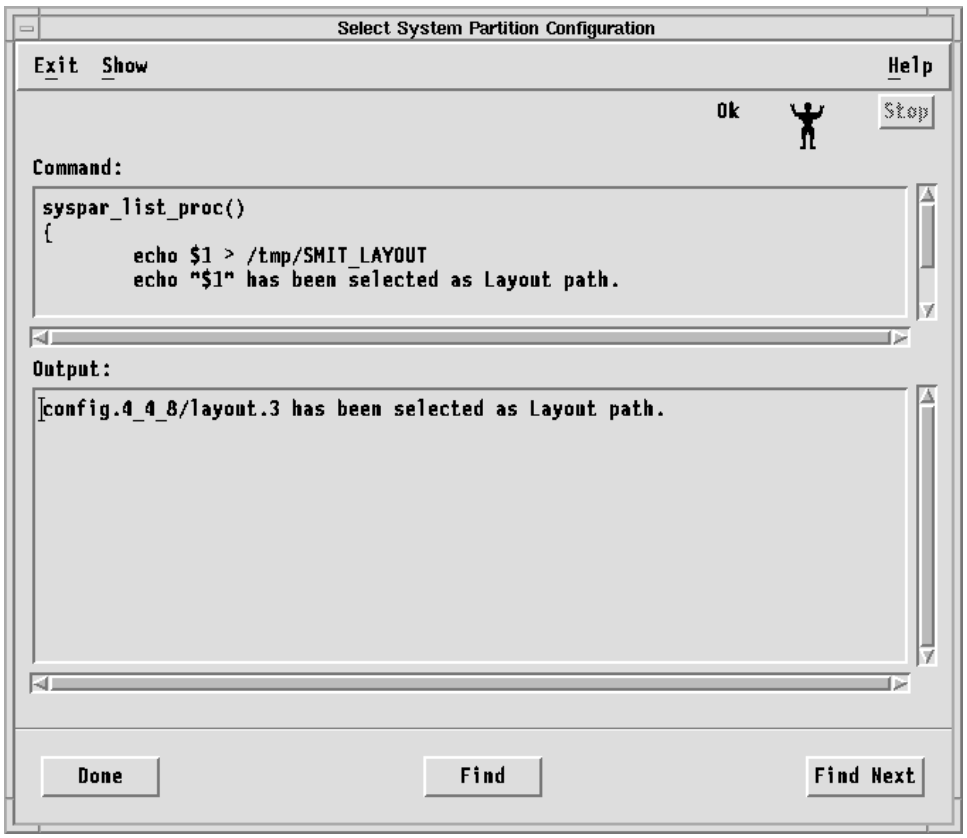


Figure 15. Select System Partition Layout Output SMIT Panel

In this example, we have chosen layout 3 for our selected 4-4-8 system partition configuration.

Step 3: Customize a System Partition

You customize a system partition by specifying information for:

- System partition name or IP address
- IBM Parallel System Support Programs for AIX code level
- Partition default installation image (optional)
- Primary node (optional)

The customization information is retained in the **custom** file for the partition. Because the customization information is retained in the **custom** file, the customization step would not have to be repeated when doing a reconfiguration if the same system partition is being used and the customization information does not need to be changed.

Using SMIT: At the System Partition Configuration menu:

SELECT Enter Customization Information for a Selected System Partition

▲ A list of the system partitions for the chosen configuration and layout appears.

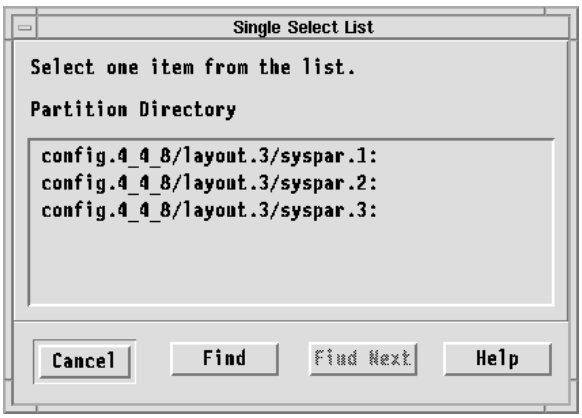


Figure 16. Partition Directory SMIT Panel

SELECT A system partition and enter customization arguments

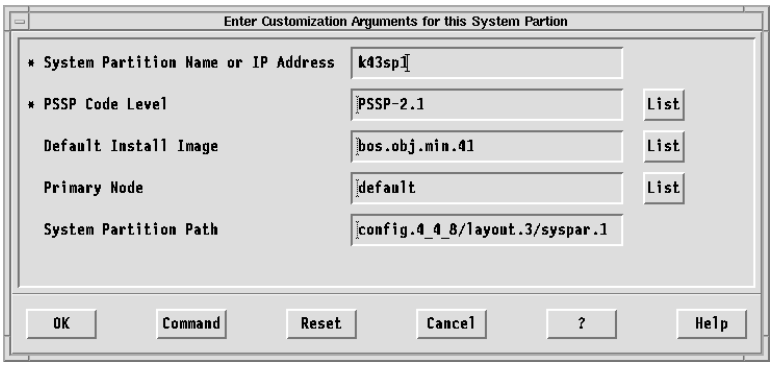


Figure 17. Enter Customization Arguments for this System Partition SMIT Panel

Note: If you have already customized this system partition this panel displays those values. Change the values as desired.

The customization arguments and their meanings are:

System Partition Name or IP Address
The name of the system partition is either the real host name (in the case of the default system partition) or the alias host name of the control workstation. The IP address of the system partition is either the real IP address (in the case of the default system partition) or the alias IP address of the control workstation.

PSSP Code Level
The IBM Parallel System Support Programs for AIX (PSSP) version and release, either **PSSP 1.2** (runs on AIX 3.2.5) or **PSSP 2.1** (runs on AIX 4.1).

Default Install Image
The name of the **mksysb** image that is used to install a node in this system partition (if the node itself doesn't have an installation image value).

Primary Node The node on the High Performance Switch that initializes, recovers, provides diagnosis, and performs other operations to the switch network. The default is the first node in the **nodelist** file.

System Partition Path
The full path name for the configuration, layout, and system partition name.

Using the command line: To list the system partitions for our chosen 4-4-8 system partition configuration, layout 3, enter:

```
spdisplay_config config.4_4_8/layout.3
```

To enter customization information (as shown in our example SMIT panel) for system partition 1 of our chosen 4-4-8 system partition configuration, layout 3, enter:

```
spcustomize_syspar -n k43sp1 -l PSSP-2.1 -d bos.obj.min.41\
-e default config.4_4_8/layout.3/syspar.1
```

Repeat this step, using the appropriate values, for each system partition in the configuration.

Step 4: Shut Down All Nodes in Changing Partitions

When you are partitioning or repartitioning the system, system partitions are considered the same if all of the following are identical:

- Node list files
- Topology files
- IBM Parallel System Support Programs for AIX code level
- Default install image
- IP address
- Partition name

If any of the above items are not identical, the system partition is being changed and nodes in that partition should be shut down before applying the system partition configuration.

Note: Modifying only the primary node will not result in a changed system partition. If you want to change the primary node, use **Eprimary** followed by **Estart**. See the **Eprimary** command in the *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference* for more information.

You can use the Verify option on the Apply System Partition Configuration SMIT panel to determine what nodes are changing. The output for the Verify option will list changing system partitions as well as any nodes in those system partitions that are not shut down.

Using SMIT: At the System Partition Configuration menu:

SELECT Apply System Partition Configuration

▲ The previously selected system partition configuration and layout appears

SELECT The system partition configuration and layout

▲ The Apply System Partition Configuration menu appears.

SELECT Verify only. from the list of System Partition Apply Options

SELECT OK

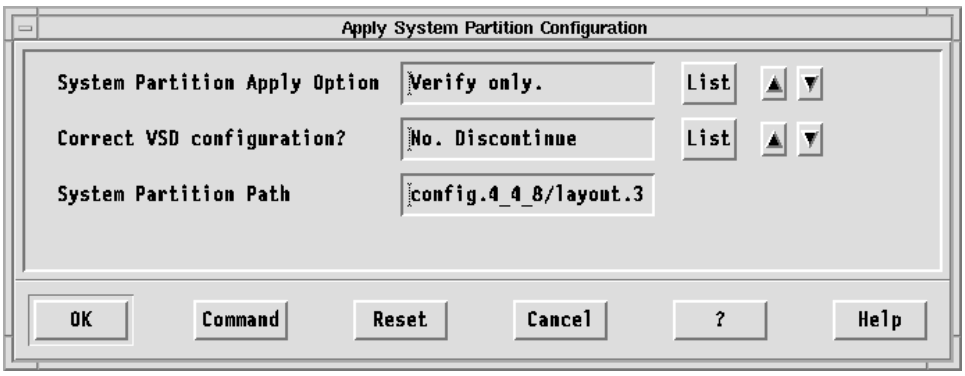


Figure 18. Verify System Partition Configuration SMIT Panel

The Apply System Partition Configuration arguments and their meanings are:

System Partition Apply Option
The options are provided to either only verify the system partition configuration or to actually apply the system partition configuration.

Correct VSD configuration?
The option to correct or discontinue when nonfatal errors are encountered in the IBM Virtual Shared Disk subsystem in the application of the specified system partition configuration.

System Partition Path
The full path name for the configuration, layout, and system partition name.

Using the command line: Enter:

```
spapply_config -v config.4_4_8/layout.3
```

Step 5: Apply a System Partition Configuration

When you apply or commit a system configuration, the following occurs:

- The existence of all required files (**nodelist**, **topology**, **custom**) for each system partition in the specified layout directory is verified. In addition, the contents of the **custom** file is checked for completeness.
- Any new system partitions are created on the control workstation.
- The system partition object for each system partition is created in the SDR.
- Affected node objects in the SDR (**Node**, **Adapter**, **host_responds**, **switch_responds**) are moved to the newly created partition.
- New **hbd**, **hrd**, and **sdrd** daemons are created and started.
- The **Syspar_map** information in the SDR is set for each node slot in the partition.
- The **verparvsd** command is invoked to verify the IBM Virtual Shared Disk data for system partitioning. The command extracts all IBM Virtual Shared Disk data from nodes involved in the system partitioning and writes SDR commands to update the IBM Virtual Shared Disk data to an output file. The output file reloads the IBM Virtual Shared Disk SDR data into the correct new partitions. (Note that the **verparvsd** command can also be used prior to applying a system partition configuration to test the desired layout for the existing IBM

Virtual Shared Disk configuration defined in the SDR. See the *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference* for more details.)

- Old unused system partitions on the control workstation, along with their **hbd**, **hrd**, and **sdrd** daemons are deleted.
- Updated SDR information is checked for accuracy against the information in the supplied system partition configuration files.

To apply a system partition configuration,

Using SMIT: At the System Partition Configuration menu:

SELECT Apply System Partition Configuration

▲ The previously selected system partition configuration and layout appears

SELECT The system partition configuration and layout

▲ The Apply System Partition Configuration menu appears.

SELECT Apply this config.

SELECT OK

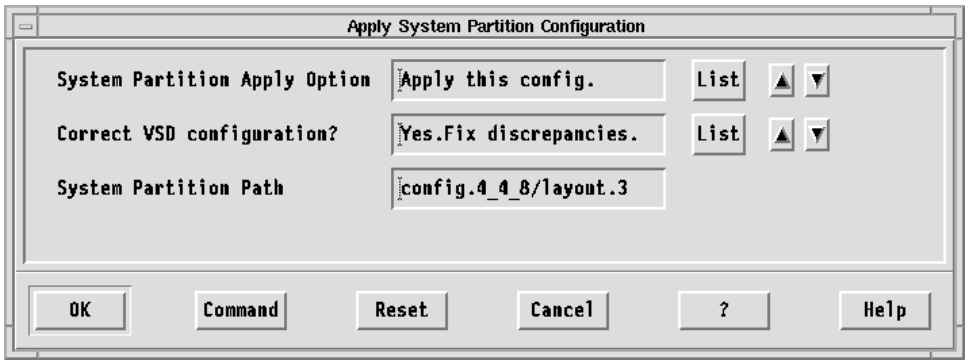


Figure 19. Apply System Partition Configuration SMIT panel

Using the command line: To apply our chosen 4-4-8 system partition configuration, layout 3, enter:

```
spapply_config -F config.4_4_8/layout.3
```

Important

If a failure occurs at this step, your system is put into an inconsistent state. Use the **sprestore_config** command or Restore System Partition Configuration SMIT menu option to restore the backed-up SDR and appropriate **hbd** and **hrd** daemons.

Step 6: Reboot All Nodes in Changed Partitions

Reboot any nodes that were shut down in “Step 4: Shut Down All Nodes in Changing Partitions” on page 110.

Displaying System Partition Configuration Information

You can display the following information for the system partitions in your current configuration:

- A list of the system partitions in your current configuration
- For each system partition in your current configuration:
  - System partition name
  - IP address
  - Install image
  - Layout directory name
  - IBM Parallel System Support Programs for AIX code version

Using SMIT:

TYPE	**smit**
	▲ The System Management menu appears.
SELECT	9076 SP System Management
	▲ The 9076 SP System Management menu appears.
SELECT	9076 SP Configuration Database Management
	▲ The 9076 SP Configuration Database Management menu appears.
SELECT	List Database Information
	▲ The List Database Information menu appears.
SELECT	List System Partition Database Information
	▲ The list of system partitions along with information for each of the system partitions appear.

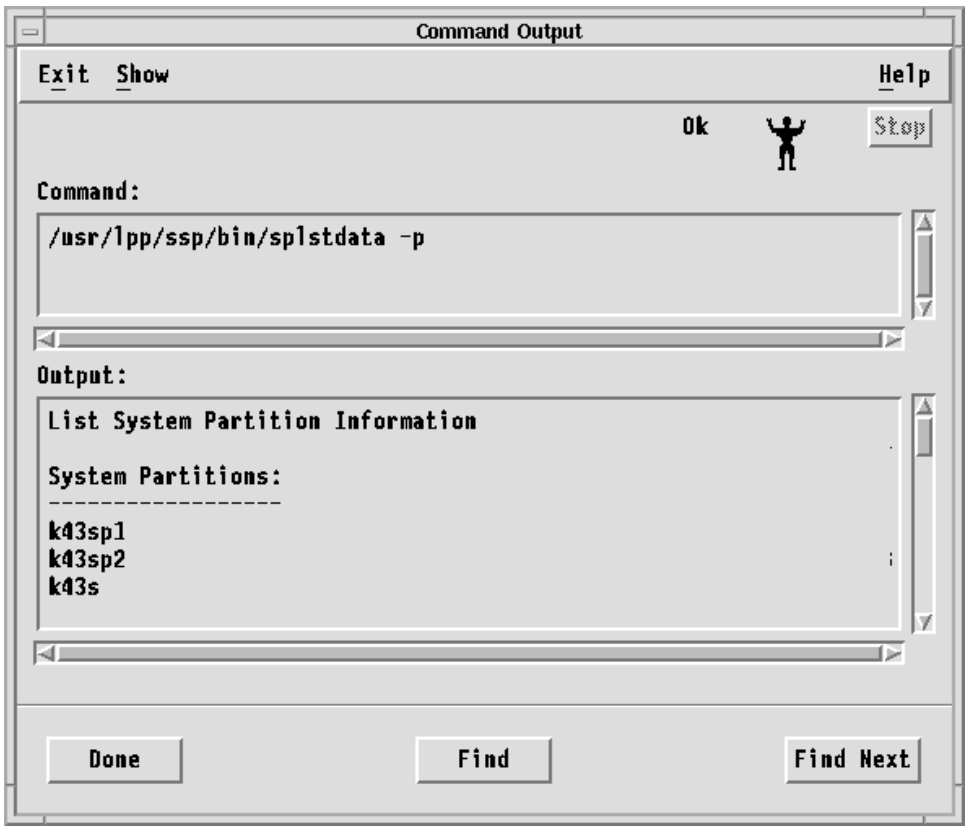


Figure 20. List System Partition Database Information Output

(This example shows just the first screen of the system partition configuration information for the system partition configuration we have applied.)

The fastpath invocation for the List System Partition Configuration Information menu is:

```
smit list_syspar
```

Using the command line: To display information for the system partitions in your current configuration, enter:

```
splstdata -p
```

Verifying System Partition Configuration

You can verify that the system partition information in the SDR is logically consistent.

Using SMIT

TYPE **smit**

▲ The System Management menu appears.

SELECT 9076 System Management

▲ The 9076 SP System Management menu appears.

SELECT 9076 SP Installation/Configuration Verification

▲ The 9076 SP Installation/Configuration Verification menu appears.

SELECT SSP System Partition Configuration

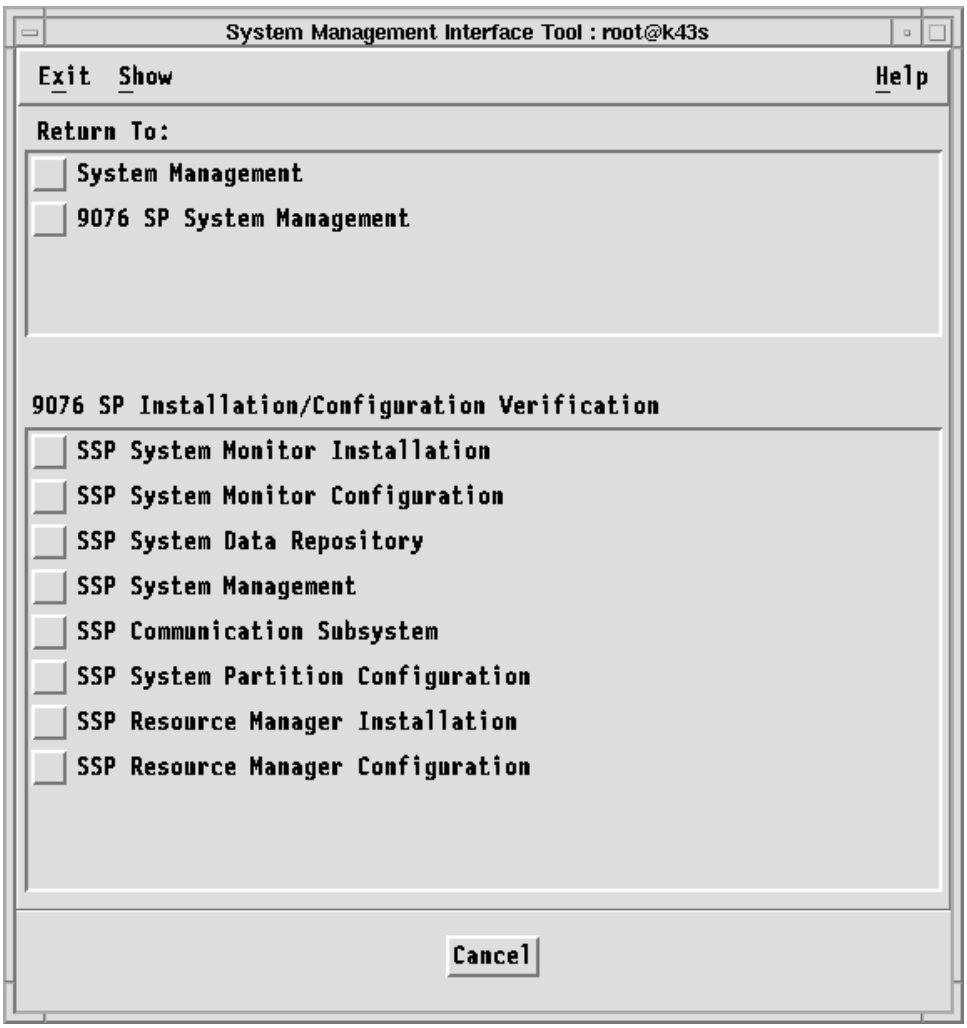


Figure 21. Verify SSP System Partition Configuration SMIT Panel

The fastpath invocation for the Verify SSP System Partition Configuration SMIT menu is:

```
smit SP_verify
```

Using the command line: Enter:

```
spverify_config
```

If the **spverify_config** command runs successfully, the SMIT status indicator will display **OK**.

Chapter 7. Using the SP System Monitor

The SP System Monitor provides a way for authorized operators and administrators to operate and monitor the system hardware. There are four main parts to the SP System Monitor:

Hardware Monitor A set of commands and a daemon used to monitor and control the SP hardware platform.

SP System Monitor GUI A graphical user interface to the Hardware Monitor that provides a point-and-click method to view the system status and interact with the frame, node, and switch controls. The SP System Monitor GUI provides the option of viewing an individual system partition or viewing the system globally (all system partitions).

Command interface The **spmon**, **hmmon**, **hmcmds**, **s1term**, and **nodecond** commands with their flags and parameters, provide a way to monitor and control the system at an individual system partition level or at a global (all system partitions) level. These can be used from an ASCII terminal, such as those used on a dial-up line, from scripts, or from another terminal.

Logging daemon The **splogd** logging daemon logs SP hardware state changes, reports SP hardware errors, and provides a state change alert.

See Figure 22 on page 118 for an overview of the SP System Monitor.

This chapter provides conceptual information and procedures for configuring and using the SP System Monitor. It also shows you how to operate the system controls and monitor the system activity using the SP System Monitor GUI.

© Copyright IBM Corp. 1995

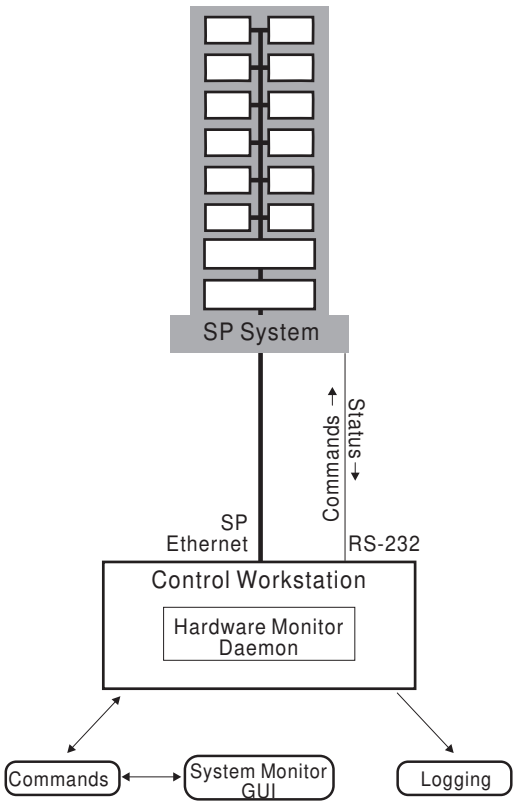


Figure 22. The SP System Monitor

Understanding the Hardware Monitor

The Hardware Monitor consists of a daemon, named **hardmon**, and a set of client commands. The daemon executes on the control workstation and, using the RS-232 lines to each frame, polls the frames for the state of the hardware within the frame. The daemon also sends hardware-level commands to each frame to change the state of the hardware within the frame in some way. The client commands **spmon** and **hmmon** interact with the daemon to obtain the current or changing state of the hardware. The client commands **spmon** and **hmcmds** also interact with the daemon to change the state of (that is, control) the hardware. The SP System Monitor Graphical User Interface issues the **spmon** command internally to control and monitor the hardware. The **splogd** logging daemon is also a client of the hardware monitor for logging purposes.

The **hardmon** daemon reports variables representing the state of the frame, node and High Performance Switch hardware. These state variables are made available through the **spmon** and **hmmon** client commands and through the SP System Monitor GUI. You can find a list of these variables in Appendix D, “System Monitor Variables” on page 415. This list provides variable names, display attributes and descriptions for all variables available in the SP System Monitor. The **hmmon** command, with the appropriate flag, also provides a list of variable names and descriptions. You will see these variable names in the SP System Monitor log. They also serve as parameters to the **spmon** and **hmmon** commands.

Frame Hardware State

The **hardmon** daemon reports variables to *monitor* the frame hardware status. This information includes:

- Switch failure
- Power LEDs
- Environment LEDs
- Temperature
- Voltage
- State of the power supply

The **hardmon** daemon allows the administrator to *control* the frame hardware state. This information includes:

- Power on/off

Node Hardware State

The **hardmon** daemon reports variables to *monitor* the node hardware status. This information includes:

- Three-digit display
- Power and Environment LEDs
- Temperature
- Voltage
- Fan failure

The **hardmon** daemon allows the administrator to *control* the node hardware state. This information includes:

- Power on/off
- Key Mode Switch
- Reset button

Switch Hardware State

The **hardmon** daemon reports variables to *monitor* the switch hardware status. This information includes:

- Temperature
- Voltage
- Fan failure
- Power and Environment LEDs

The **hardmon** daemon allows the administrator to *control* the switch hardware state. This information includes:

- Power on/off
- Change multiplexor clock setting

Authority

There are four classes of authority for the SP System Monitor:

1. Monitor authority permits users to display system statistics.
2. Control authority permits users to display system statistics and control the hardware.
3. S1 authority permits read and/or write access to a node's serial port.
4. Administrative authority permits a user to issue administrative commands to **hardmon**.

Configuring the SP System Monitor

The system administration tasks for configuring the SP System Monitor include authorizing users and configuring the error log.

Step 1: Authorize Users for the SP System Monitor

The SP System Monitor Access Control Lists (ACLs) are found in **/spdata/sys1/spmon/hmacls** on the control workstation. Edit this file if you wish to add users for your system. The **/spdata/sys1/spmon/hmacls** file is initially set up giving all levels of authority (including administrator) to the same user that is defined as the primary authentication services administrator by the **setup_authent** command (refer to Chapter 3, "Security Features of the SP System" on page 17 for more information). The **hardmon** principal is initially set up with monitor authority (for use by **splogd**).

The fields for each entry in the **/spdata/sys1/spmon/hmacls** file are

object name permissions

where:

object A frame number or hostname (where the **hardmon** daemon is running)

name A Kerberos principal name and optional instance

permissions

- **v** Virtual Front Operator Panel (VFOP). This gives permission to issue commands to the hardware.
- **s** Serial link. This gives permission to read and write to a serial port.
- **m** Monitor. This gives permission to receive state changes.
- **a** Administrative. This gives authority to control **hardmon**.

Refer to Chapter 3, "Security Features of the SP System" on page 17 for more information on security considerations.

Step 2: Configure the SP System Monitor Error Log

When the hardware supervisors indicate a warning or shutdown condition, the SP System Monitor writes a message using the AIX **syslog** facility and the AIX error log facility. For example, when the hardware supervisors determine that a fan has failed, the SP System Monitor writes a precise message into the log file that includes the time, node, type of error, variable name, and, in some cases, associated values.

The installation process creates the default syslog file **/var/adm/SPlogs/SPdaemon.log** on the control workstation. You may want to configure your system to send the system log information to other locations. For example, you may want to send the **SPdaemon.log** messages to another workstation for convenience. You can do this using the *@hostname* parameter in the **/etc/syslog.conf** file. For more details, refer to *IBM AIX Version 4.1 Files Reference*. The facility name for the SP System Monitor is **daemon**.

Customizing SP System Monitor GUI Default Resources

1. The default resources are in **/usr/lib/X11/app-defaults/Spmon**
2. To customize, add the contents of **/usr/lib/X11/app-defaults/Spmon** to the **.Xdefaults** file in your $HOME directory. **Note:** If you have any comments in the **.Xdefaults** file, add a : to the comment line.

Refer to "/usr/lib/X11/app-defaults/SPmon" on page 414 for a copy of this file.

Using the Command Line Interfaces

The SP system includes command line interfaces that allow you to perform the SP System Monitor functions without using a graphical user interface. With the **spmon** and **hmmon** commands you can query the value of a state variable or monitor that variable for a change in state. With the **spmon** and **hmcmds** commands you can control the nodes, frames and switches. The **hmmon** and **hmcmds** commands provide equivalent function to the **spmon** command but provide a more compact interface than **spmon** that can be appropriate in scripts. The **spmon** command is provided for compatibility with earlier SP software releases.

By default, these commands operate at a system partition level. That is, they work only on nodes *in the current system partition*. (To determine the current partition, issue the **spget_syspar** command.)

Note: Hardware Monitor queries or commands using a wildcard character do not return an error code if nodes outside of the current system partition are specified by the wildcard. Nodes outside of the current partition are treated as if they don't exist.

Each of the commands has a **-G** flag that removes system partition boundaries, allowing the command to work on any hardware in the SP system. The **-G** option is *always* needed to monitor or control frame or switch hardware. Frames and switches are considered to be outside all system partitions.

Before using any SP System Monitor command your identity must be authenticated. Use the **kinit** command to login to the SP authentication services. If your site uses an alternative to the SP authentication services, you must use authentication procedures appropriate to such an alternative. (Refer to Chapter 3, "Security Features of the SP System" on page 17 for more information on SP authentication services.)

See the *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference* for a complete description of all SP System Monitor commands.

Using the SP System Monitor GUI

The SP System Monitor GUI is a graphical user interface that allows you to view hardware objects at either a system partition level or across system partitions and operate the frame, node, and High Performance Switch controls. Using a point-and-click method, you can select from menus and icons to display information in a concise visual format. The SP System Monitor GUI windows use different

colors to indicate variations in status. The windows are updated as the state of the hardware changes.

SP System Monitor GUI Windows

The SP System Monitor GUI offers several ways for you to obtain monitoring information. You can get a high level overview of the system activity and status or just focus on one frame or slot (which may contain a switch or processing node). You have a similar choice when operating the controls and can initiate actions across the entire system, across one frame, or on just one slot.

Following is a list of the SP System Monitor GUI windows and an explanation of the information they display. The order in which you approach system monitoring depends on the task you are performing. Once you are logged in, you can start with any of these windows and move freely between them.

Topology Displays variables representing SP LANs (local area networks) connecting SP processor nodes. It uses colors to indicate the status of the adapters connecting the nodes to the LANs.

All Node Summary Display
Displays the state of one variable for slots 1–16. Allows you to monitor status of items such as the three-digit display or power LEDs for each slot in a single partition or across all partitions (Global view).

Display Layouts Displays windows for a particular frame, node, or the High Performance Switch. Allows you to monitor the status of several variables and interact with the controls.

Global Controls Provides access to the front panel controls of nodes. Allows you to issue actions on several nodes at once, within a single partition or across all partitions (Global view).

Dynamic Configuration

If a node is not created in the All Node Summary, Topology, or in the Global Controls display, or if a node or frame does not appear in a Display Layouts selection list, then there are two possible explanations:

- The variable is not valid for your hardware type. Not all variables are valid for both thin and wide nodes. (This only occurs on the All Node Summary Display or Topology display.)
- Some nodes were not included in the SP System Monitor configuration. Any SP frame that is not connected through RS-232 when the **hardmon** daemon is initialized will not be recognized in the system monitor configuration and its nodes will not be selectable. Once the RS-232 is connected and responds to the **hardmon** daemon poll, it will be added to the SP System Monitor configuration.

  If this is the case, the All Node Summary, Topology, and Global Controls windows must be closed and reopened for the entire configuration to be displayed. Display Layouts will become selectable for frames and nodes once they are included in the SP System Monitor configuration.

If all frames don't come up in the GUI, you may not be authorized for them. Refer to “Step 1: Authorize Users for the SP System Monitor” on page 120 for information.

Any active GUI sessions should be terminated before deleting a system partition. Failure to do so may result in an SDR connection failure within the GUI logic or in obsolete windows containing a view of the deleted system partition.

Window Elements

The SP System Monitor GUI windows contain elements common to many graphical user interfaces, such as menu bars, scroll bars, and radio buttons. If you are not familiar with these elements, refer to *IBM Getting Started: Using IBM RISC System/6000* for an explanation of how to manage the windows and interact with the common interface elements.

Interface Instructions

The tasks in the section are presented with step-by-step instructions. The instructions use a format that distinguishes between the user action and the system response.

User actions appear in uppercase bold type.

PRESS Cancel

Selections from a menu bar are indicated with an →.

SELECT SP → Display Layouts

The system response to an action follows the ▲ symbol.

▲ The diagnostic display appears.

Closing Windows to End the SP System Monitor GUI Session

In addition to the window menu button in the upper-left corner (provided by the window manager), every SP System Monitor GUI window has either a push button or a menu bar choice to close the window.

Starting the SP System Monitor GUI

Follow these steps to start the SP System Monitor GUI after logging in to the control workstation.

Note

If you are using a system other than the control workstation as the X-server, you need to export the SP System Monitor GUI. The following example shows you how to do it in **ksh**.

On the control workstation, set the DISPLAY to your server. Enter:

```
export DISPLAY=displayhost:0
```

displayhost is the name of the X-server.

On the X-server, enter:

```
xhost +controlhost
```

controlhost is the name of control workstation.

Step 1: Get Authentication to Use the Hardware Monitor.

If you have not already been authenticated to use the hardware monitor server, use the **kinit** command to login to the SP authentication system.

Refer to Chapter 3, “Security Features of the SP System” on page 17 for more information on authentication.

Step 2: Start the SP System Monitor GUI

From the control workstation:

TYPE **/usr/lpp/ssp/bin/spmon -g**

Note: For a Global view (all system partitions), type **/usr/lpp/ssp/bin/spmon -G -g**

▲ The SP System Monitor GUI main window (in this example, for system K43s, View for partition K43sp2) appears.

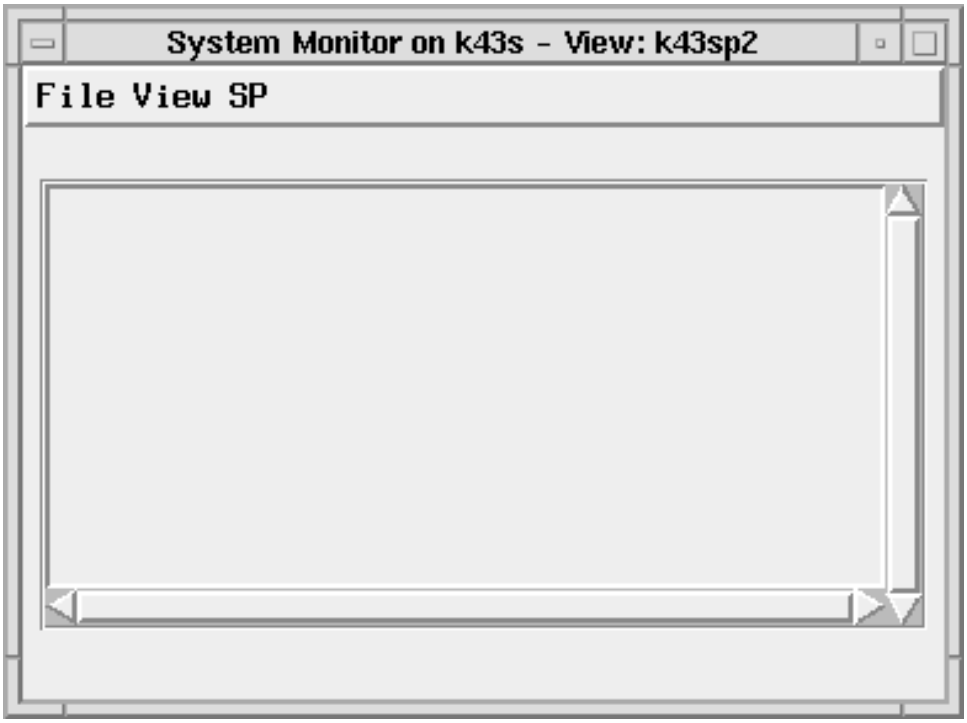


Figure 23. SP System Monitor GUI Main Window

The main window contains:

- A title that includes your control workstation name and the default view (either for an individual system partition or for a Global view) the SP System Monitor will display. If you have High Availability Control Workstation (HACWS) function installed, the title also includes the word 'primary' or 'backup' to indicate which control workstation is active.
- A menu bar with three choices, **File** (that allows you to exit your SP System Monitor GUI session), **View** (that allows you to specify an individual system partition or Global view), and **SP** (a pulldown menu that displays other window selections).
- A window area that echoes commands issued internally by the SP System Monitor GUI.

Step 3: Select View

SELECT **View**,

▲ A pulldown menu appears, shown in Figure 24 on page 126.

The pulldown menu lists the Global (All System Partitions) option and individual system partition names. The view option selected serves as the default view of the system.

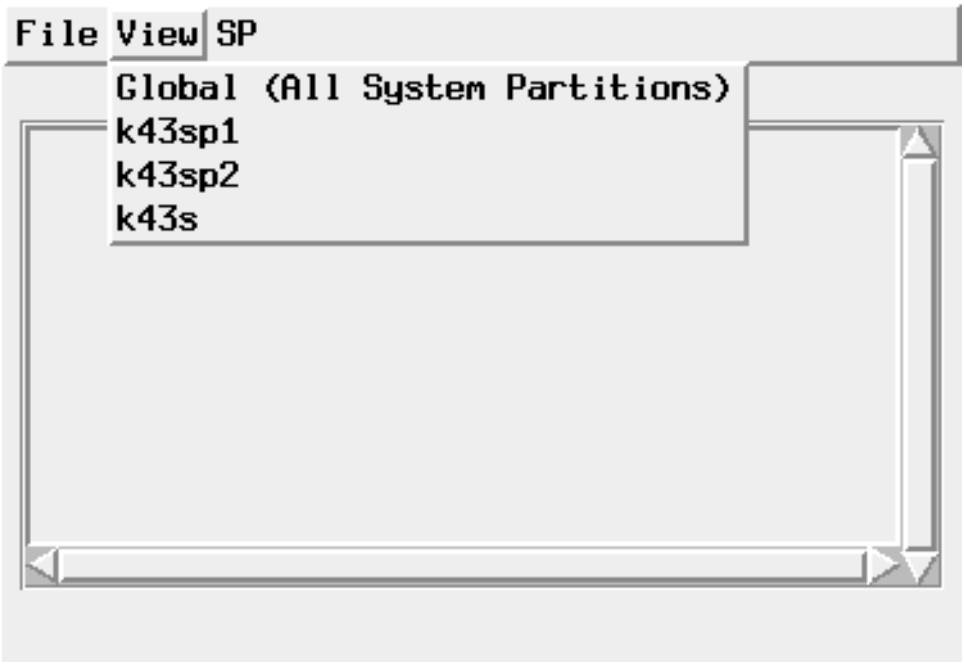


Figure 24. View Pulldown Menu

Step 4: Select SP

SELECT **SP**,

▲ A pulldown menu appears, shown in Figure 25.

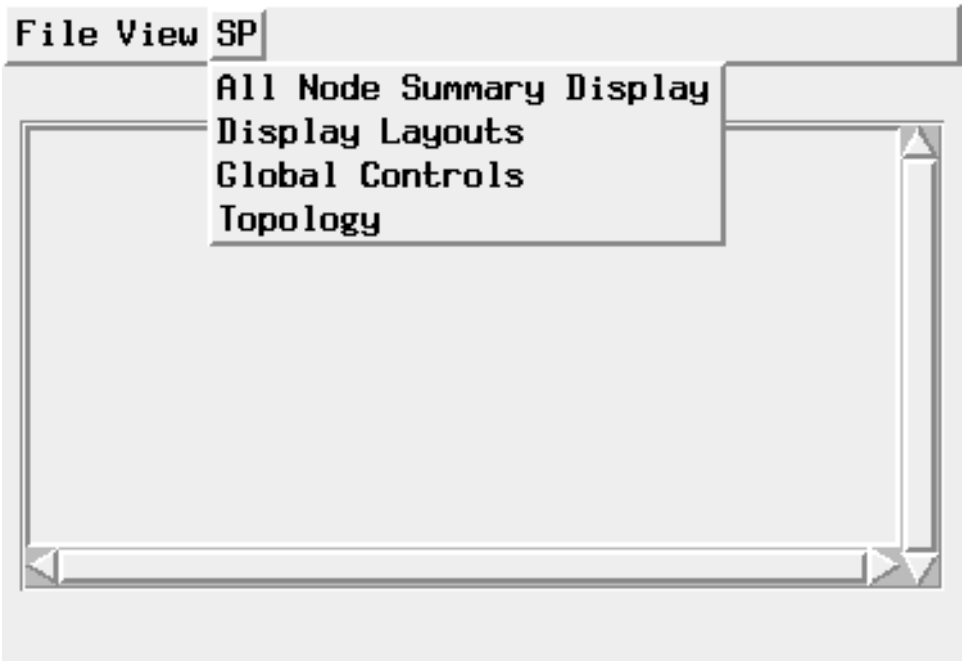


Figure 25. SP Pulldown Menu

Displaying Frame, Node, and Switch Layouts

Start at the menu bar on the SP System Monitor GUI to access the Display Layouts window.

From the SP System Monitor GUI Main Window:

SELECT SP → Display Layouts

▲ The display layout selection window appears.

SELECT View

▲ A pulldown menu lists defined system partitions and 'Global' (all system partitions) selections (in a multipartitioned system; if a system is not partitioned, 'View' won't appear on the menu bar)

SELECT The desired view of the system

SELECT The number of the frame or the number of the frame and slot in the option menu at the top of the screen

▲ The layout lists become active and you can select choices valid for the specified frame/node type.

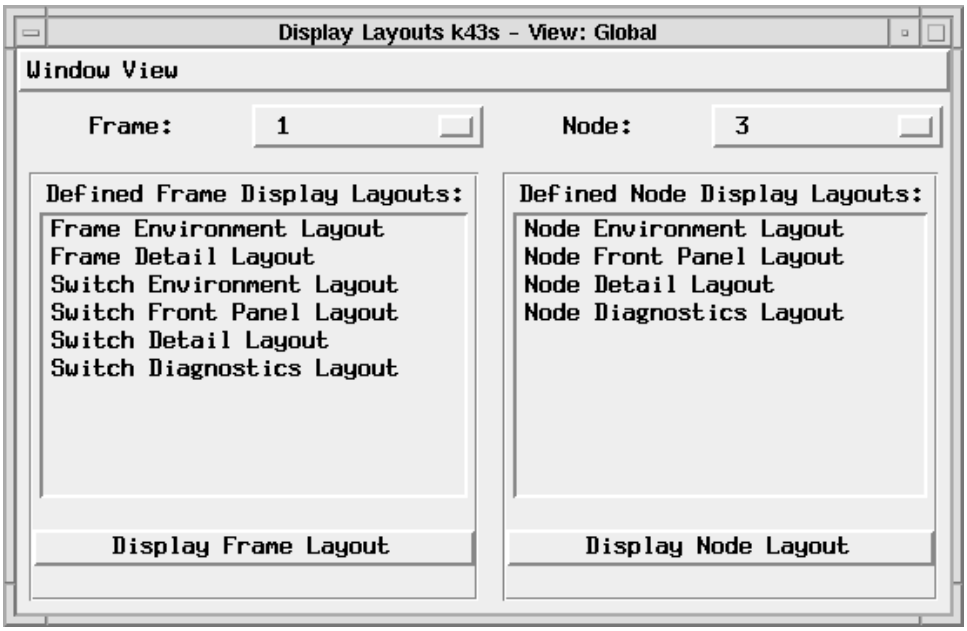


Figure 26. SP System Monitor GUI Display Layouts Selection Window

The Display Layouts window lists all the available layouts for the selected frame and node. Each layout is a window and displays a grouping of variables and controls. Layout windows provide you with detailed information about the hardware and, if you have the authorization, allow you to operate the system controls.

Note: On SP systems with more than one system partition, you must use the 'Global' view of the system to control frames and switches.

To select and display a layout:

SELECT The layout of your choice from one of the lists

PRESS Display Frame Layout or Display Node Layout

▲ The layout window of your choice appears.

Table 8 summarizes the information available on the layouts that come with the SP system.

Defined Layouts

Table 8. Frame, Node, and Switch Layouts

Select:	To get:
Frame Environment	Information about frame power and communications Control for the frame power supplies
Frame Detail	Information on type, code version, etc.
Switch Environment	Temperature and voltage indicators
Switch Front Panel	Power, environment, and communications indicators, etc.
Switch Detail	Information on type, code version, etc.
Switch Diagnostics	Information about the cause of switch failures
Node Environment	Temperature and voltage status
Node Front Panel	Three-digit display, power, environment, and communications indicators Controls for power, key mode switch, reset button, fencing, and opening tty
Node Detail	Information on type, code version, etc.
Node Diagnostics	Information about the causes of node failures

Monitoring and Controlling the High Performance Switch

You can use the Switch Front Panel layout to view the High Performance Switch indicators and monitor and control the High Performance Switch.

Note: On SP systems with more than one system partition, you must use the 'Global' view of the system to monitor and control the High Performance Switch. A 'Global' view is required to monitor and control switches residing in slots 1–16.

On a Frame with Nodes

At the Display Layouts window:

SELECT The number of the frame that the switch is on from the option menu

SELECT Switch Front Panel Layout

PRESS Display Frame Layout

▲ The Switch Panel for that frame appears.

On an All Switch Frame

The available layouts for an all switch frame will appear on the right side of the menu where node selections are displayed.

SELECT The number of the frame

SELECT The number of the slot with the switch you want to display

SELECT Switch Front Panel Layout

PRESS Display Node Layout

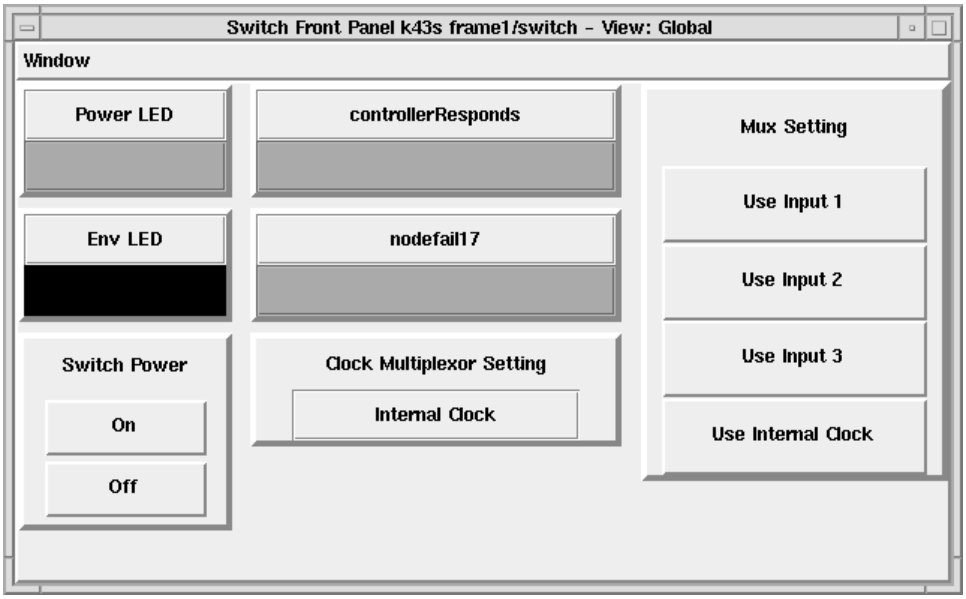


Figure 27. SP System Monitor GUI Switch Front Panel Layout Window

This window displays the following indicators:

Power LED — Status of the switch power light. Green (running), gray (off but power is available), or black (power not available).

Env LED — Status of the switch environment light. Black (environment light is off, no indication of failure), gray (environment light is flashing a failure), or yellow (environment light is on indicating an environment warning).

nodeFail*X* — *X* is the slot where the switch is. Indicates if the switch supervisor is responding to requests from the frame supervisor. Green (responding) or red (not responding).

controllerResponds — Indicates if the frame supervisor is responding to requests from the control workstation. Green (responding) or red (not responding).

Switch Power — Power controls for the switch.

Mux Setting — Used to select clock source at each frame. **Warning**: If you have more than one frame, use the **Eclock** command to select the clock source.

Clock Multiplexor Setting Reflects the current clock source selected.

Note: We recommend terminating all applications that use the switch before powering off the switch.

Operating Frame Environment Controls

The Frame Environment Layout window monitors the status of your system frame and allows you to turn the power on or off for the entire frame.

Note: On SP systems with more than one system partition, you must use the 'Global' view of the system to control frames. The number of indicators varies according to your frame supervisor. The Type 18/22 frame supervisors (Scalable Electrical Power Base Unit) have more indicators than the Type 17/20 (Power Distribution Unit) frame supervisors.

Warning

Powering off the frame power supply turns off the nodes and switch. Perform **Shutdown** before turning off the power supplies. See "Using Global Controls" on page 139.

Invoking the Frame Environment Layout

At the SP System Monitor GUI main window,

SELECT SP → Display Layouts

▲ The Display Layouts window appears. This window is shown with "Displaying Frame, Node, and Switch Layouts" on page 127.

SELECT The number of the frame you want to see

SELECT Frame Environment Layout

PRESS Display Frame Layout

▲ The Environment window for that frame appears.

Frame Environment Indicators and Controls

controllerResponds	Indicates if the frame supervisor is responding to requests from the control workstation. Green (responding) or red (not responding).
rs232 Active	Status of the RS-232 communication with the control workstation. Yellow (RS-232 activity) or black (no RS-232 activity) or gray (RS-232 error).
Bus Error	Status of communication with the nodes over the supervisor bus. Gray (no nodes responding) or yellow (1 or more nodes responding) or black (all nodes responding).
DC OK	Status of the DC power. Green (DC OK) or gray (DC Failure).

Type 17 (Power Distribution Unit) Frame Supervisor Indicators

AC OK	Status of the AC power. Green (AC OK) or gray (AC Failure).
Power_DC_x	Indicators for the four frame power supplies A–D. Green (OK) or red (Failure).
On/Off	Power control for frame power supplies A–D. **Warning**: When powering off a main PCI frame, other frames may also be powered off. See *IBM RISC System/6000 Scalable POWERparallel Systems: Site Planning*, SC23-3869, for more information on PCI.

Type 18 (Scalable Electrical Power Base Unit) Frame Supervisor Indicators

Although as many as four power supply modules may be installed in the Scalable Electrical Power Base Unit, all frame power is controlled with a single setting.

Modules Present	Power supply modules installed. Green (at least one power module is present) or gray (no power modules present).
ModulePowerX	Status of power supply modules installed. Green (OK) or red (Failure).
On/Off	Single power control for entire frame. **Warning**: When powering off a main PCI frame, other frames may also be powered off. See *IBM RISC System/6000 Scalable POWERparallel Systems: Site Planning*, SC23-3869, for more information on PCI.

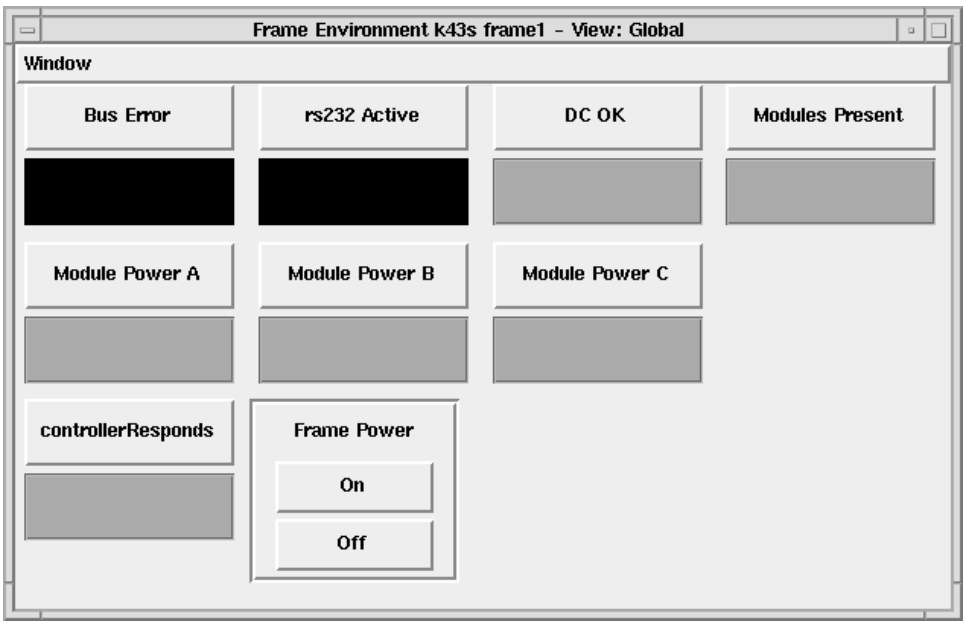


Figure 28. Frame Environment Layout for Type 18 Frame Supervisor

Type 20 Frame Supervisor Indicators

The Type 20 frame supervisor has the same indicators as the Type 17 (except for rs232 Active), along with the rs232 Active1, rs232 Active2, and controllerTail X as described for Type 22.

Type 22 Frame Supervisor Indicators

The Type 22 frame supervisor has the same indicators as the Type 18 (except for rs232 Active), along with:

rs232 Active1	Status of the RS-232 Active1 communication with the control workstation. Yellow (RS-232 activity) or black (no RS-232 activity) or gray (RS-232 error).
rs232 Active2	Status of the RS-232 Active2 communication with the control workstation. Yellow (RS-232 activity) or black (no RS-232 activity) or gray (RS-232 error).
controllerTail *X*	The monitor is connected to the frame supervisor via S1 or S2. For whichever controller tail is used, green (active) or red (inactive).

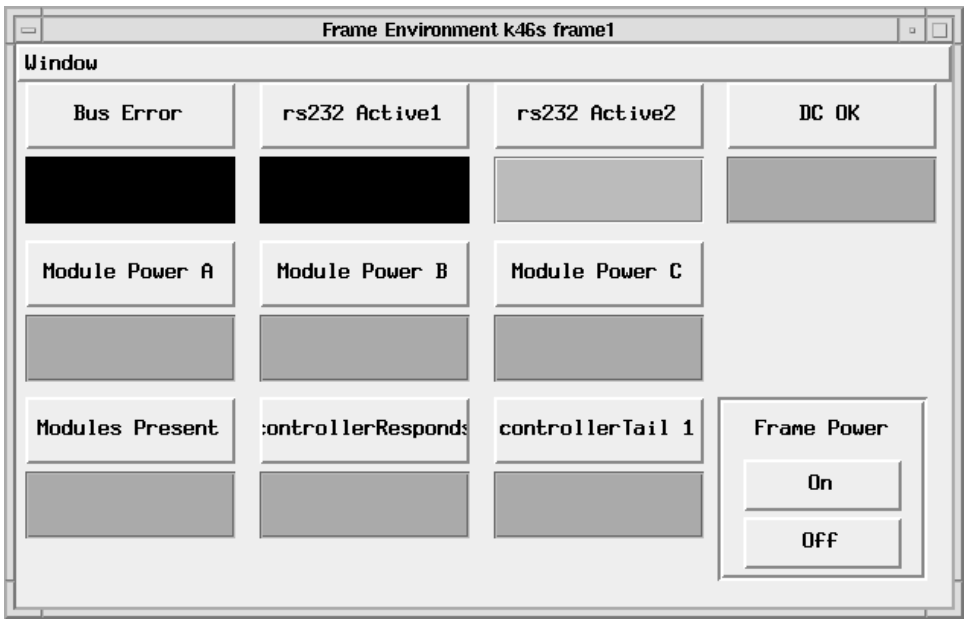


Figure 29. Frame Environment Layout for Type 22 Frame Supervisor

Operating the Front Panel Controls

If a processor node is having problems, you may want to shut it down, reset it, or prepare it for servicing. The node Front Panel window offers you the same controls that you would have on the RISC System/6000 front control panel.

You can also use Front Panel controls to isolate nodes from the High Performance Switch, in order to avoid a switch fault, if you want to shut down or reset nodes.

Invoking the Node Front Panel Layout

At the SP System Monitor GUI main window,

SELECT SP → Display Layouts

▲ The Display Layouts window appears. This window is shown with “Displaying Frame, Node, and Switch Layouts” on page 127.

SELECT The number of the frame you want to see

SELECT The number of the node within the frame

SELECT Node Front Panel Layout

PRESS Display Node Layout

▲ The Front Panel window for that node appears.

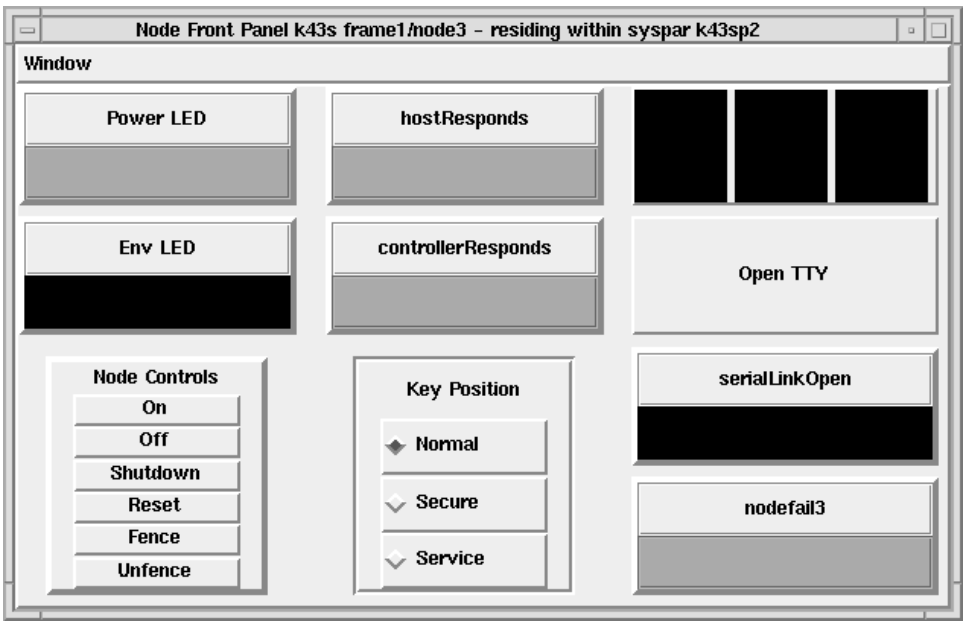


Figure 30. Node Front Panel Display

Node Front Panel Indicators

Power LED	Status of the node power light. Green (running), gray (off, but power is available), or black (power not available).
Env LED	Status of the node environment light. Black (environment light is off, no indication of failure), gray (environment light is flashing a failure), or yellow (environment light is on, indicating an environment warning).
controllerResponds	Indicates if the node supervisor is responding to requests. Green (responding) or red (not responding). Green (on) under normal operating conditions.
hostResponds	Status of communication between the processor node and the control workstation. Green (on) indicating node is up.

Three-digit display	Shows three-digit codes or black if nothing to display.
serialLinkOpen	Status of the serial port 1 (**tty** connection). Yellow (open) or black (closed).
nodeFail*X*	*X* is the slot where the node is. Indicates if the node supervisor is responding to requests from the frame supervisor. Green (responding) or red (not responding).

Node Front Panel Controls

On/Off/Reset	Power controls for the node, with Shutdown and Restart options.
Shutdown	System shutdown without powering off.
Fence	Isolates the node from the High Performance Switch
Unfence	Returns the node to the High Performance Switch
Normal/Secure/Service	Key mode switch controls for the node
Open TTY	Allows you to open an **aixterm** connection to the node

Turning Node Power Off or On

Be aware that turning off the power ends all processes running on the node and data may be lost. Perform a shutdown, either to individual nodes or globally to the SP system, before turning off power. Shutdown can be selected from the Front Panel and Global Controls display. (Refer to “Operating the Front Panel Controls” on page 132 or refer to “Using Global Controls” on page 139 for instructions on accessing these displays.)

Chapter 8, “Starting Up and Shutting Down the SP System” on page 149 describes how the **cshutdown** and **cstartup** commands work.

From the Node Front Panel:

CHECK The color of the power indicator.

- If the indicator is green, the power is on.
- If the indicator is gray, the power is available but not on.
- If the indicator is black, the power is unavailable to the node.

Turning the Power Off:

PRESS Off

▲ The Power Off menu is displayed.

Figure 31. Power Off Menu

PRESS Options:

Fence
: Isolates node from High Performance Switch

Enable AutoJoin (can be selected only in conjunction with Fence)
: Enables node to automatically rejoin the High Performance Switch when the node has restarted

Power Off
: Turns off node power.

Shutdown → Off
: Halts node, ignoring **cshutdown** dependencies, and turns off power

Cluster Shutdown → Off
: Halts node and turns off power only if **cshutdown** dependencies are met

▲ The power indicator changes to gray.

Turning the Power On:

PRESS On

▲ The Power On menu is displayed.

Figure 32. Power On Menu

PRESS One of the two options:

Power On
: Turns on power to nodes.

Cluster Power On
: Turns on power to nodes only if **cstartup** dependencies are met

▲ The power indicator changes to green and a three-digit display shows various codes while the node goes through its system start-up routine. When this completes without error, the three-digit display remains black, the hostResponds indicator turns green, and the node is ready (up).

Note: If you set *bootp_response* to `install` the powering process may take 20 minutes or more (because it is installing AIX and **pssp** on all of the nodes). If you set *bootp_response* to `customize` it will go faster. You can check the current value of *bootp_response* by querying the node class object in the SDR. Enter:

```
splstdata -b
```

See the *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference* for details on command syntax. Also, see the *IBM RISC System/6000 Scalable POWERparallel Systems: Installation Guide* for more information on the *bootp_response*.

If the system stops longer than three minutes with a three-digit code displayed, refer to information on diagnosing boot problems in the *IBM RISC System/6000 Scalable POWERparallel Systems: Diagnosis and Messages Guide* and the *IBM RISC System/6000 Problem Solving Guide* or the *IBM RISC System/6000 Scalable POWERparallel Systems: Maintenance Information Volume 1* to find out the meaning of the code and respond accordingly.

Resetting Nodes

The reset function allows you to boot the node when its key mode switch is in the Normal or Service position and to read error codes and diagnostic messages after a flashing **888** is displayed in the three-digit display. Be aware that resetting a node while it is operating can also result in corrupted or lost data.

From the Node Front Panel:

PRESS Reset

▲ The Reset menu is displayed.

Figure 33. Reset Menu

PRESS Options:

Fence
: Isolates node from High Performance Switch

Enable AutoJoin (can be selected only in conjunction with Fence)
: Enables node to automatically rejoin the High Performance Switch when the node has restarted

Reset
: Reboots the nodes

Shutdown → Restart
: Halts and reboots the nodes, ignoring **cstartup** dependencies

Cluster Shutdown → Restart
: Halts and reboots the nodes obeying **cstartup** dependencies

▲ The three-digit display shows various codes while the node goes through its start-up routine. When this completes without error the three-digit display remains black.

If the system stops longer than three minutes with a three-digit code displayed, refer to information on diagnosing boot problems in the *IBM RISC System/6000 Scalable POWERparallel Systems: Diagnosis and Messages Guide* and the *IBM RISC System/6000 Problem Solving Guide* to find out the meaning of the code and respond accordingly.

Shutting Down Nodes

The Shutdown and Cluster Shutdown options allow you to halt nodes. Be sure you have terminated all applications running on the node.

From the Node Front Panel:

PRESS Shutdown

▲ The Shutdown menu is displayed.

Figure 34. Shutdown Menu

PRESS Options:

Fence
: Isolates node from High Performance Switch

Enable AutoJoin (can be selected only in conjunction with Fence)
: Enables node to automatically rejoin the High Performance Switch when the node has restarted

Shutdown
: Halts the nodes, ignoring **cshutdown** dependencies and turns off power

Cluster Shutdown
: Halts the nodes and turns off power, obeying **cshutdown** dependencies

Isolating Nodes from the High Performance Switch

To use the Node Front Panel Display to isolate nodes from the High Performance Switch, take the following steps:

1. Select Fence
2. At the Fence Display, optionally select AutoJoin (Note that if you are doing Fence from Off, Shutdown, or Reset, and the Fence fails, the Off, Shutdown, or Reset will not execute.)
3. Select Fence

Figure 35. Fence Menu

Changing the Key Mode Switch Position

The key mode switch establishes the initial program load (boot) path and has three possible positions:

Normal
: The most common setting used for attended operation.

Secure
: Used for unattended operation to prevent unauthorized reset. The reset function is not active when the key mode switch is in this position.

Service
: Used for attended operation when hardware and software service is conducted. The reset function is active in this position.

  Note: To successfully boot in service mode, you need to open a **tty** console connecting to the control workstation.

From the Node Front Panel:

CHECK
: The key mode switch buttons to verify the current setting

PRESS
: The appropriate button for the setting you want.

Opening a tty Connection

You can open a **tty** connection to display a system console for system debugging. This connection opens /**dev**/**tty0** on the node.

From the Node Front Panel:

PRESS Open TTY

▲ A system console window appears.

Note

When using this **tty** connection, the appropriate terminal type is **aixterm.**

This uses the RS-232 and supervisor bus connections and competes with node controller traffic. It is not appropriate for general node access. Limit its use to debugging purposes only.

If you use more than one **tty** connection per frame, the buffer may be overrun resulting in lost data.

Using Global Controls

The level (within a system partition or across system partitions) at which commands issued from the Global Controls panel operate is dependent on the View of the system you've selected.

You can issue a command to operate the controls on several nodes at once. You can turn the node power on or off, reset nodes, or change their key mode switch positions. You can select the nodes by frame or individually, on a node to node basis. Any switches occupying slots 1–16 in a frame will be colored gray. Use the All Node Summary (“Step 1: Display the All Node Summary” on page 141) to display changing LED status. Refer to “Node Front Panel Controls” on page 134, which describes how each of these options works, with the exception of Network Boot, or to “Monitoring and Controlling the High Performance Switch” on page 128 as appropriate.

You can issue commands from the Global Controls window to isolate nodes from the High Performance Switch. Refer to “Isolating Nodes from the High Performance Switch” on page 141 for the steps to perform node isolation.

You can invoke a network boot from the Global Controls display panel. Network Boot causes booting over the SP Ethernet LAN, allowing installation, customization, or maintenance. You might find this useful after powering down a node, when the bootlist goes back to disk.

Refer to the *IBM RISC System/6000 Scalable POWERparallel Systems: Installation Guide* for more information.

Accessing the Global Controls Display:

Note

These choices can have wide ranging effects. Be cautious when using the global controls.

At the SP System Monitor GUI main window:

SELECT SP → Global Controls

▲ The Global Controls panel appears with all the slots deselected.

SELECT The slots to control

SELECT The action you want to perform

Choose one of the available actions such as On, Off, or Reset.

PRESS Do Command

▲ The SP System Monitor performs this action on all the selected slots.

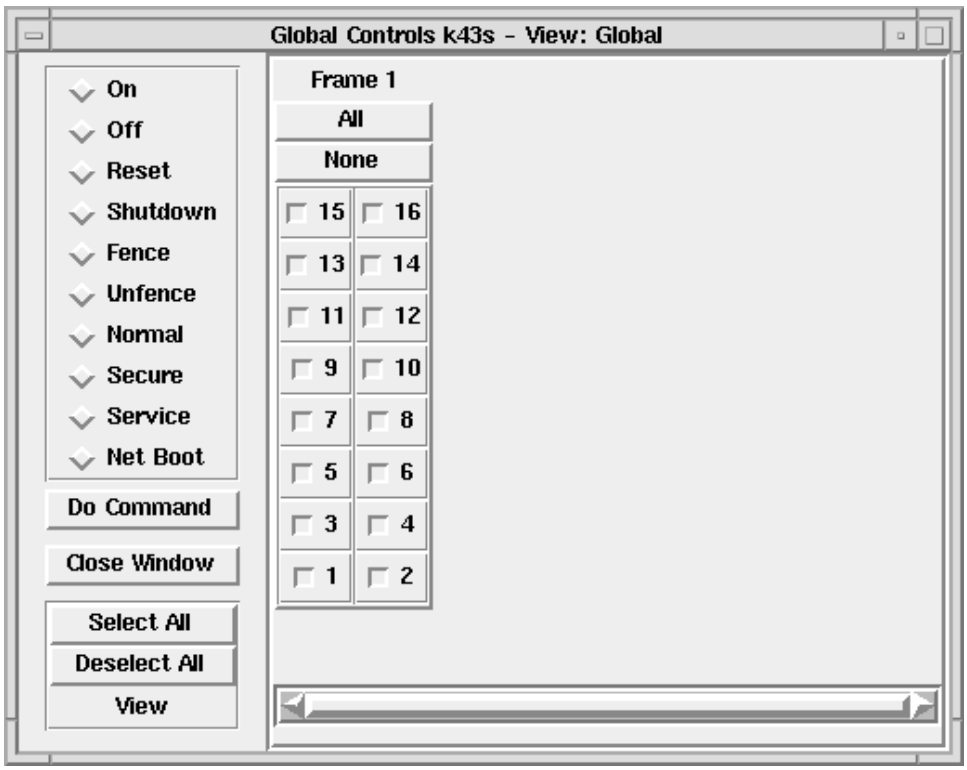


Figure 36. Global Controls Display

Frame 1 in this display contains sixteen thin nodes.

Note: The 'View' selection is available only for a multipartitioned system; if a system is not partitioned, 'View' won't appear. If 'View' is present and selected, a new Global Controls Display is opened for the selected view.

Isolating Nodes from the High Performance Switch

To use the Global Controls Display to isolate nodes from the High Performance Switch, take the following steps:

1. Select nodes
2. Select Off, Shutdown or Reset
3. Select Fence and, optionally, AutoJoin (Note that if you are doing Fence from Off, Shutdown, or Reset, and the Fence fails, the Off, Shutdown, or Reset will not execute.)
4. Select the type of Power Off, Reset, or Shutdown
5. If Enable AutoJoin was not selected, select Unfence to return nodes to the High Performance Switch once nodes have restarted.

For additional information on isolating nodes from the High Performance Switch, refer to Chapter 9, “The High Performance Switch” on page 161.

Determining If Nodes Are Operating: All Node Summary

With the SP System Monitor GUI you can monitor the system to determine, at a glance, if processor nodes are operating. Switches in slots 1–16 will also appear in an All Nodes Summary display, if you are in a Global view.

The All Node Summary provides you with a way to globally monitor the system and determine if nodes are operating. The View pulldown menu provides the option of viewing an individual system partition or viewing the system globally (all system partitions). From the All Node Summary window, you can select an SP System Monitor variable and display its values for nodes. Certain variables, such as *hostResponds*, the three-digit display, power, and environment lights, can provide you with system status.

Not all variables are valid for all node types. When a variable is not supported by the target node, that variable is blanked out. Refer to Appendix D, “System Monitor Variables” on page 415 to determine which variables your node type supports.

Step 1: Display the All Node Summary

At the SP System Monitor GUI main window:

SELECT SP → All Node Summary Display

▲ The selection window for node variables appears.

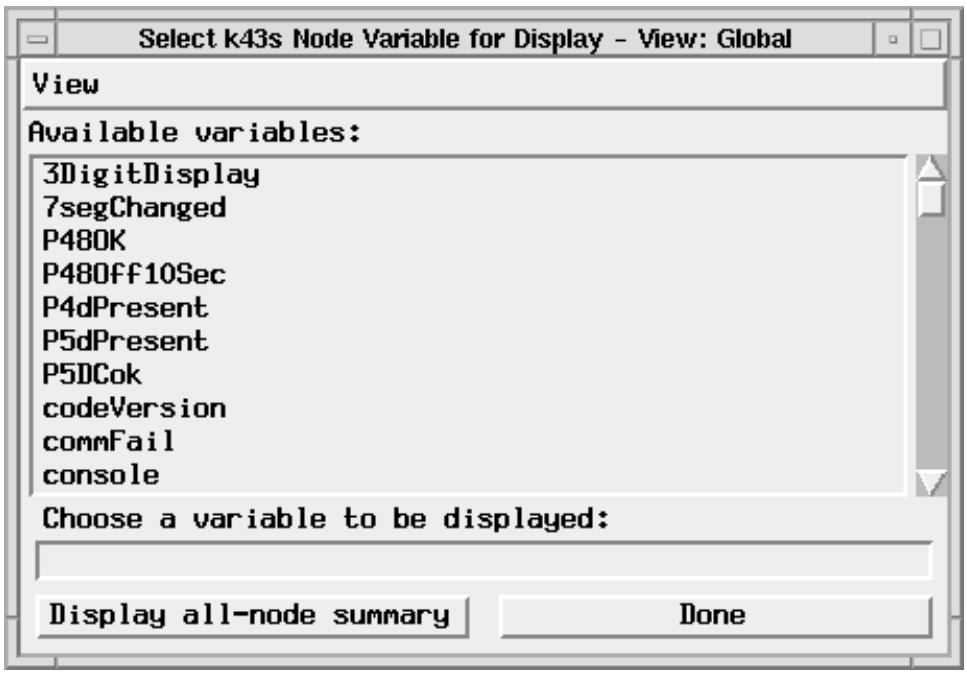


Figure 37. Node Variable Selection Window

This window contains:

- View pulldown menu that allows you to select an individual system partition or Global view of the system (in a multipartitioned system; if a system is not partitioned, 'View' won't appear on the menu bar)
- Scrolling list, in alphabetical order, of SP System Monitor variables that pertain to node hardware
- Entry field where you can type the name of a SP System Monitor variable
- Two push buttons: one to display the All Node Summary, one to close the window.

Step 2: Display Node Variables

At the Node Variables display:

SELECT 3DigitDisplay from the list of variables

PRESS Display all-node summary

▲ The three-digit display summary appears.

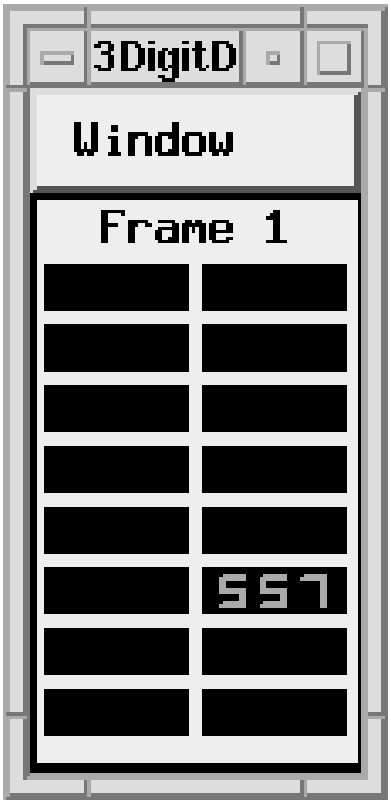


Figure 38. 3-Digit Display

Displaying hostResponds, powerLED, and envLED gives you an at-a-glance view of the status of your system. Use the scroll bar to find these other variables and repeat the selection process.

Every All Node Summary display has the same structure. The window title bar identifies the variable. The node format follows the physical organization of nodes within the frame. This means that, for a particular frame, the bottom left node in the window corresponds to the bottom left node within that frame.

Step 3: Check the System Status

Together, these summaries give you a logical view of the front panel indicators on all your nodes. Table 9 on page 144 explains how to interpret these summaries:

Note

The variable `console` has control functions. Clicking on this variable executes the function if you have serial link authority.

Table 9. Default Variable Colors

Variable Name	Front Panel Indicator	Color and Meaning
3DigitDisplay	Three-digit display	Any number: Can be an informational message, a warning, or failure. Check the codes in the *IBM RISC System/6000 Problem Solving Guide*. Pressing the left mouse button in a frame causes the slot numbers to appear in the node areas. Pressing the right mouse button in a frame causes the node number (relative to the whole SP system) to be displayed. These numbers are displayed only while the mouse button is pressed. Black: If nothing is displayed and hostResponds is green, this indicates the node is running (up)
hostResponds	hostResponds light	Green: The host node is responding. Red: The host node is not responding.
powerLED	Power light	Green: Power light is on. Node is running. Gray: Power light is flashing. Node is off but power is available. Black: Power light is off. Node is not running. Check the power switch on the front panel.
envLED	Environment light	Black: Environment light is off. No indication of failure. Gray: Environment light is flashing a failure. Yellow: Environment light is on. Indicates an environment warning.
Key Mode Switch	Key position	Green: normal Yellow: secure Red: service

Determining the State of Network Adapters within the SP System

The Topology windows provide you with a way to globally determine which network adapters are configured for TCP/IP within each SP processor node and determine if the adapters are operating. The window allows you to select a variable associated with each configured LAN in the SP system and display its values for all the processor nodes.

Not all LANs can be configured for all processor nodes. When a LAN is not configured within a target node, that variable is blanked out. Determination of which LANs are configured for each target node is based on network adapter information contained within the System Data Repository combined with communications with the SNMP (Simple Network Management Protocol) agent daemons running in each processor node. Until communication with an agent daemon in a target node is completed, the state of the variable is reflected as unknown for the target node.

Step 1: Display the Topology

At the SP System Monitor GUI main window:

SELECT SP → Topology

▲ The Topology display appears.

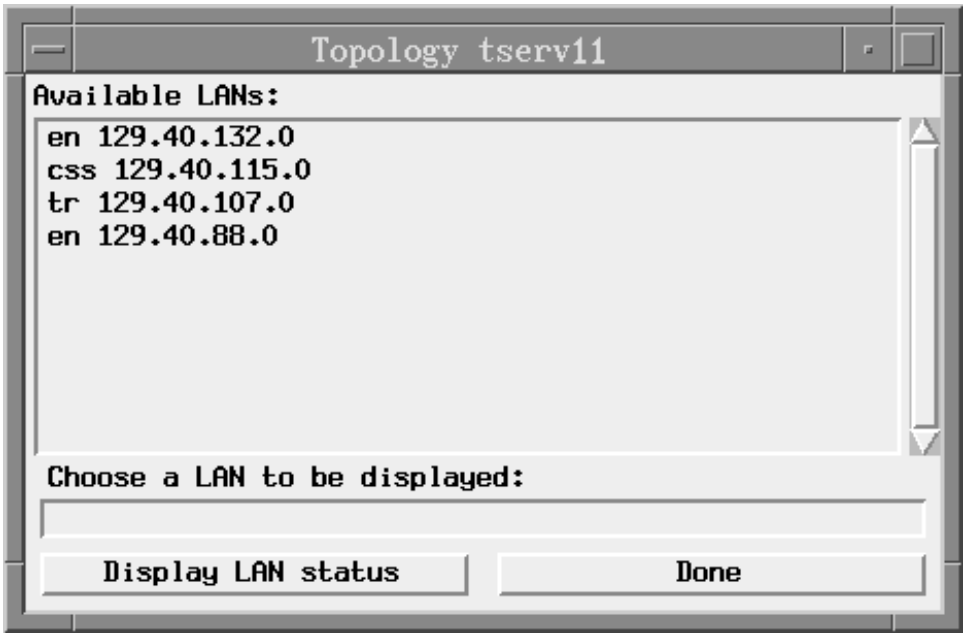


Figure 39. LAN Topology Display

This window contains:

- A scrolling list of SP System Monitor LAN variables that pertain to configured LANs.
- The entry field where you can type the name of a SP System Monitor LAN variable
- Two push buttons: one to display the LAN status, one to close the window.

Step 2: Display the LAN Status

At the Topology display:

SELECT A LAN ID

PRESS Display LAN Status

▲ The LAN Status display summary appears.

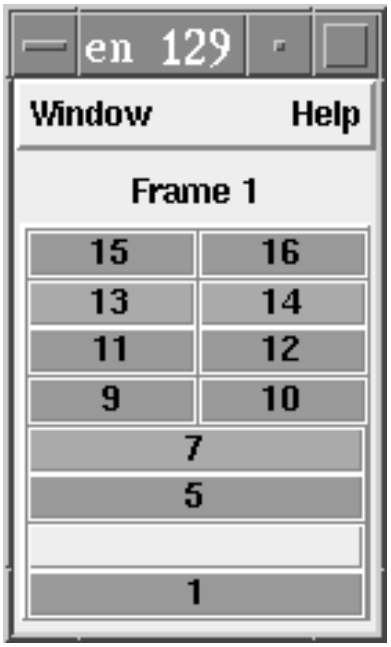


Figure 40. LAN Status Display

This display shows that the node residing in slots 3 and 4 is not configured for the chosen LAN. The darker colored nodes indicate the state of the adapter (see Table 10).

Graphical Element Colors

The Topology window displays use color to show the state of adapters and communications with the SNMP agents running in the processor nodes. The colors are orange, yellow, red, and green.

Table 10. Graphical Element Colors

Color	Adapter Status	Meaning
Orange	Initial status	Communication with the SNMP agent running in the target node has not been successful.
Yellow	Adapter exists: state unknown	Communication with the SNMP agent indicates an adapter connecting to the LAN being displayed exists within the target node. The state of the adapter is being determined or is not yet defined on the processor node.
Red	Adapter is not operational	Communication with the SNMP agent running in the target node indicates the adapter is configured within the target node but is not operational.
Green	Adapter is operational	Communication with the SNMP agent running in the target node indicates the adapter is configured within the target node and is operational.

Note: A pulldown help menu that contains a summary of the colors and their meaning is available within each Topology window.

Communication with the SNMP agent running in each processor node occurs serially. After communication with all of the processors completes, it is reinitiated after a delay of 1 minute. When communication with the agent running within a target node is unsuccessful, the target node is placed on an “unresponsive list”. Communication with each node on the “unresponsive” list is attempted every 5 minutes.

LAN configuration changes occurring while the LAN window is open is not reflected until Topology is reentered from the main window.

Setting Up the Logging Daemon on Separate Workstations

Step 1: Install the **ssp.clients** option on your workstation. Refer to the appropriate steps in the chapter on installing and configuring the SP system in the *IBM RISC System/6000 Scalable POWERparallel Systems: Installation Guide*.

Step 2: Edit the /**spdata**/**sys1**/**spmon**/**hwevents** file to configure the logging options you want. Refer to **splogd** in the *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference* for more information.

Step 3: If you have set up **hwevents** to do error logging (SP_ERROR_LOG keyword), run /**usr**/**lpp**/**ssp**/**bin**/**setup_logd**. (Refer to **setup_logd** in the *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference* for more information.) If you are installing the logging daemon only to call your own executable when a state change is reported, just add an entry to **inittab** or start **splogd** in the background:

```
splogd -b
```

Step 4: If you have run /**usr**/**lpp**/**ssp**/**bin**/**setup_logd** or added **splogd** to the **inittab** file in Step 3, then you must add the following to the /**etc**/**environment** file:

```
SP_NAME=control_workstation
```

where *control_workstation* is the hostname of the control workstation. Otherwise, the environment variable SP_NAME must be defined to the hostname of the control workstation prior to executing **splogd**.

Step 5: For each frame in the SP system, add the following line to the /**spdata**/**sys1**/**spmon**/**hmacls** file on the control workstation:

frame_number **hardmon.***workstation_name* **m**

where *frame_number* is the number of the frame and *workstation_name* is the short hostname, as returned by **hostname -s**, of the workstation where you have installed the logging daemon.

Setting Up the Hardware Monitor Clients on Separate Workstations

Step 1: Install the **ssp.clients** option on your workstation. Refer to the appropriate steps in the chapter on installing and configuring the SP system in the *IBM RISC System/6000 Scalable POWERparallel Systems: Installation Guide*.

Step 2: Prior to executing any Hardware Monitor client command, set the environment variable SP_NAME to the hostname of the control workstation. Before using any command you must be authenticated to use the Hardware Monitor. Use

the **kinit** command to login to the SP authentication system. If your site uses an alternative to the SP authentication services, you must use authentication procedures appropriate to such an alternative.

Chapter 8. Starting Up and Shutting Down the SP System

This chapter describes how to start up and shut down nodes on the SP system. It explains how to specify the sequence in which the nodes and subsystems in your system are started up and shut down. It also describes the tools that save you time and effort by operating on all or some of the nodes in your SP system with a single command. These tools include the **cstartup**, **cshutdown**, and **seqfile** commands, plus the files they use to control the sequence of startup or shutdown processing.

In addition, this chapter describes the interface that lets you coordinate the shutdown of a subsystem that operates across multiple nodes.

Overview of System Startup and Shutdown

You control the operation of the SP system with commands issued at the control workstation.

Use the **cstartup** command to power on SP nodes and boot up the operating system. You can also reset SP nodes with the **cstartup** command.

Use the **cshutdown** command to halt or reboot SP nodes.

Your SP may be a complex group of nodes, each dependent on other nodes for some resources. Startup and shutdown operations must occur in the proper sequence so that these dependencies are honored. As a simple example, a server node may have to be running before its client nodes are started up. A server node should not shut down before its clients have been shut down. The **seqfile** command defines sequencing relationships that result from using boot/install servers. You can define additional dependencies.

To provide best performance, **cstartup** and **cshutdown** can operate on multiple nodes concurrently.

Similar considerations apply to the shutdown of *special subsystems* (see “Creating a Special Subsystem” on page 159) that run on multiple nodes in the SP system. An example of a subsystem is a file system that spans several nodes within a system partition. Subsystems may need to do special processing to prepare for the shutdown of one or more nodes, or a set of subsystems may need to shut down in a specific order.

Controlling the Sequence of Startup and Shutdown Operations

You must define the sequencing relationships between your nodes. For example, boot/install servers must start up before their clients. Other programs on your SP system may create additional sequencing relationships.

The node sequence files, **/etc/cshutSeq** and **/etc/cstartSeq**, have lines that describe these dependencies. In these files, you define the groups of nodes and the sequence in which the groups are started up or shut down. If you don't have a node sequence file, some sequencing is performed automatically. The **seqfile** command helps you create these files by generating definitions for groups of nodes

© Copyright IBM Corp. 1995

and their sequencing order for dependencies relating to boot/install servers and their clients.

You can also create a subsystem sequence file, **/etc/subsysSeq**, which lists the subsystems, defines groups of subsystems, and defines relationships between subsystems. If you don't have a subsystem sequence file, no subsystems are notified of an impending node shutdown.

The sequence files, which reside on the control workstation, are:

/etc/cstartSeq Startup. The list defining groups of nodes and the order of startup processing.

/etc/cshutSeq Shutdown. The list defining groups of nodes and the order of shutdown processing.

/etc/subsysSeq Special subsystems. The list defining special subsystems, groups of special subsystems, and the order of shutdown processing.

The node sequence files are separate to allow for different startup and shutdown dependencies. However, if your startup and shutdown dependencies are mirror images, you can maintain a single file; create a symbolic link to provide both file names.

"Defining Startup and Shutdown Sequence Files" on page 155 describes the format of the sequence files and their interactions with command line flags.

Default Node Sequencing

The SP system can do some sequencing automatically during system startup or shutdown. The **seqfile** command shows you the default sequencing relationships.

If you don't create the **/etc/cstartSeq** file, the **cstartup** command uses a default sequence based on the System Data Repository (SDR). Any nodes defined in the SDR as boot/install servers are started up concurrently. After the servers are running, nodes defined as boot/install clients are started up. If you create an empty **/etc/cstartSeq** file, then all nodes are started up concurrently.

If you don't create an **/etc/cshutSeq** file, **cshutdown** uses a default sequence based on the SDR. If you create an empty **/etc/cshutSeq** file, then all nodes are shut down concurrently. If you don't have a **/etc/subsysSeq** file or the file is empty, no special subsystem shutdown occurs.

Starting the SP System

To start the SP system, login on the control workstation and issue the **kinit** command, specifying a principal name for which there is an entity in the **hardmon** acls file with control authorization for the frames to be started up. Then, use the **cstartup** command. The **cstartup** command powers on specified nodes that are not powered on, and may reset specified nodes that are already powered on. Those actions are equivalent to manually pressing the hardware switches. Once the hardware-equivalent actions are done, control is passed to **boot** and **init**.

Caution!

The **cstartup** command attempts to power on nodes that are powered off. This has safety implications if someone is working on the nodes. Proper precautions should be taken when using this command.

To restart a SP system that is already running, use **cshutdown -r -G**; this does *not* power on any nodes.

Note: If some nodes cannot be started in a specific sequence, or if some cannot be started, the system may still be usable.

The complete syntax of the **cstartup** command is described in the *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference*.

Starting or Resetting All Nodes

You can start or reset nodes with a single **cstartup** command, as shown in Table 11.

Table 11. Specifying Nodes to Start with cstartup

Command	Meaning
cstartup ALL	Starts all nodes in the current system partition.
cstartup *target_nodes*	Starts specific nodes in the current system partition. All nodes listed must be in the current system partition or the command will fail.
cstartup -G ALL	Starts all nodes in the physical SP system.
cstartup -G *target_nodes*	Starts specific nodes in the physical SP system, regardless of system partitioning.

If a node is powered off, this command powers it on. If the node is already powered on and is not running, this command resets it. The nodes are started in the sequence specified by **/etc/cstartSeq**. If this file doesn't exist, the nodes are started up in the default sequence.

To start all nodes in the physical SP system concurrently, ignoring the sequence file if it exists or bypassing the default sequencing if the file does not exist, use the **-E** flag:

```
cstartup -E -G ALL
```

Information about creating the **/etc/cstartSeq** file is in "Defining Startup and Shutdown Sequence Files" on page 155.

Starting or Resetting Selected Nodes

To start or reset a specific node within the current system partition, put its name on the **cstartup** command line. The following example starts the single node in the current system partition named **cs6**:

```
cstartup cs6
```

The command checks for dependencies before starting up the node. It uses the **/etc/cstartSeq** file, if one exists, or the default sequence rules.

If node **cs6** is already running, you must specify either **-z** or **-Z** to force the node to be reset. This example resets the node **cs6** in the current system partition without checking any sequencing dependencies that **cs6** has with other nodes:

```
cstartup -Z cs6
```

You can also identify nodes by node number or range of node numbers using the **-N** flag. The following command starts node numbers 4 through 12 in the current system partition.

```
cstartup -N 4-12
```

Shutting Down the SP System

To shut down the SP system, login on the control workstation and issue the **kinit** command, specifying a principal name for which there is an entry in the **hardmon** acls file with control authorization for the frames to be shutdown. Then, use the **cshutdown** command. The SP **cshutdown** command is similar to the workstation **shutdown** command, except it has these advantages over using **shutdown** to shut down each node of an SP:

- **cshutdown** provides a single point of control.

  Using one **cshutdown** command on the control workstation, you can shut down all or selected nodes.

- The sequencing of node shutdown and reboot is automated.

  You can use the /**etc**/**cshutSeq** file to control the order in which nodes are shut down, or you can let the system determine the order based on System Data Repository information about servers and clients.

- Special subsystems can be notified of impending node shutdown.

  The /**etc**/**subsysSeq** file lists these special subsystems and describes any sequencing relationships between them.

Note: The **cshutdown** command powers off the node after the **shutdown** process completes.

You can shutdown nodes with a single **cshutdown** command as shown in Table 12.

Table 12. Specifying Nodes to Shut Down with cshutdown.

Command	Meaning
cshutdown ALL	Shuts down all nodes in the current system partition.
cshutdown *target_nodes*	Shuts down specific nodes in the current system partition. All nodes listed must be in the current system partition or the command will fail.
cshutdown -G ALL	Shuts down all nodes in the physical SP system.
cshutdown -G *target_nodes*	Shuts down specific nodes in the physical SP system, regardless of system partitioning.

The complete syntax of the **cshutdown** command is described in the *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference*. Understanding how shutdown processing works may help you

understand some of **cshutdown**'s options; this is described in “The Phases of Shutdown Processing” on page 154.

Halting Nodes

You can halt all nodes in the current system partition with one command, using:

```
cshutdown ALL
```

The **-h** (halt) flag is the default; you don't need to use it when you invoke **cshutdown**. The nodes are shut down in the sequence specified by **/etc/cshutSeq**. If this file doesn't exist, the nodes are shut down in a default sequence based on the SDR. If **/etc/subsysSeq** exists, those subsystem interfaces are invoked during shutdown.

Halting selected nodes is similar. To halt nodes **cs3**, **cs5**, and **cs7** in the current system partition, use the command:

```
cshutdown cs3 cs5 cs7
```

The command checks for dependencies before shutting down the nodes. If **/etc/cshutSeq** exists, those rules apply; otherwise, **cshutdown** applies default sequencing rules based on the SDR. If **/etc/subsysSeq** exists, those subsystem interfaces are notified of the shutdown of the target nodes.

Rebooting Nodes

Reboot all nodes in the current system partition by using the **cshutdown** command with the **-r** (reboot) flag:

```
cshutdown -r ALL
```

The target nodes are halted according to the sequencing rules in **/etc/cshutSeq**, and restarted according to the rules in **/etc/cstartSeq**. If these files don't exist, the default sequencing rules are determined at shutdown and startup.

You can control how the system starts during reboot by passing **cstartup** flags with the **-C** flag. For example, to pass the **-E** flag to tell **cstartup** to ignore the sequence file when rebooting current system partition nodes numbered 4 and 5, use this command:

```
cshutdown -r -C "-E" -N 4-5
```

To pass multiple startup flags, you must escape any space characters in the string. The double quotes in this example (`"-E"`) do that, although they are not strictly required with a single flag.

Rebooting from Maintenance Mode

If you bring a node down to maintenance mode, you must ensure file system integrity before you reboot the node.

The **cshutdown** command, which runs from the control workstation, will be unable to **rsh** to the node to perform the node shutdown phase processing which includes synching of the file systems. We recommend that you issue the **sync** command three times in succession from the node console before running **cshutdown**. This is especially important if any files were created while the node was in maintenance mode.

To determine which nodes are in maintenance mode, run the **spmon -d** command and look for the combination of **nodePower** on and **hostResponds** no.

Notifying Users of an Impending Shutdown

You probably want to warn your users if nodes are going down, so they can wrap up any work in progress. By default, when you issue a **cshutdown** command, a message is sent to all users logged into the affected nodes, telling them how long they have before shutdown. You can customize the message as well as the time when the system shuts down.

To power off the entire SP system at 9 pm, issue the following command:

```
cshutdown -T 21:00 -M "All nodes will power off at 9 pm." -G ALL
```

To power off all nodes in the current system partition at 9 pm, issue the following command:

```
cshutdown -T 21:00 -M "All nodes will power off at 9 pm." ALL
```

Suppose you need to do an emergency reboot of the pair of nodes named **cs1** and **cs2** in the current system partition, but want to allow 10 minutes for jobs to complete. You can notify your users and schedule the reboot with this command:

```
cshutdown -r -T 10 \
          -M "Reboot on cs1&2 in 10 minutes--don't submit new jobs." cs1 cs2
```

The Phases of Shutdown Processing

cshutdown command processing has these phases:

1. Notifying users and terminating nonroot processes
2. Subsystem shutdown
3. Node shutdown
4. Restarting nodes, if requested

Notifying Users and Terminating Nonroot Processes

User processes that are not started by root and are running on the target nodes are sent a SIGTERM followed, 30 seconds later, by a SIGKILL. This gives user processes that handle SIGTERM a chance to do whatever cleanup is necessary.

Subsystem Shutdown

The subsystem shutdown phase tells special subsystems to terminate their own operations on the target nodes before the node phase shuts down the target nodes. Groups for which /**etc**/**subsysSeq** doesn't define a sequencing relationship are invoked concurrently. Subsystems not listed in /**etc**/**subsysSeq** are not notified of the pending system shutdown.

If **cshutdown** is issued without the **-Y** flag and a nonzero (failure) code is returned, you receive a prompt allowing you to continue, to quit, or to start a subshell so that you can check /**var**/**adm**/**SPlogs**/**cs**/**cshut.***MMDDhhmmss***.***pid* for messages and refer to the documentation for the subsystem. When you leave the subshell, you are prompted with the same choices.

The subsystem phase of shutdown waits for all subsystems to return before continuing with the node phase. If a subsystem does not return, the **cshutdown** command waits indefinitely. Use the **-W** flag to specify a time-out value.

Node Shutdown

The node phase operates on nodes specified on the **cshutdown** command line. The **/etc/cshutSeq** file controls the sequencing of the node phase operation.

Node shutdown has potentially two levels of coordinated concurrent operations:

- More than one group of nodes can be shut down concurrently.
- Within each one of those groups, more than one node can be shut down concurrently.

They are coordinated in that one group may wait for the completion of some other specific groups before starting to shut down.

In a halt operation, the node phase applies the standard workstation shutdown on the target nodes. The nodes are halted in the sequence specified in **/etc/cshutSeq**, if it exists, or in the default sequence. Nodes not listed in **/etc/cshutSeq** may be halted concurrently with any other nodes.

Restarting the Nodes, If Requested

If you request a system reboot with the **-r** flag, the node phase halts the nodes in the sequence specified in **/etc/cshutSeq** and then restarts the nodes consistent with the initialization sequence specified in **/etc/cstartSeq**. If these files do not exist, the default sequence is used. It does not power on any nodes that were powered off when the **cshutdown** command was issued. If you use the **-r** flag, the only nodes powered on during reboot are those powered off during shutdown.

Checking the Status of Startup and Shutdown Requests

Each time you invoke **cstartup** or **cshutdown**, the system creates a log of activities related to that command. The logs contain an entry for the initiation and completion of operations on each node. **cshutdown**'s log includes information about subsystem shutdown operations. These logs are in the directory **/var/adm/SPlogs/cs**. Two types of log files are created:

cstartup **/var/adm/SPlogs/cs/cstart.***MMDDhhmmss.pid*

cshutdown **/var/adm/SPlogs/cs/cshut.***MMDDhhmmss.pid*

Each log file's name contains the date and time that the command was issued, in the format *MMDDhhmmss.pid*, where:

MMDDhhmmss The timestamp

pid The process ID of the **cstartup** or **cshutdown** command

Defining Startup and Shutdown Sequence Files

Relationships between nodes must be defined in the **/etc/cstartSeq** and **/etc/cshutSeq** files. You define groups of nodes in these files. Nodes in the same group are started up or shut down concurrently. You also define the order in which groups of nodes are processed. Sequencing files can contain nodes from any and all system partitions; sequencing rules are applied without regard to system partitions. Groups without specified sequencing relationships are started up concurrently. Nodes not listed in **/etc/cstartSeq** may be started up concurrently with any other nodes. Nodes not listed in **/etc/cshutSeq** may be shut down concurrently with any other nodes.

The sequencing of special subsystems is handled similarly. Subsystems and groups of subsystems are defined in the **/etc/subsysSeq** file. Subsystems without specified sequencing relationships are shut down concurrently. Subsystems must be defined in this file. If they are not defined, they are not invoked during shutdown.

Planning for Concurrent Startup and Shutdown

It's important that you allow the **cstartup** and **cshutdown** commands to do as much work as possible concurrently. This ensures optimal performance when starting up and shutting down a system with many nodes and/or subsystems. There are potentially two levels of coordinated concurrent operations:

- More than one group of nodes or subsystems can be started up or shut down concurrently.
- Within each one of those groups, more than one node or subsystem can be started up or shut down concurrently.

They are coordinated in that you can specify that one group must wait for the completion of some other specific groups before being started up or shut down.

Format Rules for Sequence Files

A line in **/etc/cstartSeq**, **/etc/cshutSeq**, and **/etc/subsysSeq** can be one of the following:

- Comment line starting with a pound sign (#). A comment can also start within a line.
- Group definition line containing a group name followed by a colon (:), followed by one or more node names (**/etc/cstartSeq**, **/etc/cshutSeq**) or subsystem names (**/etc/subsysSeq**). To put multiple nodes or subsystems in a group, separate their names with commas (,) and, optionally, blanks.
  - In **/etc/cstartSeq** and **/etc/cshutSeq**, a node name is the en0 Ethernet hostname of the node. These are also known as **reliable_hostname** in the **Node** object class in the System Data Repository.
  - In **/etc/subsysSeq**, a group definition line is the arbitrary subsystem group name, followed by a colon (:), followed by the full path name of the command you want issued at shutdown.

  The group names in the file are arbitrary names and treated as tokens.

  Each node or subsystem can be in only one group. A group can consist of from 1 to all your nodes or subsystems.
- Sequence line containing a greater than symbol (>). Thus, in **/etc/cstartSeq**, `groupA > groupB` says to start groupA before starting groupB. And, in **/etc/cshutSeq**, `groupA > groupB` says to shut down groupB before shutting down groupA.

  A group cannot both precede and trail the same group. That is, the sequence `A > B > A` is not valid.

The lines can be in any order. Each node must be assigned to a group if you wish to control its sequencing.

A group or node name cannot contain a blank, comma, newline, tab character, greater than symbol (>), colon (:), parentheses (() or ()), or pound sign (#). Case is significant in node and group names.

Testing Your Sequence Files

You can verify that the **/etc/cstartSeq** file has the correct format without powering on or resetting any nodes by specifying:

```
cstartup -k ALL
```

The **-k** flag tells **cstartup** not to reset any nodes, just check the sequence file. If the sequence file is correct, you'll get the following message:

```
cstartup: No circular dependencies were detected
```

Verify that the **/etc/cshutSeq** and **/etc/subsysSeq** files have the correct format without shutting down any nodes by specifying:

```
cshutdown -kF ALL
```

The **-k** flag tells **cshutdown** not to shutdown any nodes, just check the sequence file. The **-F** flag tells **cshutdown** not to issue any messages to users.

Getting Started: Capturing the Default Sequences

An easy way to create your **/etc/cstartSeq** and **/etc/cshutSeq** files is to capture the default group definitions and sequencing relationships that are based on the boot/install server information in the System Data Repository. You can then add to or modify them as required for your installation.

Create the default **/etc/cstartSeq** file using this command:

```
seqfile -b > /etc/cstartSeq
```

The **-b** flag tells **seqfile** to include boot/install relationships, which are important only during system startup.

To create the default **/etc/cshutSeq** file, use this command:

```
seqfile > /etc/cshutSeq
```

After you've created the default files, you can edit them to reflect sequencing relationships based on other aspects of your installation's configuration.

Sample Startup File /etc/cstartSeq

Here is a simple file for sequencing startup:

```
ServerA > ClientsOfA
ServerA: node2
ClientsOfA: node3, node4, node5, node6
```

The line `ServerA: node2` defines node 2 as the group ServerA. The line `ClientsOfA: node3, node4, node5, node6` defines nodes 3 through 6 as a group that can be started up concurrently. The sequence line `ServerA > ClientsOfA` says that ServerA must be completely started up before starting up the nodes that are clients of server A (nodes 3 through 6).

This sample file is slightly more complex:

```
groupA > groupB
groupA > (groupF, groupE) > groupD > groupC
groupD > groupG
groupA: node0  #node0 is the node that node1-node7 depend on.
groupB: node1
groupC: node2
groupD: node4
groupE: node5
groupF: node6, node7
groupG: node3
```

The line `groupA > ( groupF, groupE ) > groupD > groupC` says to start up groupA. When groupA is up, concurrently start up groupF and groupE. When they are up, start up groupD. Finally, when groupD is up, start up groupC. The line `groupA > groupB` says that groupB depends on groupA, but may be started concurrently with groupE and groupF. No other groups have dependencies on groupB.

If your command line specifies a node but omits another that is listed in your **/etc/cstartSeq** or **/etc/cshutSeq** files as its leading node, the sequencing rules still apply unless overridden by the **-X** flag.

Sample Shutdown File /etc/cshutSeq

Here is one of the sample startup sequence files, interpreted now as a shutdown file:

```
ServerA > ClientsOfA
ServerA: node2
ClientsOfA: node3, node4, node5, node6
```

The line `ClientsOfA: node3, node4, node5, node6` defines nodes 3 through 6 as a group that can be shut down concurrently. The line `ServerA: node2` defines node2 as the group (of one node) called ServerA. The line `ServerA > ClientsOfA` says that the nodes that are clients of server A (nodes 3 through 6) can be shut down concurrently, but must be shut down before node2 in group ServerA.

Sample Subsystem File /etc/subsysSeq

A sample file for subsystem shutdown could be:

```
ssgroupA: /full/pathname/subsystemA
ssgroupB: /local/subsystemB, /local/subsystemC
ssgroupA > ssgroupB
```

The line `ssgroupA: /full/pathname/subsystemA` defines the group ssgroupA, consisting of the single subsystem invoked by the /full/pathname/subsystemA command. The line `ssgroupB: /local/subsystemB, /local/subsystemC` defines a group consisting of the two subsystems /local/subsystemB and /local/subsystemC.

The line `ssgroupA > ssgroupB` says to first notify ssgroupB before nodes are shut down. When both subsystems in ssgroupB have returned, then notify the single subsystem in ssgroupA.

Creating a Special Subsystem

Read this section if you write your own special subsystem and want the **cshutdown** command to notify the subsystem when nodes are being halted or rebooted. The SP system has no predefined special subsystems.

The special subsystem interface is a user-provided command that is invoked during the subsystem phase of SP system shutdown processing. The command is defined in the **/etc/subsysSeq** file and must reside on all nodes of the SP system; the **cshutdown** command invokes the subsystem interface on a randomly selected node in the current system partition.

The following sections define the interface to a special subsystem. This interface consists of:

- The inputs that the **cshutdown** command passes to your special subsystem
- The actions your subsystem must perform.

Note: The special subsystem interface is called *after* nonroot processes are killed. If you want your subsystem to take advantage of shutdown notification, some part of it must be running as root.

Input to the Special Subsystem Interface

The **cshutdown** command invokes the special subsystem interface with the following standard parameters:

-r or **-h** Indicates whether the **cshutdown** command is halting (**-h**) or restarting (**-r**) the target nodes.

-b *IPaddress* Specifies, in dotted decimal form, the IP address of the calling program, in this case, the control workstation.

-f /var/adm/SPlogs/cs/cshut.*MMDDhhmmss***.***pid*
Specifies the log file to which the subsystem interface program should append a record of status and activities.

target_nodes The list of node numbers of the target nodes specified on the **cshutdown** command line. The numbers are passed at the end of the argument list.

For example, if the **/etc/subsysSeq** file contains the following line:

```
ssgroupA: /full/pathname/subsystemA
```

and this **cshutdown** command is issued on the control workstation whose IP address is 129.40.64.70:

```
cshutdown -h -N 1 2 3
```

the **cshutdown** command issues the following command on a randomly-selected node during the subsystem phase of shutdown processing:

```
/full/pathname/subsystemA -h -b 129.40.64.70 \
                          -f /var/adm/SPlogs/cs/cshut.time.pid \
                          1 2 3
```

What Your Special Subsystem Must Do

In addition to any processing specific to the subsystem, the special subsystem must call the **csLogger** command to record its activities. Your program must call **csLogger** with the following required parameters:

-b *IPaddress* Specifies, in dotted decimal form, the IP address of the calling program, in this case, the control workstation.

-f /var/adm/SPlogs/cs/cshut.*MMDDhhmmss***.***pid*
Specifies the log file to which the subsystem interface program should append a log of status and activities.

logString Specifies a string that will be appended to the specified log file. This is the last parameter.

Using the example started above, if the **cshutdown** command calls your program with this command:

```
/full/pathname/subsystemA -h -b 129.40.64.70 \
                          -f /var/adm/SPlogs/cs/cshut.time.pid \
                          1 2 3
```

then the **/full/pathname/subsystemA** command could make calls similar to these, to log the start and completion of subsystem shutdown:

```
csLogger -b 129.40.64.70 \
         -f /var/adm/SPlogs/cs/cshut.time.pid \
         subsystemA:  initiating shutdown at 15:43.55 on 1 Apr 94.

csLogger -b 129.40.64.70 \
         -f /var/adm/SPlogs/cs/cshut.time.pid \
         subsystemA:  shutdown complete at 15:46.23 on 1 Apr 94.
```

Chapter 9. The High Performance Switch

The optional High Performance Switch offered with the SP system provides high bandwidth, low latency, location-independent communication between nodes. The High Performance Switch adapter is the powerful High Performance Switch Adapter-2. This chapter presents information on:

- Switch communication modes
- System Partitioning and the High Performance Switch
- Selecting switch clocks
- Managing the switch topology file
- Starting the switch
- Isolating nodes from the High Performance Switch

Switch Communication Modes

Two communication modes are offered for the High Performance Switch Adapter-2:

1. AIX sockets using standard IP protocol
2. Single process (user space) message passing

Both modes can be used concurrently, as if the High Performance Switch Adapter-2 were actually two separate adapters.

Switch Communication Using IP (TCP/IP and UDP/IP)

The switch adapters are configured for IP by default during the installation process. You can assign specific IP network addresses statically during installation or dynamically using ARP (Address Resolution Protocol). When you assign the IP address statically, there are restrictions on mapping of nodes to low order addresses (see the information on switch node numbering in the *IBM RISC System/6000 Scalable POWERparallel Systems: Installation Guide*). When you assign the IP address range dynamically using ARP, there are no restrictions.

Any application can communicate over the High Performance Switch by opening a standard AIX socket and specifying appropriate IP addresses. Additional system facilities based on IP, such as NFS, and licensed programs such as AFS, can also be configured to use the High Performance Switch. With the possible exception of applications that depend on network-specific functions (LAN broadcasts, for example), most applications work over the High Performance Switch without modification.

Notes:

1. The **iptrace** command is not fully supported for the High Performance Switch adapter; it provides interface-level packet tracing for incoming packets only.
2. The standard AIX commands **ifconfig** and **arp** have been modified to support the High Performance Switch adapter. For further information, refer to the *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference*.
3. Static routes associated with the High Performance Switch will not be automatically reconfigured during node reboot. The routes for the switch are associated with the default interface (**en0**) upon reboot because the High Performance Switch is not available during the phase of system initialization

© Copyright IBM Corp. 1995

when the static routes are reconfigured. If you want the routes added automatically, we recommend that you put them in a script and have the script called as the last entry in **/etc/inittab**.

Switch Communication Using User Space Message Passing

The user space message passing mode supports parallel applications requiring maximum bandwidth and minimum latency. This mode is used by PVMe and the Parallel Environment for AIX to provide their exported message passing interfaces. In this mode, only a single user-space process may access the adapter, and the nodes to be used must be allocated to the parallel application via the Resource Manager. The Resource Manager should group nodes into named resource pools to facilitate node scheduling.

The High Performance Switch Adapter-2 provides multiplexing for and automatic availability of both communication modes. Although the user space message passing mode may be used by only one process per node, TCP/IP and UDP/IP messages can be processed concurrently (for example, NFS I/O).

System Partitioning and the High Performance Switch

As discussed in Chapter 6, “Managing System Partitions” on page 95, the SP system can be organized into groups of nodes for various purposes. System partitioning has the following implications for the High Performance Switch:

- Commands related to the High Performance Switch:
  - **Eannotator** - Operates only within the current system partition
  - **Eclock** - Always operates globally (across system partitions)
  - **Eduration** - Operates only within the current system partition
  - **Efence** - Can operate globally (**-G**) or only within the current system partition
  - **Eprimary** - Operates only within the current system partition
  - **Estart** - Operates only within the current system partition
  - **Etopology** - Operates only within the current system partition
  - **Eunfence** - Can operate globally (**-G**) or only within the current system partition
- Switch faults and message traffic are contained *within a system partition*.
- The subset of the High Performance Switch defined by the topology file for the system partition is viewed as a logical High Performance Switch.
- There is no connectivity over the High Performance Switch *between system partitions*. (A gateway node with routing set up to the High Performance Switch network may require routing changes if the gateway is to remain a gateway for more than one system partition. Use the explicit host routes on the gateway node or enable ARP on all system partitions and redefine the IP addresses within a system partition as a different subnetwork.)
- There remains one High Performance Switch clock source, regardless of system partitioning.

Selecting Appropriate Switch Clocks

One of two switches in the system provides the master switch clock to the rest of the system. If the current master clock fails, or is removed from the configuration, the alternate master clock must be selected and the switch reinitialized.

The **Eclock** command establishes a master clock source after the system is powered up or when an alternate must be selected. It can set the appropriate clock input for every switch in the system or for a single switch after power on. After **Eclock** completes, the **Estart** command must be run.

Refer to the *IBM RISC System/6000 Scalable POWERparallel Systems: Maintenance Information Volume 1* for information on removing and restoring the High Performance Switch resources.

Warning: Running **Eclock** after the switch is started may cause nodes to lose their clock source. Refer to **Eclock** in the *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference* for more information.

Selecting the Switch Clock Source

To set the High Performance Switch switch clock source using SMIT:

TYPE **smit**

▲ The System Management menu appears

SELECT 9076 SP Systems Management

▲ The 9076 SP Systems Management menu appears.

SELECT 9076 SP Cluster Management

▲ The 9076 SP Cluster Management menu appears.

SELECT Perform Switch Operations

▲ The Perform Switch Operations menu appears.

SELECT Change/Show Switch Clock Source Settings

▲ The Change/Show Switch Clock Source Settings menu appears.

At this point, you can select options:

```
Change/Show Switch Clock Source Settings

□ List Switch Clock Source Settings
□ Initialize Switch Clock Source Settings
□ Initialize Switch Clock Source Settings for a Switch
□ Restore the Switch Clock Source Settings
□ Select Alternate Switch Clock Source Settings
□ Create an Eclock Topology File
```

Figure 41. The Change/Show Switch Clock Source Settings SMIT menu.

When you choose a switch clock source setting option, you are prompted for a topology file name, and then the **Eclock** command is invoked.

The fastpath invocation for this menu is:

```
smit clock_src
```

Managing the Switch Topology File

The switch topology file describes the wiring configuration for the High Performance Switch; it contains node-to-switch or switch-to-switch cable information. You choose a switch topology file based on the number of **switches** in your SP system. (For more information on choosing a switch topology file, refer to the *IBM RISC System/6000 Scalable POWERparallel Systems: Installation Guide*.) You can annotate connection labels onto the selected switch topology file with the **Eannotator** command. Once you have selected and annotated the switch topology file, you store it in the System Data Repository (SDR) with the **Etopology** command. When you start the switch, the switch initialization code reads the switch topology file stored in the SDR. (Refer to the *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference* for more information on the **Eannotator** and **Etopology** commands.)

An **expected.top** file in the /**etc**/**SP** directory of the primary node overrides the topology file specified in the SDR. (This is used when servicing or debugging the High Performance Switch. Refer to the *IBM RISC System/6000 Scalable POWERparallel Systems: Maintenance Information Volume 2* for more information.)

Annotating a Switch Topology File

Before a switch topology file is stored in the SDR, it should be annotated with **Eannotator**. **Eannotator** will update the switch topology file's connection labels with their current physical locations. If the **-O yes** flag is specified, the topology file is stored in the SDR. Running **Eannotator** makes the switch hardware easier to debug because the switch diagnostics information is printed based on physical locations. The base topology files contain logical locations. **Eannotator** is not a necessary step for switch operations. For more details on **Eannotator**, refer to the *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference*.

Using SMIT:

TYPE	**smit**
	▲ The System Management menu appears
SELECT	9076 SP Systems Management
	▲ The 9076 SP Systems Management menu appears.
SELECT	9076 SP Cluster Management
	▲ The 9076 SP Cluster Management menu appears.
SELECT	Perform Switch Operations
	▲ The Perform Switch Operations menu appears.
SELECT	Topology File Annotator
	▲ The Topology File Annotator menu appears.

At this menu:

- Input the name of the topology file to be annotated

- Input the name to which you want the annotated topology file stored
- Enter **yes** to storing the topology file in the SDR

Using the Command Line: To annotate a switch topology file, use the command:

```
Eannotator -F input_file -f output_file -O yes
```

For example, to annotate a topology file for a two-switch or maximum 32-node system:

```
Eannotator -F expected.top.2nsb.0isb.0 -f expected.top.2nsb.0isb.0 -O yes
```

Storing the Switch Topology File in the SDR

Use **Etopology** to store the topology file in the SDR.

Using SMIT:

TYPE **smit**

▲ The System Management menu appears

SELECT 9076 SP Systems Management

▲ The 9076 SP Systems Management menu appears.

SELECT 9076 SP Cluster Management

▲ The 9076 SP Cluster Management menu appears.

SELECT Perform Switch Operations

▲ The Perform Switch Operations menu appears.

SELECT Fetch/Store Topology File

▲ The Fetch/Store Topology File menu appears.

SELECT Store a Topology File

▲ The Store a Topology File menu appears.

Input the topology file for your configuration.

Using the Command Line: To store a topology file, use the **Etopology** command:

```
Etopology topology_filename
```

For example, to store a topology file for a two-switch or 32-node maximum configuration:

```
Etopology expected.top.2nsb.0isb.0
```

Starting the High Performance Switch

Once all the processor nodes are running, you can start the High Performance Switch.

1. Check that all processor nodes are running (hostResponds indicator). Refer to "Determining If Nodes Are Operating: All Node Summary" on page 141 for a discussion of the ways you can use the System Monitor to do this.
2. Start the High Performance Switch. On the control workstation, enter:

   ```
   Estart
   ```

(**fault_service_Worm_RTG**, the fault-handling daemon for the High Performance Switch that checks, initializes, and prepares the switch for operation, must be running on the primary node before you do an **Estart**. Use the **rc.switch** command to start the daemon.) When **Estart** finishes, it sends a message similar to the following example verifying that it completed:

```
Switch initialization started on tserv10.hpssl.kgn.ibm.com
Initialization successful for 16 nodes
Switch initialization completed
```

If the switch initialization process encounters any problems, other messages may appear. If this happens, refer to diagnosis information in the *IBM RISC System/6000 Scalable POWERparallel Systems: Diagnosis and Messages Guide* and resolve any errors. See the *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference* for more information on **Estart**.

If the primary node is unavailable (powered-down, rebooted, and so forth), **Eprimary** should be run to designate a new primary. (This should be followed by **Estart** to restart the High Performance Switch. On a primary node failure, the High Performance Switch and the "switchResponds" System Monitor display are in an inconsistent state until **Eprimary** and **Estart** are run.) Whenever a new switch primary is assigned, the old primary must be rebooted or powered off.

You can monitor switch connectivity to the nodes by using the switchResponds indicator available through the Node Layout window under the SP System Monitor or by querying the **switch_responds** object in the SDR, as follows:

```
SDRGetObjects switch_responds
```

A value of 1 indicates the node is available on the switch.

You can also monitor switch status with the Switch Front panel layout, explained in "Monitoring and Controlling the High Performance Switch" on page 128.

Isolating Nodes from the High Performance Switch

Node isolation function provides for isolating nodes that will be rebooted or powered-off from the High Performance Switch in order to prevent such nodes from causing a switch fault. The capability of allowing the node to rejoin the High Performance Switch once the node is again operational is also provided. Isolating the node from the High Performance Switch is referred to as "fencing". Allowing the node to rejoin the High Performance Switch is known as "unfencing".

Note: The primary node cannot be fenced.

Node isolation can be accomplished through either a command interface or with the System Monitor GUI. See Chapter 7, "Using the SP System Monitor" on page 117 for information on using the System Monitor GUI to isolate nodes. This section presents the task flow and commands for isolating nodes with the command interface. **Efence** and **Eunfence** can operate on nodes regardless of system boundaries if the global option **-G** is used.

Step 1: Isolate the Node from the High Performance Switch

On the control workstation, use the **Efence** command to isolate nodes. For example:

```
Efence node1 node2
```

When you issue the **Efence** command, you can use **-autojoin** to optionally specify if the nodes should rejoin the High Performance Switch once they are again operational. For example:

```
Efence -autojoin node1 node2
```

A return code will indicate success or failure. In addition, a message will be displayed to indicate success or failure.

Note: If you specify the **-autojoin** option, a node cannot return to the High Performance Switch network if a switch initialization (**Estart**) or switch fault occurred while the node was isolated. If this condition occurs, you must perform another switch initialization to return the node to the High Performance Switch.

Step 2: Deactivate the Node

Enter the appropriate command to reset, power off, or shut down the node.

Step 3: Return the Node to the High Performance Switch

If you did not specify that the node automatically rejoin the High Performance Switch in "Step 1: Isolate the Node from the High Performance Switch," use the **Eunfence** command to return the node to the High Performance Switch network. For example:

```
Eunfence node1 node2
```

A return code will indicate success or failure. In addition, a message will be displayed to indicate success or failure.

Note: A node cannot return to the High Performance Switch network if a switch initialization (**Estart**) or switch fault occurred while the node was isolated. If this condition occurs, you must perform another switch initialization to return the node to the High Performance Switch.

Setting the Run Phase of the High Performance Switch

Node isolation requests are queued until they can be serviced. Such requests can be serviced only at the end of the current "run phase" of the High Performance Switch. Therefore, the duration of the run phase determines how quickly the system responds to node isolation requests. Use the **Eduration** command to set the run phase for the switch in days, hours, or minutes. For example:

```
Eduration [days days] [hours hours] [minutes minutes] [-h]
```

Optionally, you can use a SMIT panel to set the run phase duration.

Using SMIT

TYPE **smit**

▲ The System Management menu appears

SELECT 9076 SP Systems Management

▲ The 9076 SP Systems Management menu appears.

SELECT 9076 SP Cluster Management

▲ The 9076 SP Cluster Management menu appears.

SELECT Perform Switch Operations

▲ The Perform Switch Operations menu appears.

SELECT Change/Show Switch Run Phase Duration

▲ The Change/Show Switch Run Phase Duration menu appears.

SELECT Set Switch Run Phase Duration

▲The Set Switch Run Phase Duration menu appears.

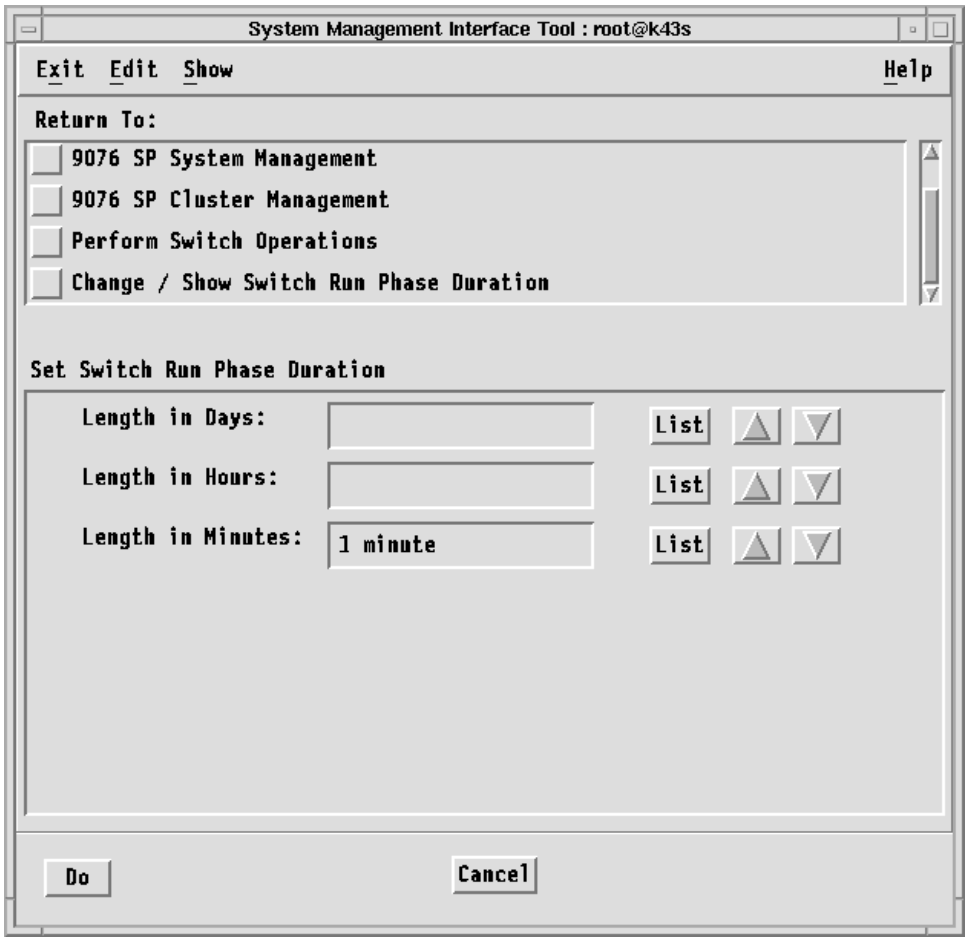


Figure 42. The Set Switch Run Phase Duration SMIT menu.

See the *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference* for more information on the **Efence**, **Eunfence**, and **Eduration** commands.

Chapter 10. Parallel Management Commands

This chapter describes commands provided with the SP system that are useful for performing common tasks on multiple nodes in parallel.*

These commands, along with the **sysctl** function and SP authenticated remote commands, provide for management of the SP System from a single point of control.

Using dsh to Execute Parallel Management Commands

You can use **dsh** from a single point of control to execute commands on:

- All nodes in the current system partition (**-a** option)
- All nodes in the physical SP system (**-aG** option)
- Selected nodes (**-w** option)

The target nodes need not be limited to those in your SP System. **dsh** will execute commands on any host in your network to which you can issue commands through the SP authenticated remote command **rsh** (see “Security Considerations” on page 171). Although **dsh** uses **rsh** to issue remote commands, **dsh** offers better performance than an **rsh** loop because the remote commands run concurrently.

Inputting to dsh

dsh provides two ways for you to specify input:

1. **dsh** reads lines from standard input and executes each line as a command via **rsh**. Specify commands in **rsh** syntax. Pipes, filters, output redirection, and paths can be used on the remote nodes in the same way they are used in **rsh**.

   Any input line beginning with an exclamation point is passed directly to the shell on the local host.

   As an example, the following **dsh** command targets three nodes, reads commands from stdin, and filters the output of the **ps** command remotely:

   ```
   dsh -w host1,host2,host3
   dsh > ps -ef "|" grep root
   ```

2. **dsh** accepts commands specified on the command line. For example, to issue the **ps** command on three nodes and filter the results remotely, specify:

   ```
   dsh -w host1,host2,host3 ps -ef "|" grep root
   ```

* The parallel management commands described in this chapter were influenced by (but are not an implementation of) the tools described in Gropp, William and Ewing Lusk. 1994. Scalable UNIX tools on parallel processors. In *Proceedings of the Scalable High Performance Computing Conference*, 56-62. IEEE.

© Copyright IBM Corp. 1995

Targeting Nodes for dsh

dsh sends commands to a node set called the **working collective**. **dsh** assembles the working collective from the first existence of one of these sources:

1. A list of hostnames you specify on the command line with the **-a**, **-G**, **-w** flags.
2. The contents of a file you specify in the **WCOLL** environment variable.

If neither of these exist, an error has occurred and no commands are issued.

You can have commands run on the working collective concurrently or in sequence. Default processing is done concurrently, but you can specify a maximum number of nodes for concurrent execution to prevent system degradation. When this maximum number, or *fanout* is reached, results from outstanding **rsh**s are awaited before any further commands are run. These **rsh** results are displayed as soon as they return from the remotely executed commands.

This example specifies a maximum of 8 concurrent commands even though the working collective is defined as the entire system partition.

```
dsh -f 8 -a  cat /var/adm/SPlogs/sysman/"*"config.log"*" | pg
```

Note: Because **dsh** starts processes for each host in the working collective, there may be cases where a smaller fanout will result in less contention for resources on the local machine, and thus better overall performance, than the greater concurrency of a larger fanout.

Output from dsh

dsh displays returned information grouped by hostname. Stdout of the remotely executed commands goes to the stdout of **dsh**, and stderr of remotely executed commands goes to the stderr of **dsh**.

All **dsh** stderr and stdout lines are prefixed by the name of the host that sent them. You can format **dsh** stderr and stdout lines by piping them to **dshbak**, which strips off the hostnames and displays the lines grouped by host in alphabetic sequence, as shown in this example:

```
dsh -w host1,host2,host3 cat /etc/passwd 2>&1 | dshbak
```

Alternatively, **dshbak** can be specified with the **-c** option. This will cause any identical output from two or more nodes to be shown only once, with the hostnames displayed above the output. Remember, however, that the most efficient way to filter large amounts of output from parallel commands is to filter on the nodes before the output is returned to the workstation from which the parallel command was issued.

Error Handling

No special error recovery is provided for **dsh** command failure on remote hosts. If **dsh** finds that a node in the working collective is down, no further commands will be sent to that node unless you specify the **-c** (continue) flag on the command line. If hosts are down, the underlying **rsh** will time out in approximately 2.5 minutes.

Security Considerations

dsh uses the SP authenticated remote command **rsh**. This version of **rsh** attempts to authenticate users via Kerberos. If authentication fails, the AIX version of **rsh** is used instead. The authenticated version of **rsh** allows system administration without the requirement for the existence of a root level **.rhosts** file.

The hostlist Command

The **hostlist** command can be used to specify hostnames for parallel command execution via **dsh** or **sysctl**. The command writes hostnames to stdout based on various criteria:

- The contents of a **dsh** working collective file
- The contents of a POE hostname file
- All the nodes in the current system partition (**-a** option)
- All the nodes in the physical SP system (**-aG** option)
- Ranges of slots on particular frames
- A set of nodes to exclude
- A range of node numbers
- Explicitly listed hostnames

Before the names are written, there is an optional check of whether the nodes are responding.

The output from **hostlist** can be written to a file, which can then be used as a working collective file for **dsh**. Both **dsh** and **sysctl** can read hostnames from stdin, so the output of **hostlist** can be piped directly to **dsh** or **sysctl**.

Examples

To issue a command to the node in the first slot in each of the first 4 frames:

```
hostlist -s 1-4:1 | dsh -w - command
```

To run the program on the nodes on all slots for frame 1 and slots 1–3 for frame 3, and also on host **otherone**:

```
hostlist -n 1-16,33-35 -w otherone > /tmp/wcoll
WCOLL=/tmp/wcoll dsh program
```

To determine all the nodes in the SP system on which one is authenticated, and to bypass the nodes that are not responding and “badnode”, issue:

```
hostlist -av -e badnode | sysctl -c - -L whoami | dshbak -c
```

Note:

> The **-L** option on **sysctl** is necessary if the **dshbak** filter is used. It causes the output lines of **sysctl** to be formatted with the hostname preceding each line.

Output may look like this:

```
HOSTS===========================================================
host1             host2             host3             host4
host5             host6             host8
================================================================
joe.@A.B.C

HOSTS===========================================================
host7
================================================================
unknown
```

If the **-c** option is not specified, output will not be collapsed:

```
HOST: host1
===========
joe.@A.B.C

HOST: host2
===========
joe.@A.B.C
⋮
```

List of Parallel Management Commands

The SP system provides several parallel management commands. They all have a common syntax and are based on **dsh**, **sysctl**, **hostlist**, and **dshbak**. More complete descriptions of each is provided in the *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference*.

The syntax of the parallel management commands is:

command_name remote_host_specification cmd_args_flags

where

command_name The name of a parallel command

remote_host_specification
Indicates which hosts the command should be executed on. Can be a range of slots, a list of hostnames, a set of flags to the **hostlist** command, or read from stdin.

cmd_args_flags The flags and arguments to the command

Where appropriate, the output from the parallel management commands is filtered as described above.

The following parallel management commands are based on **dsh**.

pexec

This application issues a command in parallel on the specified hosts. Any arguments to the commands are passed along with the command name. Results are filtered as described in “Output from dsh” on page 170. For example, to show users on nodes in slots 5–8:

```
pexec 5-8 who
```

pexscr

This application executes commands in parallel on different hosts, including the local host. Unlike **pexec**, the commands may differ on different hosts. Lines consisting of *hostname: command* are read from stdin. When the lines have been read, each command is executed on the corresponding host in parallel. Remote commands are executed via **rsh** and local commands are executed directly. For example, you could create the following file /**etc**/**doit** on r05n09:

```
r05n09: /usr/lpp/ssp/rcmd/bin/rcp /tmp/bozoid tserv11:/tmp/bozoid
r05n09: /usr/lpp/ssp/rcmd/bin/rcp /tmp/bozoid r05n01:/tmp/bozoid
r05n01: backup_script
r05n15: backup_script
```

Then, to copy the local file /**tmp**/**bozoid** to tserv11 and r05n01 and simultaneously start backup scripts on r05n01 and r05n15:

```
pexscr < /etc/doit
```

pcp

This command distributes a local file or directory to remote hosts in parallel. It uses the SP authenticated remote command **rcp**. To use this command, you must be authorized to issue commands of the form:

```
rcp localfile remotehost: remotefile
```

for each of the hosts to which the file is to be copied. To recursively copy /**etc**/**new**/ on the local host to /**etc**/**new1**/ on the host in the first slot of each of the first four frames:

```
pcp "-s 1-4:1" -r /etc/new/ /etc/new1/
```

p_cat

This application issues the **cat** command in parallel on the specified hosts. Any arguments to the command are passed along with the command name. Results are not filtered. For example,

```
p_cat "-w host1,host2,host3" /etc/file > localfile
```

copies the contents of the /**etc**/**file** files on each of the hosts to the **localfile** file on the local host. **localfile** would consist of unmodified lines from /**etc**/**file** on **host1**, on **host2**, and on **host3** concatenated together.

pls, pfind, prm, pps, pmv

These commands issue the **ls**, **find**, **rm**, **ps**, or **mv** commands in parallel on the specified hosts. Any arguments to the AIX commands are passed along with the command name. Results are filtered. For example,

```
pls "-av -e badnode" /
```

lists the contents of the root directory on all the responding hosts in the SP system with the exception of “badnode”. Note that the **-i** flags to **mv** and **rm** are not supported because **dsh** doesn't propagate the stdin of the remote hosts.

Examples: For example, a working collective could be set up with **hostlist**:

```
hostlist -av > /tmp/wcoll
export WCOLL=/tmp/wcoll

pps "" -ef
```

would list information about all the processes on the nodes in the working collective.

```
pfind "" /usr/lpp/ssp/bin/  -name "pfind" -print
```

may result in the following:

```
HOSTS ===========================================================================
r05n01              r05n03dx            r05n05              r05n07
r05n09              r05n10              r05n13              r05n14dx
r05n15              r05n16
 ===============================================================================
 /usr/lpp/ssp/bin/pfind
```

```
pmv "" /tmp/foo /tmp/newfoo
```

would rename /**tmp**/**foo** on all the working collective.

```
pls "" /tmp/\\*foo
```

could result in the following output:

```
HOSTS ===========================================================================
r05n01              r05n03dx            r05n05              r05n07
r05n09              r05n10              r05n13              r05n14dx
r05n15              r05n16
===============================================================================
/tmp/newfoo
```

Note that the shell metacharacter is escaped twice, once to protect it from the shell, and once to protect it when the **rsh**'s are done by **dsh**.

```
prm "" /tmp/newfoo
```

would remove /**tmp**/**newfoo** from all the nodes in the working collective.

ppred

This application issues a **ksh** test on each node in parallel and runs commands based on the results of the test. For example,

```
ppred 1-16 '-f /etc/foo' 'echo Here' 'echo Not here'
```

could result in output like the following:

```
HOSTS============================================================
host1             host2             host3             host4
host5             host6             host8
=================================================================
Here

HOSTS============================================================
host7             host9             host10            host11
host12            host13            host14            host15
host16
=================================================================
Not here
```

pfps

pfps is a parallel process find application. It takes arguments similar to those of the **find** command, but operates on processes instead of files. In addition, access control list-based permissions to **nice** or **kill** processes is provided. For example, to signal all the "daemond" processes in the SP system partition in parallel, a user that was in the /**etc/sysctl.pfps.acl** file on each node could issue the following:

```
pfps "-a" -tn sysctld -kill HUP
```

To list the **sysctl** daemon processes on nodes **r05n15** and **r05n16**, the following command:

```
pfps "-w r05n15,r05n16" -tn sysctld -print
```

produces this output:

```
HOSTS ===========================================================================
r05n15
===============================================================================
USER       PID %CPU %MEM   SZ  RSS    TTY STAT    STIME  TIME COMMAND
root     11172  0.0  1.0  452  420      - S    11:54:47  0:00
/usr/lpp/ssp/bin/sysctld -l /var/adm/SPlogs/sysctl/sysctld.log
root     12497  0.0  1.0  300  360      - S    06:24:19  0:00
/usr/lpp/ssp/bin/sysctld -l /var/adm/SPlogs/sysctl/sysctld.log

HOSTS ===========================================================================
r05n16
===============================================================================
USER       PID %CPU %MEM   SZ  RSS    TTY STAT    STIME  TIME COMMAND
root      7089  0.0  1.0  448  420      - S    11:54:48  0:00
/usr/lpp/ssp/bin/sysctld -l /var/adm/SPlogs/sysctl/sysctld.log
root      8423  0.0  1.0  296  360      - S    13:01:31  0:00
/usr/lpp/ssp/bin/sysctld -l /var/adm/SPlogs/sysctl/sysctld.log
root     10418  0.0  1.0  512  504      - S
```

pdf

pdf is a parallel command that performs function similar to **df**. It provides more information than the **df** command. To get filesystem information from all the nodes on the first slots of each of four frames:

```
hostlist -s 1:1-4 > /tmp/wcoll
export WCOLL=/tmp/wcoll
pdf ""
```

To find information on the /**var** filesystem on the nodes in the working collective:

```
pdf "" /var
Filesystem               Size-KB Used-KB Free-KB %Free  iUsed  iFree %iFree
==================       ======= ======= ======= =====  =====  ===== ======
HOST: r05n01
============
/var                        8192    2452    5740   71%    279   1769    87%

HOST: r05n03dx
==============
/var                        8192    1988    6204   76%    214   1834    90%

HOST: r05n05
============
/var                        8192    2080    6112   75%    223   1825    90%

HOST: r05n07
============
/var                        8192    2080    6112   75%    223   1825    90%

HOST: r05n09
============
/var                        8192    2088    6104   75%    223   1825    90%

HOST: r05n10
============
/var                        8192    2448    5744   71%    286   1762    87%

HOST: r05n13
============
/var                        8192    2220    5972   73%    229   1819    89%

HOST: r05n14dx
==============
/var                        8192    2012    6180   76%    221   1827     90%

HOST: r05n15
============
/var                        8192    2368    5824   72%    220   1828    90%

HOST: r05n16
============
/var                        8192    2100    6092   75%    225   1823    90%
```

pfck

pfck displays file system information in parallel based on various usage criteria. For example, to find information on all the file systems on the hosts in POE host list file **hostlist** that have used more than 100000 KB:

```
MP_HOSTFILE=./hostlist pfps "" -s 100000
```

To display the filesystems on hosts **r05n16** and **r05n15** that have more than 90% of their space used:

```
pfck "-w r05n15,r05n16" -pu 90
Filesystem                Size-KB Used-KB Free-KB %Free  iUsed  iFree %iFree
==========                ======= ======= ======= =====  =====  ===== ======
HOST: r05n15
============
/usr                       331776  327640    4136    2%  17673  66295    79%

HOST: r05n16
============
/usr                       331776  327640    4136    2%  17673  66295     79%
```

To find all the /**var** filesytems in the SP system that are 95% full:

```
pfck "" -pu 95 | egrep "(var|HOST)"

 Filesystem                Size-KB Used-KB Free-KB %Free  iUsed  iFree %iFree
 ==========                ======= ======= ======= =====  =====  ===== ======
 HOST: r05n10
 /var                       335872  331412    4460    2%  17819  66149    79%
 HOST: r05n13
 /var                       335872  331632    4240    2%  17820  66148    79%
 HOST: r05n14dx
 /var                       335872  331388    4484    2%  17817  66151    79%
```

Chapter 11. Managing SP Resources

This chapter describes the Resource Manager, its administration, and use. Task information is presented for:

- Understanding the Resource Manager
- Installing the Resource Manager
- Defining the Resource Manager configuration
- Starting the Resource Manager
- Reconfiguring the Resource Manager
- Stopping the Resource Manager

For information on Accounting related to Resource Manager, refer to “Accounting for Exclusive Use of Node Jobs” on page 268.

For many of the tasks involved in managing the Resource Manager you have the option of using either a command or the SMIT graphical user interface. Note that when you are using SMIT there is a list option (PF4) available for dialog screens and that help screens are available for all Resource Manager menus and fields.

Understanding the Resource Manager

The Resource Manager is a server that runs on one of the SP nodes. It prevents parallel jobs from interfering with each other and reports job related node information.

When applications such as the Parallel Operating Environment (POE) and LoadLeveler make a request, the Resource Manager identifies a group of nodes that meet the requested criteria. In doing so it:

- Allows POE applications to request dedicated or shared use of nodes or adapters on nodes. LoadLeveler always requests shared use of nodes unless the High Performance Switch adapter is used in user space mode, in which case it requests shared use of the node but dedicated use of the adapter. Requesting dedicated use of an adapter stops other jobs from using the adapter specified by the job (one of Ethernet, FDDI, FCS, Token Ring, the High Performance Switch used in user space mode and the High Performance Switch used in IP mode). It does not stop the node from being used by another job using a different adapter. This means that if a job requests and is allocated shared use of nodes A and B but dedicated use of the High Performance Switch-Adapter2's user space part on those nodes, then no other job will be allocated the High Performance Switch-Adapter2's user space part on A and B for as long as the first job is running. However, another job needing shared use of nodes A and B but dedicated or shared use of the High Performance Switch-Adapter's IP part on those nodes, will be allocated those nodes since the request does not conflict with the first job's resource criteria.
- Allows applications to be run interactively or submitted via a batch scheduler (LoadLeveler only).
- Controls accounting for dedicated node usage.
- Can create CSS authentication tables on identified nodes so that the authorized CSS user space job can run without interference from other jobs.

© Copyright IBM Corp. 1995

- Allows you to lock out users from nodes unless they request access via the Resource Manager. Refer to “Controlling User Login” on page 203 for more information.

A single point of control is provided for starting, stopping, and configuring the Resource Manager.

When the Resource Manager server is started, it automatically tries to start a backup on another node. If the primary Resource Manager server fails, this backup becomes the primary server and starts a backup on another node. Jobs will continue to run unaffected.

When the Resource Manager selects nodes for use by an application, it might have to execute certain functions (such as accounting, login control, or CSS authentication tables creation) on the selected nodes. To do so, it contacts the **inetd** daemons on those nodes which in turn spawn a Resource Manager daemon to perform the task.

A flat UID space is required for all clients connecting to the Resource Manager.

There are two Resource Manager log files:

- **/var/adm/SPlogs/jm/jmd_out** contains general information about the operation of Resource Manager daemons
- **/var/adm/SPlogs/jm/jmd_err** contains errors encountered by Resource Manager daemons

System Partitioning

Resource Manager function, as described above, is supported *within individual system partitions*. The Resource Manager operates within the scope of each system partition environment. Each partition may have its own Resource Manager server, but no functionality crosses system partition boundaries. The Resource Manager gets the correct node and adapter information for each system partition from the SDR automatically. Each system partition has its own configuraton database named on the control workstation. If the system is not partitioned, the *syspar_name* is the short host name of the control workstation and can be found by issuing **spget_syspar -n**.

Installing the Resource Manager

If you are not using the SP for parallel processing, you do not need to install the Resource Manager.

To install the Resource Manager, use **installp** to install **ssp.jm** of the IBM Parallel System Support Programs for AIX. **ssp.jm** may have been installed on the control workstation during system installation as described in the *IBM RISC System/6000 Scalable POWERparallel Systems: Installation Guide*. (To check if **ssp.jm** has already been installed on the control workstation, enter **lslpp -m | grep ssp.jm**)

If you choose to selectively install the Resource Manager, you must install it on the control workstation and every node of your SP System. Refer to the chapter on software maintenance in the *IBM RISC System/6000 Scalable POWERparallel Systems: Installation Guide*.

When you install the Resource Manager, the **/etc/services**, **/etc/inetd.conf**, and **/etc/inittab** files are updated. Backup copies of the original files are stored in **/etc/services.bak.jm**, **/etc/inetd.conf.bak.jm**, and **/etc/inittab.bak.jm**, respectively.

- Two lines identifying Resource Manager ports are added to **/etc/services**:

  ```
  jmd             4086/tcp
  ijmd            4173/tcp
  ```

  These port numbers may be changed to any other unused port number as long as it is consistent on all the nodes of your SP System. Also, any other machines needing services from the Resource Manager on your SP must update their **/etc/services** file to contain this jmd port number.

- The following entry is added to **/etc/inetd.conf**:

  ```
  ijmd    stream  tcp      nowait  root    /usr/bin/jmd jmd
  ```

- The following entry is added to **/etc/inittab**:

  ```
  jm_setupcss:2:boot:/usr/lpp/ssp/bin/jm_setupcss
  ```

You can verify that all the Resource Manager executables and required links have been properly installed on the control workstation and SP nodes, either from SMIT panels or by using the the verification script, **jm_install_verify**.

Using SMIT:

TYPE	**smit SP_verify**
	▲ The Installation/Configuration Menu appears.
SELECT	SSP Resource Manager Installation
PRESS	Enter

Using the Command Line: This command must be executed from the control workstation as root and resides in **/usr/lpp/ssp/bin**.

Enter the command as follows:

```
jm_install_verify
```

Optional flags

-l *filename*	Specifies log file name (Default: /var/adm/SPlogs/jm_install_verify.log).
-q	Suppress messages.
-h	Print Usage information

See the *IBM RISC System/6000 Scalable POWERparallel Systems: Diagnosis and Messages Guide* for more information and examples.

Defining the Resource Manager Configuration

Begin by classifying the nodes on the SP System as either:

- Nodes not used for parallel jobs (**serial nodes**)

or

- Nodes used for parallel jobs (**parallel nodes**).

Serial Nodes

The Resource Manager does not allocate serial nodes; however, you may wish to record:

- Where the batch serial jobs run (**batch serial nodes**)
- Where the users are allowed to log into (**interactive serial nodes**)
- Where either of the above will be allowed (**general serial nodes**)

These serial nodes form the serial pool which is identified by a *pool_id* of -1 and an associated description string.

A node defined to the Resource Manager as a member of the serial pool cannot also be defined as a member of any parallel pools described below.

Parallel Nodes

You should group parallel nodes into parallel pools, since parallel applications indicate their resource requirements by specifying the pools from which nodes are to be allocated.

There can be multiple parallel pools. Any number of nodes may form a pool, and any meaningful group of characteristics can be the basis for defining a pool. You might form one pool from all wide nodes and another from thin.

Each node can belong to only one pool. Each parallel pool is identified by a unique pool_id, greater than -1, and an associated description string. Each node in the parallel pool should now be identified as being able to run:

- Parallel jobs submitted through LoadLeveler (member of the **batch subpool**)
- Parallel jobs started interactively (member of the **interactive subpool**)
- Either of the above (member of the **general subpool**)

If you identify all nodes in a pool as general nodes (belonging to the general subpool), you may experience delayed batch processing due to heavy interactive use. For this reason it may be preferable to set aside some of the nodes in the pool for batch use only. Similarly, interactive users may be denied access to nodes due to heavy batch load, so you may choose to set aside some of the nodes in the pool for interactive use only.

When the Resource Manager allocates nodes for interactive use, it searches through the interactive nodes (members of the interactive subpool) of the specified pool first, and then the general nodes, to satisfy the user's request. Similarly, when the Resource Manager allocates nodes for batch use, it searches through the batch nodes (members of the batch subpool) of the specified pool first, followed by the general nodes.

Refer to “Step 3: Configure the Resource Manager” on page 184 to specify the pool and subpools for your nodes.

Using the Resource Manager

You can use the Resource Manager for both interactive and batch support.

Interactive Support

Consider the user who is logged onto one of the serial interactive nodes, and is using POE to run a parallel job interactively on the SP System. The POE partition manager conveys the job's node requirements to the Resource Manager. The Resource Manager, in turn, allocates the appropriate parallel nodes from the parallel pools. Refer to *IBM AIX Parallel Environment: Operation and Use* for more information about using the POE.

Batch Support

If users want to submit parallel jobs to LoadLeveler, the System Administrator must perform additional LoadLeveler configuration for interaction with the Resource Manager as described in *IBM LoadLeveler: Administration and Installation*.

Example Configuration

Figure 43 shows a a possible configuration of a SP System with 16 nodes. This system has three parallel pools. The pool with id=0 has six nodes: nodes 1 and 2 assigned to the general subpool, nodes 3 and 4 assigned to the batch subpool and nodes 5 and 6 assigned to the interactive subpool. The pool with id=1 has seven nodes: nodes 7, 8 and 9 assigned to the general subpool, nodes 10 and 11 assigned to the batch subpool and nodes 12 and 13 assigned to the interactive subpool. The pool with id=2 has three nodes, 14, 15, 16, all assigned to the general subpool.

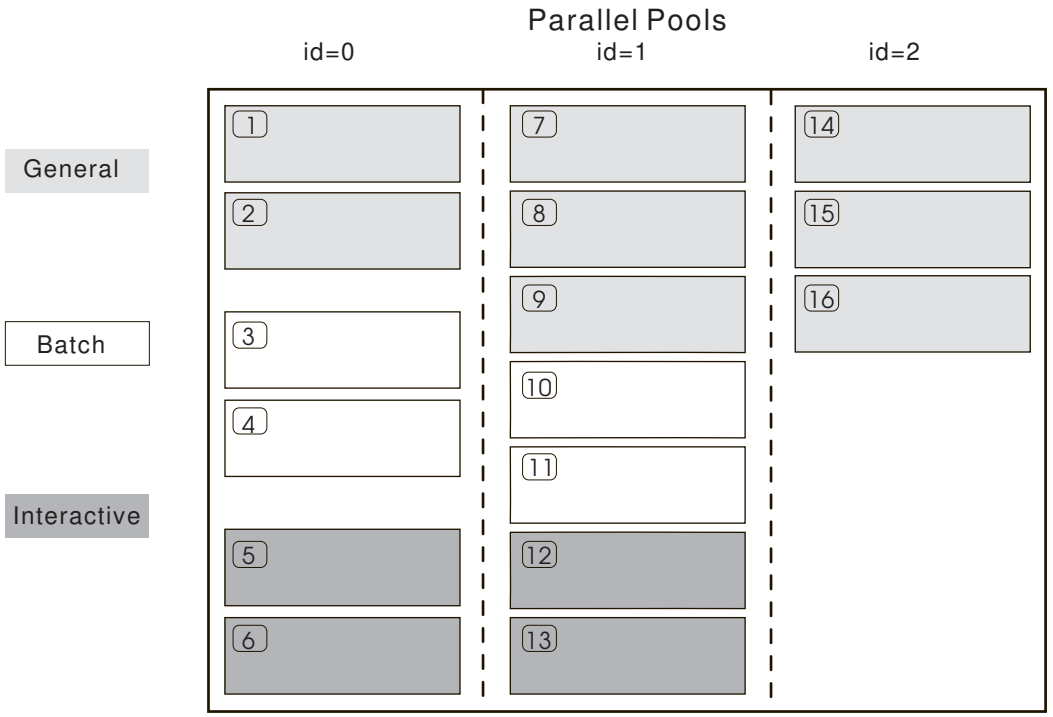


Figure 43. Example Resource Manager Configuration

Given this configuration, if an interactive job requested two nodes from Pool 1, it would receive nodes 12 and 13, if they were available. If an interactive job requested 4 nodes from pool 0, it would receive nodes 1, 2, 5, and 6. If a batch job requested 2 nodes from pool 2, it would receive two out of nodes 14,15,16. If a batch job requested 5 nodes from pool 1, it would receive nodes 7, 8, 9, 10 and 11.

Starting the Resource Manager

Follow these steps to start the Resource Manager within a system partition on your SP system.

Step 1: Run Login Access Control

If you wish to block users from parallel nodes unless they request access via the Resource Manager, then run the **spacs_cntrl** command as discussed in "Controlling User Login" on page 203. (See **spacs_cntrl** in the *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference* for command syntax.)

Step 2: Ensure SDR Is Running

Run the verification test that checks for correct operation of the System Data Repository. You can run this verification test from SMIT panels or from the command line.

Using SMIT:

TYPE	**smit SP_verify**
	▲ The Installation/Configuration Menu appears.
SELECT	SSP System Data Repository
PRESS	Enter

Using the Command Line:

Enter:

```
SDR_test
```

Refer to the *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference* if verification test fails.

Step 3: Configure the Resource Manager

The Resource Manager keeps all configuration data in an ASCII file named **/etc/jmd_config**. The system partition data is differentiated by adding a suffix to this file which corresponds to the current system partition environment (this environment can be determined by issuing **spget_syspar -n**). To define the Resource Manager configuration, edit the **/etc/jmd_config**.*syspar_name* file, which represents the current system partition environment, and assign your SP nodes to pools and subpools as described in "Defining the Resource Manager Configuration" on page 181. If you edit the **/etc/jmd_config**.*syspar_name* file manually, you must make sure the system partition named is correct (it should match the output of the **spget_syspar -n** command).Appendix C, "Sample Files" on page 399 shows a sample **jmd_config** file.

Using SMIT: As an alternative to editing the **/etc/jmd_config**.*syspar_name* file, you can use SMIT menus (available on the control workstation only) to define Resource Manager configuration. If you edit the **/etc/jmd_config** file using the SMIT interface, the correctly named **/etc/jmd_config**.*syspar_name* file is created for you. Once you have installed **ssp.jm** on all the nodes, you can configure the Resource Manager at the same time you use SMIT to configure the rest of the SP

System. Detailed information on every field in the SMIT panels can be obtained via Help.

Note: Any data defined via SMIT is recorded in the /**etc**/**jmd_config**.*syspar_name* file. Therefore, the **jmd_config**.*syspar_name* file should always be consistent with the current configuration data.

To *initially* configure the Resource Manager:

TYPE **smit**

▲ The System Management menu appears.

SELECT 9076 SP System Management

▲ The 9076 SP System Management menu appears.

SELECT 9076 SP Resource Manager

▲ The 9076 Resource Manager menu appears.

The fastpath invocation for the 9076 Resource Manager menu is:

```
smit RM_options
```

```
                  9076 SP Resource Manager

□ Change/Show Configuration Data
□ Start the Resource Manager
□ Reconfigure the Resource Manager
□ Stop the Resource Manager
```

SELECT Change/Show Configuration Data

▲ The Configuration Data menu appears.

The fastpath invocation for the Configuration Data menu is:

```
smit RM_configure
```

```
                   Configuration Data

□ Change/Show Resource Manager Server List
□ Change/Show Access Control
□ List All Pool Data
□ Add a Pool
□ Delete a Pool
□ Change/Show a Pool
```

SELECT Change/Show Resource Manager Server List

▲ The Change/Show Resource Manager Server List menu appears.

The fastpath invocation for the Change/Show Resource Manager Server List menu is:

```
smit server_list
```

```
                Change/Show Resource Manager Server List

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

     1st Node in Server List              [none]   + required
     2nd Node in Server List              [none]   +
     Additional Nodes in Server List      [none]     +
     Update Configuration Now ?           [no]       +
```

- The dialog fields show the current values found within the SDR. The initial configuration would show values of **none**.
- Enter the node names or use the list option to select the node names desired for each field.
- Enter **no** for `Update Configuration Now?`

SELECT Change/Show Access Control

▲ The Change/Show Access Control menu appears.

The fastpath invocation for the Change/Show Access Control menu is:

```
smit access_control
```

```
                        Change/Show Access Control

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

     Access Control                [no]          +
     Update Configuration Now ?    [no]          +
```

- The default value of the `Access Control Field` will be **no**.
- Use the list to select the value **yes**.
- Enter **no** for `Update Configuration Now?`

SELECT Add a Pool (for each pool to be defined)

▲ The Add a Pool menu appears.

The fastpath invocation for the Add a Pool menu is :

```
smit add_pool
```

```
                          Add a Pool

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

□ Pool ID                          []
□ Description/Attribute            []
□ Delete nodes if previously defined ?      [yes] +
□ Batch Nodes                  [none]      +
□ Interactive Nodes            [none]      +
□ General Nodes                [none]      +
□ Update Configuration Now ? [no]      +
```

- The data fields initialize to blank or **none**.
- Enter the *pool_id* and *attribute* string.
- Enter the node names for the subpools or use the list option to select from valid nodes on the system.
- Enter **no** for `Update Configuration Now?`

(After using SMIT to define the initial configuration data, see the "Using SMIT" section of Step 4 to start the Resource Manager.)

Step 4: Start the Resource Manager

The **jm_start** command must be executed by root on the control workstation and operates within the scope of the current system partition environment.

Enter:

```
jm_start {-i}
```

The **jm_start** command reads the data in the /**etc**/**jmd_config**.*syspar_name* file into the SDR and starts the primary Resource Manager server on one of the nodes indicated. This server then starts up a backup on the next available node in the list of server candidates.

jm_start returns a message stating which **jmd_config** file (for which system partition) was used by the command.

If **jm_start** successfully starts the primary Resource Manager server, it returns a message indicating its location. If a backup is not successfully started, an additional message is issued and the primary Resource Manager server continues to run, attempting periodically to start the backup. If you do not have a backup running, and the primary Resource Manager server dies for any reason, there will be no recovery and the Resource Manager server must be restarted manually. Refer to "Resource Manager Recovery" on page 192 for more information.

Use the **-i** flag only if the **jm_start** command cannot determine whether the Resource Manager server is already running. The **-i** flag does not check if the Resource Manager server is already running, therefore it is best not to use this option unless you are sure the Resource Manager server is not running on any node.

Using SMIT: To start the Resource Manager using SMIT, select the Start the Resource Manager option from the 9076 SP Resource Manager menu.

The fastpath invocation for the 9076 Resource Manager menu is:

```
smit RM_options
```

```
                  9076 SP Resource Manager

□ Change/Show Configuration Data
□ Start the Resource Manager
□ Reconfigure the Resource Manager
□ Stop the Resource Manager
```

Step 5. Verify that the Resource Manager Server Is Ready to Accept Requests

You can verify that the Resource Manager server is ready to accept requests from clients either from SMIT panels or by using the the verification script, **jm_verify**.

Using SMIT:

TYPE	**smit SP_verify**
	▲ The Installation/Configuration Menu appears.
SELECT	SSP Resource Manager Configuration
PRESS	Enter

Using the Command Line: You can use the verification script, **jm_verify**, to verify that the Resource Manager server is ready to accept requests. This command resides in /**usr**/**lpp**/**ssp**/**bin** and can be run on any node by any user and operates within the scope of the current system partition environment.

Enter the command as follows:

```
jm_verify
```

Optional flags

-l *filename*	Specifies log filename (Default: /var/adm/SPlogs/jm_verify.log).
-q	Suppress messages
-h	Print Usage information

See the *IBM RISC System/6000 Scalable POWERparallel Systems: Diagnosis and Messages Guide* for more information and examples.

Reconfiguring the Resource Manager

At some point you may decide to redefine your subpools and assign different nodes for batch, interactive, or general use, or change the list of candidate nodes for running the Resource Manager server, or change the setting of login access control. You must change the **jmd_config**.*syspar_name* file to reflect the new configuration and issue the **jm_config** command. (**jm_config** will return a message stating the name of the **jmd_config** file (for which system partition) the command is using.)

If you decide to enable (or disable) SP Exclusive Use Accounting, thereby modifying the **spacct_excluse_enable** attribute in the SP object of the SDR, when the Resource Manager is already running, be sure to reconfigure the Resource Manager by issuing **jm_config**. If you modify any SDR attributes that affect Node or Adapter objects, such as adding or deleting a frame, when the Resource Manager is already running, you must stop the Resource Manager and restart it in order to reread the SDR data.

Step 1: Edit the Resource Manager Configuration File or Update Configuration Information

Edit the /**etc/jmd_config**.*syspar_name* file to specify which nodes belong to the batch, interactive, and general subpools of your serial and parallel hardware pools. You can also change the list of nodes to be used for running the Resource Manager server, and the setting for Login Control.

Using SMIT: If you use the SMIT to interface to reconfigure the Resource Manager, the new and correctly named /**etc/jmd_config**.*syspar_name* file is created for you. To *change* the current Resource Manager pool configuration using SMIT, use the fastpath invocation for the Change/Show a Pool option:

```
smit configure_pool
```

▲ The Change/Show a Pool menu appears.

```
                        Change/Show a Pool

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

     Pool ID                          1
     Description/Attribute                  [widenode_userspace]
     Delete nodes if previously defined ?      [yes]    +
     Batch Nodes                  [none]      +
     Interactive Nodes            [r07n01 r07n03]
     General Nodes                    [none]      +
     Update Configuration Now ?       [no]  +
```

- Select a previously defined pool and view the current set of defined data.
- Enter the node names for the subpools or use the list option to select from valid nodes on the system.
- Enter **yes** for `Update Configuration Now?`

- If you do not wish to update configuration at this point, enter **no** for `Update Configuration Now?` At a later time you can use the Reconfigure the Resource Manager option on the 9076 SP Resource Manager menu to reconfigure according to the values you have just entered in the fields.

Step 2: Run jm_config

This command, which must be run from root on the control workstation, and operates within the scope of the current system partition environment, reads configuration data from the **jmd_config**.*syspar_name* file into the SDR. It checks the SDR for the location of the primary Resource Manager and then sends the Resource Manager a message telling it to reread the configuration data from the SDR. The new configuration takes effect with the next client request.

Enter:

```
jm_config
```

jm_config returns a message stating the name of the **jmd_config** file (for which system partition) the command is using.

Using SMIT: To reconfigure the Resource Manager using SMIT, select the Reconfigure the Resource Manager option from the 9076 SP Resource Manager menu. The fastpath invocation for the 9076 Resource Manager menu is:

```
smit RM_options
```

```
                    9076 SP Resource Manager

□ Change/Show Configuration Data
□ Start the Resource Manager
□ Reconfigure the Resource Manager
□ Stop the Resource Manager
```

The currently executing clients will not be affected by the change in configuration. The server recognizes the new configuration for new clients making a connection with the Resource Manager server.

Stopping the Resource Manager

To stop the primary and backup Resource Manager servers, use the **jm_stop** command. This command must be run by root on either the control workstation or on any of its associated nodes and operates within the scope of the current system partition environment.

Enter:

```
jm_stop {-q}
```

The q flag is optional. If not specified, the primary and backup Resource Manager server exit immediately.

The system can be quiesced in a less disruptive fashion by specifying the **-q** option, which tells the Resource Manager server to first wait for all existing jobs to terminate before exiting. The **jm_stop -q** command will return immediately. Any

subsequent Resource Manager job requests, as well as **jm_status** and **jm_config** commands, will fail. When all jobs have finished, the primary and backup Resource Manager server will exit. Issue **locate_jm** to determine if the Resource Manager servers have exited. Anytime after **jm_stop -q** has been issued, you can issue **jm_stop** (without **-q**) to cause the Resource Manager servers to exit immediately.

Using SMIT: The fastpath invocation for the Stop the Resource Manager menu is:

```
smit RM_stop
```

```
                  Stop the Resource Manager

□ Stop the Resource Manager immediately
□ Wait for existing clients to exit, then stop the Resource Manager
```

Locating the Resource Manager

To identify the nodes on which the primary and backup Resource Manager servers are running, use the **locate_jm** command. This command must be executed by root and has no parameters. It operates within the scope of the current system partition environment.

Querying the Resource Manager

Use the **jm_status** command to get information about defined pools or about jobs running on nodes allocated by the Resource Manager. **jm_status** can be executed by any user, on any workstation.

Enter the command as follows:

```
jm_status {-n control_workstation_name} {-P|-p pool_id|-j} {-v}
```

1. The **-n** flag indicates the name, or alias name, of the SP control workstation. This flag is optional. If you choose not to specify the **-n** flag, you can set the **SP_NAME** environment variable to indicate the control workstation name. If neither is specified, the control workstation name is obtained from the local **/etc/SDR_dest_info** file.
2. One, and only one, of the **-P**, **-p**, or **-j** flags must be specified.
   - The **-P** flag requests information about all the pools in the SP System.
   - The **-p** flag requests information about the pool specified by **pool_id**. A **pool_id** value of -2 requests information about all the parallel pools. A **pool_id** value of -3 requests information about all pools and is equivalent to specifying **-P**.
   - The **-j** flag requests information about all the jobs running on the system.
3. The **-v** (verbose) flag is always accepted, but no special action is taken when it is specified together with **-j**. If **-P** or **-p** is specified:
   - Without the **-v** flag, only information about pool IDs, pool attributes and member nodes is displayed.

- With the **-v** flag, additional information is displayed about jobs running in the specified pools.

Resource Manager Recovery

When the Resource Manager is started, the primary server automatically tries to start a backup server on another node, identified in the configuration file. If it is unable to do so, it will retry at regular intervals. Once the backup server has been started, the primary server keeps an open IP socket connection with it, and polls it periodically to check that the backup server process is alive. In addition, it checks the SDR to see if the backup server node is alive, since a node crash will not cause any message to appear on the IP socket. If it is determined at any point in the primary server's operation that the backup server process has died, the primary server will attempt to start another backup. The primary server records node allocation in the SDR.

The backup server periodically polls its IP socket connection with the primary server to check that the primary server process is alive. For the same reasons as described above, it also checks the SDR to see if the primary server node is alive. If it is determined that the primary server has died, the backup takes over as the new primary server by reading configuration and job data from the SDR. It waits a period of three minutes for all existing clients to validate their existence by reconnecting to it. During this time, the new primary server will accept only connection and reconnection requests from clients. After the three-minute interval, the job data in the SDR pertaining to invalid clients (those that did not reconnect) is removed. The backup server is now ready to accept all client requests and operate as the primary server. With this recovery scheme, jobs should be able to run unaffected despite the failure of the primary Resource Manager server.

The Resource Manager requires that the SDR be operational. If the control workstation is rebooted, or any other activity occurs that causes the SDR to be restarted, the Resource Manager will no longer be usable, and will eventually die. Under these circumstances the administrator should issue **jm_stop**. Once the SDR is again operational, issue **jm_start**.

If the primary and backup servers die at the same time, there is no recovery. Should this occur:

1. Quiesce the system (wait till all currently running parallel jobs terminate).
2. If the Resource Manager was configured with ACCESS_CONTROL=yes, run the **spacs_cntrl** command to reset user node access to its original value. The Resource Manager might have left it in an undesirable state upon failure.
3. Execute **jm_start** from the control workstation to restart the Resource Manager.

Chapter 12. Managing User Accounts

The SP system supplies user administration commands, **spmkuser**, **spchuser**, **sprmuser**, and **splsuser**. This chapter shows you how to add, change, delete, or list user accounts using both the SP commands and the SMIT interface.

Task information included in this chapter includes:

- Understanding SP user management
- Adding user accounts
- Changing user accounts
- Listing a user account
- Removing user accounts
- Changing passwords
- Controlling user login

Understanding SP User Management

The SP system provides the ability for you to manage your user accounts by adding new users, deleting existing users, or changing the user account information from a single point of control. SP User Management ensures that users have the same account, home directory, and environment across all the nodes in the SP system. SP User Management is optional, and it is designed to fit into your existing user management system.

The objective of any user management system is to ensure that the user account files, namely **/etc/passwd**, **/etc/group**, **/etc/security/passwd**, and **/etc/security/group**, are consistent across all of your nodes. You can accomplish this by using NIS, by distributing these files using file collections, or by some other means.

If you use NIS, the system maintains NIS maps to provide each user with a single image of their user account information across all the nodes. This information includes a user's login name, password, group information, default shell, and the name of the host where the home directory resides. If you use Amd, the system also maintains maps of mount points for the home directories. The SP User Management commands can be used in conjunction with NIS, and will create new NIS maps and push these maps when used on the NIS master.

If you are not using NIS, the SP User Management commands use the **user.admin** file collection to propagate the user administration files to the nodes of the system. You may wish to use this file collection to keep other user management related files consistent across the nodes. For more information refer to the “user.admin Collection” on page 216.

In NIS, users can change their own passwords, as needed, while logged in any SP node. In non-NIS environments users must log into the host where the password file resides. The full pathname for the password file is set with the **passwd_file** parameter of the **spsitenv** command, or through SMIT panels during installation. Administrators can change passwords for any user account from the NIS master.

After you complete the installation and customization procedures, you establish an initial set of users from the existing user management files in your network. The

© Copyright IBM Corp. 1995

SP system provides the ability for you to manage your user accounts by adding new users, deleting existing users, or changing the user account information.

SP User Management Options

Using the SP user management commands, you can add and delete users, change account information, and set defaults for your users' home directories. Specify the user management options you wish to use in your site environment during the installation process, or change them later, either through SMIT panels or by using the **spsitenv** command. The *IBM RISC System/6000 Scalable POWERparallel Systems: Installation Guide* contains detailed instructions for entering site environment information.

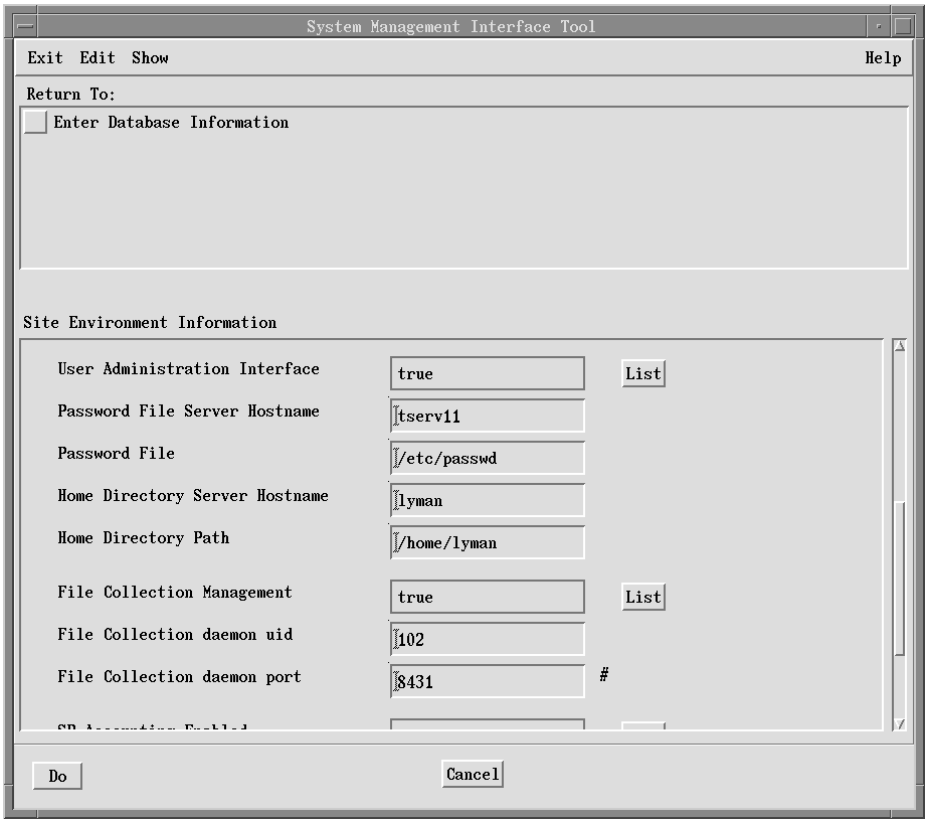


Figure 44. SMIT Panel for Site Environment Information

The following options, displayed in the SMIT Panel for Site Environment Information, pertain to user management:

User Administration Interface
Whether or not you want to use SP User Management to add, delete, and change user account information.

Password File
The path of your password file. The user commands add, modify, and delete entries in this file. If you modify the default path of **/etc/passwd** and you are using NIS, ensure that you also modify your NIS makefile to build the password map from the new password file.

Password File Server Hostname
The hostname of the machine where your password file resides. The initial value is the control workstation.

Home Directory Server Hostname
The name or IP address of the machine where the user home directories reside.

You can specify a default host for users' home directories. If you use Amd, the user management commands will use this host when building Amd maps to mount and link the home directories. If you do not specify a default, the initial value of the control workstation is assumed.

You can override the value in Hostname Home Directory Server when adding or modifying a user account with the **spmkuser** and **spchuser** commands.

If the users' home directories are served by a machine within the SP system and you follow the authentication setup described in Chapter 3, "Security Features of the SP System" on page 17, then you need not do anything more to have the SP user management commands run correctly.

However, if the home directory server is outside the SP system, there are two cases:

1. The machine that serves the home directory is in the same authentication realm as the SP system. In this case, the administrator's principal name must be added to the **.klogin** file of that machine.
2. The machine that serves the home directory is outside the realm and nonauthenticated. In this case, add a line to the **.rhosts** of the machine with the following data:
   - The hostname where the commands run
   - The user name **root**

   In this case, when the SP user management commands are run, informational messages are displayed to show that authenticated commands could not be run. However, the user management task will be performed.

Home Directory Path
This is where you specify the default path of users' home directories. The default is /**home**/*control_workstation_name*/.

Enter another value to set another path as the default for your site. You can override the default path with the **home** attribute, **spmkuser**, and **spchuser** commands.

Any host that is serving user directories must have its /**etc**/**exports** file modified to grant access to all hosts that need to mount these directories. Export the file systems using **exportfs -a** after modifying the /**etc**/**exports** file.

SP User Management Commands

The SP User Management commands reside in the **/usr/lpp/ssp/bin directory** on the control workstation. If you wish to use these commands in an NIS environment and your control workstation is not your NIS master, you must mount these commands on the NIS master where you will be executing them. (Your NIS master must be an RS/6000.)

You can add, change, delete, or list user accounts using either the SP commands or, optionally, the SMIT interface. For example, you can add new users to the SP system in any of the following ways:

- Enter the **spmkuser** command with the appropriate attributes and values.
- Enter **smit spmkuser**.
- Enter **smit**, select *Security & Users*, and then select *9076 SP Users*.
- Enter **smit**, select *9076 SP System Administration*, and then select *9076 SP Users*.

The examples in this section show the first two methods. You can find the complete syntax for the commands in the *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference*.

Adding User Accounts

To add new users to the SP system with the SMIT interface:

TYPE **smit spmkuser**

▲ The Add a User window appears.

Figure 45. SMIT Panel for Adding an SP User

Only the *User name* is required. You can accept the defaults or leave the other fields blank. The defaults for primary group, secondary groups, and the initial program are in **/usr/lpp/ssp/bin/spmkuser.default**. The default for the home directory is stored in the System Data Repository. The following table shows how the fields in this window relate to the parameters in the **spmkuser** command.

SMIT Field	Command Parameter	Description
User name	Name	User login name
User ID	id	ID of the user
LOGIN user?	login	Indicates whether the user can login to the system with the **login** command. This option does not change the **/etc/security/user** file but instead alters the user's password field in **/etc/security/passwd**.
Primary GROUP	pgrp	Principle group of the user
Secondary GROUPS	groups	The groups to which the user belongs
HOME Directory	home	Host name of the file server where the home directory resides and the full path name of the directory separated by a colon
Initial PROGRAM	shell	Shell program run when the user logs in
User Information	gecos	General information about the user

When you add a new user, the system generates a random password for the user and stores it in **/usr/lpp/ssp/config/admin/newpass.log**. The **root** user has read and write permission to this file. It is the administrator's responsibility to communicate this password to the new user and periodically delete the contents of this file.

Example

The following example shows how to enter the information for a user named **joe** using SMIT:

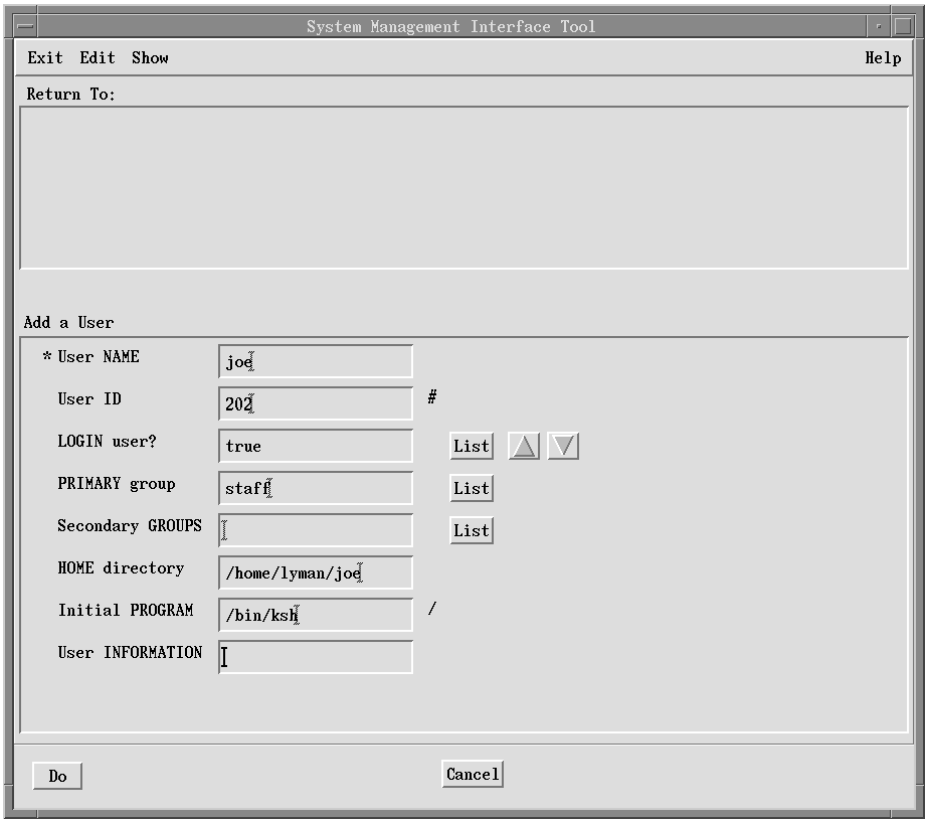


Figure 46. Example of Adding an SP User

TYPE The appropriate information in each field

PRESS **Do** to store the data.

To store the same information using the **spmkuser** command, enter:

```
spmkuser id=202 pgrp=staff home=lyman:/home/lyman shell=/bin/ksh joe
```

Changing User Accounts

To change user information with the SMIT interface:

TYPE **smit spchuser**

TYPE The name of the user whose information you want to change

▲ The Change/Show Characteristics window appears

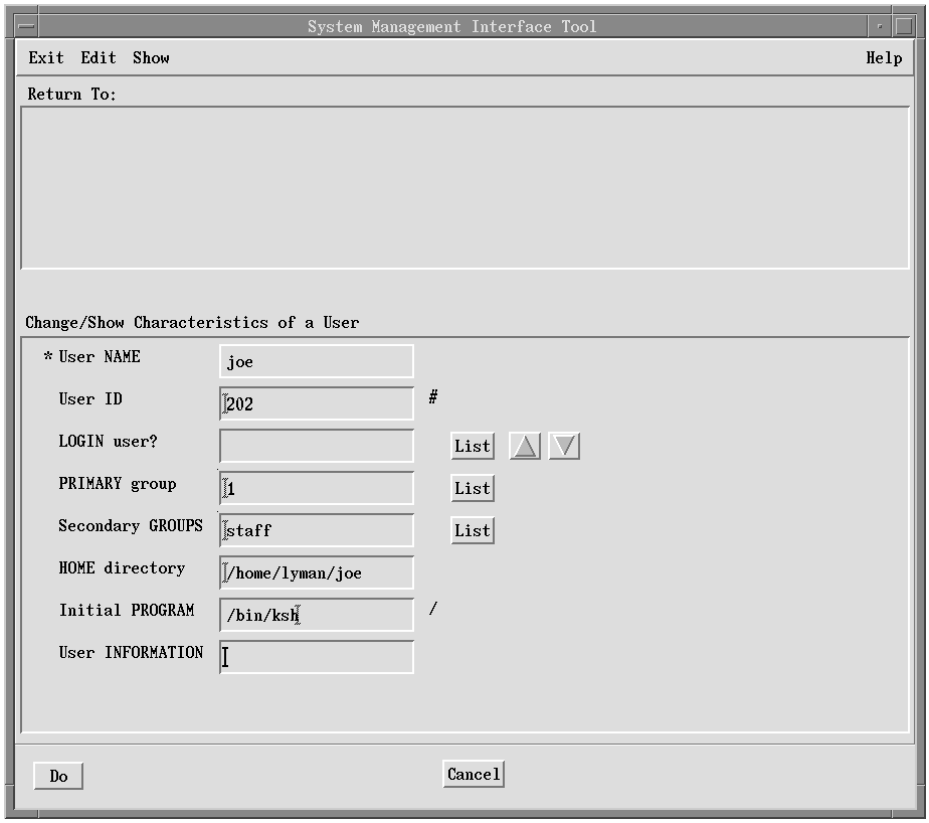


Figure 47. SMIT Panel for Changing SP User Information

To change information, type the new values in the appropriate fields. The only field you cannot change is *User name*. You can scroll to see the information in the home directory field.

Example

To change information using SMIT:

TYPE The new values

PRESS **Do** to store the data.

To change the home directory for **joe** using the **spchuser** command, enter:

```
spchuser home=lyman:/home/lyman/joe joe
```

Removing User Accounts

To remove an existing user account from the SP system with the SMIT interface:

TYPE **smit sprmuser**

TYPE The name of the user whose information you want to remove

▲ The Remove a User window appears.

Figure 48. SMIT Panel for Removing an SP User

You can choose to remove the user's home directory and the authentication data when /**etc**/**security**/**passwd** is utilized.

Example

To remove the **joe** user account with the **sprmuser** command but leave the authentication data and home directory intact, enter:

```
sprmuser joe
```

Listing a User Account

You can display the information for a user with the SMIT interface using **smit spchuser**.

TYPE **smit spchuser**

TYPE The name of the user whose information you want to view

▲ The SP Change/Show Characteristics window appears. This window is shown on page 199.

Be sure to cancel after viewing the information.

You can also list user information using the **splsuser** command. For example to list information for a user named **ken** in colon-separated records, enter:

```
splsuser -c ken
```

Changing Passwords

Your method of account management determines which steps are necessary to change passwords. The procedure for NIS is different than for a system with file collections. If neither NIS nor file collections is used, you will probably use **rdist**.

Note: If you have configured the SP USER ADMINISTRATION INTERFACE to **true** and you are not using NIS, the following base AIX commands are linked to SP commands on the nodes:

- **/bin/chfn → /usr/lpp/ssp/config/sp_chfn**
- **/bin/chsh → /usr/lpp/ssp/config/sp_chsh**
- **/bin/passwd → /usr/lpp/ssp/config/sp_passwd**
- **/bin/su → /usr/lpp/ssp/config/sp_su**
- **/bin/sbin/tsm → /usr/lpp/ssp/config/sp_tsm**

The original AIX files have been saved in their original location with extension **.aix** added. For example, the original AIX **passwd** command is in **/bin/passwd.aix**.

The linked **sp_** commands inform users what machine to log in to when, or if, they need to change password, gecos, or shell information.

Changing Passwords on an NIS System

NIS users can modify their passwords using either the **yppasswd** or **passwd** command. They must run these commands within the NIS domain that serves their user account.

Note

If you maintain an NIS password file outside of an NIS environment, you should continue your current process for changing passwords and disable the **yppasswdd** daemon on the NIS master:

1. Enter **smit chmaster**.
2. Set *START the yppasswdd daemon?* to **no**.
3. Set *CHANGE server now, at system restart or both* to **both**.

As a result, whenever users try to change passwords, they receive a message stating that the master is not running.

Changing Passwords for a File Collection System

If you use file collections rather than NIS, you must change the password in the master password file. Although this file may not be maintained on the control workstation, any changes to passwords must be made known to the control workstation.

1. Change password on host where master password file resides.
2. If the master password server is not the control workstation, copy the files **/etc/passwd** and **/etc/security/passwd** from the password server machine to the control workstation.
3. New password will be propagated throughout SP system within one hour.

Restricting the Use of the Control Workstation for General Users

If you use file collections on your system, users must be able to log in to the machine designated as the password file server to change their passwords. If you use the control workstation for the password server, you must allow users to log in to it to change their passwords. However, you may wish to have only administrative tasks performed on the control workstation, and therefore, do not want general users to be able to do anything other than change their passwords.

A script is provided that allows full login on the control workstation to the users you designate and restricts full login to all others. Users not designated can only log in to change their passwords and then are logged out. The script can be incorporated into the profile that is run during every login to the control workstation.

The script is located in **/usr/lpp/ssp/config/admin/cw_restrict_login**. To use the script, take the following steps:

1. Edit **/usr/lpp/ssp/config/admin/cw_allowed**

   The **cw_allowed** file defines to the **cw_restrict_login** script which user names are allowed to fully log in to the control workstation. **cw_allowed** is shipped containing three administrators' user names to show how the file should be formatted. These sample names should be removed from the file.

   Start each user name in the left most column and place one user name per line. Place no comments in the file. You do not need to list the root user in the file; the script is written to allow the root user to fully log in to the control workstation.

2. Integrate the restricted login script into the control workstation's login process by adding the following lines to the beginning (or at the most appropriate place) of the control workstation's /**etc**/**profile** file:

```
# Allow general users to login to control workstation to only change
# their password then log them out.
/usr/lpp/ssp/config/admin/cw_restrict_login
```

3. Verify that you have set up correctly:
   a. Log in to the control workstation as root and make sure you can fully log in.
   b. Log in as a user that should be allowed (listed in **cw_allowed**) to log in to the control workstation. Make sure this user can fully log in.
   c. Log in as a user that is not allowed (not listed in **cw_allowed**) to fully log in to the control workstation. Make sure this user can change his password and then is logged out.

At any time you wish to remove this restrictive login, you can do so simply by commenting out (adding a # at the beginning of the script execution line) or by deleting the lines you added from the control workstation's /**etc**/**profile** file.

Changing Passwords on Systems without NIS or File Collection

If neither NIS nor file collection is used, you must distribute these updated password files across the SP system after you change the password.

1. /**etc**/**passwd**
2. /**etc**/**security**/**passwd**

Controlling User Login

Login Control is used to dynamically prevent interactive login of users on a node basis. Preventing interactive login of users on nodes running parallel jobs is desirable for performance purposes. Login Control is intended for use with the Resource Manager. Once you have restricted access for users on a node using Login Control, the Resource Manager uses Login Control to unrestrict interactive access to those users running a parallel job scheduled through the Resource Manager. Once the job has completed, the Resource Manager uses Login Control to return the user to the original state.

If you have correctly configured the Resource Manager to use Login Control, it can automatically control user access to your SP nodes on a per job basis.

Login Control can be used to control interactive user access on nodes that have not been assigned to the Resource Manager for running jobs. You may want to use Login Control to temporarily restrict all but a few users on a node for a specific purpose.

You can issue commands to temporarily restrict and unrestrict users' interactive access to nodes running parallel jobs. However, this function should be used with caution because it can interfere with the coordination between the Resource Manager and Login Control processes.

Understanding Login Control

User login completes successfully under AIX after checking attributes in the /**etc**/**security**/**user** file. The attributes are *login* and *rlogin*. If these values are set to `true`, the user is allowed to log in. Conversely, if these values are set to `false`, the user cannot log in. Login Control updates this file dynamically for a user or several users when a request is received.

A special file, **spacs_data**, is created when a user access change request is made more than once by a job submissions system using the keywords **allow** or **deny**. It can be said that this file holds the state of requests for a user. After the initial allow or deny request has been proccessed in /**etc**/**security**/**user**, the next allow or deny request sets a request count for the user. The same number of subsequent opposite requests will check the count, decrement it and remove the user, if present, from **spacs_data**. The use of the block and unblock keywords by the System Administrator updates **spacs_data**, clearing all multiple job submission system allow/deny requests for a user before changing the state in /**etc**/**security**/**user**. These keywords are intended to set the user to a known state (either restricted or unrestricted) in the /**etc**/**security**/**user** file and in the **spacs_data** file. The **spacs_data** file is owned by root and is located in the /**var**/**spacs** directory.

Login Control recognizes requests made with the **spacs_cntrl** command. (See the *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference* for syntax.) This command is run on each parallel node and can be used with **dsh** for this purpose.

Login Control uses a lock on the **spacs_cntrl.lock** file to prevent more than one instance of Login Control from updating the files. This file is created the first time the **spacs_cntrl** command is executed. This file also resides in the /**var**/**spacs** directory.

The **spacs_cntrl** command takes four state keywords that manage user access to a parallel node:

.

1. **block**
2. **unblock**
3. **allow**
4. **deny**

Block and Unblock

These keywords are used by a System Administrator to dynamically restrict and unrestrict access to a node for a user or a group of users. These keywords are intended to set the user to a known state (either restricted or unrestricted) in the /**etc**/**security**/**user** file and in the **spacs_data** file. The block keyword can be used to initially restrict user access to a node before Resource Manager submits user jobs. These keywords should not be used on a node while Resource Manager is submitting jobs to the node; otherwise, the user state may be corrupted.

Allow and Deny

These keywords are used by a job submission system (such as the Resource Manager) on a transactional basis to update the user state. Multiple usage of **spacs_cntrl** with one of these keywords for a user results in the request state being updated in **spacs_data**.

A user can be in one of four states when these keywords are used:

1. Allowed interactive login
2. Denied interactive login
3. Allowed login after a deny request
4. Denied login after an allow request

State 3 can occur when other jobs by this user are still running. A deny request removes one allow request: in this case, the first allow from the first job.

State 4 can occur when more than one deny request is issued for a user. A deny request removes one allow request.

Preventing rsh Access to the Node

The use of **rsh** with a command bypasses the AIX **login** and **rlogin** executables, and therefore, cannot be restricted with Login Control. Restricting **rsh** can be important because users can use this command to run commands and perhaps start long-running jobs on a parallel node.

Restricting **rsh** access to a node.

You can restrict **rsh** to a node by leaving the /**etc**/**hosts.equiv** file empty and disallowing the use of **.rhosts** files by the user community. You should run a **cron** job that periodically searches and removes **.rhosts** files from users' home directories. If you wish to restrict users and still allow **.rhosts** files, see the next paragraph.

Restricting a particular user's **rsh** to a node.

You can restrict access for one or more users by adding the following line to the /**etc**/**hosts.equiv** file on the nodes.

\+ *-userid*

The + is located in the host field and means **all hosts**. The *-userid* is in the user field and means **access denied for rsh for this user**. Together, the meaning is **restrict rsh** access on this node for this user no matter what host the user is on. This allows certain users to use **rsh** if necessary and other users to have access restricted. This will restrict access for a user even if the user has an **.rhosts** file in the /**home** directory.

Using the Login Control Sample File

A sample Perl script called **block_usr_sample** is located in the /**usr**/**lpp**/**ssp**/**samples** directory. This script creates a file of user IDs, /**tmp**/**usr.input**, that can be used with **spacs_cntrl -f** to restrict user access. It serves as a model for formatting other files for use with the **spacs_cntrl** command:

```
userid1
userid2
userid3
```

Once you are familiar with Login Control, **spacs_cntrl** can be run within this script by removing the comments surrounding the command. A copy of this file is located in Appendix C, “Sample Files” on page 399.

Before using this script you must edit the **threshold UID ($uidstart)** in the script to be large enough so that system IDs and SP system IDs are not restricted. These IDs include **bin**, **admin**, **lpd**, **supman**, and **prtid**. The default threshold UID is set at 125. The **block_usr_sample** script can be added to the **script.cust** file or to the node's **inittab** file.

Adding the script to run from the **script.cust** file will create the /**tmp**/**user.input** file and run **spacs_cntrl block** when you install the node to initially restrict interactive access for all users. Users would remain in this state across a reboot unless specifically unrestricted by the System Administrator or by the job submission system. New users that are added to the system would not be included until the **block_usr_sample** script was run again at some later time.

Adding the **block_usr_sample** script to the **inittab** file will run this script at node boot time. This will include any new users on the node but may also increase the time to boot the node. This script must be added **AFTER** the **rc.sp** script in the **inittab** file. **rc.sp** removes the login control lock file, **spacs_cntrl.lock**, if present, when booting a node to ensure login control will run if a node is rebooted after a crash.

Using Login Control

There are many ways you can use Login Control on your SP system. You can restrict all or a subset of users on a parallel node to prevent the users from gaining interactive access. When the Resource Manager is not configured to use login control, the restriction remains in effect until you specifically use the unblock keyword to unrestrict access for the restricted users. Any jobs scheduled to the nodes controlled by the Resource Manager will return an error for those users whose access is restricted.

This enables you to set aside a parallel node for the use of select users for interactive and parallel jobs. You may list the users on the command line or use a file containing the user names in a column. Note that this file must reside on the parallel node in order to be accessible to the login control command.

Here are a couple of suggestions to get you started. Whichever you decide to use, ensure that the Resource Manager is configured correctly. You can use the **locate_jm** command to identify the primary and backup Resource Manager servers. The **jm_status** command gets information about resource pools and nodes allocated for Resource Manager jobs. See “Using the Resource Manager” on page 182.

Restricting a User on a Single Node

To restrict user Betty on a particular node, on that node issue:

```
spacs_cntrl block betty
```

Restricting Users on Multiple Nodes

1. Run the **block_usr_sample** file after adjusting threshold UID.
2. Send file to the node(s). Note this example would require **rsh** priviledges on the nodes.

   ```
   dsh -a rcp  root@mynode:/tmp/usr.input /tmp/usr.input
   ```

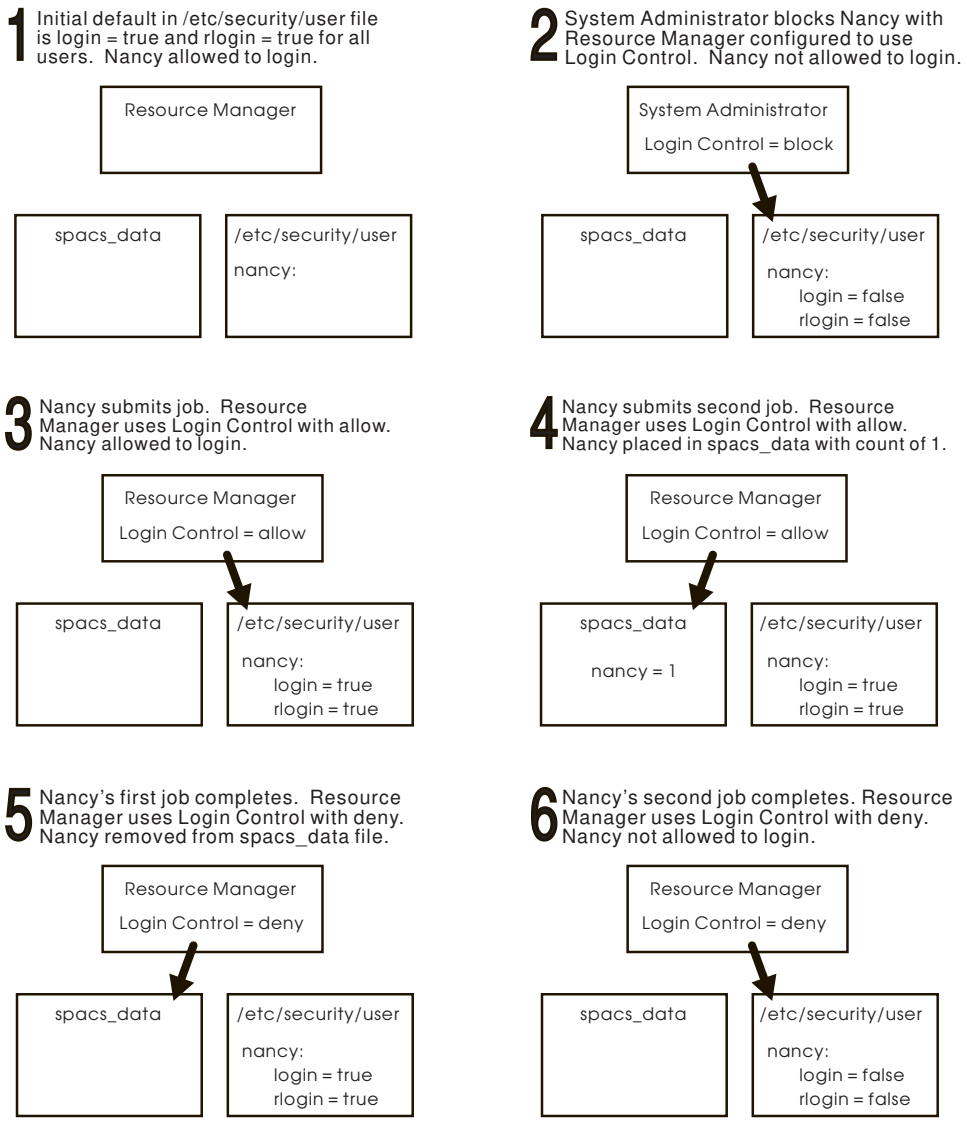
3. Issue login command to restrict interactive access for users to all the nodes.

   ```
   dsh -a spacs_cntrl -f /tmp/usr.input block
   ```

Login Control and the Resource Manager

The Resource Manager is used to submit parallel jobs to the SP System. When the Resource Manager has control of a node for scheduling parallel jobs, it must be able to unrestrict a user's access on the system in order to run the user's parallel job. Note that during the running of a parallel job, that user can also use the node interactively. Once the user is unrestricted, the AIX **login**, **rlogin**, and **telnet** executables cannot distinguish between the Resource Manager scheduling the job using the user's ID and the user logging into a node.

The following example shows how the Resource Manager and Login Control interact. For this example, the System Administrator has restricted access for users on this node before allowing the Resource Manager to run.



Using Login Control with the Resource Manager

You may configure the Resource Manager to use Login Control in order to dedicate a node to running parallel jobs. Once it has control of the node for scheduling jobs, the Resource Manager will use Login Control to allow a user access in order to schedule that user's parallel job. Once the job is finished, the user will be placed back in the original state.

Since the System Administrator's keywords, block and unblock, clear all multiple allow/deny requests in **spacs_data**, you should not use it on the nodes under the Resource Manager's control. You can initially block users on a node and then start the Resource Manager to handle the access for parallel jobs. Conversely, the Resource Manager should be quiesced before running **spacs_cntrl** again on those nodes it controls.

The only exception is blocking a new user on a parallel node. This should be done before allowing the user to submit parallel jobs to avoid conflict.

You can configure the Resource Manager to use Login Control whether or not you choose to initially block users on a parallel node.

Maintaining Login Control

Minimal maintenance is required after the Resource Manager is configured to use Login Control on nodes running parallel jobs. However, the following information is important for Login Control to function properly.

Since the **/etc/security/user** file is used by Login Control for state information, and by the AIX system for restricting and unrestricting login for users, this file is machine dependent and should not be overwritten. The **/etc/security/user** file should not be distributed through file collections or any other mechanism throughout the SP system.

The **spacs_cntrl** command may generate a large log file, depending on which flags you specify. The default for trimming the log file is 400 lines and is done every night through a **cron** job. If you wish to save the log for some reason, you can rename the file and a new log file will be created.

Usage Notes

You should not use the Login Control command, **spacs_cntrl**, while the Resource Manager is using Login Control. This is because the System Administrator keywords, **block** and **unblock**, return the user to a known state and remove status kept in the **spacs_data** file. The Resource Manager has no way to tell this has been done and may leave the user in an unpredictable state for running jobs.

You must issue the **spacs_cntrl** command to block new users added to a node if you have previously blocked all users on that node. Users denied login will see the standard message, “You have entered an invalid login name or password.”

The Login Control utility will not prevent **LoadLeveler** from running jobs submitted by blocked users because **LoadLeveler** logs in as **root** and then switches to the user. Root is never blocked on a node.

Each time the log is opened, a datestamp is written. This may or may not be followed by messages depending on the options chosen on the command line. The following **spacs.log** example shows the output when using the **-d** and **-l** flags on the **spacs_cntrl** command line.

Sample Log

```
"Tue Apr 26 10:11:47 EDT 1994"
spacs_cntrl:  Command Invocation: spacs_cntrl -d -l deny nancy.
spacs_cntrl:  CPFILE called for /etc/security/user and /etc/security/user.17008.
spacs_cntrl:  ACS_FILE/lock called for /var/spacs/spacs_data.lck.
spacs_cntrl:  List of users in deny list: .
spacs_cntrl:  List of users in allow list: ,mroz.
spacs_cntrl:  Deny processing.
spacs_cntrl:  List of users in send list: nancy.
spacs_cntrl:  MUSRFILE called.
spacs_user:  Called with parameters:  /usr/lpp/ssp/bin/spacs_user -l -d deny nancy.
spacs_user:  Attribute for user nancy changed to requested value.
spacs_cntrl:  List of users in return list: nancy:0.
spacs_cntrl:  CPFILE called for /var/spacs/spacs_data and /var/spacs/spacs_data.17008.
spacs_cntrl:  ACS_FILE/unlock called for /var/spacs/spacs_data.lck.
"Tue Apr 26 10:11:52 EDT 1994"
spacs_cntrl:  Command Invocation: spacs_cntrl -d -l deny pepper.
spacs_cntrl:  CPFILE called for /etc/security/user and /etc/security/user.20102.
```

```
spacs_cntrl:  ACS_FILE/lock called for /var/spacs/spacs_data.lck.
spacs_cntrl:  List of users in deny list: .
spacs_cntrl:  List of users in allow list: ,mroz.
spacs_cntrl:  Deny processing.
spacs_cntrl:  List of users in send list: pepper.
spacs_cntrl:  MUSRFILE called.
spacs_user:  Called with parameters:  /usr/lpp/ssp/bin/spacs_user -l -d deny pepper.
spacs_user:  Attribute for user pepper changed to requested value.
spacs_cntrl:  List of users in return list: pepper:0.
spacs_cntrl:  CPFILE called for /var/spacs/spacs_data and /var/spacs/spacs_data.20102.
spacs_cntrl:  ACS_FILE/unlock called for /var/spacs/spacs_data.lck.
```

Chapter 13. Managing NTP

The SP system uses Network Time Protocol (NTP) to synchronize the time-of-day clocks on the control workstation and processor nodes. There are several ways you may currently be handling time-of-day synchronization in your environment. You may already be using NTP, either locally or through the Internet, or you may be using some other time service software. It is also possible that you have no established method for synchronizing the system clocks in your computing environment.

The SP system gives you several options for time-keeping:

- If you have an established NTP time server, you can use it to synchronize and manage time on the SP system.
- You can choose an NTP time server from the Internet.
- You can run NTP locally on the SP system to generate a consensus time.
- You can choose not to use NTP at all, relying on other methods at your site. **The SP machines do not have system batteries. If you choose not to use NTP, you must have another way to manage clock synchronization.**

NTP Options

There are three attributes to set for NTP. *ntp_version* defaults to a value of **3**, the version shipped with the SP System Support Programs. If your installation is using an earlier version of NTP, change this value. The other two attributes are described in Table 13.

Table 13. Time Service Choices

	Worksheet Entries to Be Filled In	
To do this....	*ntp_config*	*ntp_server*
Use your site's existing NTP time server to synchronize the SP system clocks.	**timemaster**	*hostname* of your current NTP time server
Use an NTP time service from the Internet to synchronize the SP system clocks.	**Internet**	*hostnames* of time servers on the Internet*
Run NTP locally on the SP to generate a consensus time.	**consensus** (default)	
Do not use NTP on the SP; instead, use some other method to synchronize system clocks.	**none**	

Note:

- Change default attribute values to suit your environment.
- Blank entries imply that you make no substitutions for these values.
- *Reference **/usr/lpp/ssp/ssp.public.README** for information on Internet time servers.

When you choose any of the options that involve using NTP, the installation scripts determine the IP address of the host specified as **ntp_server** and creates the **/etc/ntp.conf** file on each of the control workstation, boot/install server, and processor nodes. This file indicates which hosts can provide time service to which other hosts using the designations of *server* and *peer*, where

© Copyright IBM Corp. 1995

server Indicates a host from which you can request time.

peer Indicates a host with which this host can synchronize time. either peer host can request time from the other.

The following example shows a sample /**etc**/**ntp.conf** file:

```
# Server ip_addr version <ntp version>
server 9.8.77.66 version 3      # server is host_one.conrad.ibm.com
server 123.45.678.9 version 3   # server is host_two.ursula.ibm.com
peer 9.8.77.55 version 3        # server is host_x.conrad.ibm.com
peer 9.8.77.44 version 3        # server is host_y.conrad.ibm.com
peer 9.8.77.33 version 3        # server is host_z.conrad.ibm.com
```

Assuming this is the /**etc**/**ntp.conf** file on *your_host*, the following is true:

- *your_host* can request time from *host_one*, *host_two*, *host_x*, *host_y*, and *host_z*
- *your_host* can provide time, on request, to *host_x*, *host_y*, and *host_z*
- *your_host* cannot provide time to *host_one* or *host_two*.

The following table shows which host addresses will be used for servers and peers in the /**etc**/**ntp.conf** files created on the control workstation, file servers, and processor nodes for each **ntp_config** choice.

ntp_config	Control Workstation	Boot/Install Servers	Processor Nodes
timemaster	SERVER: your time server PEER: all boot/install servers	SERVER: your time server PEER: control workstation PEER: all boot file servers	SERVER: your time server SERVER: control workstation SERVER: all boot file servers
Internet	SERVER: Internet server	SERVER: control workstation PEER: other boot file servers	SERVER: control workstation SERVER: all boot file servers
consensus	SERVER: 127.127.1.10*	SERVER: control workstation PEER: other boot file servers	SERVER: control workstation SERVER: all boot file servers
none	no file created	no file created	no file created
Note: * This address signifies local time service to NTP and makes the control workstation a stratum 10 time server allowing you to hook into a higher stratum Internet time server at a later date.			

Related Information: The manual pages for NTP provide additional information on how NTP synchronizes time across a network. They are available in /**usr**/**lpp**/**ssp**/**man**/**man8**. The man pages are xntpd and xntpdc.

For more information on the installation scripts, see the *IBM RISC System/6000 Scalable POWERparallel Systems: Installation Guide*.

For directions on updating the site environment see the *IBM RISC System/6000 Scalable POWERparallel Systems: Installation Guide*.

Chapter 14. Managing File Collections

The SP system installs file collection technology by default, to simplify the task of maintaining duplicate files on multiple machines. In the delivered system, the files that are required on the control workstation, boot/install servers, and processor nodes belong to file collections. Grouping these files into file collections and using the provided tools to manage them allows you to easily maintain their consistency and accuracy on multiple nodes in the SP system.

This chapter explains the file collections delivered with the system. It shows you how to report on, update, and maintain the system file collections, as well as how to build file collections for your own purposes. It also explains the basic concepts you need to understand before working with file collections, such as what file collections are, how they differ from standard directories and files, and how they are used in the SP system.

The file collection technology provides you with the means to perform both beginning basic tasks and optional advanced tasks. For the basic tasks you should understand the fundamental concepts and be familiar with the delivered file collections. The more advanced tasks require an understanding of the master files, what they contain, and how they work.

Information is presented on the following basic file collection tasks:

- Understanding file collections
- Reporting file collection information
- Verifying file collections using **scan**
- Updating files in a file collection
- Adding and deleting files in a file collection

Information is presented on the following advanced file collection tasks:

- Building a file collection
- Installing a file collection
- Refusing files in a file collection
- Modifying the file collection hierarchy
- Removing a file collection

Terminology

The following terms are used when defining file collections:

Resident A file collection that is installed in its true location and able to be served to other systems

Available A file collection that is not installed in its true location but able to be served to other systems

© Copyright IBM Corp. 1995

Understanding File Collections

A *file collection* is a set of files and directories that are duplicated on multiple machines in a network and managed by tools that simplify their control and maintenance. Included with the administrative software is a program called **supper**. The **supper** perl program (/**var**/**sysman**/**supper**) uses the Software Update Protocol (SUP) to manage the SP file collections and transfer them across the system.

File collections have unique features:

- A file collection directory does not contain the actual files in the collection. Instead, it contains a set of master files to define the collection. Some master files contain rules to define which files can belong in the collection and others contain control mechanisms, such as time stamps and update locks. These files are explained in "Directory and Master Files" on page 217.
- You handle files in a collection with special procedures and **supper** commands rather than with the standard AIX file commands. The **supper** commands interpret the master files and use the information to install or update the actual files in a collection. You can issue these commands in either batch or interactive mode. The procedures and use of the **supper** commands are detailed throughout this chapter. You can find the command syntax and descriptions in the *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference*.
- File collections can be either *primary* or *secondary*. Primary file collections can contain other secondary file collections. When a primary collection that contains secondary collections is installed, the secondary collections are *available* but not *resident*. This means they can be served from this location but not actually executed or used there. "File Collection Types" on page 215 provides a further explanation of these concepts.
- File collections require special entries in the /**var**/**sysman**/**file.collections** and /**var**/**sysman**/**sup**/**.resident** files to define them to the **supper** program. They also require a symbolic link in the /**var**/**sysman**/**sup**/**lists** file pointing to their **list** master file. See "Building a File Collection" on page 228 for an explanation of these entries.
- File collections also require a unique, unused user ID for **supman**, the file collection daemon, along with a unique, unused port through which it can communicate. The default installation configures the user ID, *supman_uid*, to 102 and the port, *supfilesrv_port*, to 8431. You can change these values using SMIT or the **spsitenv** command.
- The file collection daemon, **supman**, requires read access permission to any files that you want managed by file collections. For example, if you want a security file such as /**etc**/**security**/**limits** managed, you must add the **supman** ID to the security group. This provides **supman** with read access to files that have security group permission and allows these files to be managed across the SP by file collections. You can add **supman** to the security group by adding **supman** to the security entry in the file /**etc**/**groups**.

File Collection Types

File collections can be either primary or secondary. A primary file collection can be a stand-alone collection or it can contain a secondary collection. Having a secondary collection allows you to keep a group of files available on a particular machine to serve to other systems without having those files installed.

For example, if you want to have one **.profile** on all nodes and another **.profile** on the control workstation, consider using the **power_system** collection delivered with the IBM Parallel System Support Programs for AIX. This is a primary collection that contains **node.root** as a secondary collection.

- Copy **.profile** to the /**share**/**power**/**system**/**3.2** directory on the control workstation.
- If you issue **supper install power_system** on the boot/install server, the **power_system** collection is installed in the /**share**/**power**/**system**/**3.2** directory. Because the **node.root** files are in that directory, they cannot be executed on that machine but are available to be served from there. In this case, **.profile** is installed as /**share**/**power**/**system**/**3.2**/**.profile**.
- If you issue **supper install node.root** on a processor node, the files in **node.root** collection are installed in the **root** directory and, therefore, can be executed. Here, /**share**/**power**/**system**/**3.2**/**.profile** is installed from the file collection as /**.profile** on the node.

Predefined File Collections

Looking at the file collections delivered with the SP system can help you to understand file collection concepts. You should become familiar with the delivered file collections before you begin to update or add to them.

The SP system provides four predefined file collections:

- **sup.admin**
- **user.admin**
- **power_system**
- **node.root**

You can display information about each collection on a particular machine using the **supper status** command. This reports whether or not the file collection is resident on that machine. To do this, log in as **root** or issue the command remotely. Enter:

```
/var/sysman/supper status collection_name
```

You can see a list of the files in a file collection by looking at the **scan** file. See "Verifying File Collections Using scan" on page 223 for instructions on how to do this. The **scan** file contains an entry for each file in the collection that could be installed on a boot/install server or processor node. Certain files in the collection may not exist on one particular system depending on the contents of the **refuse** file on that system. You can find more information about the **scan** and **refuse** files in "Directory and Master Files" on page 217.

SP File Collection Summary

The following table summarizes where these file collections are *resident* or *available* in the SP default configuration.

Control Workstation		Boot/Install Servers		Processor Nodes	
Resident	**Available**	**Resident**	**Available**	**Resident**	**Available**
sup.admin user.admin power.system	node.root	sup.admin user.admin node.root power_system		sup.admin user.admin node.root	

sup.admin Collection

The **sup.admin** file collection is a primary collection that is *resident* on the control workstation, the boot/install servers, and on each processor node.

This file collection is important because it contains the files that define the other file collections. It also contains the file collection programs used to load and manage the collections. Of particular interest in this collection are:

- /**var**/**sysman**/**sup** which contains the directories and master files that define all the file collections in the system
- /**var**/**sysman**/**supper** which is the Perl code for the **supper** tool
- /**var**/**sysman**/**file.collections** which contains entries for each file collection.

user.admin Collection

The **user.admin** file collection is a primary collection that is *resident* on the control workstation, the boot/install servers, and on each processor node. It contains the user management files such as /**etc**/**amd**/**amd-maps**/**amd.u**. This collection is a good place to add other files related to user management, such as /**etc**/**environment**, /**etc**/**security**/**limits**, and /**etc**/**passwd**. If you want to keep user characteristics the same across nodes, you may want to add /**etc**/**security**/**user** to this collection also. However, you may **not** want to have user characteristics the same across nodes. For instance, you may want a user to be able to log in to some nodes, but not to others. In this case, do not add the /**etc**/**security**/**user** file to this collection.

power_system Collection

The **power_system** file collection is used for files that are system dependent. It is a primary collection that contains one secondary collection called the **node.root** collection. The **power_system** collection contains no files other than those in the **node.root** collection.

When the **power_system** collection is installed on a boot/install server, the **node.root** file collection is *resident* in the /**share**/**power**/**system**/**3.2** directory and can be served from there.

node.root Collection

This is a secondary file collection under the **power_system** primary collection. The **node.root** collection is *resident* on the boot/install servers and processor nodes, not the control workstation. It contains key files that are node-specific.

The **node.root** file collection is *available* on the control workstation and the boot/install servers under the **power_system** collection so that it can be served to all the nodes. We do not install **node.root** on the control workstation because the files in this collection might conflict with the control workstation's own **root** files.

Primary and secondary files are defined in the **file.collections** file (see “Step 5: Update the file.collections File” on page 230).

File Collections Organization

When you complete the SP installation and configuration process, the delivered file collections are organized in a hierarchy with the control workstation as the master server for all the collections.

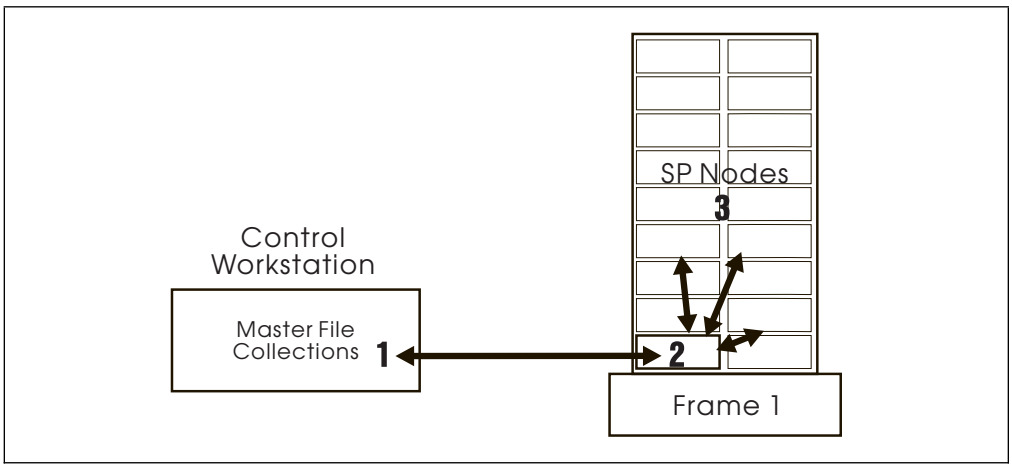


Figure 49. SP Default File Collection Hierarchy

The commands to update the file collections are set up in a **crontabs** file to run hourly in a staggered sequence. Figure 49 shows this organization. Files are updated on the control workstation in the master file collections (**1**). The boot/install servers use **supper update** to request the changes from the control workstation (**2**) and the nodes use **supper.update** to request the changes from the boot/install servers (**3**).

Understanding the File Collection Structure

Before you work with the delivered file collections or build a file collection of your own, you need to understand the directories, master files, what they contain, and how they work.

Directory and Master Files

The **/var/sysman/sup** directory contains master files for all the file collections. Figure 50 on page 218 shows the structure for this directory.

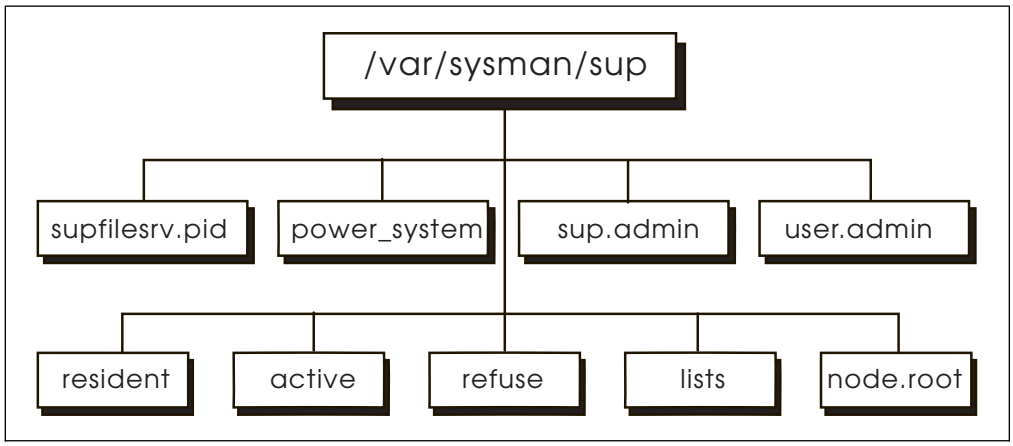


Figure 50. /var/sysman/sup Files and Directories

Following is an explanation of the files and directories.

.active A file identifying the active volume group. You can modify this file with the **supper activate** command. (This file is not on the control workstation.)

.resident Lists each file collection in the SP system. This file needs to be updated when you are adding a new file collection. (This file is not on the control workstation.)

lists A directory that contains links to the **list** files in each file collection.

node.root Directory of the master files in the **node.root** collection.

power_system Directory of the master files in the **power_system** collection.

refuse On a client system, this is a user-defined (you must define the file; it is not supplied) text file containing a list of files to exclude from all the file collections. This allows you to customize the file collections on each system.

You list the files to exclude by their fully qualified names, one per line. You can include directories, but you must also list each file in that directory you want excluded.

sup.admin Directory of the master files in the **sup.admin** collection.

supfilesrv.pid The process ID of the SUP **supfilesrv** process.

user.admin Directory of the master files in the **user.admin** collection.

Each individual file collection is composed of a directory and several master files. Figure 51 on page 219 shows the structure for the **sup.admin** file collection in the **/var/sysman/sup/sup.admin** directory. It includes a sample of the file contents. This example shows eight master files. Other file collections may have fewer master files.

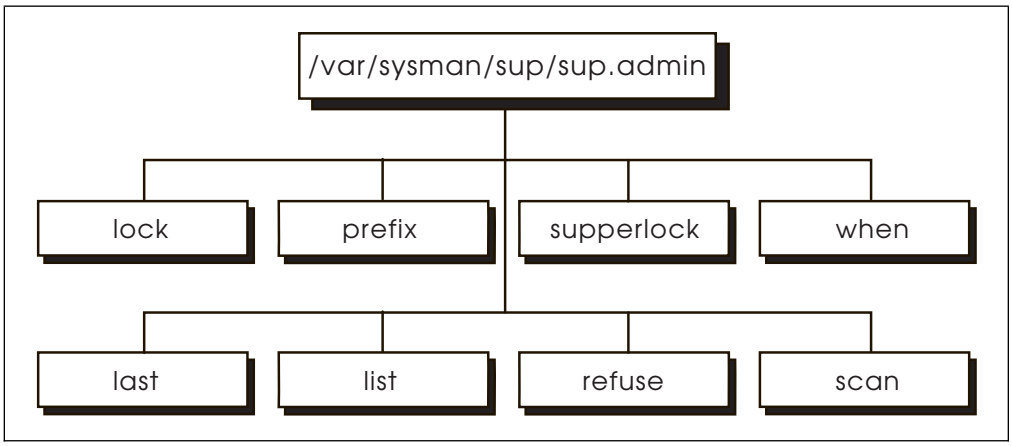


Figure 51. sup.admin Master Files

Following is an explanation of the master files shown in Figure 51.

last A system-maintained list of files and directories that have been updated. This helps identify files to be deleted on a local machine when they are no longer part of the collection.

The **supper files** command reads the **last** file to obtain a current list of files in the collection.

list A user-defined file containing special commands. These commands are interpreted by the **supper** scan process to define which files belong to the file collection relative to the base directory specified in the **prefix** file. You can find the specific format for these commands in "supper list File Syntax, Keywords, and Operands" on page 236

lock An empty file used by SUP to lock a collection and prevent more than one update of the same collection at the same time.

prefix A file containing the name of a directory to be used as the base directory for file references and the starting point for the **supper** scan process. This file is created by the system from data in the **file.collections** file.

refuse A system-created file that contains a list of all the files excluded during the update process. If there are no files for this collection listed in the **refuse** file in the **/var/sysman/sup** directory, the **refuse** file in this directory will have no entries.

scan A system-created file containing a list of files that comprise the collection and their permissions and time stamps. This file is the result of the **supper** scan process. It is optional but recommended for all collections and can cause maintenance problems if you do not keep it current.

supperlock Similar to **lock**, an empty file used by **supper** to lock a collection during updates.

when A system-maintained file containing the time of the last file collection update. This file protects against accidental update of a collection with an old version of the files. You can reset it using the **supper reset** command if you need to override this restriction.

How the Master Files Work

The **supper** master files play a key role during the install, update, and scan processes.

When you run the **supper scan** command for a file collection, the search begins at the point in the directory specified in the **prefix** file and traverses the directory tree to find all the files that meet the criteria defined in the **list** file. The output of this process is the **scan** file which lists the files in the collection. The **scan** file is optional but it documents the file collection contents and eliminates the need for the install and update processes to do the directory search.

The **supper install** and **update** commands use the **scan** file, if it is present, to identify the files to install or update. If a **scan** file is not present, **supper** performs the same directory search to identify the files as it performs when it creates the **scan** file. These commands also check the **refuse** file in **/var/sysman/sup** at each location and bypass any files it contains. As a result, these commands create a **refuse** file in the file collection's directory, listing the files that were bypassed.

During updates, the **lock** and **supperlock** files prevent update access by other SUP or **supper** commands. **supper** checks the update time in the **when** file against the actual file time stamps to verify that you are not updating with an old version of the files and, if so, does not update. It uses the **last** file for additional verification when deleting files that no longer exist in the file collection.

Master File Processing Order: Whether invoked with the **scan, update,** or **install** commands, **supper** performs the search process in the following manner:

prefix Reads this file first to obtain a directory from which it will start the search

list Reads this file next and processes its commands in the following order:

- **omit** Marks these files for omission
- **upgrade** Marks these files for upgrade when updating or installing
- **always** Overrides any files marked for omission

refuse Reads this file last when it is present on a client (boot/install server or processor node) and does not upgrade any of the specified files on that client.

Automatic Updates: The **supper update** commands are included in the **crontabs** file with the default set to run an update on each of the file collections hourly. You can modify the **crontabs** file to run the **supper update** more or less frequently. Refer to “Updating crontabs on the SP Nodes” on page 281 for more information.

File Collection Tasks

The file collection technology provides you with the means to perform both beginning basic tasks and optional advanced tasks. For the basic tasks you should understand the fundamental concepts and be familiar with the delivered file collections. The more advanced tasks require an understanding of the master files, what they contain, and how they work.

Basic File Collection Tasks

In order to maintain the delivered file collections you will need to perform some basic tasks such as reporting on file collection information, updating files that belong to an existing file collection, and possibly adding or deleting files within an existing file collection. For example:

- To become familiar with the delivered collections you might list their name, access point, and whether or not they are resident. You might also want to verify the complete list of all the files in a collection.
- You might update files such as /**etc**/**profile** to change the user default profile.
- You might add files such as /**etc**/**environment** and have them managed by the file collection processes.

You can find basic file collection tasks on pages 214 through 226.

Advanced File Collection Tasks

Once you are familiar with file collection technology, you may want to build your own collections to manage other files on your system that you would like duplicated on multiple machines. You might have a group of files such as application data that you want installed on each node or local tools you want installed on each boot/install server.

The task of building and distributing your own collections is more advanced. It includes creating the directory and master files, adding entries to the **file.collections** and **lists** files, and possibly creating **scan** and **refuse** files. Once you complete these tasks and build your file collection, you can install it on the machines in your system. You can also install a file collection from a server other than its default server.

You can find advanced file collection tasks on pages 233 through 235.

Reporting File Collection Information

supper has subcommands that report information about file collections. These subcommands are useful for verifying information before performing file collection manipulations or for checking results at some point during a procedure.

The following table shows the **supper** subcommand to use, where you would run the command, and the data you can report. See the *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference* for additional command options.

Use:	On:	To Report This Data:
status	Control workstation, boot/install servers, and nodes	Name, resident status, and access point of all available file collections, plus the name and estimated size of their associated file systems
files	boot/install servers and nodes	All the resident files resulting from a **supper update** or **install** command
serve	Control workstation and boot/install servers	All the collections that can be served from your machine

Use:	On:	To Report This Data:
where	Control workstation	Current boot/install servers for collections
when	boot/install servers and nodes	Last update time of all resident collections
diskinfo	Control workstation and boot/install servers	Available disk space and active volume for your machine
log	boot/install servers and nodes	Summary of the current or most recent **supper** session
rlog	boot/install servers and nodes	Raw output of the current or most recent **supper** session

The following examples show the output produced by some of these commands.

From the Control Workstation:

```
[tserv11][/]> supper status

 Collection          Resident  Access Point          Filesystem           Size
==============================================================================
 node.root              -     /                        -                  -
 power_system           -     /share/power/system      -                  -
 sup.admin              -     /var/sysman              -                  -
 user.admin             -     /                        -                  -
==============================================================================
```

From a boot/install server:

```
[r05n01][/]> supper status

 Collection          Resident  Access Point          Filesystem           Size
==============================================================================
 node.root             Yes    /                        -                  -
 power_system          Yes    /share/power/system      -                  -
 sup.admin             Yes    /var/sysman              -                  -
 user.admin            Yes    /                        -                  -
==============================================================================
```

From a processor node:

```
[r05n10][/]> supper status

 Collection          Resident  Access Point          Filesystem           Size
==============================================================================
 node.root             Yes    /                        -                  -
 power_system           -     /share/power/system      -                  -
 sup.admin             Yes    /var/sysman              -                  -
 user.admin            Yes    /                        -                  -
==============================================================================
```

Example of a **supper where** command:

```
[r05n01][/var/sysman/sup/node.root]> supper where
supper:  Collection node.root would be updated from server r05cw.
supper:  Collection power_system would be updated from server r05cw.
supper:  Collection sup.admin would be updated from server r05cw.
supper:  Collection user.admin would be updated from server r05cw.
```

Verifying File Collections Using scan

The **scan** file provides an inventory of all the files and directories in the file collection. By looking at the **scan** file you can quickly see which files are in a collection. You can find the **scan** file in the file collection's directory in /**var**/**sysman**/**sup**.

Although **scan** files require maintenance whenever files are added, deleted, or changed in the master collection, they save processing time on larger file systems and are recommended.

Note

You must update the **scan** file each time you modify the file collection or you risk accidentally updating the collection from an outdated master file.

Example: To check for an existing **scan** file in the **user.admin** file collection, enter:

```
ls /var/sysman/sup/user.admin
```

To create a **scan** file for the **user.admin** file collection, enter:

```
/var/sysman/supper scan user.admin
```

Following is a sample of a **scan** file for the **user.admin** collection. It shows the name of each file preceded by three sets of numbers. The first set is the file permissions, the second set indicates when the collection was installed, and the third set indicates when the collection was last updated.

```
V2
240755 740845355 740845355 .
100644 740760294 740760294 .kshrc
100644 740760294 740760294 .profile
240755 740845315 740845315 etc
240755 740760295 740760295 etc/amd
100754 740760294 740760294 etc/amd/amd
100754 740760295 740760295 etc/amd/amd.start
100754 740760295 740760295 etc/amd/amq
100754 740760295 740760295 etc/amd/refresh_amd
100644 740943550 740943550 etc/motd
240755 740760294 740760294 var
240755 740760294 740760294 var/sysman
240755 740760295 740760295 var/sysman/etc
100754 740760295 740760295 var/sysman/etc/kernswitch
100754 740760295 740760295 var/sysman/etc/ntpdate
100754 740760295 740760295 var/sysman/etc/sup
100754 740760295 740760295 var/sysman/etc/supfilesrv
100754 740760295 740760295 var/sysman/etc/supscan
100754 740760295 740760295 var/sysman/etc/xntpd
100754 740760295 740760295 var/sysman/etc/xntpdc
```

Updating Files in a File Collection

The SP file collections can be used to contain system control files that you will need or want to change. When you update files that are part of a file collection, you do not have to make the same update on each replication of the files. Instead, you make the update to the copy on the master server for that collection and run the **supper update** command on all systems you want updated. The **supper** program does the updates for you.

The **supper update** command is included in the **crontabs** file with the default set to run hourly. You can modify the **crontabs** file to run the **supper update** more or less frequently. Refer to Chapter 19, "Maintaining the crontabs File" on page 281 for more information.

Steps in this task:

Step 1. Make the changes to the files
Step 2. Run the **supper scan** command
Step 3. Run the **supper update** command

Step 1: Make the Changes to the Files

1. **Be sure you are working with the master files.**

   In a file collection there are many replications of the same file. It is possible for copies of a file to exist on the control workstation, on several boot/install servers, and on all the processor nodes.

   In the default system, the control workstation is the master server for all file collections. If you have not changed anything in this configuration, this is where you will always make updates.

   Note

   If you changed the default configuration, be aware of the hierarchy of servers in your system and be sure you are modifying the copy of the file that is on the master server for the collection.

   Consult "File Collections Organization" on page 217 to review file collection hierarchy.

2. Change the files as appropriate.

Example: In our default configuration, the master files for all file collections are on the control workstation.

Log in to the control workstation as **root** and copy a new version of /**.profile** into the /**share**/**power**/**system**/**3.2** directory.

Step 2: Run the supper scan Command

To keep the **scan** file current, you must run the **supper scan** command on the master collection every time you make a change.

Example: Run the **supper scan** command on the control workstation. Enter:

```
/var/sysman/supper scan power_system
```

Step 3: Run the supper update Command

You need to run the **supper update** command in every location that the collection is installed and where you want the change to be effective. For example, if you change the files on a server that is the master for eight processor nodes and you want each node to have the changes, you need to run the update on all eight nodes.

You can accomplish this update in several ways:

1. You can wait for the scheduled update. The **supper update** command is included in the **crontabs** file to be run on a scheduled basis.
2. You can issue the commands immediately, logging in to each machine or remotely issuing the commands with **rsh** or **rexec**.

Example:

1. Log in to each server requiring the update. The **.profile** file is in the **node.root** collection as a secondary collection available within the **power_system** collection. This means on the boot/install servers you update the **power_system** collection. Enter:

   ```
   /var/sysman/supper update power_system
   ```

2. Log in to each of the nodes requiring the update. On the nodes you update the **node.root** collection. Enter:

   ```
   /var/sysman/supper update node.root
   ```

3. Check the **supper** messages. **supper** returns the number of files updated, removed, and the number of errors produced as shown in this message which was returned after changing **.profile** and performing the update on a processor node.

   ```
   ===================================================================
   [r02n07][/]> supper update node.root
   Updating collection node.root from server mrcs-r1.
   File Changes:  1 updated, 0 removed, 0 errors.
   ===================================================================
   ```

   supper also writes summary messages to to the following file:
   /var/adm/SPlogs/filec/sup.<**month**>.<**day**>.<**year**>.<**hour**>.<**min**>
   Detailed information is written to the following file:
   /var/adm/SPlogs/filec/sup.<**month**>.<**day**>.<**year**>.<**hour**>.<**min**>**.r**
   File collection log files older than 48 hours are removed at midnight by a **crontab** entry. To see all the messages in the log, you have to display or print the **logs** file. You can check the **supper** log for messages from the last install or update using a **supper** command. The following example shows a **supper update**, followed by a **supper rlog** command:

```
[r05n01][/var/adm/SPlogs/filec]> supper update node.root
Updating collection node.root from server r05cw.
File Changes:  2 updated, 0 removed, 0 errors.


[r05n01][/var/adm/SPlogs/filec]> supper rlog
SUP 7.24 (4.3 BSD) for file /tmp/.sf13689 at May  6 12:01:57
SUP Upgrade of node.root at Fri May  6 12:01:57 1994
SUP Fileserver 7.12 (4.3 BSD) 26295 on r05cw.hpssl.kgn.ibm.com
SUP Locked collection node.root for exclusive access
SUP Requesting changes since Fri May  6 12:00:07 1994
SUP Receiving file .kshrc
SUP Receiving file .profile
SUP Upgrade of node.root completed at Fri May  6 12:01:58 1994
```

Adding and Deleting Files in a File Collection

Adding and deleting files in a file collection are a bit more complicated than adding and deleting files in a standard directory. The **prefix, list,** and **refuse** master files contain criteria that define which files are in a particular collection. The master files are read and processed by the **scan, install**, and **update** commands. When you add or delete files in a file collection, you need to review the contents of these master files. For more information about the master files and how they work, see "Building a File Collection" on page 228.

The examples in this section show how to add the **/etc/environment** file to the **node.root** collection.

Steps in this task:

Step 1. Add or delete files in the file system
Step 2. Run the **supper scan** command (optional)
Step 3. Run the **supper update** command

Step 1: Add or Delete Files in the File System

1. **Be sure you are working with the master files.**

   The same considerations you make to update files also apply here. Verify that you are on the machine that is the master server for the collection you are changing. Consult "File Collections Organization" on page 217 to review file collection hierarchy.

2. Add or delete files in the file system using standard AIX file commands.

   When adding files, you need to consider whether the files are in a primary or secondary collection, what the **prefix** and **list** files in that collection contain, and the position of the new files in the file tree. Remember, the **supper** commands begin their search starting at the point defined in the **prefix** file. All files encountered that pass the criteria defined in the **list** file are considered part of the file collection.

Examples:

1. If you wanted to add a file to the **node.root** collection which is a secondary collection under the **power_system** collection, you would copy the file to the

/share/power/system/3.2 directory. The **/var/sysman/sup/node.root/prefix** file shows that the directory search for this collection begins at **/share/power/system/3.2**. The **/var/sysman/sup/node.root/list** contains an entry (**upgrade .**) indicating that all files encountered during the search will be upgraded:

```
symlinkall
omit    ./usr
upgrade .
```

To add **/etc/environment** in our default configuration, log in to the control workstation and copy **/etc/environment** to the **/share/power/system/3.2** directory:

```
cp /etc/environment /share/power/system/3.2/etc/environment
```

2. If you wanted to add a file to the **user.admin** collection, you would proceed differently. The **/var/sysman/sup/user.admin/prefix** file shows that the directory search starts at the root of the file tree, therefore, the directory search would find the **/etc/environment** file. The **/var/sysman/sup/user.admin/list** file shows that only those files specifically listed will be upgraded instead of the entire directory.

   In this case, you do not need to copy the file. Instead, add a specific entry to the **list** file to upgrade it.

Note

To add a file to a collection, the file must be readable by others.

To delete a file from a file collection, you would either remove it on the master server or delete its entry in the **list** file and proceed with the other steps in this task.

For example, to remove **/etc/environment** from the **node.root** collection, log in to the control workstation and enter:

```
rm /share/power/system/3.2/etc/environment
```

Step 2: Run the supper scan Command (optional)

The **scan** is not a required file and not all file collections have one. Check the file collection directory to see if the collection you are updating has a **scan** file.

- If you have a **scan** file in this collection and you want to maintain it, you must run the **supper scan** command on the master collection every time you make a change. When a **scan** file is present, the **update** command reads it as an inventory of the files in the collection and does not do the directory search.

  You also have the option of deleting the **scan** file and allowing **supper** to process the file collection without it.

  Note

  If you keep the **scan** file and fail to run the **supper scan** command when you add files, the **scan** file will not be current and the new files will not be included in the collection.

- If you do not have a **scan** file in this collection, the **update** command will search the directory, apply the criteria in the master files and add the new file.

Example:

1. Run the **supper scan** command on the control workstation:

   ```
   /var/sysman/supper scan power_system
   ```

   This builds a new **scan** file for the collection.

2. Check the **scan** file to verify that it contains the additional /**etc**/**environment** file.

Step 3: Run the supper update Command

Run the **supper update** command everywhere you want this change to be effective. Refer to "Step 3: Run the supper update Command" on page 225.

Example:

1. Log in to each server requiring the new file. The /**etc**/**environment** file is in the **node.root** collection as a secondary collection available within the **power_system** collection. Enter:

   ```
   /var/sysman/supper update power_system
   ```

2. Log in to each of the nodes requiring the new file. On the nodes you update the **node.root** collection. Enter:

   ```
   /var/sysman/supper update node.root
   ```

3. Check the **supper** messages to verify the updates worked.

Building a File Collection

Once you become familiar with file collections and have worked with the delivered system collections, you may want to create your own. You can build a file collection for any group of files that you want to have identically replicated on nodes and servers in your system. These boot/installs must all reside on the control workstation or a boot/install server. Some good candidates for file collections are application data files or local tools.

The basic procedure to build a file collections is:

- Begin by logging in as **root** to the server where the files reside.
- Create the directory and create a set of master files by copying them from an existing file collection. The master files define the rules for including and excluding files in the collection.
- Optionally, run the **supper scan** command to search the file tree, apply the rules, and build a **scan** file.
- Set up a link in the /**var**/**sysman**/**sup**/**lists** for the new file collection's **list** file.
- Add an entry to the **file.collections** file to define the file to the **supper** program.

After you build your file collections, you can install them on other servers or nodes.

Steps in this task:

Step 1. Identify the files you want to collect
Step 2. Create a file collection directory
Step 3. Create the master files
Step 4. Add a link to the **lists** file
Step 5. Update the **file.collections** file
Step 6. Update the **.resident** file
Step 7. Build the **scan** file (optional)

This section provides examples showing you how to build a primary file collection for a set of local tools in the /**usr**/**local** directory for the control workstation and boot/install servers.

Related Information

Before you build a file collection of your own, be sure you understand the directory and master files, what they contain, and how they work. Refer to "Understanding File Collections" on page 214 if you need to review these concepts.

Step 1: Identify the Files You Want to Collect

The files you want to collect as a group must all reside on the same system from which you want to serve this collection and their permissions must allow them to be readable by everyone.

Example: In this example, the files will be served from the control workstation so they must all reside there in /**usr**/**local**.

Step 2: Create a File Collection Directory

Create a directory in /**var**/**sysman**/**sup** that will be the name of your file collection.

Example: This file collection will be called *tools*.

```
cd /var/sysman/sup
mkdir tools
```

Step 3: Create a list File

Create a **list** file to describe the rules for including and excluding files in that directory. Refer to "supper list File Syntax, Keywords, and Operands" on page 236 for details.

Example: The easiest way to create these files is to copy them from an existing file collection directory, delete those you do not need or the system creates, and modify them as required.

```
cp sup.admin/* tools
cd tools
rm when last scan
```

Modify the **list**, adding these commands:

```
symlinkall
omitany *~
omit ./usr/local/admin
upgrade ./usr/local
always ./usr/local/admin/control_file
execute /usr/local/refresh_fonts (./usr/local/fonts)
```

This instructs **supper** to:

- Copy links rather than resolve them
- Omit files that start with the backup notation
- Omit administrative tools
- Upgrade all files in the local tools directory
- Override an omit instruction and always update the control file
- Execute the script to refresh the font library whenever the /**usr**/**local**/**fonts** file is changed

Step 4: Add a Link to the lists File

The **lists** file in the /**var**/**sysman**/**sup** directory contains a symbolic link to the **list** file in each file collection. When you create a new file collection you must add a link to this file.

Example: Add a link pointing to the **list** file in the *tools* collection:

```
ln -s /var/sysman/sup/tools/list  /var/sysman/sup/lists/tools
```

Step 5: Update the file.collections File

Edit /**var**/**sysman**/**file.collections** using your text editor. Add the name of your new file collection as either a primary or secondary file.

Example: The *tools* collection is a primary collection and requires an entry similar to **server.root**. Edit /**var**/**sysman**/**file.collections** and add these lines to the end:

```
# tools - boot/install collection to manage local tools files on the file servers
primary tools - / - / EDO power no
```

The nine fields in this entry have specific meaning to **supper**:

Field 1 Defines the file as a primary or secondary collection

Field 2 Specifies the name of the file collection

Field 3 For a primary collection, this specifies the name of the file system associated with the collection. This file system will be created when the file collection is installed. For a secondary collection, this specifies the name of its primary collection. The notation (-) indicates no file system will be created or that it is a secondary collection.

Field 4 Specifies the name of the directory by which the files are normally accessed.

Field 5 If the collection has a file system, this specifies the file system size. The notation (-) indicates no file system is associated with this collection or that it is a secondary collection and the file system is associated with its primary collection.

Field 6 Specifies the prefix directory at which the **scan** process starts.

Field 7 Specifies SUP options. In our example,

- E allows **list** file **execute** statements
- D deletes files that no are longer in the collection.
- O indicates that all files (even those older than the last update) should be evaluated.

Field 8 Specifies the system architecture. **power** indicates RISC System/6000 architecture.

Field 9 Specifies if this file collection can be installed on a different architecture.

Step 6: Update the .resident File

The **.resident** file contains a list identifying all the SP file collections. Edit the **.resident** file on the control workstation or your master server and add your new collection.

Example: Add the tools file collection to the **/var/sysman/sup/resident** file on the control workstation:

```
sup.admin 0
user.admin 0
power_system 0
node.root 0
tools 0
```

Step 7: Build the scan File

The **scan** file provides you with a list of files in the collection that you can use for verification and eliminates the need for **supper** to do a directory search on each update. If your directory tree is extensive, this can save processing time on large file systems

Note

You must keep the **scan** file current. When a **scan** file is present, the **update** command reads it as an inventory of the files in the collection and does not do the directory search. If you fail to create a new **scan** file when you add, modify, or delete files in the master collection, the file will not be current and **supper** will not upgrade the collection correctly.

Example: To create a **scan** file for the tools file collection, enter:

```
/var/sysman/supper scan tools
```

Installing a File Collection

During the initialization and customization process, the required SP file collections are first built on the control workstation and then installed on your boot/install servers and processor nodes. If you create your own file collections you have to install them yourself on each server or node.

To install a collection on a boot/install server or processor node, you must be logged in as root or issue remote commands to run **supper** there.

Steps in this task:

Step 1. Update the **sup.admin** file collection
Step 2. Run the **supper install** command
Step 3. Add the **supper update** to **crontabs**

Step 1: Update the sup.admin File Collection

The **sup.admin** file collection contains the files that identify and control all the file collections. It contains the **file.collections** and **.resident** files. Whenever you make changes to these files, you need to update the **sup.admin** collection to distribute these updates.

You can wait for the scheduled update in the **crontabs** file or issue the command yourself on every boot/install server and node that needs the new collection.

Example: Log in to each boot/install server and node or issue the command remotely and enter:

```
supper update sup.admin
```

Step 2: Run the supper install Command

Run **supper install** on each boot/install server or node that needs this collection. The **install** command assumes the dedicated server to be the default.

Example: In this example, the tools file collection will only be installed on the boot/install servers and not the processor nodes.

Log in to each boot/install server or issue the command remotely and enter:

```
/var/sysman/supper install tools
```

Step 3: Add the supper update to crontabs

You need to add the **supper update** command to the **crontabs** file on each server or node that has this collection. This insures that the new collection will be updated on schedule with the other file collections.

Example: To add the command so it runs hourly, edit **crontabs** and add this line:

```
0 * * * * /var/sysman/supper update tools 1>/dev/null 2>/dev/null
```

Until the update runs, the files will not be available at each location. If you need the files on these systems before the scheduled update, you can issue the update command on each server and processor node yourself.

Example: Log in to each boot/install server or issue the command remotely, and enter:

```
/var/sysman/supper update tools
```

Refusing Files in a File Collection

The **refuse** file allows you to customize the file collection at different locations. It is possible for you to create a file collection with one group of files and have different subsets of that group installed on the boot/install servers and the processor nodes.

You do this by creating a **refuse** file in the **/var/sysman/sup** directory on the boot/install server or processor node. This master file lists all the files in every file collection that you do not want installed or updated on that system.

Example: In this example, we want one boot/install server to have the full file collection including the test files and another to have the boot/install collection minus the test boot/installs. Log in to the boot/install server that should not have the test files and create a **refuse** file with the following entries:

```
./usr/local/draw.test
./usr/local/crunch.test
```

This instructs **supper** to never include these test files when installing or updating this file collection *at this location.*

Note

There are two kinds of **refuse** files. One is in the **/var/sysman/sup** directory and contains the files that will not be installed on that server or node. These **refuse** files are optional and can vary at each location. The other **refuse** file is in the file collection's directory and contains a list of all the files refused when the file collection is installed or updated. This **refuse** file is system-created and may be empty if you always install the entire collection.

Modifying the File Collection Hierarchy

The delivered file collections have a default hierarchy shown in Figure 49 on page 217. You can modify the hierarchy in several ways. One way is to make a given boot/install server function as the master server for a collection. This enables you to maintain a different set of files on a group of nodes that request their **supper** updates from that boot/install server. You do this by eliminating the logical path from the control workstation to the boot/install server for one or more boot/install collections. This is called taking the boot/install server *offline* and you do it with the **supper offline** command. Figure 52 on page 234 shows a boot/install server that is offline for the **power_system** collection.

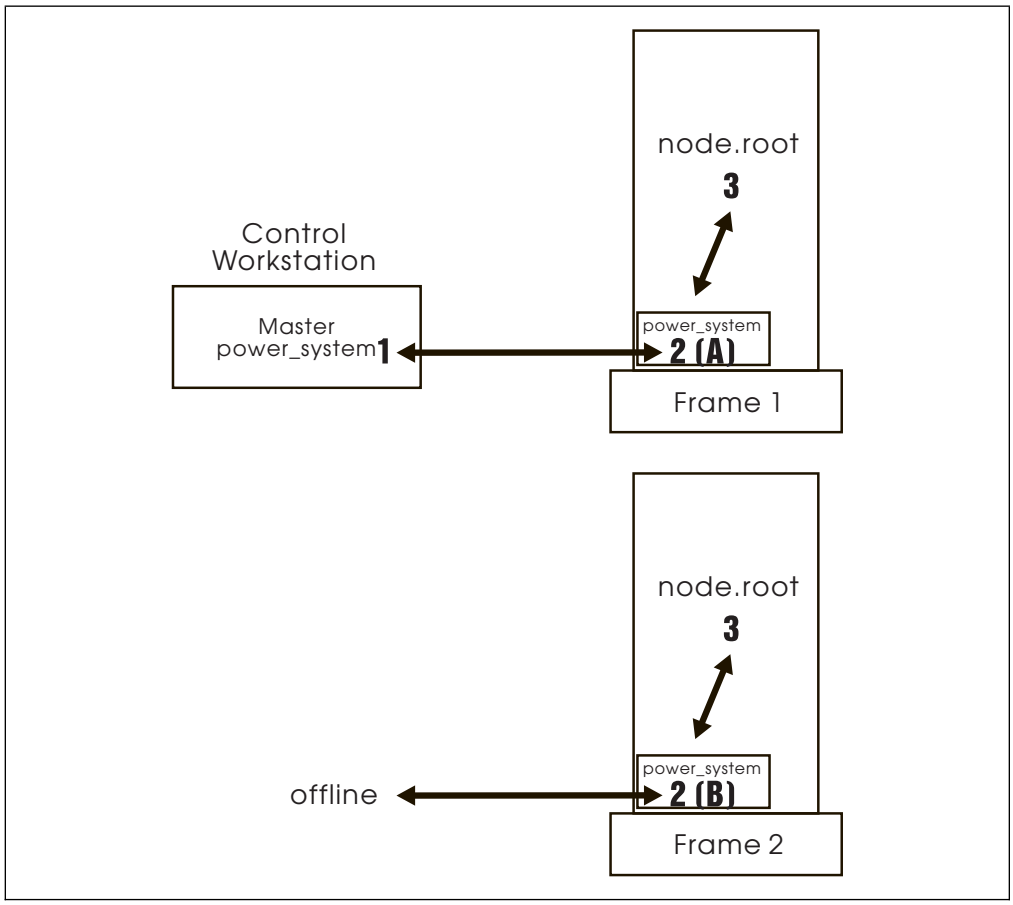


Figure 52. SP Hierarchy of an Offline Boot/install Server

In this case, Boot/install Server A and Frame 1 get any changes made on the control workstation; Boot/install Server B and Frame 2 do not. Only changes made directly to the files on Boot/install Server B affect the files on Frame 2.

Note

Whenever you change the hierarchy, you must be aware of the location of the master files for every file collection. In Figure 52, there are two master file collections for **power_system**. The master files for the **node.root** collection on Frame 2 reside on Boot/install Server B and the master files for the **node.root** collection on Frame 1 reside on the control workstation.

- If you want to change a file in the **node.root** collection for every node, you must make that change in two places, on Boot/install Server B and on the control workstation, and then run the **supper update** command on each node.
- If you want to change a file in the **node.root** collection for just one frame of nodes, you must make that change in the master file collection for that frame, either Boot/install Server B or the control workstation, and then run the **supper update** command on each node in that one frame.

The following example shows you the procedure and commands to change the default hierarchy as represented in Figure 49 on page 217. It shows you how to take the **power_system** collection offline, change a file in the **node.root** file collection, and distribute that update to the nodes on Frame 2.

Example: This example assumes that **.profile** is in the **node.root** file collection.

1. To take the boot/install server offline from the control workstation, log in to Boot/install Server B as **root** and enter:

   ```
   /var/sysman/supper offline power_system
   ```

2. On Boot/install Server B, modify the **.profile** in the /**share**/**power**/**system**/**3.2** directory.

   Remember, any changes made on the control workstation will not be updated on Boot/install Server B because it is offline. You must make the changes on Boot/install Server B.

3. Issue remote commands to update the **node.root** file collection on each of the nodes.

   ```
   /var/sysman/supper update node.root
   ```

4. Boot/install Server B no longer gets updated files from the control workstation. Unless you remove the **supper** command to update the **power_system** from the **crontabs** file on Boot/install Server B, the command will execute every hour and fail. Edit the **crontabs** file and remove:

   ```
   0 * * * * /var/sysman/supper update power_system 1>/dev/null 2>/dev/null
   ```

Removing a File Collection

> **Warning:**
> Never remove any of the file collections that come with the SP system!

The file collections that come with the SP system are required. If you remove them, the System Support Programs cannot run properly.

Removing a file collection does not delete it. It removes it from the place it was installed. To completely delete a file collection, you must remove it from every place it was installed. Be sure you do not need any files in a file collection before you remove it.

You can use the **supper scan** command to build a **scan** file for the file collection you are going to remove. The **scan** file shows you a list of all the files in a collection. You can check it to verify that you do not need any of these files before you remove the collection. See “Verifying File Collections Using scan” on page 223 for more information about the **scan** file.

Example: To build a **scan** file for verification, enter this command on the boot/install server where the *tools* collection is installed:

```
/var/sysman/supper scan tools
```

If you are sure you do not need any of the files, continue with the next command.

To remove the *tools* boot/install collection, on the same boot/install server enter:

```
/var/sysman/supper remove tools
```

This removes all the files in the collection only from that machine. If you also issue this command on the control workstation or the master server for the collection, you will no longer be able to propagate these files in your system.

---

supper list File Syntax, Keywords, and Operands

The following commands are used in the **supper list** file to define which files in a directory to include or exclude from a file collection. Refer to "Directory and Master Files" on page 217 for information about the **supper list**.

Syntax Rules

- Each command contains a keyword and a number of operands separated by spaces.
- The order of the commands does not matter and you can include blank lines between the commands.
- *filename* can be a single file, directory, or a list of files or directories separated by blanks. The names can include wild cards and metacharacters as used by **csh**, including *, ?, [....] and {...}.

Command Keywords and Operands

- **upgrade** *filename*

  The specified files or directories are included in the list of files to be upgraded unless they are also specified by an **omit** or **omitany** command or are in the **refuse** file. If a directory name is given, it recursively includes all subdirectories and files within that directory.

- **always** *filename*

  The specified files or directories are included in the list of files to be upgraded, regardless of the **omit** or **omitany** commands. They will not be included if they are in the **refuse** file.

- **omit** *filename*

  The specified files or directories will be excluded from the list of files to be upgraded. For example, by specifying **upgrade /usr/vision** and **omit /usr/vision/exp**, the generated list of files would include all subdirectories and files of **/usr/vision** except **/usr/vision/exp** and its subdirectories and files.

- **omitany** *pattern*

  The specified patterns are compared against the files specified in the **upgrade** entry. If a pattern matches, the file is omitted. The **omitany** command currently supports all wild card patterns except {...}. Also, the pattern must match the entire file name, so a leading */, or a trailing /*, may be necessary in the pattern.

- **backup** *filename*

The specified files are marked for backup. If they are upgraded, backup copies are created. Directories cannot be specified and no recursive file name construction is performed; you must specify the names of the specific files to be backed up before upgrading.

- **noaccount** *filename*

  The accounting information (permissions and time stamps) of the specified files is not preserved.

- **symlink** *filename*

  The specified files are to be treated as symbolic links. They are transferred as links and not followed. By default, symbolic links are followed. **symlinkall** transfers all symbolic links.

- **execute** *exec-command* (*filename*)

  The command specified is executed on the client process whenever any of the files listed in parentheses are upgraded. A special token, %s, may be specified in the command and is replaced by the name of the file that was upgraded. For example, if you use `execute ranlib %s (libc.a)`, whenever `libc.a` is upgraded, the client machine will execute `ranlib libc.a.`.

- **include** *listfile*

  The specified *listfile* is read at this point. This is useful when one collection subsumes other collections; the larger collection can simply specify the files for the smaller collections contained within it.

Related Information

The *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference* presents the full **supper** command syntax with all the parameters and options. It also explains how to use **supper** in batch or interactive mode.

You can find additional information about **supper** and SUP in their manual pages.

Chapter 15. Managing Amd

Understanding the NFS and Amd Implementation

NFS is a distributed file system that allows you to access files and directories located on remote systems and treat those files and directories as if they were local.

Amd, the BSD automounter is optionally used to manage NFS mounting of user home directories and other directories. When configured, an Amd daemon runs on each node, mounting directories on demand and unmounting them after a period of inactivity. The mounted directories can come from SP boot/install servers or any workstation or server in the network.

Amd manages directories specifically defined in the Amd map files. The Amd daemon runs on the control workstation, boot/install servers, and every processor node in the SP system. Amd works with NFS to mount or with AFS to link these directories on demand when they are first referenced and unmount or unlink them after a period of inactivity.

You can use Amd to provide remote access to users' home directories and their files or you can rely on other methods of automounting established at your site by specifying your choice when defining your site environment, either through SMIT panels or using the **spsitenv** command. If you set **amd_config** to `false` when you initially install the SP system, and decide at a later date to use Amd, you can use SMIT or **spsitenv** to change the variable to `true`. Should you decide to unconfigure Amd at a later time, use SMIT or **spsitenv** to change the variable to `false`. When you change **amd_config** from `true` to `false`, the Amd daemons are killed only when you reboot the SP system or take other actions to kill the daemons.

Note: If you choose to unconfigure Amd, make sure that all user home directories, as well as any other directories previously served by Amd, are distributed and made available by some other means.

If you choose Amd as your automounter, you can also use it to provide similar access to any other directories that you would like to have automatically mounted on demand.

Changing Amd Defaults

Amd groups all the NFS mount points into the /**a** directory by default and it unmounts a directory after two minutes (120 seconds) of no activity. You can change the default directory and the waiting period for unmounting by following these steps:

1. **cd** /**etc**/**amd** on the control workstation and all nodes and edit the line which starts Amd in the **amd_start** file:

   ```
   system("nice --4 $amd -t 16.120 -x all -l $amdlog -p >\
   /etc/amd/amd.pid @amdargs") && exit(0);
   ```

2. Add the **-a** *DirectoryName* or the **-w** *wait-timeout* flags to the command. Enter the wait time in seconds.

© Copyright IBM Corp. 1995

3. If you are using file collection, copy the file to **/share/power/system/3.2/etc/amd/amd_start** on the control workstation. This puts it in the **node.root** secondary collection to be served by the **power_system** file collection.

   If you are not using file collections, use another method to distribute the Amd maps to each node, and then use **dsh** to run **amd_start**. Go to step 5.

4. Run **supper update** on the boot/install servers and **supper update user.admin** on the nodes to propagate the update to all the nodes.
5. Stop and restart Amd (refer to information on starting and stopping Amd in the *IBM RISC System/6000 Scalable POWERparallel Systems: Diagnosis and Messages Guide* for instructions) or reboot the nodes.

See Chapter 14, “Managing File Collections” on page 213 for more information on the file collection processes.

Amd Maps

An Amd map is a file in the **/etc/amd/amd-maps** directory that associates the name of the directory being referenced to its true directory or file system. An Amd map file can also contain:

- The host name where the true directory or file system resides
- Logical tests to decide whether to use a link or an NFS mount when accessing the referenced directory

If Amd and the SP User Management interface are configured for your site environment, **/u** is automatically managed by Amd. The Amd map for **/u** is **/etc/amd/amd-maps/amd.u**. If you use file collections, the map is included in the **user.admin** file collection. It contains a stanza for each user added through SP User Administration, giving the location of the user's home directory. Existing users must be added manually. The entries in these stanzas give instructions so that:

- If a user is on the server where a directory resides, the directory is linked
- If a user is on another machine, the directory is mounted by NFS
- If users' files are accessed using AFS, then the directory is linked

Note: An entry is also added to **amd.u** for the net install ID's home directory. This ID is an AIX requirement for a net install.

This happens as part of every log in process when users are first connected to their home directories but it also happens anytime someone changes to a directory or references a file within a directory defined in the Amd maps. The following example illustrates how this works.

Example: In the SP system, by default, the users' directories reside in **/home**/*hostname*, where *hostname* is the short name for the host system. The following stanza shows the entries in the Amd map for a user, *jws*, whose home directory resides on *luna*.

```
jws     host==luna;type=link;fs:=/home/luna
        host!=luna;type=nfs;rfs:=/home/luna;rhost:=luna
```

In this case, if a user is on **luna** and enters **cd** /**u**/**jws**, Amd will link /**home**/**luna** to /**u**/**jws**.

```
/u/jws→/home/luna/jws
```

But, if a user is on any machine other than *luna* and enters **cd** /**u**/**jws**, Amd mounts /**home**/**luna** and establishes a link from /**u**/**jws** to the mount point.

```
/u/jws→/a/luna/home/luna/jws
```

Note: /**home**/*hostname/username* is the default for home directories in these examples. Your site may already have a different implementation which you can use in the Amd maps. For example, if your default is /**home**/*username*, the Amd map entries for the user *msml* whose home directory resides on the host *starship* would be:

```
msml    host==starship;type=link;fs:=/home
        host!=starship;type=nfs;rfs:=/home;rhost:=starship
```

Users may also have their home directories in AFS. A map for a user with an AFS home directory would look like this:

```
tberry          type:=link;fs:=/afs/kgn.ibm.com/usr2
```

In this case, when a user enters **cd** /**u**/**tberry** Amd links /**afs**/**kgn.ibm.com**/**usr2**/**tberry** to /**u**/**tberry**.

```
/u/tberry→/afs/kgn.ibm.com/usr2/tberry
```

Amd Diagnostics

Amd writes messages to an error log on the control workstation, /**var**/**adm**/**SPlogs**/**amd**/**amd.log**. If you are having problems with Amd, you may need to stop it, check the diagnostic information, and restart it. You can find a description of the Amd error log, potential messages, and instructions for stopping and starting Amd in the *IBM RISC System/6000 Scalable POWERparallel Systems: Diagnosis and Messages Guide*.

Establishing an Amd Map for Users' Directories

If you choose to use Amd as your automounter, the installation process configures Amd for you on the SP control workstation and processor nodes, and includes Amd maps in the **user.admin** file collection, if file collection is used. This means that Amd map updates are managed by the file collection software and are automatically distributed throughout the system. Chapter 14, “Managing File Collections” on page 213 provides more information on how the file collection distribution works.

To provide access to the SP users' home directories, you need to modify the existing **amd.u** map. You can use the **mkamdent** command to build the entries in this map or you can edit and modify the map manually.

Modifying the amd.u Map

The Amd map for /**u** is /**etc**/**amd**/**amd-maps**/**amd.u**. It contains a stanza for each user giving the location of the user's home directory. As part of the integration, you will need to add entries to this map for each of the SP users. If you are not using file collections, the **amd.u** map that resides on the install server must be copied to the nodes before Amd is started.

Using mkamdent

The **mkamdent** command allows you to specify a list of users to add to the **amd.u** map. The command builds a stanza for each user in the list using site environment values as the defaults. **homedir_server** is the default server for the users' home directory and the default base path for the home directory is **homedir_path**. You can override these defaults with the **-s** flag. See the *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference* for more information.

Modifying the Map Manually

To make the changes manually, edit the original **/etc/amd/amd-maps/amd.u** map file on the control workstation. It contains a sample template for you to follow and entries for the user account names reserved for SP processes. Part of this file is shown in Figure 53.

```
#
# To access the home directory on the system that it resides,
# use the following example:
#
#       host==servername;type:=link;fs:=filesystem
#       host!=servername;type:=nfs;rfs:=filesystem;rhost:=servername
#

prtid          host==tserv11;type:=link;fs:=/home/tserv11 \
               host!=tserv11;type:=nfs;rhost:=tserv11;rfs:=/home/tserv11
```

Figure 53. Sample /etc/amd/amd-maps/amd.u File

If you use file collections, the **amd.u** file is part of the **user.admin** file collection. It will be updated on the boot/install servers and all the nodes by the **supper** update process, as scheduled in the **crontabs** file.

Related Information

Amd documentation is in a compressed **tar** file, **/usr/lpp/ssp/src/amd.tar.Z**. It provides details about the Amd functions. To read it, you must first expand it with the **zcat** command and then extract the files with **tar -xvf**. You can find information about the **user.admin** collection and file collection distribution in Chapter 14, “Managing File Collections” on page 213.

Using Amd for Other File Systems

If you choose Amd as your automounter, you can also use it to provide automounting for other file systems in your network. Each file system you want managed by Amd requires an Amd map in the **/etc/amd/amd-maps** directory. These maps associate the name of a directory being referenced to its true directory or file system.

The examples in this section show how to provide Amd automounting for the **/vol** file system. Because this process requires you to reboot all the nodes, you should plan to add all your additional file systems to Amd at a time when this would not impact your system schedule.

Exceptions

Be aware that you cannot have Amd manage a directory and still maintain the directory on the same system with the same name. For instance, in our example, **amd.vol** will mount over anything that is in /**vol** with an empty directory. You would not be able to maintain data in /**vol** and have Amd use the **amd.vol** map to manage /**vol** on the same system.

You can work around this by moving the source directory, making it available under another directory, or putting it in a secondary file collection so that it is only managed on the processor nodes.

Steps in this task:

Step 1. Create an Amd map
Step 2. Add the map to the file collections
Step 3. Distribute the map to the boot/install servers and nodes
Step 4. Reboot all systems that have the new map

Step 1: Create an Amd Map

The easiest way to create an Amd map is to copy an existing one and modify it for the new file system. The Amd naming convention appends the file system name to **amd**. The map for the /**u** file system is **amd.u**, therefore, the map for the /**vol** file system would be **amd.vol**.

Example:

1. On the control workstation, copy the existing **amd.u** map and name it **amd.vol**, keeping it in the /**etc**/**amd**/**amd-maps** directory. Enter:

   ```
   cp /etc/amd/amd-maps/amd.u /etc/amd/amd-maps/amd.vol
   ```

2. Replace the map entries with the directory and host information for /**vol** as shown here:

```
/defaults    type:=nfs;opts:=ro,soft,rsize=4096,wsize=4096
inst.images  rhost:=images1;rfs:=/inst.images
techlib      rhost:=filesrv1;rfs:=/share/${key}
local        type:=link;fs:=/afs/site.edu/share/local
```

This example results in the following mounts:

- If a user issues **cd** /**vol**/**inst.images**, **inst.images** is mounted from **images1** linking /**vol**/**inst.images**→/**a**/**images1**/**inst.images**
- If a user issues **cd** /**vol**/**techlib**, /**share**/**techlib** is mounted on /**a**/**filesrv1**/**share**/**techlib** linking **vol**/**techlib**→/**a**/**filesrv1**/**share**/**techlib**
- If a user issues **cd** /**vol**/**local**, the system performs the link because **type=link** overrides the default **type=nfs**. The result is /**vol**/**local**→/**afs**/**site.edu**/**share**/**local**.

Step 2: Add the Map to the File Collections

Add the new map to the file collections so that it can be distributed and managed by the file collection programs. Choose a file collection that is resident on all the systems you require. For instance, the **user.admin** collection is resident on the processor nodes as well as on the control workstation and boot/install servers. Refer to "SP File Collection Summary" on page 216 for a complete listing.

Example: Add **amd.vol** to the **user.admin** file collection so that it is resident on the control workstation, boot/install servers, and processor nodes. To do this, edit the **list** file in the **/var/ssp/sup/user.admin** directory and add an entry requesting an upgrade of this new map:

Before the change:

```
symlinkall
upgrade ./etc/amd/amd-maps/amd.u
execute /etc/amd/refresh_amd (./etc/amd/amd-maps/amd.u)
```

After the change:

```
symlinkall
upgrade ./etc/amd/amd-maps/amd.u
upgrade ./etc/amd/amd-maps/amd.vol
execute /etc/amd/refresh_amd (./etc/amd/amd-maps/amd.u ./etc/amd/amd-maps/amd.vol)
```

Step 3: Distribute the Map to the Boot/Install servers and Nodes

Update the **scan** file on the control workstation. Enter:

```
supper scan user.admin
```

You can wait for the scheduled **supper update** to update the **user.admin** file collection as specified in the **crontabs** file. It will update each node to include the new map file.

If you want to distribute the map immediately, use a remote command to run **supper update user.admin** on each file server and node.

Step 4: Reboot All Systems That Have the New Map

To complete this process, you need to reboot all the systems where this new map will reside.

Example: In this example, reboot the control work station, the boot/install servers, and all the processor nodes.

Chapter 16. Managing Print Service

The SP Print Management System sends all print jobs to a machine designated as *Print Host*. Using the Print Host to centralize the print process simplifies print management in the following ways:

- You do not need to run the **lpd** daemon on each node.
- You do not need to establish print queues on each node.
- You can check the queue status on a Print Host for all the nodes that it serves.

The machine you select to be a Print Host may be the control workstation, a wide node within the SP system, or a host outside the SP system. (If you want a non-RISC System/6000 to serve as Print Host, see "Using a Non-AIX Workstation as Print Host" on page 251.)

The SP Print Management System supports a subset of print commands and options listed in "Supported Commands" on page 247. When a user issues a print command, the Print Management System executes programs that first parse the command for unsupported options and then sends the command, using **rsh**, to the Print Host to be processed.

Before you integrate SP print management into your environment, read the following brief description of how it works, what its requirements are, and how its security modes affect your current environment. The security mode you choose determines what changes you must make to system files as part of this integration process.

When you understand the implications and requirements of the SP Print Management System, "Integrating SP Print Management" on page 249 takes you through the steps necessary to set up your environment.

Print Management Security Modes

You can set up the print management system to run in either Open Mode or Secure Mode. Each mode requires different procedures for set up and each has a different impact on your installation.

Secure Mode

Secure Mode requires only one user to have **rsh** privileges. This is a special user account with a default user ID of **prtid**. The **prtid** account is responsible for transferring jobs to the Print Host. In Secure Mode, all print jobs are *owned* by this special user account.

If UNIX accounting is active on the Print Host, the accounting for printed pages falls under the Print Management ID rather than the user ID that submitted the job.

Secure Mode also requires you to add an **.rhosts** file containing permissions for all hosts and all users to the **prtid** user account on the Print Host. This replaces the more common practice of adding all users to the /**etc**/**hosts.equiv** file.

Because **prtid** is the user ID that actually sends the files to the Print Host, you must create this user ID on the SP system as well as on any Print Host outside the SP system.

© Copyright IBM Corp. 1995

If **prtid** is not a suitable name for your installation, you can choose another name as long as it is not **root** and not assigned to another user. To choose another user ID, fill in the user ID name you want to use under `Print_id` on the site environment worksheet.

The program that sends the print jobs to the Print Host runs in setuid mode with **rsh** privileges to the Print Host. It makes **prtid** the owner of all print jobs. As a result, the *Username* option on print commands is ineffective, and the **prtid** name appears on reports and listings in place of the user who submitted the job.

The ownership of all print jobs by the print management ID operating in Secure Mode may affect how your system accounting is handled. Keep this in mind when setting up your accounting (“Setting Up SP Accounting” on page 269).

To help identify print jobs, the Print Management System appends the user's name to the file name as shown in this example:

```
Queue    Dev    Status     Job Files            User   PP  %  Blks   Cp  Rnk
======   ====   ========   ==================   =====  === =  ====   =======
prt1     prt     DOWN
prt1     prt     QUEUED    551 user:small.file  prtid            1    1    1
                 QUEUED    552 user:STDIN       prtid            1    1    2
```

Maintenance Reminder: In the Secure mode, whenever a user cancels a print job or if an administrator clears the entire print queue, the message ‘`Job number xxx has been deleted from the queue`’ is sent to **prtid** on the Print Host rather than to the user who issued the command. The **prtid** user on the Print Host accumulates all mail messages for deleted files. You may want to set up a **cron** job to periodically clean up this mail.

Open Mode

This mode is termed *Open* because it requires all users to have **rsh** privileges on the Print Host, which they can also use to run other commands. If this poses a security risk or is inappropriate for your installation, choose Secure mode.

To grant users **rsh** privileges, add each user ID to the /**etc**/**hosts.equiv** file. In Open Mode each user owns his print jobs and the *Username* option works correctly, showing the user's name on all reports and listings. Print pages will be accounted for under the normal user ID on the Print Host if UNIX accounting is enabled.

Print Management Command Support

The Print Host is required to have a full print subsystem. All commands appropriate for their operating systems are available when you are logged into these machines. The SP Print Management System runs only on the SP nodes and supports a subset of the AIX commands, flags, and options.

When the installation scripts configure the Print Management System, they move the AIX print commands to *command*.**AIX**. For example, **lpr** is moved to **lpr.AIX**. The scripts also create symbolic links from the print commands to **pmswitch** in the /**usr**/**lpp**/**ssp**/**bin** directory.

When a user issues a print command, the Print Management System executes programs that first parse the command for unsupported options and then send the command, using **rsh**, to the Print Host to be processed.

The Print Host uses standard AIX print subsystem and security modes do not apply. If you run print commands on an SP node serving as a Print Host, **pmswitch** invokes the equivalent *command*.**AIX** command.

Supported Commands

A subset of the AIX print commands are supported in the Print Management System. Depending on the security mode you choose, not all flags and options for these commands may work. The following table lists the supported commands and their exceptions for the Secure and Open modes. For more information on these commands, see *IBM AIX for RISC System/6000 Commands Reference*.

Command	Not Supported in Secure Mode	Not Supported in Open Mode
lp	**-c -m -w**	**-m -w**
lpq	*Username*	
lpr	**-m -r -s**	**-m -r -s**
lprm	*Username* **-P**	
lpstat	**-u**	
qcan	**-X -P**	
qchk	**-u -w**	**-w**
qprt	**-c -C -F -m -M -n -r**	**-C -F -m -M -n -r**

Note: The flag **-#h** is accepted for the **qprt** command. However, **qhld**, the command to move the print job out of the hold state, is not supported. You must log onto the print server host in order to use **qhld** to release a held print job.

To use these commands, type the flags and options separately as shown:

Correct:	`lpstat -d -r`
Incorrect:	`lpstat -dr`

Unsupported Commands

The commands that deal with local queues are not supported on the SP nodes and produce error messages. You can, however, use them successfully on the boot/install servers or control workstation where the standard AIX print subsystem runs. These commands are:

enque
qadm
qpri
qmov
qhld

SMIT is not supported with the SP print commands.

SP Print Management System Requirements

The SP System manages its print services with scripts that send print files to the Print Host for handling. Therefore, if you intend to use the Print Management System, it is necessary for you to have a print subsystem set up on all hosts that will provide print host services. Print Hosts will need local queues for any attached printers and remote queues for any remote printers.

If you do not have a print subsystem on the Print Host, you can:

- Refer to *IBM General Concepts and Procedures for RISC System/6000* for information.
- Enter **smit**, select *Devices*, and follow the process under *Printer/Plotter* to set one up.

You may use the control workstation, a wide node within the SP system, or a host outside the SP system as a Print Host. If you want to use a non-RISC System/6000 for a Print Host, see "Using a Non-AIX Workstation as Print Host" on page 251.

Disk Space for Print Jobs

The **/var/spool/qdaemon** and **/var/spool/lpd/qdir** directories on the Print Hosts must have enough space to handle the print jobs sent from the SP nodes.

Depending on your anticipated print load, you should plan for 0.5 MB to 2 MB of additional space, per node, in the Print Host's **/var** file system. For example, to support 16 nodes, plan an additional 8MB to 32MB for **/var**.

The Print Management System and Kerberos

The Print Management System can use Kerberos when the Print Host is a node in the SP system or the control workstation for those users listed as principals in the Kerberos database. The Print Management System **must** be used in open mode as Kerberos must authenticate the user as their own ID. The secure mode would try to authenticate as the Print Management System ID and would fail Kerberos authentication.

The users listed in the Kerberos database do not need to be listed in the **/etc/hosts.equiv** file.

For more information on administering Kerberos on the SP system, refer to Chapter 3, "Security Features of the SP System" on page 17.

Print Management and the rsh Command

Print Management uses the SP **rsh** to move files. This **rsh** makes a Kerberos call first and if that fails uses the AIX **rsh**. This makes the Print Management System usable by users who have done a **kinit** and by users who have not done a **kinit** without requiring extra configuration steps.

If all of your users will not be doing a **kinit**, open or secure mode can be used on a non-Kerberos basis. The rules regarding the **.rhosts** file in the print management ID home directory and the **/etc/hosts.equiv** file apply. Note that this means users in secure mode will be able to **rsh** using the Print Management ID. This is

necessary for the Print Management programs to work properly. A restricted shell can be set up for the Print Management ID. Refer to the manuals for AIX for more information.

Using a Non SP System Host for Print Host

For Print Management with Kerberos to work when a host outside of the SP is used as the Print Host:

1. The Print Host must be a a PSSP Kerberos client or a Kerberos client that is compatible with PSSP Kerberos.
2. The Print Host must be added to the Kerberos database used by the SP system as a service provider.
3. The Print Host must be in the same Kerberos domain as the SP machine.

Integrating SP Print Management

This procedure describes how to set up SP Print Management on your system.

Print Management Programs

This section references the following print management programs:

pmswitch	Routes the print commands to the standard AIX commands or to the SP print program, depending on where the commands are issued. When you issue print commands on a Print Host that is part of the SP system, **pmswitch** routes them to *cmd.***AIX** (the standard AIX command). When you issue print commands on a node, it routes them to **pmfe**.
pmfe	Processes the print command against the valid subset of commands. If the command is not supported, it returns a message. If the command is supported, it runs the command and then invokes **pmbec**.
pmbec	Directs the print files. The permissions on this file control the security mode of the Print Management System and are set during configuration according to your choice of secure or open mode.

Step 1: Verify Installation on the SP system

If you set the **print_config** attribute to `open` or `secure` before running the installation scripts, either through SMIT or **setup_server**, verify that the commands were copied from /**usr**/**bin** to *cmd.*AIX and are linked to **pmswitch**. To do this, use the **ls** command with the **-l** option and list the print commands to verify their symbolic links. For example, **ls -l** /**usr**/**bin**/**lpr** should return:

```
lrwxrwxrwx  1 root  system 16 Apr 12 17:26 lpr -> /usr/lpp/ssp/bin/pmswitch
```

Step 2: Set Up Print Management User ID (Secure Mode Only)

Create the print management user ID. This ID is called **prtid**, by default, unless you assigned a different name to the *Print_id* attribute when defining your site environment. This ID must not be root, must not have administrative privileges, and must be solely for the use of Print Management System.

If you are using Login Control on the SP nodes, create the **prtid** user ID with a **UID** lower than the *threshold ID* in the **block_usr_sample** file. Otherwise, the user ID will be blocked and may cause problems running print management.

If you should assign a new print management ID, a copy of the **pmbe** file is put in **pmbe.pcopy** during configuration. **pmbe.pcopy** will contain the previous print ID on each change of the print user ID.

Step 3: Set Up User's Environments

In order to use print management, an environmental variable, PRINT_HOST, in the user's environment must point to the Print Host. In Open Mode, the user can choose any host to which he has **rsh** privileges and where the AIX print subsystem is set up. In Secure Mode, print management will fail if the print user ID (**prtid** by default) is not set up on the host to which the user sends the print job.

The PRINT_HOST variable should be added to the user's login configuration file, such as **.profile** and **.login**, to target the Print Host upon login. Be sure to add the PRINT_HOST variable to the default login configuration file for new users as well.

Step 4: Set Up Print Host(s)

The setup procedure for the Print Host depends on whether you have chosen Open Mode or Secure Mode. (If you want a non-RISC System/6000 to serve as Print Host, see “Using a Non-AIX Workstation as Print Host” on page 251.)

Secure Mode

First, create the print management user ID on the Print Host. This ID is called **prtid**, by default, unless you assigned a different name to the *PRINT_ID* attribute when defining your site environment. This ID must not be root, must not have administrative privileges, and must be solely for the use of Print Management System.

If you are using Login Control on the SP nodes, create the **prtid** user ID with a UID lower than the *threshold ID* in the **block_usr_sample** file. Otherwise, the user ID will be blocked and may cause problems running print management.

For the Print Management System to run properly, an **.rhosts** file must be created in the home directory of the print management ID (**prtid** by default) on the SP system and on all Print Hosts. It should contain the hostnames and user names, if necessary, of the hosts and users allowed to send jobs to this Print Host. To allow all hosts and all users, put a plus (+) sign in the host field and in the user field of the **.rhosts** file.

It is important to keep the print management user ID secure. The following two measures help you safeguard your SP Print Management System.

1. After creating the **.rhosts** file in the home directory, place an asterisk (*) in the password field of the print user ID in the **/etc/passwd file**.

2. Remove all user's **su** privileges to this ID. (You can do this through SMIT.)

With these measures, users cannot log or rlogin to the print management user ID, nor can they **su** to this user ID.

To access the print management user ID after measures have been taken, **root** must remove the asterisk in the /**etc**/**passwd** file and assign a password to the print management user ID in order to update or change the **.rhosts** file. Users will still be able to **rsh** to the user ID from outside of the SP system, however. This communication is necessary for the Print Management System to work properly. Since the AIX print subsystem displays the ID in the status field, there is no way to hide the user ID being used for print management.

When using Secure mode, mail will be sent to the print management user ID whenever a print job is cancelled. You may wish to set up a **cron** job on the Print Host to remove this mail periodically.

Open Mode

To run the Print Management System in the Open Mode, add every SP user to the /**etc**/**hosts.equiv** file. To have print capabilities on all nodes, add them globally. If you want a subset of users or nodes, add them individually.

Note: Due to AIX security, the **root** users on the SP nodes cannot use the /**etc**/**hosts.equiv** file to gain **rsh** access to the Print Host for printing. You can add an **.rhosts** file to the Print Host's /**root** directory to give the node **root** users print access.

Installing Print Management on a Control Workstation

In order to have the Print Management System configured properly on a control workstation, you must run the **print_config** script to copy and link commands and set permissions on the control workstation. This script is located in /**usr**/**lpp**/**ssp**/**install**/**bin** and must be run by **root**. The **print_config** script depends on two environmental variables:

PRINT_CONFIG Must be set to one of `open` or `secure` depending on the mode chosen. To unconfigure, set to `false`.

PRINT_ID Used only if you are changing the default print management user ID from **prtid**. This variable applies to secure mode only.

Using a Non-AIX Workstation as Print Host

If you want to use a different Print Host, the commands used and method of setup may vary according to your type of host. Perform the following steps to ensure your Print Host is configured properly:

1. Verify that the print commands are the same on the Print Host as on the SP nodes. You will need such commands as **lpr** or **qprt** and **lpstat** or **qchk**. Notify your users which commands can be used. Unsupported commands will return an error.
2. Verify that the path to the print commands is /**usr**/**bin**. If this is not the location where print commands reside, set up links in your location to point to the print commands, such as

   ```
   /usr/ucb/lpr → /usr/bin/lpr
   ```

3. Create the print management user ID and set the password to a value that does not allow login. Remove **su** authority of other users to this ID.
4. Check the space requirements in the directories where the host's installed print subsystem puts its print job files according to the information in “Disk Space for Print Jobs” on page 248.
5. Set up the proper communication paths. For Secure Mode, **rsh** access must be allowed for the print management ID on the Print Host. For Open Mode, all users need **rsh** access to the Print Host. Access can be provided through the **/etc/hosts.equiv** and **rhosts** files. Review the steps for the RISC System/6000 to see if they apply.

Chapter 17. Managing Mail Service

There are several ways you can integrate the SP system into your current AIX mail environment. This section provides instructions for three recommended options. Each option offers a solution that:

1. Avoids the maintenance of mail queues on each SP processor node.
2. Allows the SP users to send mail in a customary manner and receive it at their home directory host.
3. Requires the use of Amd when you use a SP node or control workstation as the mail acceptor host. Refer to "Understanding the NFS and Amd Implementation" on page 239 for more information.

These options differ depending on whether you are using DNS, NIS, or **/etc/hosts** for your host name resolution method.

Terminology

The following terms are used in the mail discussion:

Central mail hub	A central host that either accepts mail for all users or routes users' mail to their home directory hosts.
Home directory host	The host where a user's home directory resides and where the user reads mail.
Mail acceptor host	The host that serves as a central mail hub for SP nodes only. It must have an account for each user being serviced.
Mail exchanger	A host that accepts mail addressed to users on other hosts through the use of **MX** records and the **Cw** option in the **sendmail.cf** file.

Note: All procedures in this section pertain only to the RISC System/6000 architecture. If your environment includes systems with other architectures, the procedures may require modification.

Related Information

Refer to Chapter 14, "Managing File Collections" on page 213 for more information about the SP file collection procedures.

You can find further information about NIS, **sendmail**, and the TCP/IP Domain Name in *IBM AIX Communication Concepts and Procedures*.

You can find further information about the **/etc/aliases** file and the Domain Name Service records in *IBM AIX Version 3.2 for RISC System/6000 Files Reference*.

Tasks covered in this chapter include:

- Option 1: Integrating mail using DNS
- Option 2: Integrating mail using NIS
- Option 3: Integrating mail using **/etc/hosts**
- Distributing mail-related files

© Copyright IBM Corp. 1995

Option 1: Integrating Mail Using DNS

If you are using DNS for your external hostname resolution, you can configure a SP node, the control workstation, or a host outside the SP system to act as a mail hub for the processor nodes. You accomplish this by adding Address (**A**) and Mail Exchanger (**MX**) records to your domain data file on your primary name server. These records direct mail addressed to the SP nodes to the mail acceptor host. Mail for the control workstation can be rerouted in the same manner as mail for other SP nodes. As a result, mail sent through the network will be relayed, using **.forward** files, to the users' home directories or a site central mail hub. This alleviates the need to run the **sendmail** daemon and maintain mail queues on each node, however, the mail acceptor host **must have access** to each user's **.forward** file in order to route mail to their home directories.

When a SP node or control workstation is used as mail acceptor host, Amd is required to mount the user's home directories and access the **.forward** files. When a host outside the SP system is used a mail acceptor host, a **.forward** file must exist in each user's home directory on that host in order for mail to be relayed to its final destination.

You can also use DNS to route the mail to a central mail hub, although the use of a mail hub can have drawbacks as explained in each step.

Figure 54 on page 255 shows how LAN and local mail are routed in a DNS configuration, assuming Amd is running.

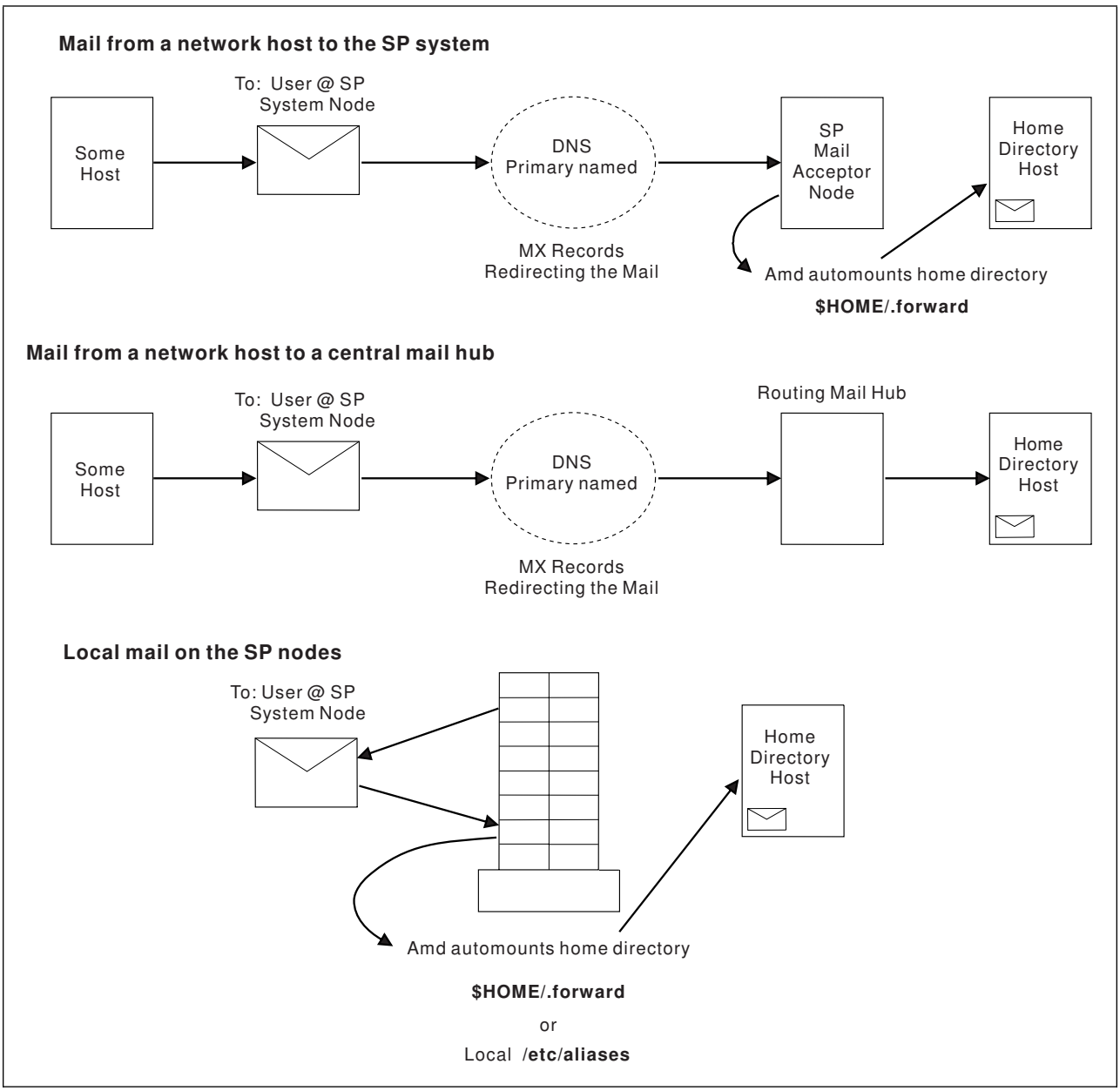


Figure 54. Option 1: Using DNS

Steps in this task:

Step 1. Modify the primary domain data file
Step 2. Modify **sendmail.cf**
Step 3. Add **.forward** files to home directories
Step 4. Turn off **sendmail** on the nodes

Step 1: Modify the Primary Domain Data File

Add Address (**A**) records and Mail Exchanger (**MX**) records to your primary domain data file. The Address records specify the address of the host serving as the mail acceptor. The Mail Exchanger records identify the control workstation and nodes whose mail is being forwarded, and the hosts to which the mail should be sent. For example, if SP node 1 serves as the mail acceptor host, then node 1 would receive all SP mail and forward it to the appropriate user home directories.

Note that you need to add **MX** records for the SP processor nodes and mail acceptor host to your **named.***domain***data** file.

Your primary domain data file resides on your primary DNS name server. The file is usually called **named.***domain***data**, where *domain* is the name of your DNS domain. For example, if your domain is called **rx7**, your domain data file would be **named.rx7data**.

Add the Records Manually

Follow this process to edit your primary domain data file and manually add the **A** and **MX** records:

1. Log in to your primary name server and edit the domain data file.
2. Add an **MX** record for each mail acceptor host and processor node, using this format:

   ```
   hostname    IN   MX    10  mail_exchanger
   ```

   where

   hostname The name of the host whose mail is being forwarded.
   mail_exchanger The name of the host to which the mail is redirected.

   All processor nodes use their mail acceptor host as their mail exchanger, and the mail acceptor node serves as its own mail exchanger. The following example shows node 1 serving as the mail acceptor host:

   ```
   spnode3.kgn.ibm.com.  IN     MX  10    spnode1.kgn.ibm.com.
   spnode2.kgn.ibm.com.  IN     MX  10    spnode1.kgn.ibm.com.
   spnode1.kgn.ibm.com.  IN     MX  10    spnode1.kgn.ibm.com.
   ```

   If you are using a central mail hub, use the mail hub host name as the mail exchanger for all nodes.
3. Add an **A** record for each mail acceptor host or mail hub (if one does not already exist) using the following format:

   ```
   hostname  IN   A     nnn.nnn.nnn.nnn
   ```

   For example:

   ```
   spnode1  IN   A    123.456.678.90
   ```

Step 2: Modify sendmail.cf

Change the /**etc**/**sendmail.cf** file in several locations, depending on your current mail configuration:

- **If you are forwarding all mail from the nodes to a mail acceptor host:**

  Add the **Cw** option at the proper location in the **sendmail.cf** file on the mail acceptor host and include the nodes it serves as aliases for that host. Follow

the format **Cw***MailacceptorHostname Node1Hostname Node2Hostname* as shown in this example where `spr1node1` serves four nodes.

```
Cwspr1node1 r1node2 r1node3 r1node4 r1node5
```

Maintenance Reminder

Whenever you expand the SP system to add new nodes, you must also add these nodes to the **sendmail.cf** file on their mail acceptor host. If file collection is used to manage the **sendmail.cf** file, note that the **sendmail.cf** file on the mail acceptor host is different from that on the other nodes in the SP system.

- **If you are forwarding all mail from the nodes to a central mail hub that routes mail:**

  Add the **Cw** option at the proper location in **sendmail.cf** on every host that will receive mail from the mail hub and include all the nodes in the SP system as aliases. If **sendmail.cf** does not include all of the processor nodes, sending a message can cause a loop. When someone sends a message to a user on the SP system, the message is intercepted and sent to the mail hub. The mail hub, in turn, sends the message to *user@homehost*. If the **sendmail** daemon on the home host does not accept the message, it forwards it to the SP system and starts the cycle again.

  Maintenance Reminder

  Whenever you expand the SP system and add new nodes, you must also add these nodes to the **sendmail.cf** file on every host that receives mail from the mail hub.

  If you have a large user base where people maintain their home directories on personal systems, this configuration may be difficult to administer and control. This might be a reason not to use a routing mail hub.

- **If you are forwarding all mail to a central mail hub that accepts mail for other hosts:**

  Add the **Cw** option at the proper location in **sendmail.cf** on the mail hub host and include the names of all the SP processor nodes and control workstation.

  Maintenance Reminder

  Whenever you expand the SP system and add new nodes, you must also add these nodes to the **sendmail.cf** file on this mail hub.

- **In all cases:**

  Enable the **OK MX** entry in **sendmail.cf** on every host in the network to allow the Mail Exchanger records to work. The default **sendmail.cf** comments this entry. Remove the comment as shown in this example:

  ```
  # The default is not to use a name server for resolving recipients.
  OK MX
  #
  ```

  Whenever you change **sendmail.cf**, you need to refresh the **sendmail** database on every host in your network, including the SP system. To do this, enter:

  ```
  /usr/lib/sendmail -bi
  /usr/lib/sendmail -bz
  ```

  Restart the **sendmail** daemon, if it is running.

Refer to “Distributing Mail-Related Files” on page 265 for suggestions on distributing the **sendmail.cf** file.

Step 3: Add .forward Files to Home Directories

Add a **.forward** file to each SP user's home directory to provide a destination for forwarding local mail on the processor nodes. When a node receives local mail, Amd mounts the user's home directory. The **.forward** file redirects the mail to the user at the host name specified in that file, which is the user's home directory host.

Add a **.forward** file to the user's home directory on the mail acceptor host if this host is outside the SP system.

Step 4: Turn Off sendmail on the non-Mail Acceptor Nodes

In this configuration, the processor nodes do not need to receive incoming mail. You can save CPU cycles by turning off the **sendmail** daemon on the nodes. Even though the **sendmail** daemon is not running, it will be invoked whenever users send mail from any processor node.

Perform the following actions once on each processor node that does not serve as a mail acceptor host.

- Edit the **/etc/rc.tcpip** file and add a comment to the line that starts the **sendmail** daemon:

```
#
#start /usr/lib/sendmail "$src_running" "-bd -q-qpi"
#
```

- Stop **sendmail** on each node using one of these methods:
  - If your site runs the System Resource Controller **srcmstr** daemon, issue:

```
stopsrc -s sendmail
```

  - If your site does not run the System Resource Controller **srcmstr** daemon, issue:

```
kill -1 `cat /etc/sendmail.pid`
```

Option 2: Integrating Mail Using NIS

If you are using NIS for external host name resolution, you can use the NIS global **mail.aliases** map to route the mail to the users' home directories. A global **/etc/aliases** file on the NIS master provides mail addresses for all users in the NIS domain. Local **/etc/aliases** files on the processor nodes contain only a minimum set of local aliases and a reference to the global file on the NIS master.

You have different steps to follow depending on whether you created a new NIS domain specifically for the SP system or already had an established NIS domain in your environment.

If you have:	Procedure:
A new NIS domain for the SP	Perform Steps 1 through 5 using the control workstation as the NIS master
An established NIS domain that does not use **mail.aliases**	Perform Step 1 to create a partial **/etc/aliases** file of the SP users Add other users to this file Proceed with Steps 2 through 5
An established NIS domain that uses **mail.aliases**	Verify that all the SP users are in the **/etc/aliases** on your NIS master Proceed with Steps 3 through 5

Steps in this task:

Step 1. Modify **/etc/aliases** on the NIS master
Step 2. Create a global NIS map
Step 3. Create and distribute local **/etc/aliases**
Step 4. Modify **sendmail.cf**
Step 5. Turn off **sendmail** on the nodes

Figure 55 on page 260 shows how LAN and local mail are routed in an NIS configuration, assuming Amd is running.

Step 1: Modify /etc/aliases on the NIS Master

The **/etc/aliases** file is part of the AIX Base Operating System. It is used to define system-wide aliases for user names or groups of users. When used with NIS, it propagates the NIS **aliases.map** from the NIS master to all clients, making aliases known throughout the NIS domain.

You must append alias records for all the SP users to the existing **/etc/aliases** file on your NIS master to create a global version of this file. You then use this global file to build an NIS map for mail aliases.

Add users to your **/etc/aliases** file by creating entries in the format *username: username@host*, where *host* specifies the node location of the user home directory being mounted to the SP nodes.

Here are sample entries in a **/etc/aliases** file:

```
fvtst1:             fvtst1@tserv5.hpssl.kgn.ibm.com
fvtst2:             fvtst2@tserv5.hpssl.kgn.ibm.com
fvtst3:             fvtst3@tserv5.hpssl.kgn.ibm.com
fvtst4:             fvtst4@tserv5.hpssl.kgn.ibm.com
saffycat:           saffycat@mroz.kgn.ibm.com
ken:                ken@tserv5.hpssl.kgn.ibm.com
wendy2:             wendy2@tserv11.hpssl.kgn.ibm.com
pepper:             pepper@mroz.kgn.ibm.com
```

If you are routing all users through a mail hub, replace the home directory hosts with the fully qualified host name of the mail hub.

Maintenance Reminder

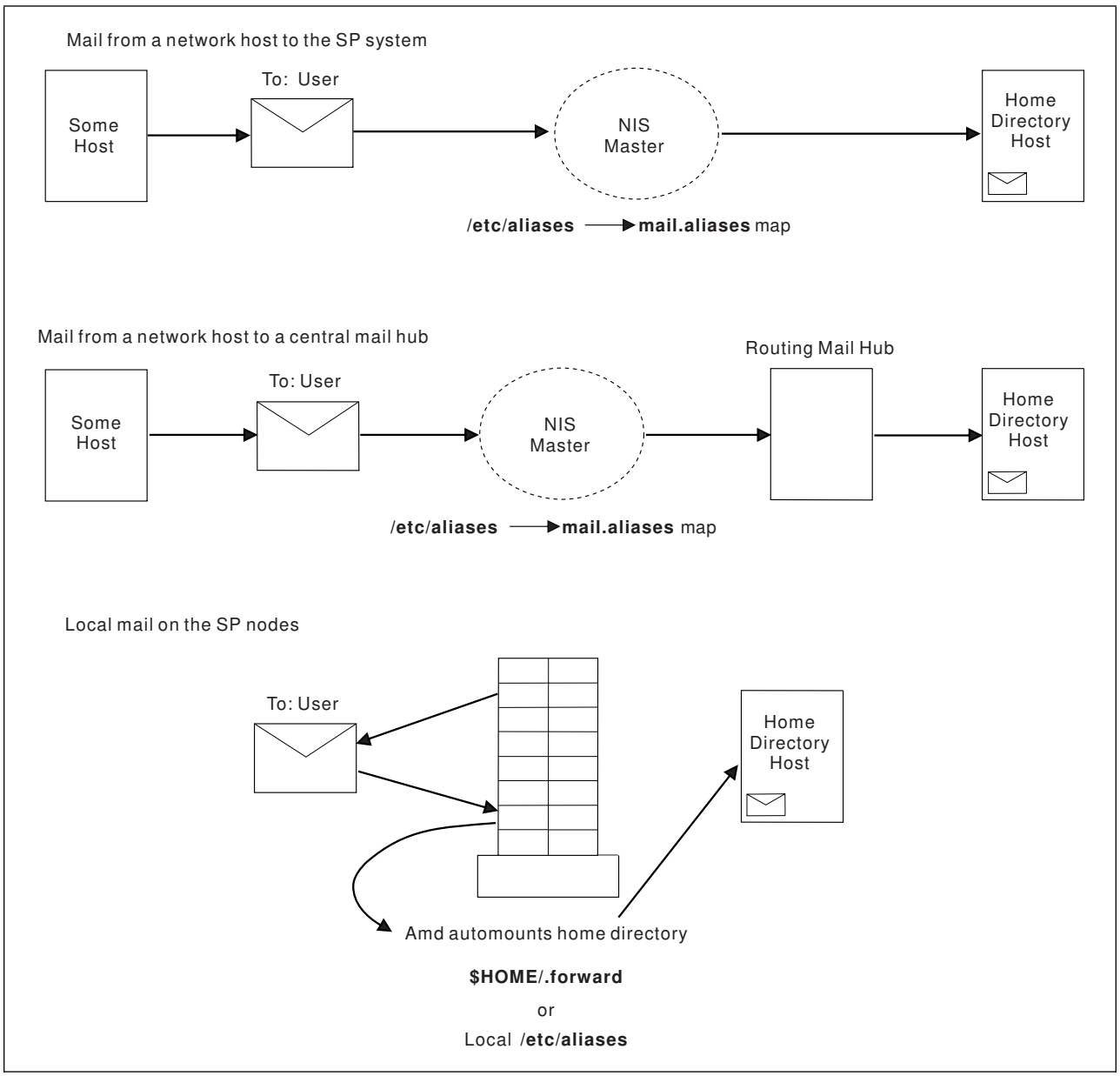


Figure 55. Option 2: Using NIS

Whenever you add a new user to your system, you must also add them to the master **/etc/aliases** file, distribute it, and refresh **sendmail** on every host.

In order for the mail services to resolve host addresses correctly, the **/etc/hosts** file must be kept up to date. Whenever you add or remove a host from your environment, you must also update this file and distribute it.

Step 2: Create a Global NIS Map

Create a global NIS map from the **/etc/aliases** file on the NIS master. When NIS is running, this map augments the **/etc/aliases** file by using the global **mail.aliases** file to provide the mail information for all the users in the NIS environment. If you did not have an established NIS domain before installing the SP system, the control workstation should be set up as your NIS master.

1. Log in to your NIS master server as **root**.

2. Enter:

```
cd /etc/yp
make PUSH=1 aliases
```

This creates the global map and propagates it to all the slaves and clients in the NIS domain.

Step 3: Modify and Distribute Local /etc/aliases

Modify the **/etc/aliases** file on every host in the NIS environment except the NIS master. Local **/etc/aliases** files must not have any entries for users specified in the global file or the local entries will override the global map. If necessary, you can maintain aliases in the local file for users that are not known globally. Each local **/etc/aliases** file requires an entry (NIS escape sequence) at the end that tells NIS to reference the global **mail.aliases** map. Follow this procedure:

1. Edit the **/etc/aliases** file on each node.
2. Remove all entries for users that are already specified in the global **mail.aliases** map.
3. Add the NIS escape sequence (+:) at the end of the file. This directs NIS to use the global **mail.aliases** map.

Step 4: Modify sendmail.cf

If you have a new NIS environment or an established NIS environment that does not use **mail.aliases**, make the following changes to the **sendmail.cf** on every AIX host in your NIS environment.

Or, if you have an established NIS environment and are using **mail.aliases**, make the following changes to the **sendmail.cf** file on the control workstation and processor nodes only.

1. Edit **/etc/sendmail.cf**.
2. Remove the comment on the option for using NIS mail aliases, as shown in this example:

```
# map name for NIS aliases
Opmail.aliases
#
```

3. Remove the comment on the option for future mail delivery. This allows the SP system to attempt to deliver mail at a later time if the NIS master is down. See the following example.

```
# If the I option is set, sendmail treats "connection refused" and "timed out"
# errors from the name server as non-fatal, leaving the message in the queue
# for future delivery attempts.
OI
#
```

Step 5: Turn Off sendmail on the Nodes

In this configuration, the processor nodes do not need to poll for incoming mail. You can save CPU cycles by turning off the **sendmail** daemon on the nodes. Refer to “Step 4: Turn Off sendmail on the non-Mail Acceptor Nodes” on page 258 for instructions on how to do this.

Option 3: Integrating Mail Using /etc/hosts

If you are using /**etc**/**hosts** for host name resolution, you can either use a local /**etc**/**aliases** file or **.forward** files to route mail to the users' home directories or a central mail hub.

Each RISC System/6000 host has a standard /**etc**/**aliases** file as part of the AIX Base Operating System. You can maintain this file as a master file of mail aliases, directing all network mail to users' home directory hosts or to a central mail hub, and distribute it to all the hosts in your network. You can use the file collection technology to maintain and distribute /**etc**/**aliases** throughout the SP system and use your standard method of file distribution for the other hosts in your network.

The other choice using the /**etc**/**hosts** file for host name resolution is to have a **.forward** file in every user's home directory pointing to the home directory host or a central mail hub. When mail is received, Amd automounts the home directory on the SP system and the **.forward** file routes the mail to the specified location. This method requires each processor node to run the **sendmail** daemon which could impact their efficiency and requires more administrative maintenance.

Figure 56 on page 263 shows how LAN and local mail are routed in an /**etc**/**hosts** configuration, assuming Amd is running.

If you want to use:	**Perform:**
local/**etc**/**aliases** files	Steps 1 through 3
.forward files	Steps 4 and 5

Steps in this task:

Step 1. Modify /**etc**/**aliases** on every host
Step 2. Update the **sendmail** database on every host
Step 3. Turn off **sendmail** on the nodes
Step 4. Add **.forward** files to home directories
Step 5. Run **sendmail** on each processor node

Step 1: Modify /etc/aliases on Every Host

Each RISC System/6000 host has a standard /**etc**/**aliases** file as part of the AIX Base Operating System. You can maintain this file as a master file of mail aliases, directing all network mail to users' home directory hosts or to a central mail hub, and distribute it to all the hosts in your network. You can use the file collection technology to maintain and distribute /**etc**/**aliases** throughout the SP system and use your standard method of file distribution for the other hosts in your network.

Add users to your /**etc**/**aliases** file by creating entries in the format *username: username@host*, where *host* specifies the node location of the user home directory being mounted to the SP nodes.

Here are sample entries in a /**etc**/**aliases** file:

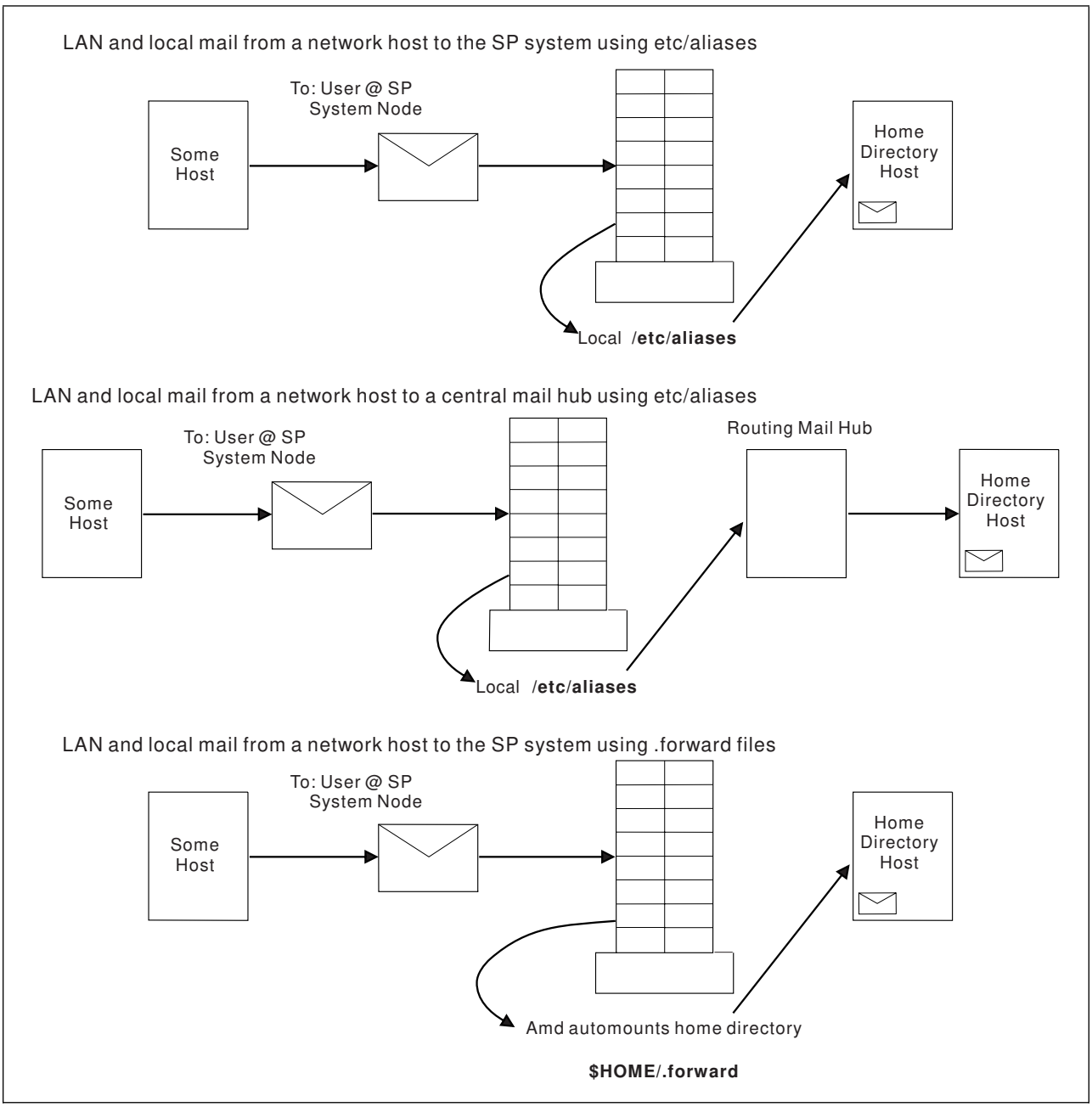


Figure 56. Option 3: Using /etc/hosts

```
fvtst1:          fvtst1@tserv5.hpssl.kgn.ibm.com
fvtst2:          fvtst2@tserv5.hpssl.kgn.ibm.com
fvtst3:          fvtst3@tserv5.hpssl.kgn.ibm.com
fvtst4:          fvtst4@tserv5.hpssl.kgn.ibm.com
saffycat:        saffycat@mroz.kgn.ibm.com
ken:             ken@tserv5.hpssl.kgn.ibm.com
wendy2:          wendy2@tserv11.hpssl.kgn.ibm.com
pepper:          pepper@mroz.kgn.ibm.com
```

Step 2: Update the sendmail Database on Every Host

AIX uses the **/etc/aliases** file to build the **sendmail** database. Whenever you change **/etc/aliases**, you need to refresh the **sendmail** database on every host in your network, including the SP system. To do this, enter:

```
/usr/lib/sendmail -bi
/usr/lib/sendmail -bz
```

Restart the **sendmail** daemon, if it is running.

Maintenance Reminder

Whenever you add a new user to your system, you must also add them to the master **/etc/aliases** file, distribute it, and refresh **sendmail** on every host.

In order for the mail services to resolve host addresses correctly, the **/etc/hosts** file must be kept up to date. Whenever you add or remove a host from your environment, you must also update this file and distribute it.

Step 3: Turn Off sendmail on the Nodes

In this configuration, the processor nodes do not need to poll for incoming mail. You can save CPU cycles by turning off the **sendmail** daemon on the nodes. Refer to “Step 4: Turn Off sendmail on the non-Mail Acceptor Nodes” on page 258 for instruction to do this.

If the SP users do not have home directories on the boot/install servers, these servers may not have incoming mail. In this case, you may also want to turn off the **sendmail** daemon on each of these servers.

Step 4: Add .forward Files to Home Directories

Request that each user adds a **.forward** file to their home directory pointing either to their home directory host or to a central mail hub. The file needs just one entry, as shown in this example:

```
marylou@hostofchoice
```

Step 5: Run sendmail on Each Processor Node

When you are using the **.forward** file configuration, the processor nodes need to poll for incoming mail and must run the **sendmail** daemon. The command to start **sendmail** is in the **rc.tcpip** file.

Most environments usually run the **sendmail** daemon by default. If this is not the case in your environment, follow these instructions to enable this daemon:

1. Edit **/etc/rc.tcpip** on the processor nodes.
2. Remove the comment from the following line:

   ```
   #
   start /usr/lib/sendmail "$src_running" "-bd -q{qpi}"
   #
   ```

3. If the **sendmail** daemon is running, refresh it by entering:

   ```
   /usr/lib/sendmail -bi
   /usr/lib/sendmail -bz
   ```

4. Reboot the nodes.

Distributing Mail-Related Files

To ease the administrative burden of maintaining mail services across the system, you can place mail-related files in a file collection for distribution to the processor nodes. The nodes must all have the same configuration for this to work. Be sure to review Chapter 14 and refer to the directions in “Adding and Deleting Files in a File Collection” on page 226.

sendmail.cf

To use DNS (option 1) to integrate the SP into your mail environment, you must modify the **/etc/sendmail.cf** file. Mail acceptor hosts have different **sendmail.cf** files from other nodes. Ensure that your file collection technology manages both kinds of **sendmail.cf** files.

To manage the **/etc/sendmail.cf** file using file collection technology:

1. Add **sendmail.cf** to the **node.root** file collection.

   ```
   cp /etc/sendmail.cf /share/power/system/3.2.3/etc/sendmail.cf
   ```

2. Add a command to refresh the **sendmail** subsystem to the **list** file for that collection.

   - If your site runs the System Resource Controller **srcmstr** daemon, add:

     ```
     execute /bin/refresh -s sendmail (./etc/sendmail.cf)
     ```

   - If your site does not run the System Resource Controller **srcmstr** daemon, add:

     ```
     execute /usr/lib/sendmail -bz (./etc/sendmail.cf)
     ```

/etc/aliases

To use **/etc/host** (option 3) to integrate the SP into your mail environment, you must modify the **/etc/aliases** file on each node. You can also add this file to the **node.root** file collection:

1. Add **aliases** to the **node.root** file collection.

   ```
   cp /etc/aliases /share/power/system/3.2.3/etc/aliases
   ```

2. Add commands to refresh the **sendmail** subsystem to the **list** file for that collection.

   - If your site runs the System Resource Controller **srcmstr** daemon, add:

     ```
     execute /usr/lib/sendmail -bi (./etc/aliases)
     execute /bin/refresh -s sendmail (./etc/aliases)
     ```

   - If your site does not run the System Resource Controller **srcmstr** daemon, add:

     ```
     execute /usr/lib/sendmail -bi (./etc/aliases)
     execute /usr/lib/sendmail -bz (./etc/aliases)
     ```

3. Restart the **sendmail** daemon if it is running on the nodes.

Chapter 18. Accounting

Your hardware resources are likely to be distributed unevenly across your SP System, with some nodes equipped for greater processing capacity and I/O than others. For this reason you may want to charge different usage fees for different nodes. You may also want to consolidate all accounting records, for the entire SP system, or for groups of nodes.

One way to make efficient use of the nodes in your SP system is to logically partition them so that each pool supports a different type of workload and is considered a separate management domain. The pooling of nodes according to configuration or workload type is the concept behind the *accounting class*. SP accounting allows you to logically groups SP nodes with similar operating characteristics into an accounting class for which the same charge fee applies.

You may also want to charge a different usage fee for compute-intensive jobs when they demand exclusive use of one or more nodes. SP System accounting provides a mechanism for treating these jobs separately as well.

Accounting by Node or by Class

An accounting class is a group of nodes for which accounting data is merged together, providing a single report for the nodes in that class. Individual nodes within a class may be enabled or disabled for accounting.

The **acct_class_id**, an attribute of each node object in the System Data Repository, is an arbitrary string you define and specify for all nodes for which you wish to group and merge accounting data.

Default values eliminate the need to define each attribute for each node. The default value of **acct_class_id** for each node is **default** Therefore, all nodes initially belong to one accounting class called **default**.

The initial value of the **acct_enable** attribute for each node is also **default**. All nodes with **acct_enable=default** are enabled or disabled for accounting based on a system-wide accounting enabled attribute (**spacct_enable**) you define for your site environment. When you set **spacct_enable** to **true**, it enables accounting on all nodes for which the individual accounting enabled attribute (**acct_enable**) is set to **default**. Likewise, when you set **spacct_enable** to **false** (the initial value at installation), accounting is disabled on all nodes that have **acct_enable=default**. Thus, accounting can be enabled or disabled for the entire SP system with individual nodes treated as exceptions.

How SP Accounting Differs from Standard AIX

Standard AIX accounting uses a script called **runacct**. This script is normally run every night by **cron** to provide daily usage reports. It also keeps a running total of usage over some fiscal period (usually a month). SP accounting splits the **runacct** function into two parts. The first part is a script that is run on each SP node every night, just as **runacct** does. This script is named **nrunacct**. The second part is a script that executes on an **accounting_master** node each night, after the **nrunacct** scripts are run. This script is named **crunacct** and it consolidates the **nrunacct** information from each node. The scripts **cmonacct** and **cprdaily** are SP versions

© Copyright IBM Corp. 1995

of standard AIX programs, modified to handle accounting class and exclusive use accounting.

nrunacct

The **nrunacct** script provides the function that merges raw data from the process, login, print, fee and disk subsystems. It also provides new support to account for the exclusive use of a node by a Resource Manager request.

nrunacct also creates an SP version of the **loginlog** file using the **var/adm/acct/sum/loginlog** files for the specified accounting cycle. Only the most recent **loginlog***yyyymmdd* file from each node is pertinent.

nrunacct provides a userexit for any site-specific accounting procedures in the **/var/adm/nsiteacct** file.

crunacct

The **crunacct** script provides the function that produces daily summary reports and accumulates data for the fiscal period, using the merged data from each node.

SP accounting merges total accounting data by user ID and login name, just as the standard AIX accounting does. This means that all SP nodes must have identical user IDs and login names. In other words, all SP nodes must be members of the same user name space, whether it is implemented through NIS, the SP User Management tools, or through other means.

cmonacct

An SP version of **monacct**, **cmonacct** creates summary files for monthly or fiscal reports.

cprdaily

An SP version of **prdaily**, **cprdaily** creates daily report files of accounting activity and cleans up files no longer needed.

Accounting for Exclusive Use of Node Jobs

While SP nodes are expected to execute exclusive parallel jobs, nonexclusive parallel jobs, and nonparallel jobs during a single accounting cycle, normal operation often results in the exclusive use of a node when executing a compute-intensive parallel job. A single job that makes such great demands on the compute resources of a node where no other jobs can share that node's resources is said to be an *exclusive use of a node* job. Such jobs can be handled separately for accounting purposes.

Standard AIX accounting produces a record for each process as the process completes. These records are later merged by user ID and login name. A user who executes one or more types of jobs on the same node has all process accounting records for that node merged together, resulting in the loss of those processes that had exclusive use of the node. It is important to track the exclusive use of a node by a user because the cost is likely to be different than the cost of shared use.

SP accounting addresses exclusive-use jobs as follows: Before starting a job that has exclusive use of a node, the Resource Manager executes a start job executable program under the real user ID of the job. When the job is finished the

Resource Manager executes an end job command, again under the real user ID of the job. These commands consist solely of an **exit()** system call in their **main()** procedure. Their purpose is to generate process accounting records for the user ID of the job that can then be used to *bracket* all of the normal process accounting records of the job. In other words, all of the process accounting records generated by an exclusive job can be identified and excluded from the other process accounting records generated on the node.

The process accounting records for the start job and end job programs are used to calculate the time the job had exclusive use of the node. The time is then charged to the user ID of the job by converting it to chargefee units. Use of the chargefee mechanism eliminates the need to generate additional accounting reports.

Accounting records for the start job and end job programs are identified by the names of the commands as found in the records. These start and stop programs are given the nonstandard names of **SJSJSJSJ** and **EJEJEJEJ** to avoid conflict with any other standard job names.

An attribute of the SP object in the SDR, **spacct_excluse_enable**, determines whether the start job program and end job program are executed for exclusive-use jobs. A node-level SDR attribute, **acct_excluse_enable**, determines whether accounting is activated to process start and end job records, and to generate chargefee records for jobs making exclusive use of that node.

Setting Up SP Accounting

IBM General Concepts and Procedures for RISC System/6000, GC23-2202, describes a procedure for setting up an accounting system that involves performing the following steps for each AIX system:

- Creating directories
- Creating and updating files
- Enabling accounting
- Adding accounting procedures to **crontabs**

In a parallel environment, where there may be hundreds of nodes, it is essential that these tasks be automated.

The following procedure helps you to tailor your accounting system by defining accounting class identifiers and accounting enabled attributes on a SP level or on an individual node basis, if necessary. Accounting class identifiers need only be specified when you want more than one accounting class. You can also specify which nodes are to have accounting enabled, however, if all nodes are to be enabled, then only SP Accounting Enabled need be set to `true` and the **acct_enable** attribute can be left to default on each Node object in the SDR.

In addition, this procedure shows you how to specify exclusive use accounting, and how to specify an Accounting Master other than the default control workstation. It also shows how to set an SP Accounting Active Node Threshold value other than the default value of 80.

You can define or change these values after installation by using SMIT or the **spacctnd** and **spsitenv** commands.

Accounting and System Partitioning

If you do not partition your SP system or if you run with a single system partition, you can assign any node or the control workstation as the accounting master.

If you configure multiple system partitions on your SP system the following restrictions apply:

- If all of your system partitions are IBM Parallel System Support Programs for AIX (PSSP) Version 2 Release 1 (running on AIX 4.1) system partitions, you can assign any node in any system partition or the control workstation as the accounting master.
- If you have configured a single PSSP Version 1 Release 2 (running on AIX 3.2.5) system partition, the accounting master can be any node in that PSSP V1R2 system partition or the control workstation. Under this configuration you cannot select a node in a PSSP V2R1 system partition to be the accounting master.
- If you have configured two or more PSSP V1R2 (running on AIX 3.2.5) system partitions, the accounting master must be the control workstation.

Step 1: Enter System Accounting Information

Use SMIT or the **spacctnd** and **spsitenv** commands to define how accounting is to be done on your SP System. This information is read by the installation scripts, which enter specific details about your site's configuration into the System Data Repository (SDR).

Note

If you are setting up accounting and modifying Node objects in the SDR on a SP System that is already running the Resource Manager, you must stop and restart the Resource Manager.

If you are only changing the SP object in the SDR on a SP System that is already running Resource Manager (to enable exclusive use accounting, for example), then you must reconfigure the Resource Manager using the **jm_config** command.

Refer to “Reconfiguring the Resource Manager” on page 189.

Using SMIT:

TYPE **smit enter_data**

▲ The Enter Database Information Window appears.

SELECT Site Environment Information

▲ The following window appears:

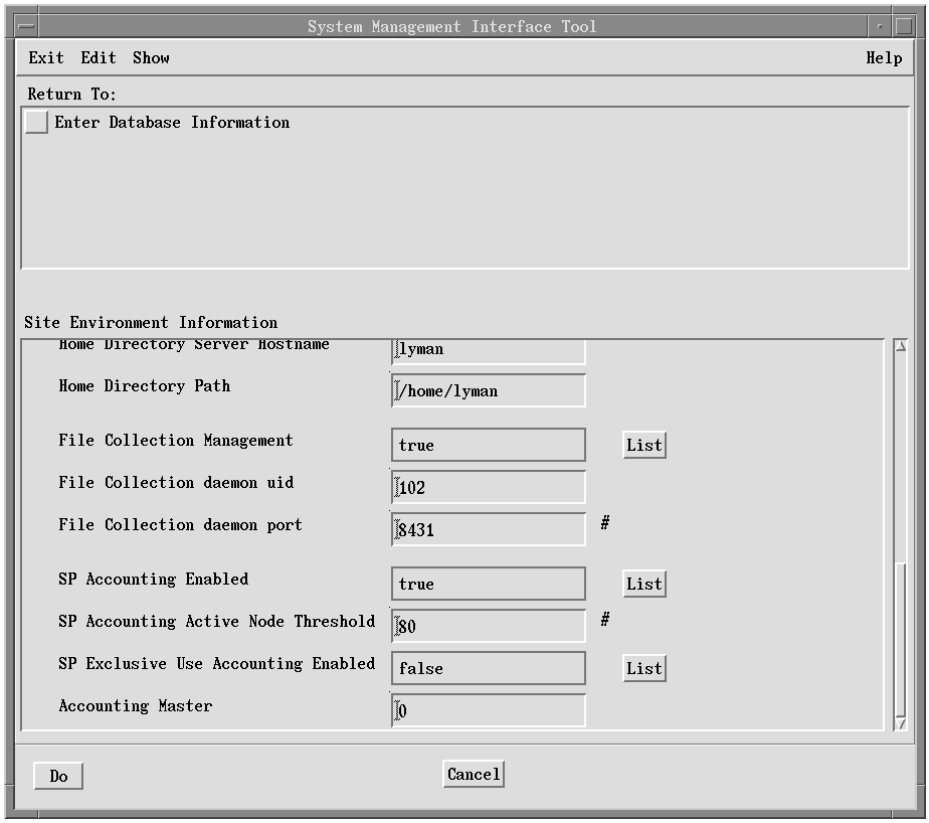


Figure 57. Accounting Fields on the Site Environment SMIT Panel

TYPE The data in the fields, as needed.

Using spsitenv::

The following example enables accounting as the default for all nodes and specifies 2 as the accounting master. It also enables exclusive use accounting.

```
spsitenv acct_master=2 spacct_enable=true spacct_excluse_enable=true
```

Accounting information on this panel includes:

SP Accounting Enabled
: Use this field to specify whether by default, accounting is enabled or disabled on all nodes that have an accounting enabled attribute of **default**. A value of **true** indicates that accounting is enabled on all nodes with an accounting enabled attribute of **default**. A value of **false** indicates that accounting is disabled on all nodes that have an accounting enabled attribute of **default**. The default value for this attribute is **false**.

  This sets the **spacct_enable** attribute in the SP object of the SDR. You can specify this attribute by using the **spsitenv** command instead of SMIT.

SP Accounting Active Node Threshold
: Use this field to specify the minimum percentage of nodes for which accounting data must be present. At the time when the new **crunacct** script runs, data must be available from at least this percentage of

nodes in order for processing to continue. The default value for this attribute is **80**.

This sets the **spacct_actnode_thresh** attribute in the SP object of the SDR. You can specify this attribute when using the **spsitenv** command instead of SMIT.

SP Exclusive Use Accounting Enabled

Use this field to specify whether to activate start and end job records, and thus chargefee records for jobs requesting the exclusive use of a node. The default value for this attribute is **false**.

This field sets the **spacct_excluse_enable** attribute in the SP object of the SDR. You can specify this attribute when using the **spsitenv** command instead of SMIT.

A value of **true** specifies that exclusive use accounting is enabled on all nodes where the **acct_excluse_enable** attribute is set to `true`. You must check each node's **acct_excluse_enable** attribute for account processing. A value of **false** (the default) specifies that exclusive use accounting is disabled on the node.

Accounting Master

Use this field to specify which node is to act as the accounting master. This is the node that executes **crunacct**. The default value for this attribute is **0** (the control workstation) but you can specify any SP node.

This sets the **acct_master** attribute in the SP object of the SDR. You can specify this attribute when using the **spsitenv** command instead of SMIT.

Step 2: Enter Node Accounting Information

Use SMIT or the **spacctnd** command to further define SP accounting at the node level.

Using SMIT:

TYPE **smit node_data**

▲ The Node Database Information Window appears.

SELECT Accounting Information

▲ The accounting information window appears:

Figure 58. SP Accounting Information SMIT Panel

TYPE The data in the fields, as needed.

Accounting information on this panel includes:

Start Frame
: Enter the frame number that corresponds to the nodes required for this operation. Specify a value between 1 and 64 inclusive.

Start Slot
: Enter the slot number of first node to be used for this operation. Specify a value between 1 and 16 inclusive. Remember that a wide node counts as two slots.

Number of Nodes

Number of nodes to be used for this operation. The node information is added for successive nodes within a frame. If the count of nodes causes the nodes in a frame to be exhausted the operation continues in the next sequential frame. Specify a value between 1 and 1024 inclusive.

Accounting Class Identifier

Use this field to change the accounting class. All nodes with the same string value constitute a class for purposes of grouping and merging accounting data. All nodes will have an initial accounting class identifier of **default**.

This sets the **acct_class_id** attribute in the Node object of the SDR. You can specify this attribute when using the **spacctnd** command instead of SMIT.

You only need to specify a value for this field when you want to change the accounting class.

Accounting Enabled

Use this field to indicate whether accounting is enabled or disabled for a node. A value of **true** indicates that accounting is enabled for the node. A value of **false** indicates that accounting is disabled for the node. All nodes have an initial value of **default** assigned to this attribute. A value of **default** indicates that the node is enabled for accounting based on your entry in the SP Accounting Enabled field in the SMIT Site Environment panel in Step 1.

This sets the **acct_enable** attribute in the Node object of the SDR. You can specify this attribute when using the **spacctnd** command instead of SMIT.

Exclusive Use Accounting Enabled

Use this type field to indicate whether accounting start and end job records, and thus chargefee records, are processed for jobs having exclusive use of the node. A value of **true** indicates that exclusive use accounting is enabled and start and end job records will be generated. A value of **false** indicates that exclusive use accounting is not enabled and start and end job records will not be generated. All nodes have an initial value of **false** assigned to this attribute.

This sets the **acct_excluse_enable** attribute in the Node object of the SDR. You can specify this attribute when using the **spacctnd** command instead of SMIT.

Accounting Job Charge Value

This value is used to determine the number of charge fee units to charge a user for exclusive use of the node and its value is in units of *seconds per charge fee unit*. All nodes have an initial value of **1.0** assigned to this attribute.

This sets the **acct_job_charge** attribute in the Node object of the SDR. You can specify this attribute when using the **spacctnd** command instead of SMIT.

Using spacctnd::

The following example adds accounting information for a system with one frame and 16 nodes. Accounting and exclusive use accounting is to be enabled for each node and 60 seconds of exclusive use by a user is to constitute one charge fee unit.

```
spacctnd -e true -j 60.0 -x true 1 1 16
```

Additional Accounting Setup Tasks

Depending on your environment, you may also wish to perform these tasks:

- Defining the file systems for which disk accounting is to be done.
- Defining the print queues for which printer accounting is to be done.
- Defining how the all print jobs owned by the print manager id (**prtid**) in Secure Mode are to be handled for accounting purposes.
- Disk accounting may be enabled for each file system as it is created. Alternatively, if many file systems already exist, then disk accounting may be turned on or off by executing the **chfs -t'yes'** <**filesystem**> command on the nodes.

Accounting Installation

The accounting installation phase is performed automatically when you select the Accounting Support option in **ssp.sysman** during the **installp** process. (Refer to the *IBM RISC System/6000 Scalable POWERparallel Systems: Installation Guide*.) This accounting installation process is also performed when you update accounting using **spsitenv**.

The accounting installation process:

1. Creates all necessary directories and files on the nodes for which accounting is enabled and on the **acct_master** node.
2. Adds the **startup** command to the /**etc**/**rc** file on the nodes for which accounting is enabled.
3. Creates the /**var**/**adm**/**acct**/**nite**/**jobcharge** file on each node for which accounting is enabled. This file contains the job charge value previously defined for the node.
4. The **holidays** file (/**etc**/**acct**/**holidays**) is placed in the **user.admin** file collection and is available on all nodes. Its source is the control workstation. If you do not use file collections, propagate this file, using whatever means is available in your environment, to all nodes in the SP System.

   Note: You may have to update the **holidays** file to reflect the correct information for the current year.
5. Update the **crontabs** file to schedule the accounting processes. See "Accounting Template for crontabs" on page 276.
6. Initialize the login and process account data files. Enter:

   ```
   /usr/sbin/acct/nulladm /var/adm/wtmp /var/adm/pacct
   ```

Accounting Template for crontabs

A default **crontab** template file, describing when each accounting command should be run, is shipped as **/usr/lpp/ssp/config/cron_template** and distributed via a file collection. If you wish to run your accounting system on a different schedule, make the appropriate changes to this template. When booting, each node for which accounting is enabled, ensures that its **crontab** file is updated using this template. If the template does not exist, it is assumed that you use file collections to distribute **crontab** and nothing is done.

The **crontab** template file, shown below, contains a partial **crontab** entry. Each line begins with a keyword specifying a command used by the accounting process.

```
# This is the crontab template file. It is read by acct_config
# to update the root crontab file with accounting-related entries.

DODISK   0 1 * * 4
CHKPACCT 5 * * * *
NRUNACCT 0 2 * * 1-6
CRUNACCT 0 4 * * 1-6
CMONACCT 15 5 1 * *
```

Figure 59. Accounting Template for crontabs

DODISK and **CHKPACCT** are standard **crontab** keywords. **NRUNACCT**, **CRUNACCT**, and **CMONACCT** are variations of standard AIX accounting commands discussed earlier in this section. Refer to Chapter 19, “Maintaining the crontabs File” on page 281 for more information on **crontabs**.

Accounting Files

All accounting data, report, and summary files reside in the **/var/adm** directory. The accounting data files belong to members of the *adm* group, and all active data files (such as **wtmp** and **pacct**) reside in the adm home directory, **/var/adm**.

Note

It is very important that the System Administrator monitor the capacity of the **/var** filesystem when exclusive use accounting is enabled. It has been found that the Resource Manager may create large **/var/adm/pacct** files, and that accounting will save the **excl_Spacct** files in the **/var/adm/acct/nite** directory. This may require the administator to periodically increase the **/var** filesystem size or remove unnecessary **excl_Spacct** files. The System Administrator may edit the **/usr/lpp/ssp/bin/nrunacct** script in the CLEANUP section to automatically remove **excl_SPacct** files.

Data Files

Data files are:

/var/adm/acct/nite/dacct
: Contains disk total accounting records, created by the **dodisk** command.

/var/adm/fee
: Contains output from the **chargefee** command, in ASCII **tacct** records.

/var/adm/pacct
Active process accounting file.

/var/adm/wtmp
Active login accounting file.

/var/adm/Spacct*i.yyyymmdd*
Contains process accounting files for *yyyymmdd* during the execution of the **nrunacct** command.

Report and Summary Files

Report and summary files reside in the **/var/adm/acct** subdirectory. You must create the following subdirectories before the accounting system is enabled.

/var/adm/acct/nite
Contains files that **nrunacct** and **crunacct** command use in daily accounting.

/var/adm/acct/sum
Contains the cumulative summary files that **nrunacct** and **crunacct** update daily.

/var/adm/cacct
Contains files that the **crunacct** command uses in daily accounting.

nrunacct Command Files

The following report and summary files, produced by the **nrunacct** command, are of particular interest.

/var/adm/nsiteacct
Contains site-specific accounting procedures for node accounting.

/var/adm/acct/nite/lineuse*yyyymmdd*
Contains usage statistics for each terminal line on the system. This report is especially useful for detecting bad lines. If the ratio between the number of logouts and logins exceeds 3 to 1, it is possible that a line is failing.

/var/adm/acct/nite/daytacct*yyyymmdd*
Total accounting file for the previous day.

/var/adm/acct/sum/daycms*yyyymmdd*
Contains the daily command summary.

/var/adm/acct/sum/loginlog*yyyymmdd*
Contains a record of the last time each user ID was used.

/var/adm/acct/nite/reboots*yyyymmdd*
Contains beginning and ending dates from the **wtmp** file and a listing of system restarts.

crunacct Command Files

/var/adm/csiteacct
Contains site-specific accounting procedures for cluster accounting.

These files are copied to the **acct_master** node during **crunacct** execution.

/var/adm/cacct/*node***/nite/lineuse***yyyymmdd*
/var/adm/cacct/*node***/nite/daytacct***yyyymmdd*

/var/adm/cacct/*node***/nite/reboots***yyyymmdd*
/var/adm/cacct/*node***/sum/daycms***yyyymmdd*
/var/adm/cacct/*node***/sum/loginlog***yyyymmdd*

The *node* level files are merged to become:

/var/adm/cacct/sum/*class***/tacct***yyyymmdd*.
/var/adm/cacct/sum/*class***/daycms***yyyymmdd*.
which in turn update cumulative summary files:
/var/adm/cacct/sum/*class***/tacct** file..
/var/adm/cacct/sum/*class***/cms** file..

Files in the /var/adm/acct/nite Directory

active
Used by the **nrunacct** command to record progress and to print warning and error messages. The file **active.***yyyymmdd* is a copy of the active file made by **nrunacct** after it detects an error.

jobcharge
Contains a floating point number used to determine the number of charge fee units to charge a user for exclusive use of the node and its value is in units of *seconds per charge fee unit*.

jobrecs
Contains *start job* and *end job* records used for accounting on jobs that make exclusive use of nodes.

ptacct.*yyyymmdd*
Contains summary version of **pacct** files.

ctmp
Contains connect session records during **nrunacct**

accterr
Contains standard error messages produced during the execution of the **nrunacct** command.

lastdate
Indicates the last day the **nrunacct** command executed, in **yyyymmdd** format.

lock
Used to control serial use of the **nrunacct** command.

statefile*yyyymmdd*
Used to record the current state during execution of the **nrunacct** command.

wtmperror
Contains **wtmpfix** error messages.

owtmp.
Contains the previous day's **wtmp** file.

Files in the /var/adm/cacct/ Directory

active
Used by the **crunacct** command to record progress and to print warning and error messages. The file **active.***yyyymmdd* is a copy of the active file made by **crunacct** after it detects an error.

lastdate
Indicates the last day the **crunacct** command executed, in **yyyymmdd** format.

lock
Used to control serial use of the **crunacct** command.

statefile*yyyymmdd*
Used to record the current state during execution of the **crunacct** command.

lastcycle
Indicates the date of the last successfully completed **crunacct** cycle.

fiscal_periods
Defines the start date of each fiscal period.

Files in the /var/adm/cacct/sum Directory

<**class**>/**cms**
Daily command summary.

loginlog
File listing current login records.

rprt*yyyymmdd*
Saved output of the **cprdaily** routine executed in **crunacct**.

<**class**>/**tacct**
Process account summaries.

Files in the /var/adm/cacct/fiscal Directory

/**var**/**adm**/**cacct**/**fiscal**/*class*/**cms***mm*
Total class summary file for the month, in binary format.

/**var**/**adm**/**cacct**/**fiscal**/*class*/**tacct***mm*
Total accounting file for month, in binary format.

/**var**/**adm**/**cacct**/**fiscal**/**fiscrpt***mm*
Report similar to that of the **prdaily** command for month, in binary format.

Chapter 19. Maintaining the crontabs File

The SP system uses the **crontabs** file to periodically update file collections and clean up log files. The installation process appends to the **crontabs** files in **/var/spool/cron/crontabs/root** on the control workstation and each of the nodes. The files contain different entries depending on their location.

- Control workstation
  - Merge accounting information
  - Clean up the logs
- Processor nodes
  - Update the **sup.admin, user.admin** and **node.root** file collections
  - Merge accounting information
  - Clean up the logs

The following example shows these entries on a control workstation:

```
 0 * * * * /usr/lpp/ssp/bin/dev/null 2>/dev/null
 0 0 * * * /usr/lpp/ssp/bin/cleanup.logs.ws
0 4 * * 1-6/usr/lpp/ssp/bin/crunacct 2>/var/adm/cacct/nite/accterr
15 5 1 * * /usr/lpp/ssp/bin/cmonacct
```

The default is to run these processes hourly, in a staggered sequence. You can customize this for your environment to make any of these processes run more or less frequently. You can also add your own entries to these files for other processes you want automated.

Updating crontabs on the SP Nodes

If you want the same **crontab** file on all SP nodes, one way to update it across the whole SP System is:

1. Get the current **crontab** from one of the nodes:

   ```
   rsh hostname crontab -l > /tmp/mycrontab
   ```

2. Edit the file:

   ```
   vi /tmp/mycrontab
   ```

3. Propagate the changed **crontabs** file to all SP nodes:

   ```
   dsh -a rcp root@mynode:/tmp/mycrontab /tmp/mycrontab
   dsh -a "crontab < /tmp/mycrontab"
   ```

If you change the default configuration to modify the hierarchy of file collection servers, you need to modify the **crontabs** files to reflect the new relationship. See “Modifying the File Collection Hierarchy” on page 233 for more information. If you create your own file collections, you will also need to add them to the **supper update** commands in the **crontabs** file to keep them on schedule with the other upgrades.

© Copyright IBM Corp. 1995

Chapter 20. Managing Error Logs

The IBM RISC System/6000 Scalable POWERparallel Systems (SP) use both the BSD syslog and AIX Error Logging facilities, as well as a number of function-specific log files to record error events on each node. Refer to the *IBM RISC System/6000 Scalable POWERparallel Systems: Diagnosis and Messages Guide* for an overview of error logging and a description of SP specific log files.

This chapter describes error log management functions provided in the **ssp.sysman** install option of the IBM Parallel System Support Programs for AIX (PSSP). (Note that error log management function is provided as part of PSSP V2R1 and cannot be run on AIX 3.2.5 nodes.) Error log management consists of:

- SMIT panel interfaces
- A set of external commands
  - **splm** for general log viewing, archiving, and service collection
  - **psyslrpt** for generating reports of BSD syslog log files
  - **psyslclr** for trimming BSD syslog log files
  - **penotify** for creating, removing, or displaying Error Notification Objects
- Sysctl-based server functions

Methods are provided for performing the following log management tasks in parallel from a single point of control:

- Managing the AIX Error Log facility
- Managing the BSD syslog facility
- Archiving error logs
- Viewing error logs
- Collecting system data for IBM Service

To access the primary SMIT menu for error log management:

TYPE	**smit**
	▲ The System Management menu appears.
SELECT	9076 SP System Management
	▲ The 9076 SP System Management menu appears.
SELECT	9076 SP Log Management
	▲ The 9076 SP Log Management menu appears.

© Copyright IBM Corp. 1995

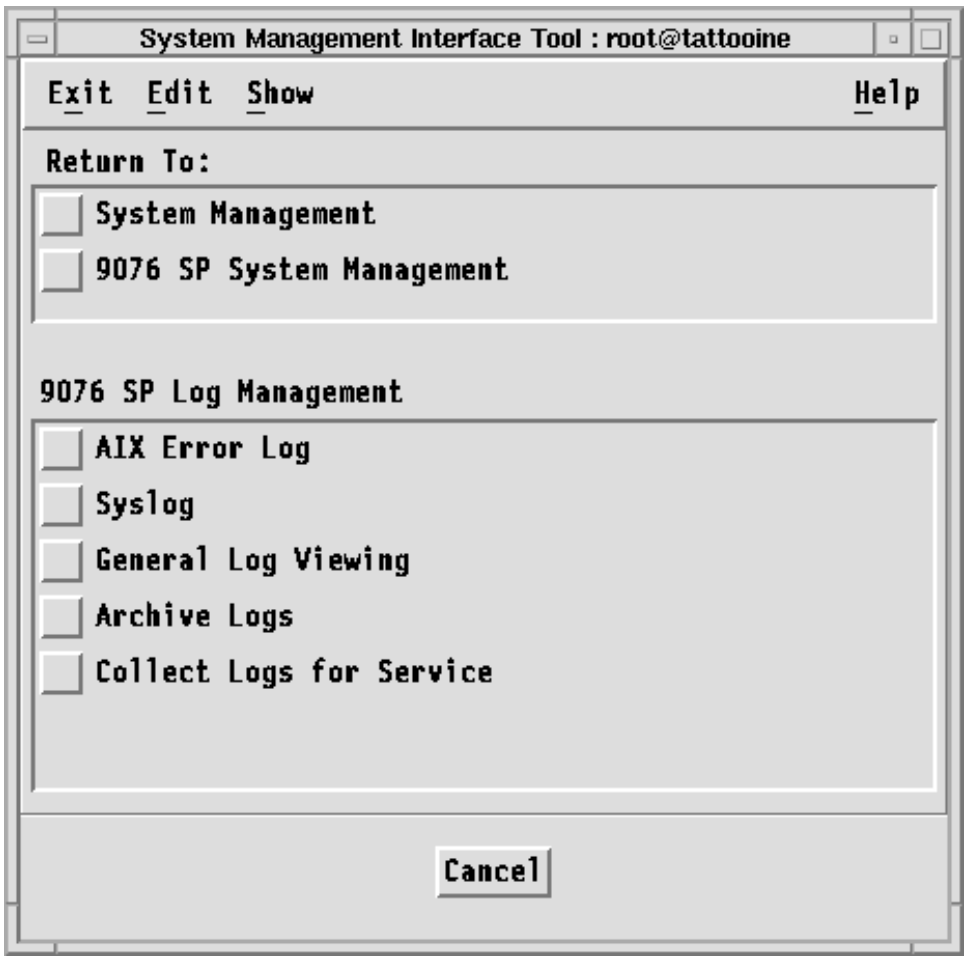


Figure 60. Log Management SMIT Panel

The fastpath invocation for the Log Management menu is:

```
smit splogmgt
```

Installing and Configuring Error Log Management

Log management functions are built upon the Sysctl facility, which uses the SP authentication services. Generating parallel AIX Error Log and BSD syslog reports and performing general log viewing require that the user issue the **kinit** command to be identified to the SP authentication services. All other log management commands additionally require that the user be defined as a principal in the **/etc/logmgt.acl** file. All users defined in the file **/etc/logmgt.acl** must also be placed in the authentication (PSSP or AFS) database as a principal. (Refer to "Adding Principals and Assigning Intitial Passwords" on page 40 for more information.) Note that the majority of log management represents administrative tasks normally requiring root authority and that a user defined in the **logmgt.acl** file will execute commands as the root user.

The following is an example of an **/etc/logmgt.acl** file:

```
#acl#
# This sample acl file for log management commands contains
# a commented line for a principal
#_PRINCIPAL root.admin@HPSSL.KGN.IBM.COM
_PRINCIPAL root.admin@HPSSL.KGN.IBM.COM
_PRINCIPAL joe@HPSSL.KGN.IBM.COM
```

The file contains entries for the user root as principal **root.admin** and user joe as principal **joe**, giving them authority to execute log management commands.

The log management server functions executed by Sysctl are located in **/usr/lpp/ssp/sysctl/bin/logmgt.cmds**. During system installation, an include statement for this file is added to the default Sysctl configuration file **/etc/syctl.conf**. If an alternate Sysctl configuration file is used, it must be updated with a statement to include the **logmgt.cmds** file, and the **sysctld** daemon must be restarted to pick up this change. Refer to Chapter 4, “Sysctl” on page 51 for a description of the Sysctl function, ACL and AUTH callback authorizations, and using an alternate Sysctl configuration file.

Managing the AIX Error Log Facility

For detailed information on the AIX Error Log facility, refer to:

- *IBM RISC System/6000 Problem Solving Guide*, SC23-2641.
- *IBM General Concepts and Procedures for RISC System/6000*, GC23-2202.

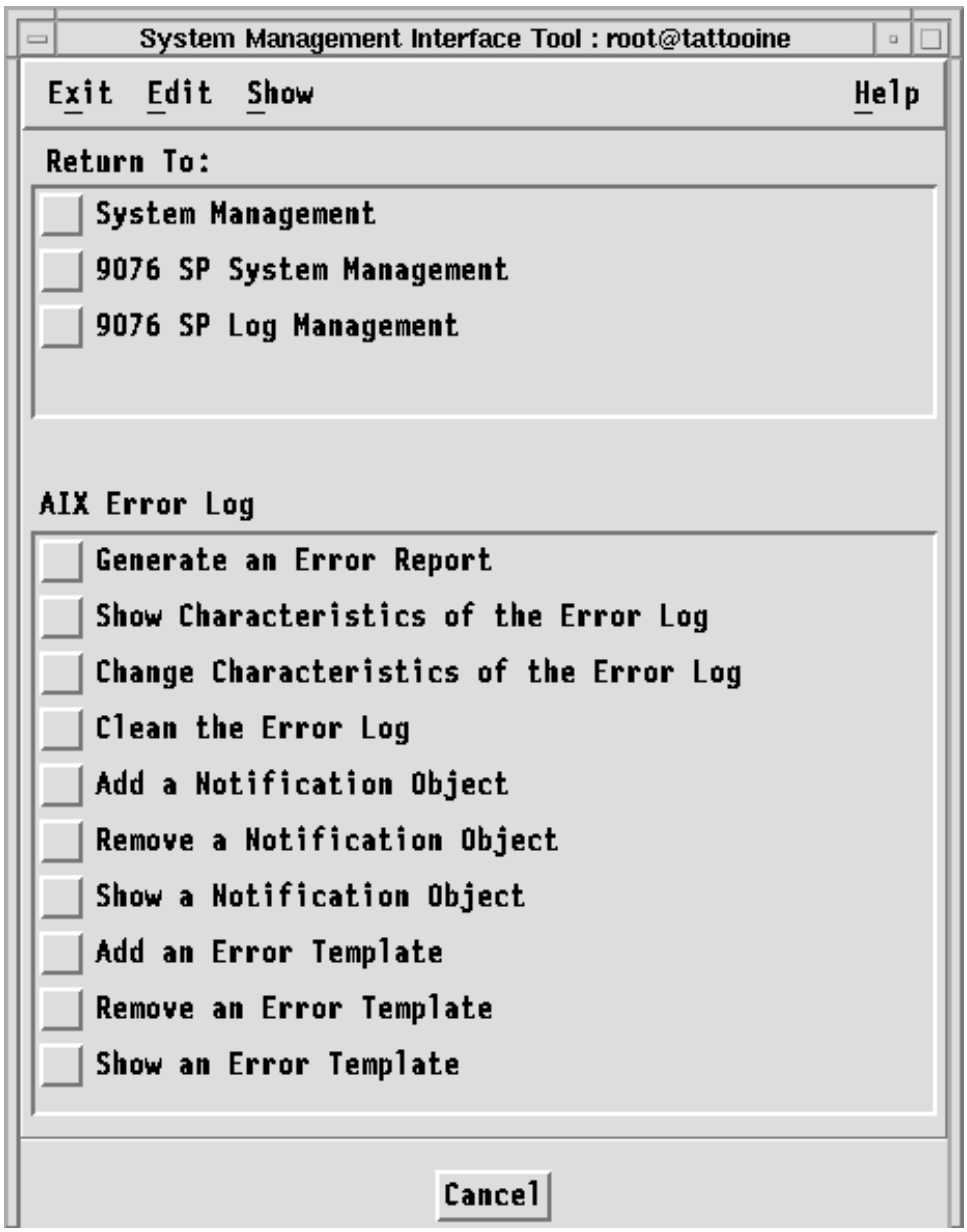
SMIT menus as well as commands are provided to help you manage the AIX Error Log facility. You can perform the following management tasks for the AIX Error Log facility:

- Generate AIX Error Log reports
- Trim AIX Error Log reports
- Configure AIX Error Logs
- Manage error notification objects
- Manage error templates

When you execute commands related to AIX Error Log management you can specify whether the command should be executed on all nodes in the current system partition or you can specify node names or the name of a file containing a list of node names. The default is the local node.

From SMIT: To access the AIX Error Log SMIT menu, enter:

```
smit sperrlog
```


System Management Interface Tool : root@tattooine
Exit Edit Show Help
Return To:
System Management
9076 SP System Management
9076 SP Log Management
AIX Error Log
Generate an Error Report
Show Characteristics of the Error Log
Change Characteristics of the Error Log
Clean the Error Log
Add a Notification Object
Remove a Notification Object
Show a Notification Object
Add an Error Template
Remove an Error Template
Show an Error Template
Cancel