IBM Parallel System Support
Programs for AIX

IBM

Administration Guide

GC23-3897-02

IBM Parallel System Support
Programs for AIX

IBM

Administration Guide

GC23-3897-02

Note!

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

Second Edition (November 1996)

This is a major revision of GC23-3897-01.

This edition applies to Version 2 Release 2 of IBM Parallel System Support Programs for AIX (5765-529), which runs on IBM RS/6000 Scalable POWERparallel Systems, and to all subsequent releases and modifications until otherwise indicated in new editions. Significant changes or additions to the text and illustrations are indicated by a vertical line (|) to the left of the change.

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 readers' comments may be provided at the back of this publication or you may address your comments to the following address:

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

FAX: (United States & Canada):1 + 914 + 432-9405
FAX: (Other Countries):

Your International Access Code+1+914+432-9405

IBMLink (United States customers only): KGNVMC(MHVRCFS)
IBM Mail Exchange: USIB6TC9 at IBMMAIL
Internet e-mail: mhvrcfs@vnet.ibm.com
World Wide Web: http://www.rs60000.ibm.com

If you would like a reply, be sure to include your name, address, telephone number, or FAX number.

Make sure to include the following in your comment or note:

Title and order number of this book

Page number or topic related to your comment

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 1996. 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 xxi
Interface Instructions xxii
Obtaining Documentation xxii
Related Publications xxiii
SP Hardware Publications xxiii
SP Software Publications xxiii
IBM Virtual Shared Disk Publications xxiii
Performance Monitor Publication xxiii
LoadLeveler Publications xxiv
Parallel Environment for AIX Publications xxiv
PVMe for AIX Publications xxiv
Client Input Output/Sockets (CLIO/S) Publications xxiv
Parallel I/O File System Publications xxiv
Network Tape Access and Control System for AIX (NetTAPE) Publications xxiv
Other IBM Publications xxv
International Technical Support Organization Publications (Red Books) . . . xxv
AIX and RS/6000 Publications xxv
Non-IBM Publications xxvi
Tcl xxvi
Ordering Additional Copies of This Book xxvi
Manual Pages for Public Code xxvi

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

Chapter 2. RS/6000 Scalable POWERparallel Systems Overview 5
Hardware Overview 5
Software Overview 8

Routine System Administration Tasks 17

Chapter 3. Starting Up and Shutting Down the SP System 19
Overview of System Startup and Shutdown 19
Controlling the Sequence of Startup and Shutdown Operations 19
Starting the SP System 20
Shutting Down the SP System 22
Checking the Status of Startup and Shutdown Requests 25
Defining Startup and Shutdown Sequence Files 26
Creating a Special Subsystem 29

Chapter 4. Parallel Management Commands 31

© Copyright IBM Corp. 1996

Using dsh to Execute Parallel Management Commands 31
The hostlist Command 33
List of Parallel Management Commands 34

Chapter 5. Managing File Collections 41
Understanding File Collections 41
Reporting File Collection Information 49
Verifying File Collections Using scan 51
Updating Files in a File Collection 52
Adding and Deleting Files in a File Collection 54
Building a File Collection 56
Installing a File Collection 59
Refusing Files in a File Collection 61
Modifying the File Collection Hierarchy 61
Removing a File Collection 63
supper list File Syntax, Keywords, and Operands 64

Chapter 6. Managing User Accounts 67
Understanding SP User Management 67
Adding User Accounts 70
Changing User Accounts 72
Removing User Accounts 73
Listing a User Account 75
Changing Passwords 75
Controlling User Login 77

Chapter 7. Managing SP Resources 85
Understanding the Resource Manager 85
Installing the Resource Manager 86
Defining the Resource Manager Configuration 88
Starting the Resource Manager 90
Reconfiguring the Resource Manager 96
Stopping the Resource Manager 97
Locating the Resource Manager 98
Querying the Resource Manager 98
Resource Manager Recovery 99

Chapter 8. Managing Time Synchronization 101
Managing NTP 101
Time Setting Options 103
Nodes and Control Workstations Out of Time Synchronization 103

Chapter 9. Managing Amd 105
Understanding the NFS and Amd Implementation 105
Establishing an Amd Map for Users' Directories 107
Using Amd for Other File Systems 108

Chapter 10. Managing Print Service 111
Print Management Security Modes 111
Print Management Command Support 112
SP Print Management System Requirements 114
The Print Management System and Kerberos 114
Integrating SP Print Management 115
Installing Print Management on a Control Workstation 118

Using a Non-AIX Workstation as Print Host 118

Chapter 11. Managing Mail Service 119

Chapter 12. Accounting 133
Accounting by Node or by Class 133
Setting Up SP Accounting 135
Accounting Files 142

Chapter 13. Maintaining the crontabs File 147

Security on the SP System 149

Chapter 14. Security Features of the SP System 151
Understanding Authentication 151
Using SP Security Services 160
Administering SP Security Services 166
Using AFS Authentication Services 182

Chapter 15. Sysctl 185
Sysctl Components 185
Relationship to the SP Authentication Services 187
Overview 187
Installation and Configuration 190
Using Sysctl 196
Developing a Sysctl Application 198
Sysctl Scripts 202
Sysctl Reference Information 204

Managing Sets of Nodes 209

Chapter 16. Using a Switch 211
Switch Communication Modes 211
Primary Node Takeover 212
System Partitioning and the Switch 213
Selecting Appropriate Switch Clocks 214
Selecting the Primary and Primary Backup Nodes 215
Managing the Switch Topology File 218
Starting the Switch 221
Determining Switch Connectivity 222
Fencing and Unfencing Nodes from the Switch 223
Preparing an SP Switch for Repartitioning 226
Monitoring Nodes on a Switch 227
Global Shutdowns and Reboots of Nodes with a Switch 227
NFS Mounts Over a Switch 228

Chapter 17. Managing System Partitions 229
Understanding System Partitioning 229
Partitioning the SP System 235
Displaying System Partition Configuration Information 249
Verifying System Partition Configuration 251
Managing System Partition-Sensitive Subsystems Using syspar_ctrl 252

Chapter 18. Managing Node Groups 257
Understanding Node Groups 257
Working with Node Groups 259
Using Node Groups with Commands 261

System Monitoring 263

Chapter 19. Using the SP Perspectives GUI 265
Getting Started 265
Understanding the SP Perspectives GUI Interface 267
Accessing Help 277
Performing Common Administration Tasks with the SP Perspectives GUI . . 278

Chapter 20. Using the SP System Monitor 281
Understanding the Hardware Monitor 282
Configuring the SP System Monitor 284
Customizing SP System Monitor GUI Default Resources 285
Using the Command Line Interfaces 285
Using the SP System Monitor GUI 286
Starting the SP System Monitor GUI 287
Displaying Frame, Node, and Switch Layouts 291
Monitoring and Controlling the Switch 292
Operating Frame Environment Controls 294
Operating the Front Panel Controls 297
Using Global Controls 304
Determining If Nodes Are Operating: All Node Summary 306
Determining the State of Network Adapters within the SP System 309
Setting Up the Logging Daemon on Separate Workstations 312
Setting Up the Hardware Monitor Clients on Separate Workstations 312

Availability and Problem Management 315

Chapter 21. Managing a High Availability Control Workstation Configuration 317
Understanding HACWS Function 318
Managing SP Subsystems for High Availability Control Workstation Function 326
Customizing HACMP/6000 332
Verifying Frame Supervisor Cables 334

Chapter 22. The Group Services Subsystem 337
Introducing Group Services 337
Group Services Components 338
Components on Which Group Services Depends 341
Group Services Configuration and Operation 342
Group Services Procedures 346

Chapter 23. The Event Management Subsystem 349
Introducing Event Management 349
Event Management Components 349
Components on Which Event Management Depends 359
Event Management Configuration and Operation 360
Event Management Procedures 365

Chapter 24. Using the Problem Management Subsystem 369
Understanding the Problem Management Daemon 370
Understanding the Problem Management Resource Monitor Daemon 376

Chapter 25. Managing SP System Events in a Network Environment . . 381
Understanding Network Management 382
Enabling Network Management for the SP System 383
Configuring the SP Proxy Agent 388
Configuring NetView for AIX for the SP System 389

Chapter 26. Managing Error Logs 391
Installing and Configuring Error Log Management 392
Managing the AIX Error Log Facility 393
Managing the BSD syslog Facility 397
Viewing Error Logs 400
Archiving Error Logs 400
Collecting System Data for IBM Service 402

System Performance 405

Chapter 27. Tuning the SP System 407
Performance Tuning and Monitoring 407
System Topology Considerations 409
SP System-Specific Tunable Recommendations 410
Tuning the Switch Device Driver Buffer Pools 417
Tuning Problems and Issues 418
Subsystem Tuning 423

Distributed Tape Management System Enablement 431

Chapter 28. Distributed Tape Management System Enablement 433
Overview 433
Magnetic Tape Extension (MTE) Pseudo Device Driver 434
Physical Tape Device Driver with the Tape Management Enhancement (TMSDD) 442

Appendixes 465

Appendix A. Using an Alternate Boot System Image 467
Enabling Alternate Boot System Image Function 467
Using the Alternate Boot System Image Function 468
Setting the Boot List Manually 468
Using Alternate System Image on the Control Workstation 469

Appendix B. SP Daemons 471

Appendix C. Sample Files 475
afsclient.cust 475
block_usr_sample 478
bootptab.info File 480
ibmSPMIB.my 481

jmd_config . . . 502
script.cust File . . . 505
tuning.cust File . . . 510
/usr/lib/X11/app-defaults/Spmon . . . 512

Appendix D. System Monitor Variables . . . 513
Frame Hardware Variables Common to All Frame Supervisors . . . 513
Frame Hardware Variables Unique to Card Type 17 and 20 Frame Supervisor 514
Frame Hardware Variables Unique to SEPBU (Type 18) and HACWS SEPBU (Type 22) Frame Supervisors . . . 515
Frame Hardware Variables Unique to HACWS (Type 20) and SEPBU HACWS (Type 22) Frame Supervisors . . . 516
Frame Hardware Variables Unique to Card Type 17 and SEPBU (Type 18) Frame Superviors . . . 517
Node Hardware Variables Common to All Node Supervisors . . . 517
Node-Specific Hardware Variables for Card Types 33 and 65 Node Supervisor 519
Node-Specific Hardware Variables for Card Type 97 Node Supervisors . . . 521
Node-Specific Hardware Variables for Card Types 81 and 83 Node Supervisors . . . 524
Node-Specific Hardware Variables For Card Type 161 Node Supervisor . . . 526
Common Switch Variables . . . 527
High Performance Switch and High Performance Switch-LC8 Card Types 49 and 50 . . . 528
SP Switch Variables for Card Type 129 . . . 529
Basecode Mode Hardware Variables for the SP Switch Card Type 128, SEPBU Frame Card Type 16 and 604 High Node Type 16 . . . 532

Appendix E. Listing SP System Resource Variables . . . 535

Appendix F. The System Data Repository . . . 541
Understanding the System Data Repository . . . 541
Accessing the System Data Repository . . . 543
Updating the host_responds Class (the hrd Daemon) . . . 550
Backing Up the SDR . . . 551
SDR Shadow Files . . . 551
System Data Repository Classes and Attributes . . . 552

Appendix G. IP Address and Host Name Changes for SP Systems . . . 575
SDR Objects with IP Address or Host Names . . . 575
SP System Files with IP Address and Host Names . . . 576
Selecting the Correct PSSP Procedure . . . 578
Changing the IP Address and Host Names for PSSP 1.2 Systems . . . 578
Changing the IP Address and Host Names for PSSP 2.1 Systems . . . 582
Execution on PSSP 2.1 Nodes . . . 589
Updating the PSSP 1.2 Boot Server Node on PSSP 2.x Systems . . . 590
Changing the IP Address and Host Names for PSSP 2.2 Systems . . . 592
Execution on PSSP 2.2 Nodes . . . 599

Glossary of Terms and Abbreviations . . . 603

Index . . . 611

Figures

1. Basic SP Configuration . . . 5
2. SP Default File Collection Hierarchy . . . 45
3. /var/sysman/sup Files and Directories . . . 46
4. sup.admin Master Files . . . 47
5. SP Hierarchy of an Offline Boot/install Server . . . 62
6. SMIT Panel for Site Environment Information . . . 68
7. SMIT Panel for Adding an SP User . . . 71
8. Example of Adding an SP User . . . 72
9. SMIT Panel for Changing SP User Information . . . 73
10. SMIT Panel for Removing an SP User . . . 74
11. Example Resource Manager Configuration . . . 90
12. Sample /etc/amd/amd-maps/amd.u File . . . 108
13. Option 1: Using DNS . . . 121
14. Option 2: Using NIS . . . 126
15. Option 3: Using /etc/hosts . . . 129
16. Accounting Fields on the Site Environment SMIT Panel . . . 137
17. SP Accounting Information SMIT Panel . . . 139
18. Accounting Template for crontabs . . . 142
19. An Overview of Kerberos Authentication . . . 155
20. Sysctl Elements . . . 186
21. The Change/Show Switch Clock Source Settings SMIT Menu . . . 214
22. Initial Value . . . 215
23. Eprimary Value . . . 215
24. Estart Value . . . 215
25. Initial Values . . . 216
26. Estart Values . . . 217
27. Backup Takeover Values . . . 217
28. Primary Takeover Values . . . 217
29. The Set Primary/Primary Backup Node SMIT Menu . . . 218
30. The Topology File Annotator SMIT Menu . . . 220
31. The Fetch / Store Topology File SMIT Menu . . . 221
32. The Fence / Unfence SP Nodes SMIT Menu . . . 224
33. The Set Switch Run Phase Duration SMIT Menu . . . 226
34. System Partition Configuration SMIT Panel . . . 237
35. Archive System Data Repository SMIT Panel . . . 239
36. System Partition Configuration List SMIT Panel . . . 240
37. Select System Partition Configuration SMIT Panel . . . 241
38. Display Information for Given Partition or Layout . . . 242
39. Display Information for Given Partition or Layout Output . . . 242
40. Select System Partition Layout SMIT Panel . . . 243
41. Select System Partition Layout Output SMIT Panel . . . 244
42. Partition Directory SMIT Panel . . . 245
43. Enter Customization Arguments for this System Partition SMIT Panel . . . 245
44. Verify System Partition Configuration SMIT Panel . . . 247
45. Apply System Partition Configuration SMIT panel . . . 249
46. List System Partition Database Information Output . . . 250
47. Verify System Partition Configuration SMIT Panel . . . 251
48. The Partition Node Groups SMIT Panel . . . 260
49. The SP Perspectives GUI Launch Pad . . . 268
50. Elements of an SP Perspectives GUI Primary Window . . . 270

© Copyright IBM Corp. 1996

51. An SP Perspectives GUI Primary Window . . . 271
52. An SP Perspectives GUI Dialog Box . . . 273
53. An SP Perspectives GUI Notebook . . . 275
54. An SP Perspectives GUI Message Box . . . 276
55. The SP System Monitor . . . 282
56. SP System Monitor GUI Main Window . . . 289
57. View Pulldown Menu . . . 290
58. SP Pulldown Menu . . . 290
59. SP System Monitor GUI Display Layouts Selection Window . . . 291
60. SP System Monitor GUI Switch Front Panel Layout Window . . . 293
61. Frame Environment Layout for Type 18 Frame Supervisor . . . 296
62. Frame Environment Layout for Type 22 Frame Supervisor . . . 297
63. Node Front Panel Display . . . 298
64. Power Off Menu . . . 299
65. Power On Menu . . . 300
66. Reset Menu . . . 301
67. Shutdown Menu . . . 302
68. Fence Menu . . . 303
69. Global Controls Display . . . 305
70. Node Variable Selection Window . . . 307
71. 3-Digit Display . . . 308
72. LAN Topology Display . . . 310
73. LAN Status Display . . . 311
74. An HACWS Configuration . . . 319
75. Addressing in an HACWS Configuration . . . 322
76. Frame Environment Layout Showing Active Control Workstation . . . 328
77. HACWS Pre- and Post-Events . . . 333
78. An Example of Problem Management Daemon Configuration . . . 371
79. Log Management SMIT Panel . . . 392
80. AIX Error Log SMIT Panel . . . 394
81. Syslog SMIT Panel . . . 398
82. General Log Viewing SMIT Panel . . . 400
83. Archive Logs SMIT Panel . . . 401
84. Collect Logs for Service SMIT Panel . . . 403
85. Simplified View of a Single Parent–Multiple Child Application . . . 422
86. SMIT Main Menu . . . 439
87. SMIT Devices Menu . . . 439
88. SMIT Tape Drive Menu . . . 440
89. Tape Drive Type Menu . . . 440
90. Add a Tape Management Pseudo Control Point Menu . . . 441
91. TMPCP COMMAND STATUS Window . . . 441
92. Configure a TMPCP Selection List . . . 441
93. Configure a Defined TMPCP COMMAND STATUS Window . . . 442
94. SMIT Main Menu . . . 448
95. SMIT Devices Menu . . . 449
96. SMIT Tape Drive Menu . . . 449
97. Tape Drive Type Menu . . . 450
98. Parent Control Unit Menu . . . 450
99. Add a 3480 Tape Drive Menu . . . 451
100. Add a 3490 Tape Drive COMMAND STATUS Window . . . 452
101. Configure a Defined Tape Drive Selection List . . . 452
102. Configure a Defined Tape Drive COMMAND STATUS Window . . . 452
103. SMIT Main Menu . . . 453
104. SMIT Devices Menu . . . 454

105. Tape Drive Type Menu 454
106. Tape Drive Submenu 455
107. Change / Show Characteristics of a 3480 Tape Drive Menu 455
108. Device Features List Submenu 456
109. Media Type List Submenu 457
110. Command Status Menu 458
111. SMIT Main Menu 459
112. SMIT Devices Menu 460
113. Tape Drive Type Menu 460
114. Tape Drive Submenu 461
115. Change / Show Characteristics of a 3490E Tape Drive Menu 461
116. Device Features List Submenu 462
117. Media Type List Submenu 463
118. Command Status Menu 464
119. Sample Output from PSSP SDRGetObjects commands 537
120. Using the Event Perspective to Get Resource Variable Information . . 539
121. Comparing Two SDR Classes 541
122. SP System Management SMIT Panel 545
123. List RS/6000 SP Configuration Information 547
124. List Node File System Information Panel 548

Tables

1. Specifying Nodes to Start with cstartup 21
2. Specifying Nodes to Shut Down with cshutdown. 23
3. Time Service Choices 101
4. Authorization Variables 201
5. Built-in Tcl Commands and Their Default Authorization Callbacks . . . 205
6. Built-in Extended Tcl Commands and Their Default Authorization Callbacks 205
7. ACL Processing Commands 206
8. Service Commands 206
9. Other Commands 207
10. Summary of System Partition-Sensitive Subsystems 254
11. Frame, Node, and Switch Layouts 292
12. Default Variable Colors 309
13. Graphical Element Colors 311
14. High Availability Control Workstation Configuration and Task Summary 325
15. Initial Values for AIX 4.1 417
16. Initial Values for AIX 3.2.5 417
17. TCP/IP(MVS) parameters/settings 427
18. LED Translation Values 519
19. Node-Specific Hardware Variables For Card Type 161 Node Supervisor 526
20. SDR Authorizations 543
21. Commands Invoked by SMIT Panels 548

© Copyright IBM Corp. 1996

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 P300
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
RS/6000
RS/6000 Scalable POWERparallel Systems

Microsoft, Windows, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

© Copyright IBM Corp. 1996

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

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

C-bus is a trademark of Corollary, Inc.

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 discusses the use of these products only 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 Parallel System Support Programs for AIX: Administration Guide* provides information to help you understand and perform SP system administration. It includes concepts and instructions pertaining to:

- Understanding IBM RS/6000 Scalable POWERparallel Systems (SP)
- Performing routine system administration tasks
- Making the system secure
- Managing sets of nodes
- Monitoring the system
- Providing for system availability and problem management
- Tuning the system

Note: The chapters "Defining and Managing IBM Virtual Shared Disks" and "Defining and Managing IBM Virtual Shared Disk Data Striping Devices" that previously appeared in this book have been moved to *IBM Parallel System Support Programs for AIX: Managing Shared Disks*, SC23-3849.

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 Parallel System Support Programs for AIX: 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 Parallel System Support Programs for AIX: Installation and Migration Guide* helps you understand and perform SP installation, configuration, maintenance, and migration. It includes concepts and instructions for:

- Installing and configuring the IBM Parallel System Support Programs (PSSP) for AIX
- Migrating to PSSP Version 2 Release 2
- Reconfiguring the system
- Performing system maintenance

The *IBM Parallel System Support Programs for AIX: Diagnosis and Messages Guide* 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 Parallel System Support Programs for AIX: Command and Technical Reference* provides detailed syntax and parameter information for all commands you can use to install, customize, and maintain the SP system.

This book applies to PSSP Version 2 Release 2. To find out what version of PSSP is running on your system, enter the following:

```
SDRGetObjects Syspar syspar_name code_version
```

In response, the system displays something similar to:

```
syspar_name   code_version
Part_1        PSSP-2.2
```

© Copyright IBM Corp. 1996

If the response indicates **PSSP-2.2**, this book applies to the version of PSSP that is running on your system.

If you are running mixed levels of PSSP in a system partition, be sure to maintain and refer to the appropriate documentation for whatever versions of PSSP you are running.

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 information related to managing the day-to-day running of the SP system, providing system security, managing sets of nodes, monitoring and operating the system, providing for system availability and problem management, and tuning the system.

Chapter 1, "Introduction" on page 1 briefly describes the new and improved features of the SP System.

Chapter 2, "RS/6000 Scalable POWERparallel Systems Overview" on page 5 describes the hardware and software components of the SP system.

Routine System Administration Tasks

Chapter 3, "Starting Up and Shutting Down the SP System" on page 19 shows how to use commands to bring up the SP system and shut it down.

Chapter 4, "Parallel Management Commands" on page 31 describes commands useful for controlling several nodes in parallel.

Chapter 5, "Managing File Collections" on page 41 describes procedures for installing, updating, and removing file collections.

Chapter 6, "Managing User Accounts" on page 67 provides information on passwords, user access, and instructions on how to give your users access to the system.

Chapter 7, "Managing SP Resources" on page 85 describes how to use the Resource Manager.

Chapter 8, "Managing Time Synchronization" on page 101 discusses how time service is used in the SP system.

Chapter 9, “Managing Amd” on page 105 discusses how to use the Automount daemon (Amd).

Chapter 10, “Managing Print Service” on page 111 describes how print management is handled in the SP system.

Chapter 11, “Managing Mail Service” on page 119 shows you how to integrate your mail system.

Chapter 12, “Accounting” on page 133 describes how to run SP Accounting in your environment.

Chapter 13, “Maintaining the crontabs File” on page 147 contains a few hints on running **cron** jobs.

Security on the SP System

Chapter 14, “Security Features of the SP System” on page 151 describes how to set up, maintain, and use authentication services.

Chapter 15, “Sysctl” on page 185 describes how to install, configure, and use the authenticated client/server system for running commands remotely and in parallel.

Managing Sets of Nodes

Chapter 16, “Using a Switch” on page 211 discusses switch operations.

Chapter 17, “Managing System Partitions” on page 229 describes the tasks necessary to partition (and repartition) your SP system.

Chapter 18, “Managing Node Groups” on page 257 presents information for understanding and working with node groups.

System Monitoring

Chapter 19, “Using the SP Perspectives GUI” on page 265 describes Scalable POWERparallel Perspectives for AIX (SP Perspectives) and provides details on starting the GUI, understanding the SP Perspectives GUI interface, and accessing help.

Chapter 20, “Using the SP System Monitor” on page 281 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.

Availability and Problem Management

Chapter 21, “Managing a High Availability Control Workstation Configuration” on page 317 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 22, “The Group Services Subsystem” on page 337 provides information on managing the Group Services subsystem.

Chapter 23, “The Event Management Subsystem” on page 349 provides information on managing the Event Management subsystem.

Chapter 24, “Using the Problem Management Subsystem” on page 369 describes the primary components of the Problem Management subsystem and how to manage it.

Chapter 25, “Managing SP System Events in a Network Environment” on page 381 describes how to issue SNMP traps in response to system events.

Chapter 26, “Managing Error Logs” on page 391 describes the functions provided for managing the AIX Error Log facility, the BSD syslog, and SP-specific log files.

System Performance

Chapter 27, “Tuning the SP System” on page 407 contains recommendations for fine-tuning your environment for better performance.

Distributed Tape Management System Enablement

Chapter 28, “Distributed Tape Management System Enablement” on page 433 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**>.

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.

Obtaining Documentation

This book is available in PostScript format. You can obtain PostScript files for the product documentation by:

- Downloading them from the **lscftp.pok.ibm.com** anonymous FTP server. The files for xyz_prod_name can be found in the directory called **/pub/pps/xyz_ftp_prod_dir**.
- Printing the PostScript documentation files that are shipped on the product media.

You can also view or download this book from the IBM RS/6000 web site at **http://www.rs6000.ibm.com**.

Related Publications

SP Hardware Publications

IBM RS/6000 Scalable POWERparallel Systems: Site Planning, GC23-3905

IBM RS/6000 Scalable POWERparallel Systems: Maintenance Information, Volume 1, Installation and CE Operations, GC23-3903

IBM RS/6000 Scalable POWERparallel Systems: Maintenance Information, Volume 2, Maintenance Analysis Procedures and Parts Catalog, GC23-3904

IBM RS/6000 Scalable POWERparallel Systems: Library Guide, GC23-3868

IBM RS/6000 Scalable POWERparallel Systems: Installation Aid Licensed Program Specification, GC23-3838

SP Software Publications

IBM Parallel System Support Programs for AIX: System Planning, GC23-3902

IBM Parallel System Support Programs for AIX: Installation and Migration Guide, GC23-3898

IBM Parallel System Support Programs for AIX: Diagnosis and Messages Guide, GC23-3899

IBM Parallel System Support Programs for AIX: Command and Technical Reference, GC23-3900

IBM Parallel System Support Programs for AIX: Event Management Programming Guide and Reference, SC23-3996

IBM Parallel System Support Programs for AIX: Group Services Programming Guide and Reference, SC28-1675

IBM Parallel System Support Programs for AIX: Licensed Program Specification, GC23-3901

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

IBM Virtual Shared Disk Publications

IBM Parallel System Support Programs for AIX: Managing Shared Disks: IBM Virtual Shared Disks, Data Striping Devices, and the IBM Recoverable Virtual Shared Disk Licensed Program, SC23-3849

IBM Recoverable Virtual Shared Disk: Licensed Program Specification, GC23-3848

Performance Monitor Publication

IBM Performance Toolbox Parallel Extensions for AIX: Guide and Reference, SC23-3997

LoadLeveler Publications

Using and Administering LoadLeveler, SC23-3989

IBM LoadLeveler: Licensed Program Specification, GH23-0040

Parallel Environment for AIX Publications

IBM Parallel Environment for AIX: Hitchhiker's Guide, GC23-3895

IBM Parallel Environment for AIX: Operation and Use, Volume 1, SC28-1979

IBM Parallel Environment for AIX: Operation and Use, Volume 2, SC28-1980

IBM Parallel Environment for AIX: Installation Guide, GC28-1981

IBM Parallel Environment for AIX: Messages, GC28-1982

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 SBOF-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, GC28-1676

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, GH34-6066

Network Tape Access and Control System for AIX (NetTAPE) Publications

IBM NetTAPE General Information, GC23-3990

IBM NetTAPE User's Guide and Reference, available from your IBM representative

Other IBM Publications

International Technical Support Organization Publications (Red Books)

IBM International Technical Support Centers Implementing High Availability on RS/6000 SP, SG24-4742

IBM International Technical Support Centers RS/6000 SP High Availability Infrastructure, SG24-4838

IBM International Technical Support Centers RS/6000 SP PSSP 2.2 Technical Presentation, SG24-4868

AIX and RS/6000 Publications

IBM AIX Version 4 Getting Started, SC23-2527

IBM General Concepts and Procedures for RS/6000, GC23-2202

IBM AIX Version 4 Files Reference, SC23-2512

IBM AIX Version 4 System Management Guide: Communications and Networks, SC23-2526

IBM AIX Version 4.1 Installation Guide, SC23-2550

IBM AIX Version 4.2 Installation Guide, SC23-1924

IBM AIX Version 4 Commands Reference, SBOF-1851

IBM AIX Versions 3.2 and 4 Performance Tuning Guide, SC23-2365

IBM AIX Version 4 Problem Solving Guide and Reference, SC23-2606

IBM AIX Version 4 Messages Guide and Reference, SC23-2641

IBM AIX Version 4.1 Network Installation Management Guide and Reference, SC23-2627

IBM AIX Version 4.2 Network Installation Management Guide and Reference, SC23-1926

IBM AIX Version 4 System Management Guide: Operating System and Devices, SC23-2525

IBM AIX Version 4 General Programming Concepts: Writing and Debugging Programs, SC23-2533

IBM AIX Version 4 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

IBM Performance Toolbox 1.2 and 2 for AIX: Guide and Reference, SC23-2625

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 (Version 4.036)	/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
Perl (Version 5.002)	Man pages are in the /usr/lpp/ssp/perl5/man/man1 directory

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

RS/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 RS/6000 technology, thousands of existing RS/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, the SP 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 2

Version 2 Release 2 of the IBM Parallel System Support Programs for AIX provides the following functional enhancements:

© Copyright IBM Corp. 1996

Installation Support

Installation software has been enhanced to provide automated recovery for various error conditions that can occur during the installation process.

System Partitioning Aid

The System Partitioning Aid is a tool that creates customized system partition configurations based on:

- A desired configuration (a list of nodes in each system partition) specified by the System Administrator
- A set of built-in rules for partitioning the switch that provide for complete isolation of system partitions, as well as reliability and availability
- An evaluation of system partition performance

When the System Administrator accepts a system partition configuration generated by the System Partitioning Aid, the tool generates the topology files, node lists, and layout description and installs the configuration in the system partitioning directory structure.

The System Partitioning Aid can be invoked from a command line interface or from the SP Perspectives GUI.

604 High Node Support

The 604 High Node introduces Symmetric Multi-Processor and PowerPC technology to the SP system. Support for the new node is provided by the System Partitioning Aid, the IBM Virtual Shared Disk, and the communication subsystem. In addition, system management support is provided that allows the 604 High Node to be managed in a way that is consistent with the management of thin and wide nodes.

Scalable POWERparallel Perspectives for AIX

Scalable POWERparallel Perspectives for AIX (SP Perspectives) is a graphical user interface (GUI) that enables you to perform system management tasks for your SP system through direct manipulation of system objects represented by icons. You can use the SP Perspectives GUI to perform the following system management tasks:

- Monitor and control hardware
- Create and monitor system events
- Define and manage IBM Virtual Shared Disks
- Generate and save system partition configurations
- Set up performance monitoring hierarchies and archiving

Node Grouping

Function is provided to group nodes together so that the group can be acted on as a single entity or object. Node groups can be created, modified, and deleted through a command line interface, SMIT panels, or through the SP Perspectives GUI.

Problem Management

Problem Management provides an infrastructure for recognizing and reporting problem events in your SP system. This infrastructure is based on an Event Management application that provides configurable access to Event Management client and resource monitor function without the necessity of writing C programs that use the Event Management APIs.

Performance Toolbox Parallel Extensions for AIX

Performance Toolbox Parallel Extensions for AIX (PTPE) is a performance monitor that provides easy access to performance information about SP system hardware and software. This information is available in both run-time (current) and archived (historical) data that you can analyze, manipulate, print, and import to a database.

For information on PTPE, refer to *Performance Toolbox Parallel Extensions for AIX Guide and Reference*.

PTPE is a separately priced feature.

Availability Services

Availability services consist of:

- Event Management services
- Group Services

Event Management services allow one set of programs, called resource monitors, to report the status or changes in status of system resources, and another set of programs, called Event Management clients, to receive notification when system resources meet certain conditions. Working together through the Event Management subsystem, resource monitors and Event Management clients can detect failures, report them, and initiate recovery actions automatically.

Group services can be used to coordinate the actions of an application that consists of multiple processes running on multiple nodes.

For information on Event Management and Group Services, refer to *IBM Parallel System Support Programs for AIX: Event Management Programming Guide and Reference* and *IBM Parallel System Support Programs for AIX: Group Services Programming Guide and Reference*.

IBM Virtual Shared Disk Usability Enhancements

A number of usability enhancements have been made to IBM Virtual Shared Disk function, including:

- One command line to define IBM Virtual Shared Disks
- One command line to undefine IBM Virtual Shared Disks
- The ability to configure a subset of IBM Virtual Shared Disks within a system partition
- A complete and self-explanatory set of error messages

For more information on these and other enhancements made to the IBM Virtual Shared Disk, refer to *IBM Parallel System Support Programs for AIX: Managing Shared Disks*.

System Partition-Sensitive Subsystem Management

The Syspar Controller is provided to manage the SP subsystems that operate within the domain of a system partition. The Syspar Controller operates through the **syspar_ctrl** command to add, start, stop and delete system partition-sensitive subsystems.

Support for Mixed Levels of PSSP in a System Partition

PSSP 2.2 supports coexistence in the same system partition with PSSP 1.2 nodes or with PSSP 2.1 nodes. This special migration support is documented in a technical bulletin available on the RS/6000 home page on the World Wide Web and through your IBM marketing representative. The bulletin, *Implementing Coexistence with PSSP 2.2*, is available at **http://www.rs6000.ibm.com/tech**.

Chapter 2. RS/6000 Scalable POWERparallel Systems Overview

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

Hardware Overview

The basic hardware components of an SP system are:

- Processor nodes
- Frames
- Optional 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.

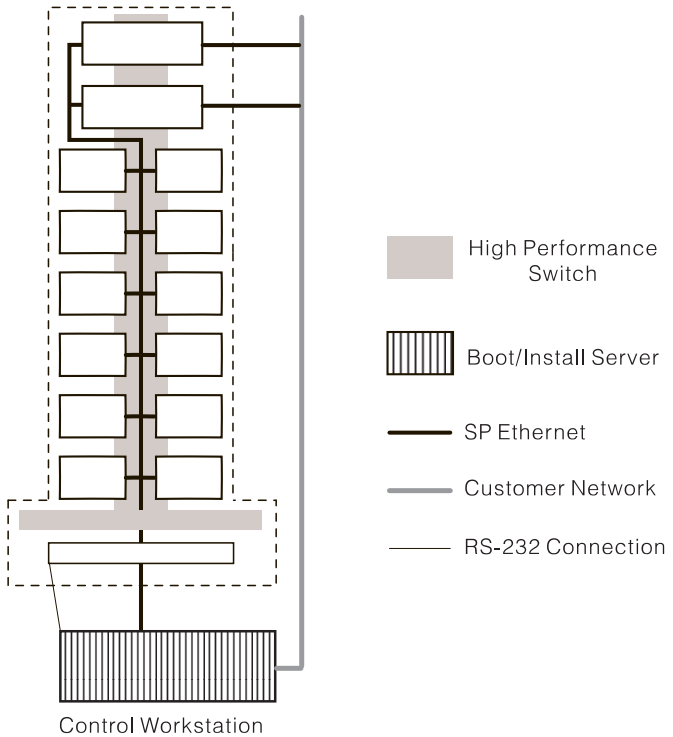


Figure 1. Basic SP Configuration

© Copyright IBM Corp. 1996

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 three types: thin nodes, wide nodes, and 604 High Nodes. Thin nodes are typically configured as compute nodes, while wide nodes are more often used as servers to provide high-bandwidth data access. 604 High Nodes provide advantages as data base servers or LAN consolidation servers. 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 RS/6000 desktop system. Half the width of the wide node, it can be packaged 16 to a frame and can contain either a RS/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 RS/6000 deskside system. It is a RS/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.

604 High Nodes

The 604 High Node is a 2-, 4-, 6-, or 8-way symmetric multiprocessing system that uses PowerPC 604 processors that operate at 112 megahertz. The 604 High Node occupies two drawers (four slots) in a frame and can be packaged up to four per frame. It has two Micro Channel buses for I/O attachment, including attachment to the High Performance or SP Switch (however, only one switch adapter will be supported).

A unique feature of the 604 High Node is the SystemGuard collection of firmware programs that run on the bringup microprocessor (BUMP). The SystemGuard and BUMP provide service processor capability. They enable the operator to manage power supplies, check system hardware status, update various configuration parameters, investigate problems, and perform tests. Refer to *IBM Parallel System Support Programs for AIX: Diagnosis and Messages Guide* for more information on SystemGuard and the 604 High Node.

Frames

An SP frame can house up to four 604 High Nodes or eight wide nodes or 16 thin nodes. 604 High Nodes and 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.

An SP Switch-8 configuration can consist of either two short 49" frames with single phase power that supports two to eight nodes in up to two frames, or one 79" frame that supports up to eight 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 16, “Using a Switch” on page 211 for more information.

Scalable POWERparallel Switch

In addition to all the capabilities provided by the High Performance Switch described previously, the SP Switch provides:

- Higher availability
- Fault isolation
- Concurrent maintenance for nodes
- Improved switch chip bandwidth

The new SP Switch Adapter connects each SP node to the SP Switch subsystem. See Chapter 16, “Using a Switch” on page 211 for more information.

Control Workstation

The SP system requires a customer-supplied RS/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 Parallel System Support Programs for AIX: Installation and Migration 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 14, “Security Features of the SP System” on page 151.)

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.

Highlights

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

Installation: Installation of AIX 4.1.4 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 Perspectives GUI provides for monitoring and controlling hardware, creating and monitoring system events, managing IBM Virtual Shared Disks, generating and saving system partition configurations, and setting up performance monitoring hierarchies and archiving.

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.

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.

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.

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.

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

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

Note: PSSP supports both the SP Switch and the High Performance Switch, but the two switch networks are not compatible and cannot be mixed within an SP system.

Availability and Problem Management:

Group Services provide a distributed coordination and synchronization service.
Event Management matches information about the state of system resources with information about resource conditions that are of interest to client programs.
Problem Management provides for pairing actions with system events.
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
ptpe

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 RS/6000 Scalable POWERparallel Systems: Site Planning* and the *IBM Parallel System Support Programs for AIX: System Planning* guides to help you prepare for installing and configuring an SP system.)

System operation and monitoring
(See Chapter 20, “Using the SP System Monitor” on page 281.)

Distributed shell
(See Chapter 4, “Parallel Management Commands” on page 31.)

Login control
(See Chapter 6, “Managing User Accounts” on page 67.)

System Data Repository (SDR)
(See Appendix F, “The System Data Repository” on page 541.)

ssp.gui: This option contains the SP Perspectives GUI as well as the graphical user interface for the System Monitor. Refer to Chapter 19, “Using the SP Perspectives GUI” on page 265 and Chapter 20, “Using the SP System Monitor” on page 281 for more information.

ssp.sysman: This option contains the following functions:

Time services
(See Chapter 8, “Managing Time Synchronization” on page 101.)

User management
(See Chapter 6, “Managing User Accounts” on page 67.)

Print support
(See Chapter 10, “Managing Print Service” on page 111.)

File collections
(See Chapter 5, “Managing File Collections” on page 41.)

Accounting
(See Chapter 12, “Accounting” on page 133.)

Parallel management commands
(See Chapter 4, “Parallel Management Commands” on page 31.)

Error Log Management
(See Chapter 26, “Managing Error Logs” on page 391.)

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 7, “Managing SP Resources” on page 85 for more information.

ssp.st: This option contains a library and commands that support an application programming interface for loading, unloading, and querying the status of job switch tables.

ssp.css: **Communications Subsystem Support** performs 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 16, “Using a Switch” on page 211 contains more details.

ssp.top: This option contains directories and files for supported system partitioning configurations, as well as the System Partitioning Aid, a tool for creating customized system partition configurations to suit your needs.

ssp.top includes:

- 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
- The System Partitioning Aid

ssp.top.gui contains the graphical user interface for the System Partitioning Aid. Refer to the *IBM Parallel System Support Programs for AIX: Installation and Migration Guide* and Chapter 17, “Managing System Partitions” on page 229 in this guide for more information.

ssp.ha: This option contains subsystems for high availability, including heartbeat, Group Services, and Event Management.

ssp.pman: This option contains the Problem Management subsystem. The Problem Management subsystem provides configurable access to Event Management client and resource monitor function. It includes daemons, commands, utilities, and scripts. Refer to Chapter 24, “Using the Problem Management Subsystem” on page 369 and Chapter 25, “Managing SP System Events in a Network Environment” on page 381 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 RS/6000 workstations that are configured as Kerberos server systems. It is not installed on the SP nodes. Refer to the *IBM Parallel System Support Programs for AIX: Installation and Migration Guide* and Chapter 14, “Security Features of the SP System” on page 151 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 15, “Sysctl” on page 185 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 RS/6000 workstation on which you desire these capabilities. Refer to the *IBM Parallel System Support Programs for AIX: Installation and Migration Guide* and Chapter 14, “Security Features of the SP System” on page 151 in this guide for more information.

ssp.perlpkg: This option contains Perl 4.036 and Perl 5.002 programs and libraries.

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 Parallel System Support Programs for AIX: Installation and Migration Guide*, the *IBM Parallel System Support Programs for AIX: Diagnosis and Messages Guide*, the *IBM Parallel System Support Programs for AIX: Command and Technical Reference*, the *IBM RS/6000 Scalable POWERparallel Systems: Site Planning* and the *IBM Parallel System Support Programs for AIX: System Planning*.

spimg

The **spimg** installp image contains a **mksysb**, or multiple **mksysb**s, of an AIX 4.1 system that you can use to install AIX 4.1 on the nodes in your SP system, after you have successfully installed and configured your control workstation. 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 Parallel System Support Programs for AIX: Installation and Migration 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 five 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.

ssp.csd.sysctl IBM Virtual Shared Disk commands that use Sysctl function.

ssp.csd.gui The graphical user interface for IBM Virtual Shared Disks.

See *IBM Parallel System Support Programs for AIX: Managing Shared Disks* 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 28, “Distributed Tape Management System Enablement” on page 433 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.

ptpe

This image provides the Performance Toolbox Parallel Extensions for AIX, a performance monitor that provides access to performance information about SP system hardware and software. The **ptpe** image is split into three options:

ptpe.docs Performance Toolbox Parallel Extension for AIX documentation, including a PostScript file and hypertext version (accessible with the **InfoExplorer** tool).

ptpe.gui The graphical user interface for the Performance Toolbox Parallel Extension for AIX.

ptpe.program The Performance Toolbox Parallel Extension for AIX program.

Note

This is an optional separately priced feature.

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.

Routine System Administration Tasks

© Copyright IBM Corp. 1996

Chapter 3. 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 29) 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. 1996

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 26 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.

Note on 604 High Nodes

The 604 High Node takes a considerably longer time to power on than thin and wide nodes. It can take a 604 High Node 10 to 20 minutes to power on, depending on the number of processors in the node and the number of adapters. You may wish to control the startup and shutdown of 604 High Nodes manually, rather than with the **cstartup** and **cshutdown** commands.

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 Parallel System Support Programs for AIX: Command and Technical Reference*.

Starting or Resetting All Nodes

You can start or reset nodes with a single **cstartup** command, as shown in Table 1.

Table 1. 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 26.

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 user-defined 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 2 on page 23.

Table 2. 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 Parallel System Support Programs for AIX: 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 24.

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:   -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:   -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 1  \
          -M "Reboot on cs1&2 in 1  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: node     #node  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 129.4 .64.7  \
                                -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 129.4 .64.7  \
                                -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 129.4 .64.7  \
        -f /var/adm/SPlogs/cs/cshut.time.pid \
        subsystemA:    initiating shutdown at 15:43.55 on 1 Apr 94.

csLogger -b 129.4 .64.7  \
        -f /var/adm/SPlogs/cs/cshut.time.pid \
        subsystemA:    shutdown complete at 15:46.23 on 1 Apr 94.
```

Chapter 4. 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 33). 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. 1996

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 Parallel System Support Programs for AIX: 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 32. 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:

```
r 5n 9: /usr/lpp/ssp/rcmd/bin/rcp /tmp/bozoid tserv11:/tmp/bozoid
r 5n 9: /usr/lpp/ssp/rcmd/bin/rcp /tmp/bozoid r 5n 1:/tmp/bozoid
r 5n 1: backup_script
r 5n15: 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 ===========================================================================
r 5n 1              r 5n 3dx            r 5n 5              r 5n 7
r 5n 9              r 5n1               r 5n13              r 5n14dx
r 5n15              r 5n16
 ===========================================================================
 /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 ===========================================================================
r 5n 1              r 5n 3dx            r 5n 5              r 5n 7
r 5n 9              r 5n1               r 5n13              r 5n14dx
r 5n15              r 5n16
===========================================================================
/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               host1               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 r 5n15,r 5n16" -tn sysctld -print
```

produces this output:

```
HOSTS ===========================================================================
r 5n15
===============================================================================
USER          PID %CPU %MEM      SZ    RSS      TTY STAT      STIME    TIME COMMAND
root        11172    .    1.    452   42          -  S       11:54:47    :
/usr/lpp/ssp/bin/sysctld -l /var/adm/SPlogs/sysctl/sysctld.log
root        12497    .    1.    3     36          -  S        6:24:19    :
/usr/lpp/ssp/bin/sysctld -l /var/adm/SPlogs/sysctl/sysctld.log

HOSTS ===========================================================================
r 5n16
===============================================================================
USER          PID %CPU %MEM      SZ    RSS      TTY STAT      STIME    TIME COMMAND
root          7 89    .    1.    448   42          -  S       11:54:48    :
/usr/lpp/ssp/bin/sysctld -l /var/adm/SPlogs/sysctl/sysctld.log
root          8423    .    1.    296   36          -  S       13: 1:31    :
/usr/lpp/ssp/bin/sysctld -l /var/adm/SPlogs/sysctl/sysctld.log
root        1 418    .    1.    512   5 4         -  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: r 5n 1
============
/var                    8192    2452    574     71%    279    1769   87%

HOST: r 5n 3dx
==============
/var                    8192    1988    62 4    76%    214    1834   9 %

HOST: r 5n 5
============
/var                    8192    2 8     6112    75%    223    1825   9 %

HOST: r 5n 7
============
/var                    8192    2 8     6112    75%    223    1825   9 %

HOST: r 5n 9
============
/var                    8192    2 88    61 4    75%    223    1825   9 %

HOST: r 5n1
============
/var                    8192    2448    5744    71%    286    1762   87%

HOST: r 5n13
============
/var                    8192    222     5972    73%    229    1819   89%

HOST: r 5n14dx
==============
/var                    8192    2 12    618     76%    221    1827    9 %

HOST: r 5n15
============
/var                    8192    2368    5824    72%    22     1828   9 %

HOST: r 5n16
============
/var                    8192    21      6 92    75%    225    1823   9 %
```

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 1
```

To display the filesystems on hosts **r05n16** and **r05n15** that have more than 90% of their space used:

```
pfck “-w r 5n15,r 5n16” -pu 9
Filesystem                Size-KB Used-KB Free-KB %Free   iUsed   iFree %iFree
==========                ======= ======= ======= =====   =====   ===== ======
HOST: r 5n15
============
/usr                      331776  32764   4136    2%      17673   66295   79%

HOST: r 5n16
============
/usr                      331776  32764   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: r 5n1
 /var                     335872  331412  446     2%      17819   66149   79%
HOST: r 5n13
 /var                     335872  331632  424     2%      1782    66148   79%
HOST: r 5n14dx
 /var                     335872  331388  4484    2%      17817   66151   79%
```

Chapter 5. 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

Important

AIX allows an "*" in the /**etc**/**password** file to prevent login to an ID. This is used for the **supman** ID. Do not use the **remove id** option of **pwdchk** as this removes the **supman** ID and stops file collections from working properly.

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.

© Copyright IBM Corp. 1996

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

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 45.

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 Parallel System Support Programs for AIX: 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 43 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 56 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 51 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 45.

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**
	node.root	sup.admin user.admin node.root power_system	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 58).

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.

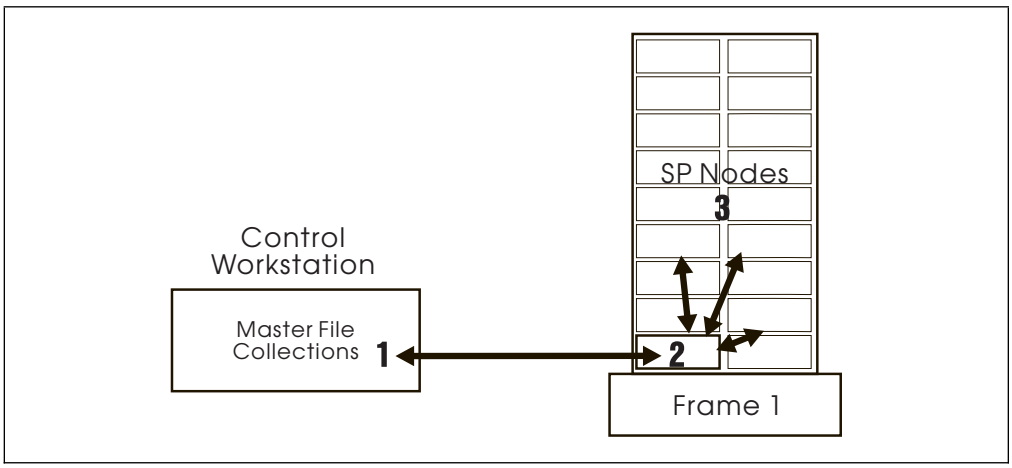


Figure 2. SP Default File Collection Hierarchy

The commands to update the file collections are set up in a **crontabs** file to run hourly in a staggered sequence. Figure 2 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 3 on page 46 shows the structure for this directory.

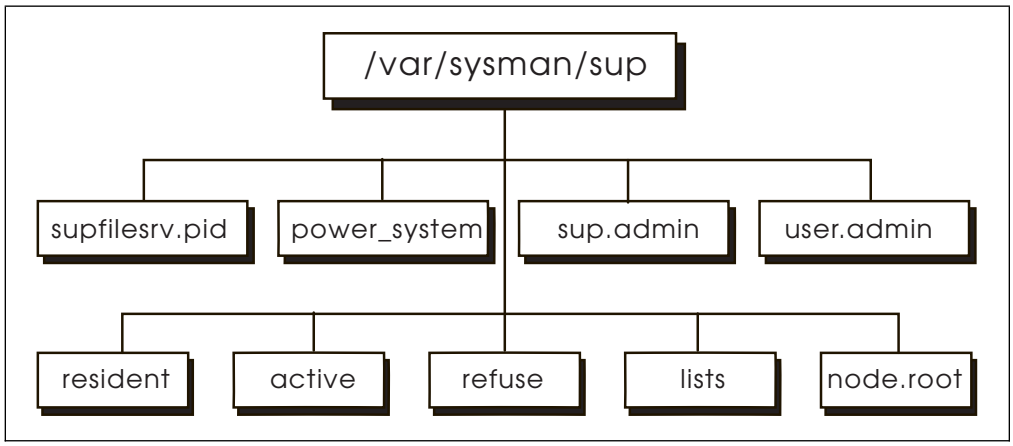


Figure 3. /var/sysman/sup Files and Directories

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 4 on page 47 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.

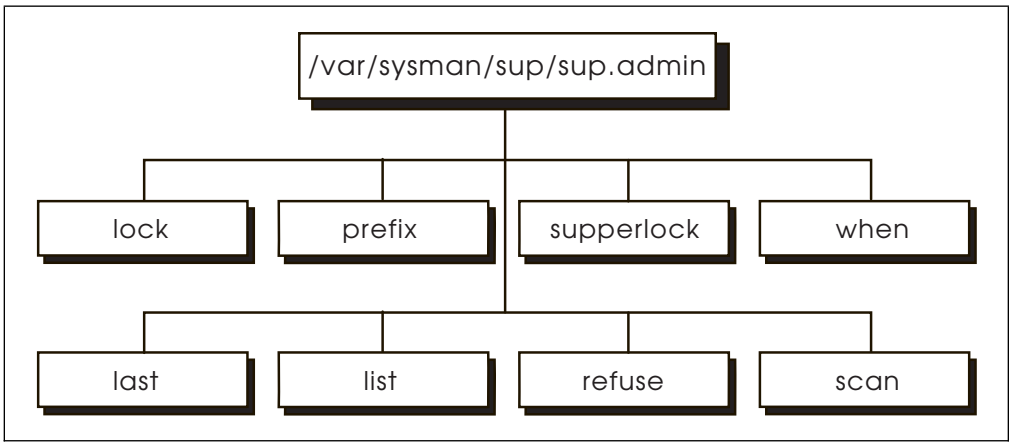


Figure 4. sup.admin Master Files

Following is an explanation of the master files shown in Figure 4.

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 64

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 147 for more information.

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 41 through 54.

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 61 through 63.

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 Parallel System Support Programs for AIX: 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
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

Use:	On:	To Report This Data:
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:

```
[r 5n 1][/]> 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:

```
[r 5n1 ][/]> 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:

```
[r 5n 1][/var/sysman/sup/node.root]> supper where
supper:   Collection node.root would be updated from server r 5cw.
supper:   Collection power_system would be updated from server r 5cw.
supper:   Collection sup.admin would be updated from server r 5cw.
supper:   Collection user.admin would be updated from server r 5cw.
```

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
24 755 74 845355 74 845355 .
1  644 74 76 294 74 76 294 .kshrc
1  644 74 76 294 74 76 294 .profile
24 755 74 845315 74 845315 etc
24 755 74 76 295 74 76 295 etc/amd
1  754 74 76 294 74 76 294 etc/amd/amd
1  754 74 76 295 74 76 295 etc/amd/amd.start
1  754 74 76 295 74 76 295 etc/amd/amq
1  754 74 76 295 74 76 295 etc/amd/refresh_amd
1  644 74 94355  74 94355  etc/motd
24 755 74 76 294 74 76 294 var
24 755 74 76 294 74 76 294 var/sysman
24 755 74 76 295 74 76 295 var/sysman/etc
1  754 74 76 295 74 76 295 var/sysman/etc/kernswitch
1  754 74 76 295 74 76 295 var/sysman/etc/ntpdate
1  754 74 76 295 74 76 295 var/sysman/etc/sup
1  754 74 76 295 74 76 295 var/sysman/etc/supfilesrv
1  754 74 76 295 74 76 295 var/sysman/etc/supscan
1  754 74 76 295 74 76 295 var/sysman/etc/xntpd
1  754 74 76 295 74 76 295 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 13, “Maintaining the crontabs File” on page 147 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 45 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.

   ```
   ==========================================================================
   [r 2n 7][/]> supper update node.root
   Updating collection node.root from server mrcs-r1.
   File Changes:    1 updated,   removed,   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:

```
[r 5n 1][/var/adm/SPlogs/filec]> supper update node.root
Updating collection node.root from server r 5cw.
File Changes:    2 updated,   removed,   errors.


[r 5n 1][/var/adm/SPlogs/filec]> supper rlog
SUP 7.24 (4.3 BSD) for file /tmp/.sf13689 at May    6 12: 1:57
SUP Upgrade of node.root at Fri May    6 12: 1:57 1994
SUP Fileserver 7.12 (4.3 BSD) 26295 on r 5cw.hpssl.kgn.ibm.com
SUP Locked collection node.root for exclusive access
SUP Requesting changes since Fri May    6 12:  : 7 1994
SUP Receiving file .kshrc
SUP Receiving file .profile
SUP Upgrade of node.root completed at Fri May    6 12: 1:58 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 56.

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 45 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 53.

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 41 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 64 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 RS/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
user.admin
power_system
node.root
tools
```

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:

```
/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, are user-created, 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 2 on page 45. 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 5 on page 62 shows a boot/install server that is offline for the **power_system** collection.

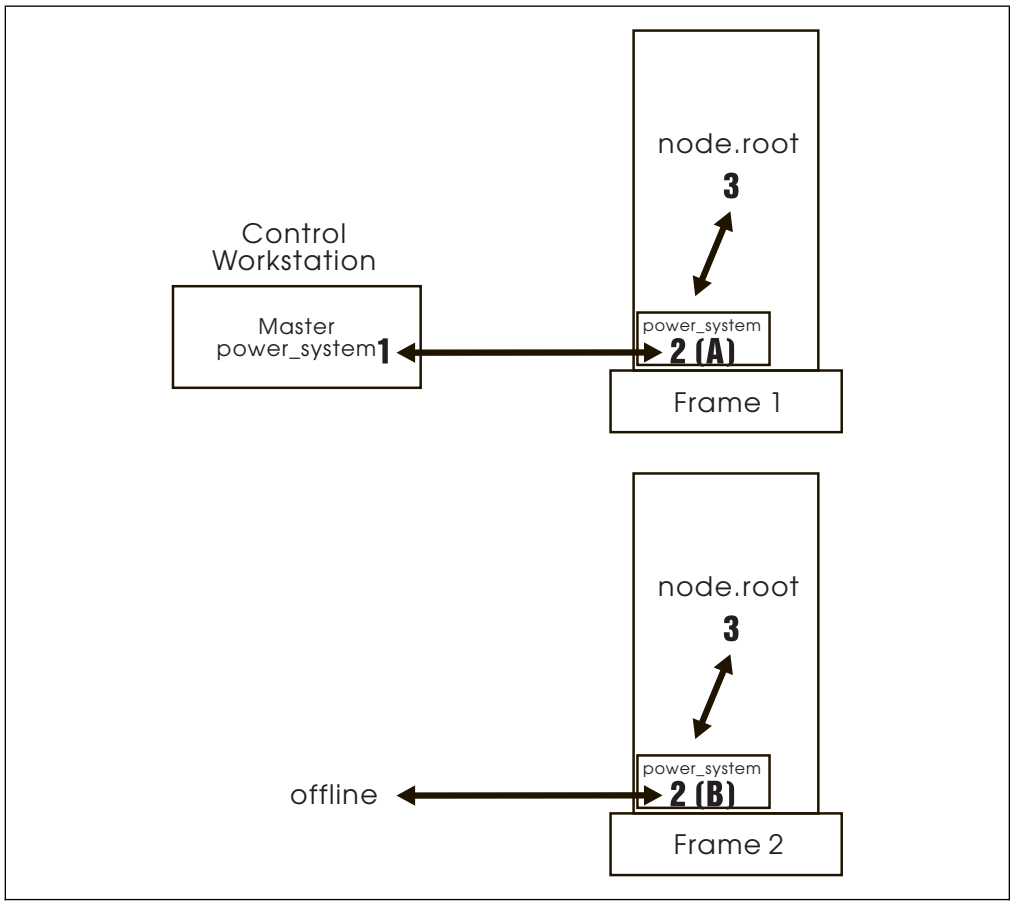


Figure 5. SP Hierarchy of an Offline Boot/install Server

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 5, 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 2 on page 45. 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:

   ```
   /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 51 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 45 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 Parallel System Support Programs for AIX: 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 6. 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 SP system references the NIS maps that are located on the NIS master machine. The NIS maps provide each user with a single image of their user account information in the NIS domain. This information includes the 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. As the system administrator, you need to create and maintain the NIS environment on the NIS master machine in the **/var/yp** directory. The SP User Management commands do not interface with the NIS environment.

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 44.

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.

© Copyright IBM Corp. 1996

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 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 Parallel System Support Programs for AIX: Installation and Migration Guide* contains detailed instructions for entering site environment information.

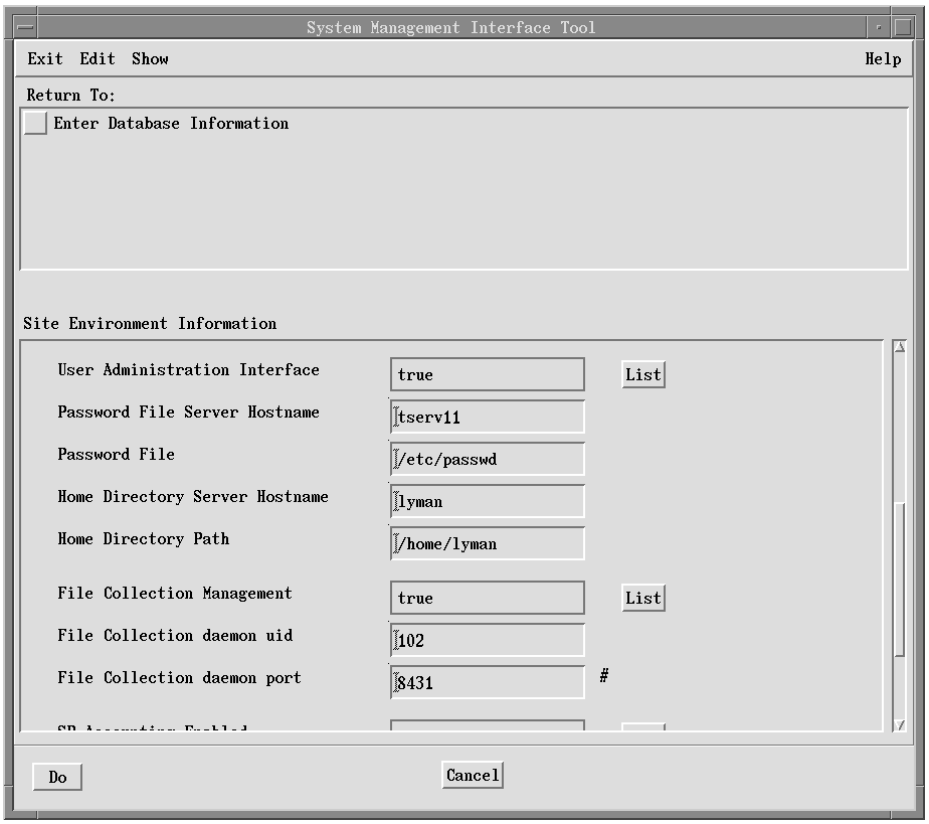


Figure 6. SMIT Panel for Site Environment Information

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 master password file resides. The initial value is the control workstation. The master password file location cannot be one of the nodes in the SP system.

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 14, "Security Features of the SP System" on page 151, 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 *RS/6000 SP Users*.
- Enter **smit**, select *RS/6000 SP System Administration*, and then select *RS/6000 SP Users*.

The examples in this section show the first two methods. You can find the complete syntax for the commands in the *IBM Parallel System Support Programs for AIX: 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 7. 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:

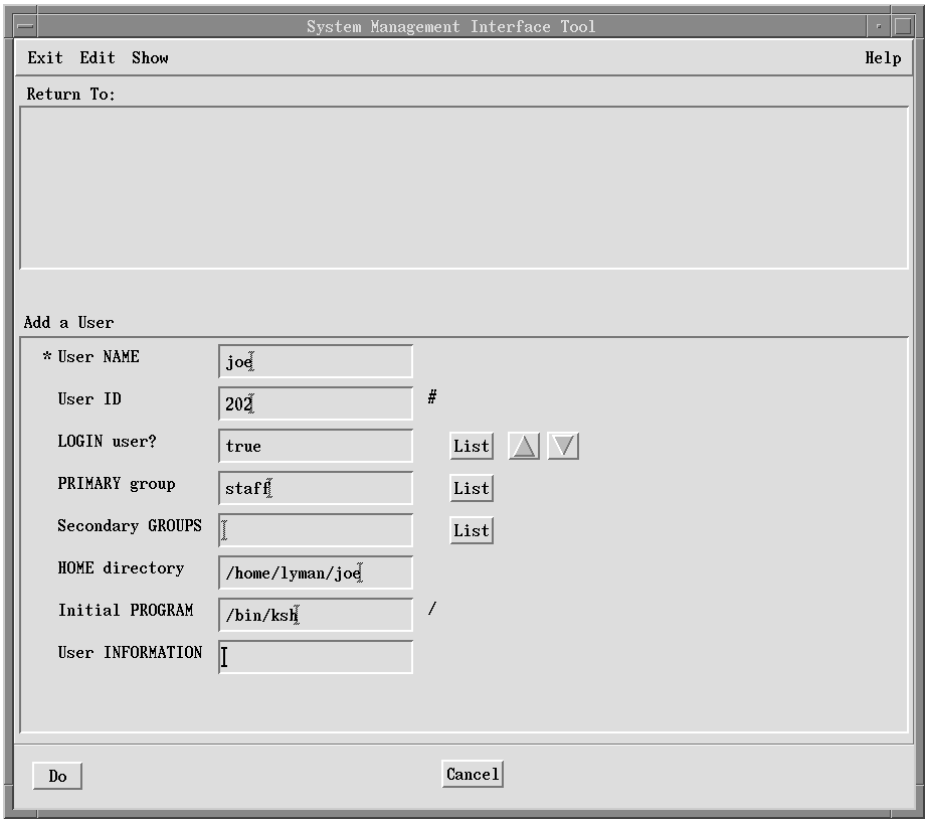


Figure 8. Example of Adding an SP User

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=2 2 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

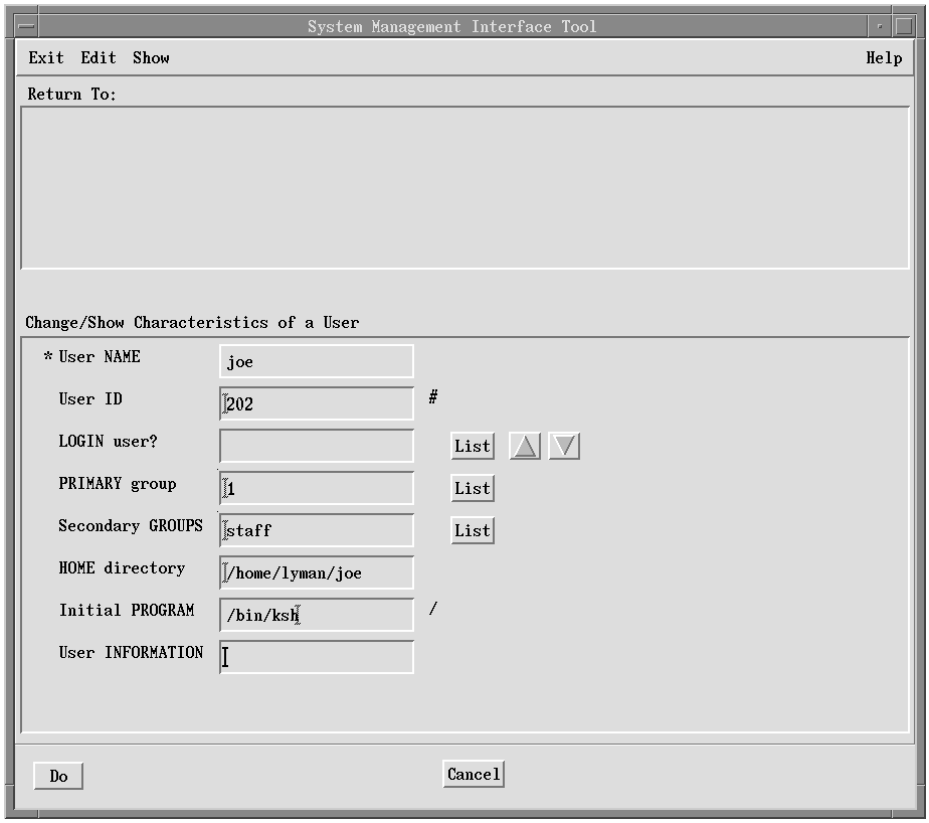


Figure 9. SMIT Panel for Changing SP User Information

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 10. 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 73.

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**.

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
   ```

   If you are using the AIX Common Destktop Environment (CDE) on the control workstation, you will also need to make a link from the restricted login script for CDE to the appropriate CDE directory:

   ```
   ln -s /usr/lpp/ssp/config/admin/cde_cw_restrict_login \
   /etc/dt/config/Xsession.d/cde_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.

If you are using CDE and wish to remove its restrictive login, simply remove the **cde_cw_restrict_login** link you created in the **/etc/dt/config/Xsession.d** directory.

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**

Changing Passwords when SP User Administration Interface is Configured

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**

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.

Important

Under certain circumstances, the user is prompted to change passwords during the login process. If this occurs, the user should log in to the machine serving the password file and make the change from there. If this is not done, the password change will be only local to the node; in addition, if File Collections is being used, the password change will be only temporary. To avoid this problem, communicate to users what machine to log in to when a password change is required during the login process. We recommend placing this information in the message of the day (MOTD).

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.

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.

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 processed 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 Parallel System Support Programs for AIX: 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.

Types of Access Disallowed

When **spacs_cntrl** block or deny is used, all of the following types of interactive access are disallowed:

login
rlogin
AIX **rsh**
AIX **rcp**
AIX **rexec**
SP **rsh**
SP **rcp**

Disallowing ftp on a Node

As system administrator, you can disallow users from using **ftp** on a node by placing users in the /**etc**/**ftpusers** file. This file can be kept in a file collection and distributed to the appropriate nodes. The **ftp** daemon can also be disabled to prevent all ftp access.

You can also use the **ruser** command to manage **ftp** usage by a user on a node:

ruser -a -f *username*

Using this command with **dsh** to the appropriate nodes allows manipulation of the /**etc**/**ftpusers** file on a per user basis.

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 475.

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 89.

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
   ```

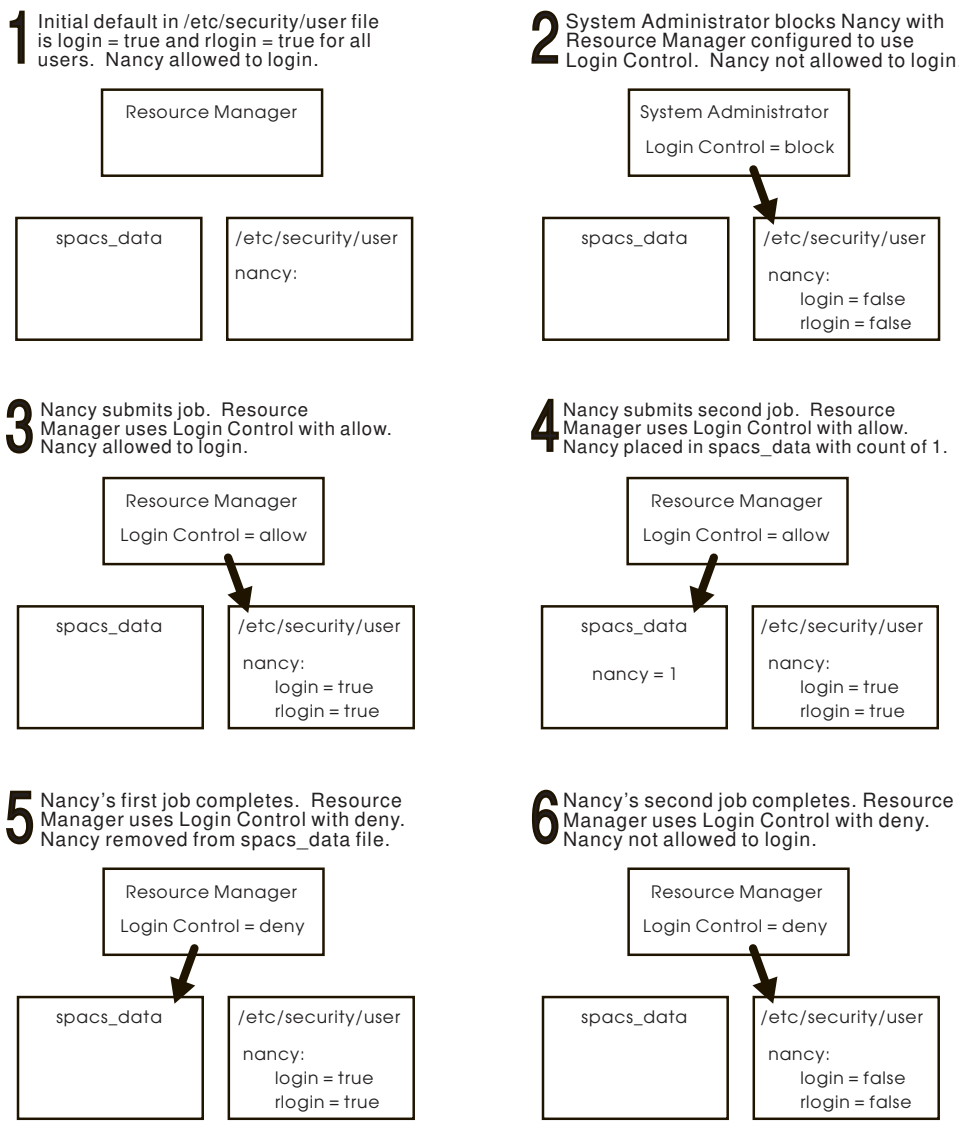
3. Issue login command to restrict interactive access for users to all the nodes.

```
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 1 :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.17  8.
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: .
spacs_cntrl:    CPFILE called for /var/spacs/spacs_data and /var/spacs/spacs_data.17  8.
spacs_cntrl:    ACS_FILE/unlock called for /var/spacs/spacs_data.lck.
"Tue Apr 26 1 :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.2 1 2.
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: .
spacs_cntrl:    CPFILE called for /var/spacs/spacs_data and /var/spacs/spacs_data.2 1 2.
spacs_cntrl:    ACS_FILE/unlock called for /var/spacs/spacs_data.lck.
```

Chapter 7. 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 134.

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.

Can select either the **en0** or **en1** adapter for Ethernet job application requests during initial configuration. The adapter type is considered to be static data and the Resource Manager must be stopped and restarted to change this information.

Allows applications to be run interactively or submitted via a batch scheduler (LoadLeveler only).

© Copyright IBM Corp. 1996

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.

Allows you to lock out users from nodes unless they request access via the Resource Manager. Refer to “Controlling User Login” on page 77 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 Parallel System Support Programs for AIX: Installation and Migration Guide*. (To check if **ssp.jm** has already been installed on the control workstation, enter **lslpp -m | grep ssp.jm**) The **ssp.jm** fileset depends on the installation of **ssp.basic** and

ssp.clients. The **libjm_client.a** library, **jm_client.shr.o** loadable object, and the **jm_status** command are now installed via the **ssp.clients** fileset. The **ssp.clients** fileset should be installed on client machines when access to the Resource Manager shared library and the **jm_status** command is desired on machines off of the SP system.

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 Parallel System Support Programs for AIX: Installation and Migration 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             4 86/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 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 Parallel System Support Programs for AIX: 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.

Note: 604 High Nodes cannot be configured as part of a parallel pool. Therefore, the Resource Manager will not allocate these nodes for parallel jobs.

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 91 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 Parallel Environment for AIX: Operation and Use*, SC28-1979 and SC28-1980, 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 *Using and Administering LoadLeveler*, SC23-3989.

Example Configuration

Figure 11 on page 90 shows 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.

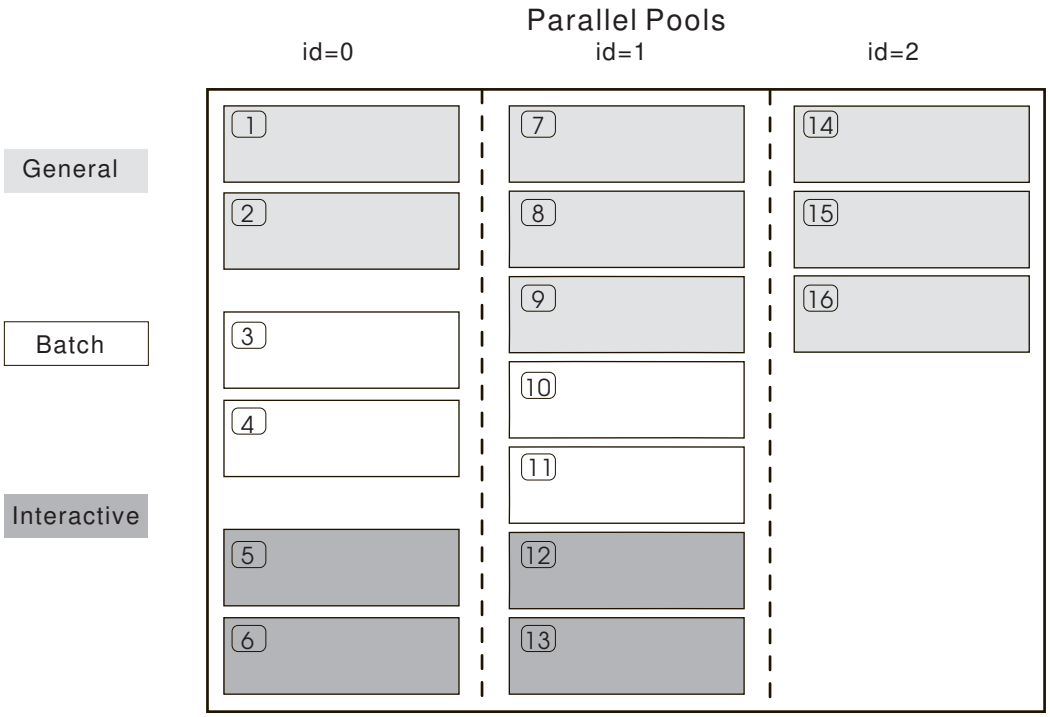


Figure 11. Example Resource Manager Configuration

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 77. (See **spacs_cntrl** in the *IBM Parallel System Support Programs for AIX: 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 System Data Repository

PRESS Enter

Using the Command Line:

Enter:

SDR_test

Refer to the *IBM Parallel System Support Programs for AIX: Command and Technical Reference* if verification test fails.

Step 3: Configure the Resource Manager

The Resource Manager keeps all general 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 88. 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 475 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 RS/6000 SP System Management

The RS/6000 SP System Management menu appears.

SELECT RS/6000 SP Resource Manager

The RS/6000 SP Resource Manager menu appears.

The fastpath invocation for the RS/6000 SP Resource Manager menu is:

smit RM_options

```
                    RS/6    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
Change/Show En Adapter
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 Change/Show En Adapter

The Change/Show En Adapter menu appears.

The fastpath invocation for the Change/Show En Adapter menu is:

smit en_adapter

```
                    Change/Show En Adapter

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

    En  Adapter                        [ ]           +
     Update Configuration Now ?    [no]          +
```

The default value of the En Adapter field will be **0** for **en0**.

Use the list to select **en0** or **en1** (if it exists).

Enter **no** for Update Configuration Now?

Update will take affect only during a **jm_start** and not a **jm_config**.

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 99 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 RS/6000 SP Resource Manager menu.

The fastpath invocation for the RS/6000 SP Resource Manager menu is:

smit RM_options

```
                    RS/6    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 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 Parallel System Support Programs for AIX: 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                         [r 7n 1  r 7n 3]
      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 RS/6000 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 RS/6000 SP Resource Manager menu. The fastpath invocation for the RS/6000 SP Resource Manager menu is:

```
smit RM_options
```

```
                    RS/6    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}
```

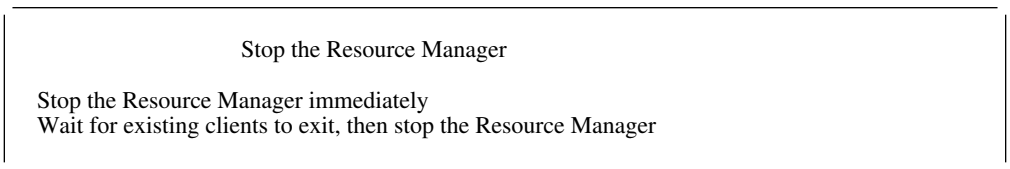
The q flag is optional. If not specified, the primary and backup Resource Manager server exit immediately.

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



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 8. Managing Time Synchronization

There are several ways you may currently be handling synchronizing time-of-day clocks on your control workstation and processor nodes. You may already be using Network Time Protocol (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.**

Managing NTP

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 3.

Table 3. 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

server Indicates a host from which you can request time.

© Copyright IBM Corp. 1996

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: 127.127.1.1	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 Parallel System Support Programs for AIX: Installation and Migration Guide*.

For directions on updating the site environment see the *IBM Parallel System Support Programs for AIX: Installation and Migration Guide*.

Time Setting Options

To change time zones, use the **chtz** *timezone_info* command. For example:

```
chtz EST5EDT
```

sets the time zone to Eastern Standard Time and 5 hours from Greenwich Mean Time and sets the machine to know about Daylight Savings Time.

Nodes and Control Workstations Out of Time Synchronization

Generally, time services will adjust clocks incrementally to keep in sync with a master. However, when time is off by a large amount, for example, an hour, it becomes difficult and time-consuming to incrementally synchronize with the time master.

To ensure that clocks stay in sync, set nodes and the control workstation to understand Daylight Savings Time. Refer to the **script.cust** sample file ("script.cust File" on page 505) for information on setting time zones.

Chapter 9. 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 16.12  -x all -l $amdlog -p >\
   /etc/amd/amd.pid @amdargs") && exit( );
   ```

2. Add the **-a** *DirectoryName* or the **-w** *wait-timeout* flags to the command. Enter the wait time in seconds.

© Copyright IBM Corp. 1996

3. If you are using file collections, 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 Parallel System Support Programs for AIX: Diagnosis and Messages Guide* for instructions) or reboot the nodes.

See Chapter 5, “Managing File Collections” on page 41 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 Parallel System Support Programs for AIX: 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 5, “Managing File Collections” on page 41 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 Parallel System Support Programs for AIX: 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 12.

```
#
# 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 12. 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 5, “Managing File Collections” on page 41.

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=4 96,wsize=4 96
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 44 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 10. 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-RS/6000 to serve as Print Host, see "Using a Non-AIX Workstation as Print Host" on page 118.)

The SP Print Management System supports a subset of print commands and options listed in "Supported Commands" on page 113. 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 115 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. 1996

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 135).

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 RS/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 RS/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-RS/6000 for a Print Host, see “Using a Non-AIX Workstation as Print Host” on page 118.

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 14, “Security Features of the SP System” on page 151.

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.

Multi-Named Host Restrictions

The Print Management System does not support use of multiple host names for the Print Host. The Print Host needs to be defined and used with only one (usually the primary) host name. Use the chosen host name for the Print Host to avoid an infinite loop.

The Print Management System uses the host name to decide whether it runs on a Print Host or a non-Print Host. If running on a Print Host, the Print Management System calls the AIX print commands directly. A loop can occur when a user uses the alternate host name for a Print Host and the PRINT_HOST variable on the Print Host is set to the primary host name. Because the alternate host name and the host name in PRINT_HOST do not match, the Print Management System assumes the request is for a remote host, and issues the command to the alternate host (really one and the same host), causing an infinite loop.

To avoid such an infinite loop, do one of the following:

Unconfigure the Print Management System on the multi-named Print Host.

Choose a host name, set PRINT_HOST to that host name, and have all users use the chosen host name when submitting print jobs.

Using a Non SP System Host for Print Host with Kerberos

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-RS/6000 to serve as Print Host, see "Using a Non-AIX Workstation as Print Host" on page 118.)

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 114.
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 RS/6000 to see if they apply.

Chapter 11. 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 105 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 RS/6000 architecture. If your environment includes systems with other architectures, the procedures may require modification.

Related Information

Refer to Chapter 5, "Managing File Collections" on page 41 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 RS/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. 1996

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 a 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 13 on page 121 shows how LAN and local mail are routed in a DNS configuration, assuming Amd is running.

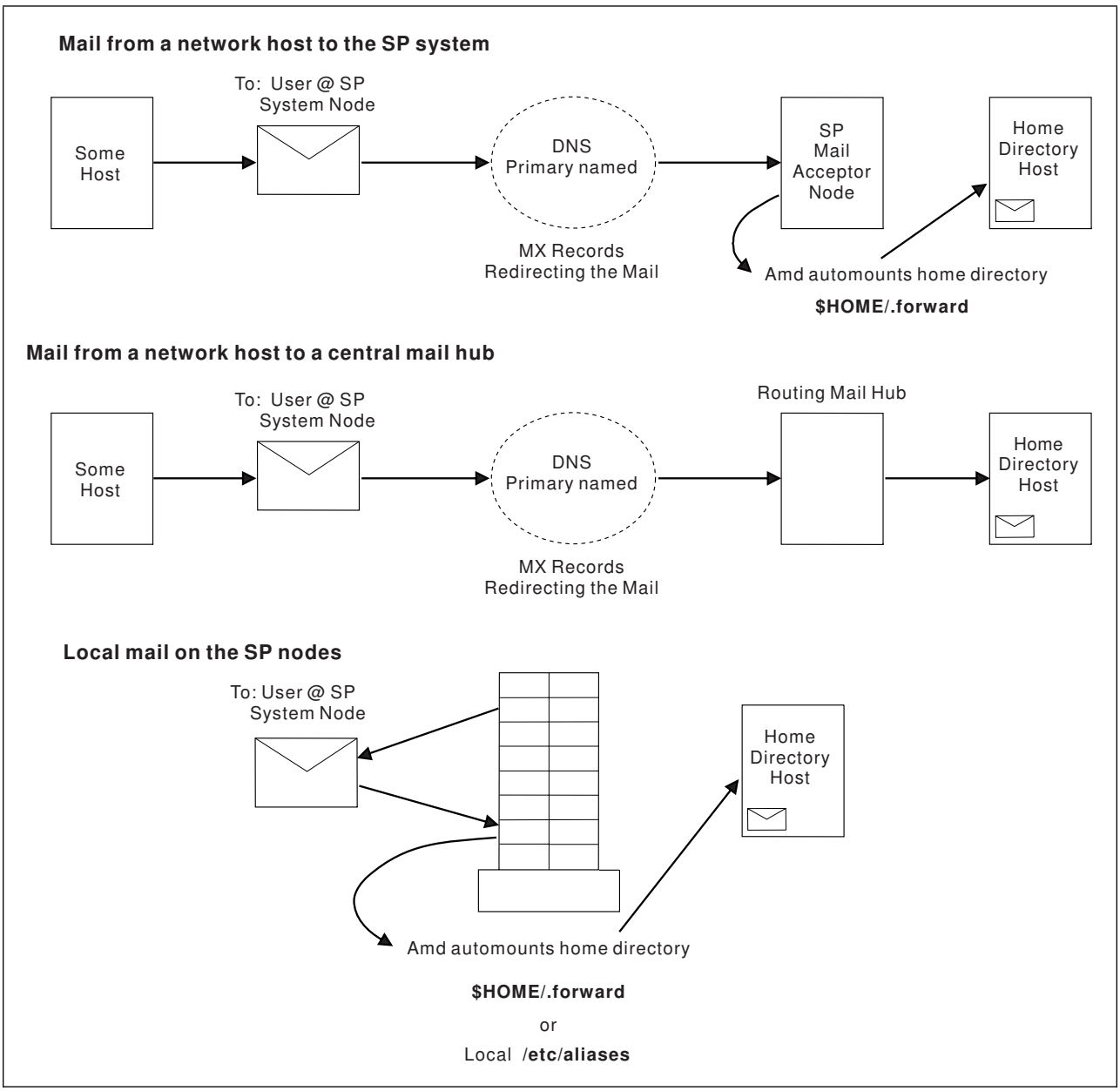


Figure 13. Option 1: Using DNS

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 1 *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  1  spnode1.kgn.ibm.com.
   spnode2.kgn.ibm.com.  IN  MX  1  spnode1.kgn.ibm.com.
   spnode1.kgn.ibm.com.  IN  MX  1  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  123.456.678.9
   ```

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 131 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 14 on page 126 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

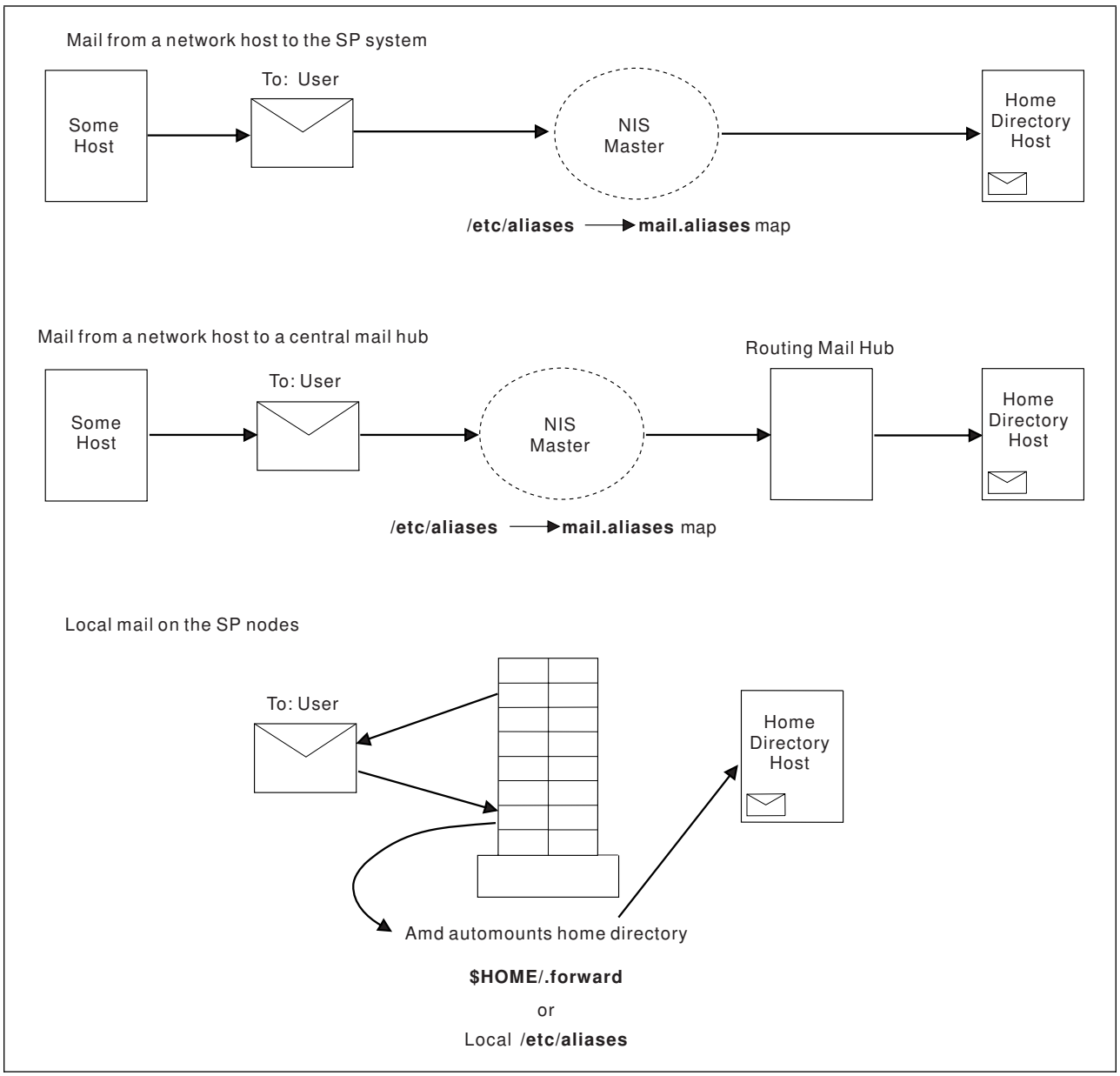


Figure 14. Option 2: Using NIS

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 124 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 RS/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 15 on page 129 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 RS/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:

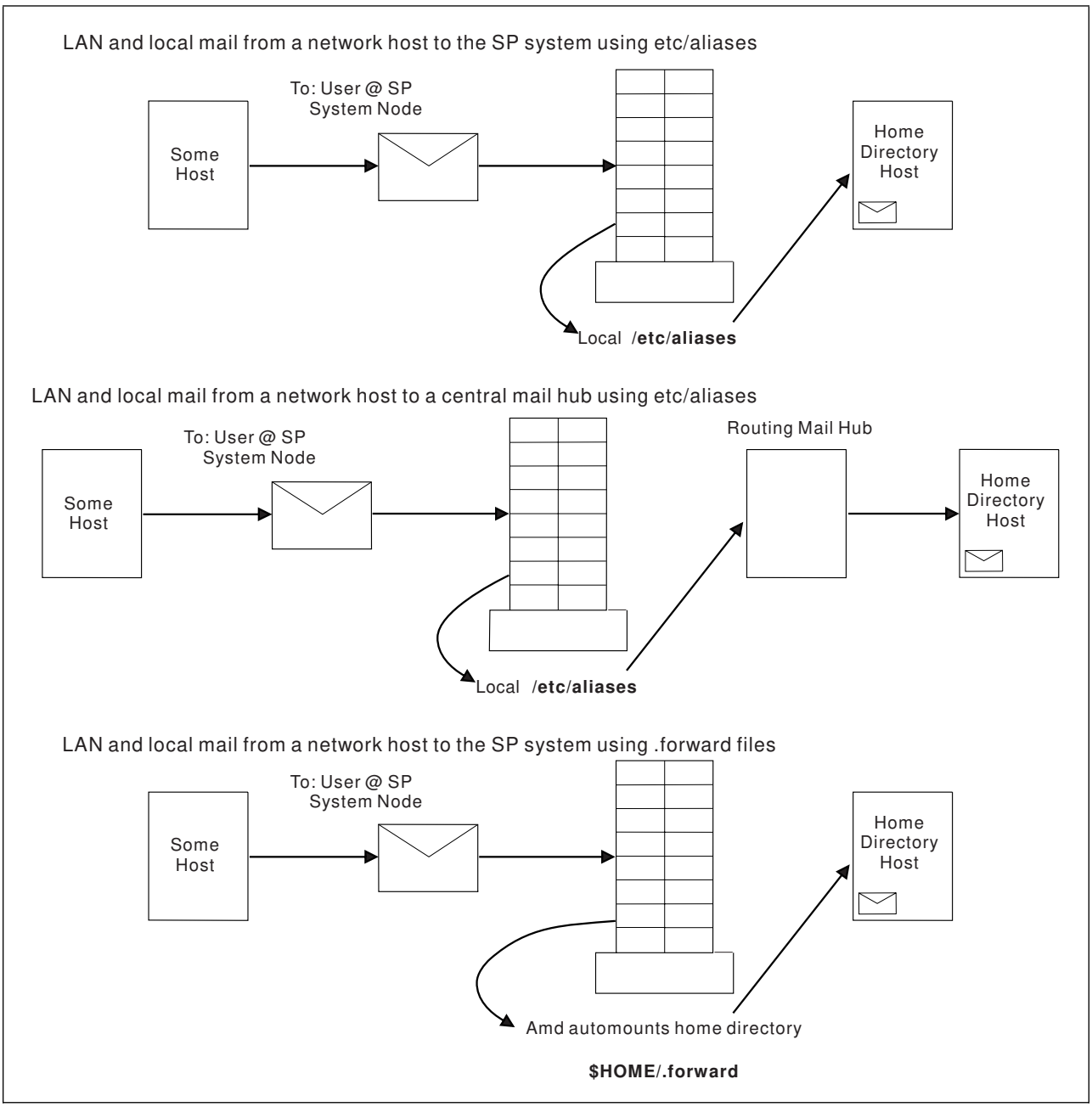


Figure 15. Option 3: Using /etc/hosts

```
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 124 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 5 and refer to the directions in "Adding and Deleting Files in a File Collection" on page 54.

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 12. 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. 1996

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 user exit 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 RS/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 96.

Using SMIT:

TYPE **smit enter_data**

The Enter Database Information Window appears.

SELECT Site Environment Information

The following window appears:

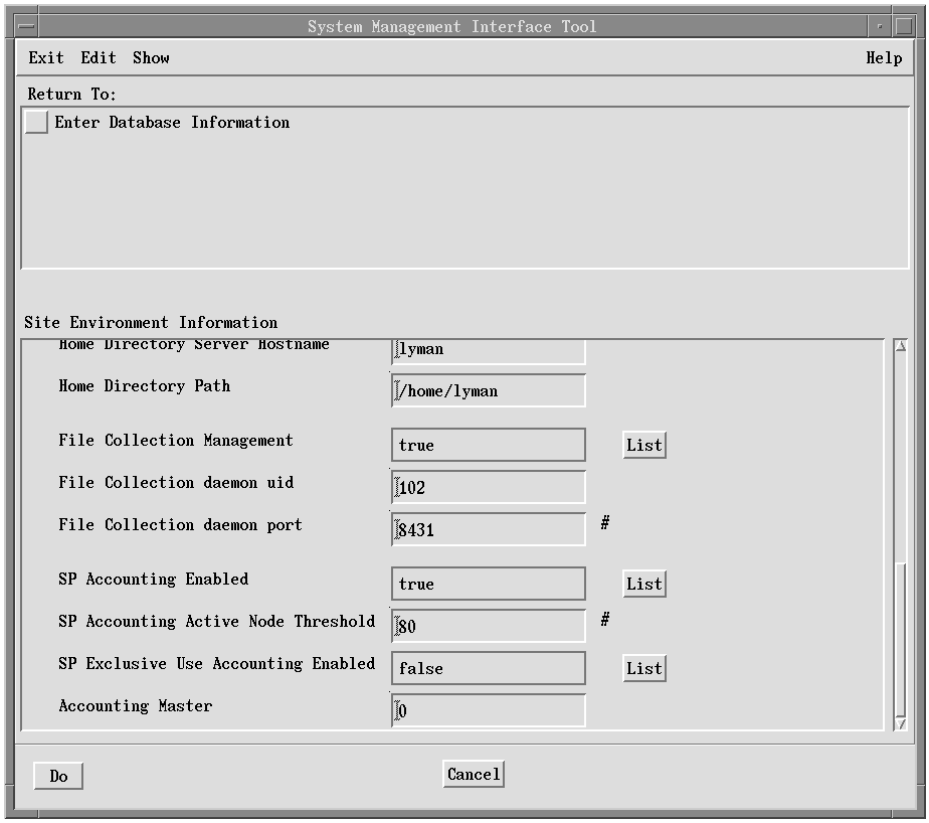


Figure 16. Accounting Fields on the Site Environment SMIT Panel

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 17. 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 6 .  -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 Parallel System Support Programs for AIX: Installation and Migration 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 142.
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      1    4
CHKPACCT 5
NRUNACCT   2    1-6
CRUNACCT   4    1-6
CMONACCT 15 5 1
```

Figure 18. 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 13, “Maintaining the crontabs File” on page 147 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 13. 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 entries 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:

```
          /usr/lpp/ssp/bin/dev/null 2>/dev/null
          /usr/lpp/ssp/bin/cleanup.logs.ws
 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 61 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. 1996

Security on the SP System

© Copyright IBM Corp. 1996

Chapter 14. Security Features of the SP System

This chapter includes information on the security services provided with the IBM RS/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 160 describes the tasks involved in running authenticated client-server applications supplied with the SP system.

"Administering SP Security Services" on page 166 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 182 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 Parallel System Support Programs for AIX: Installation and Migration Guide*. For information about diagnosing authentication problems, refer to the *IBM Parallel System Support Programs for AIX: 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. 1996

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 19 on page 155.

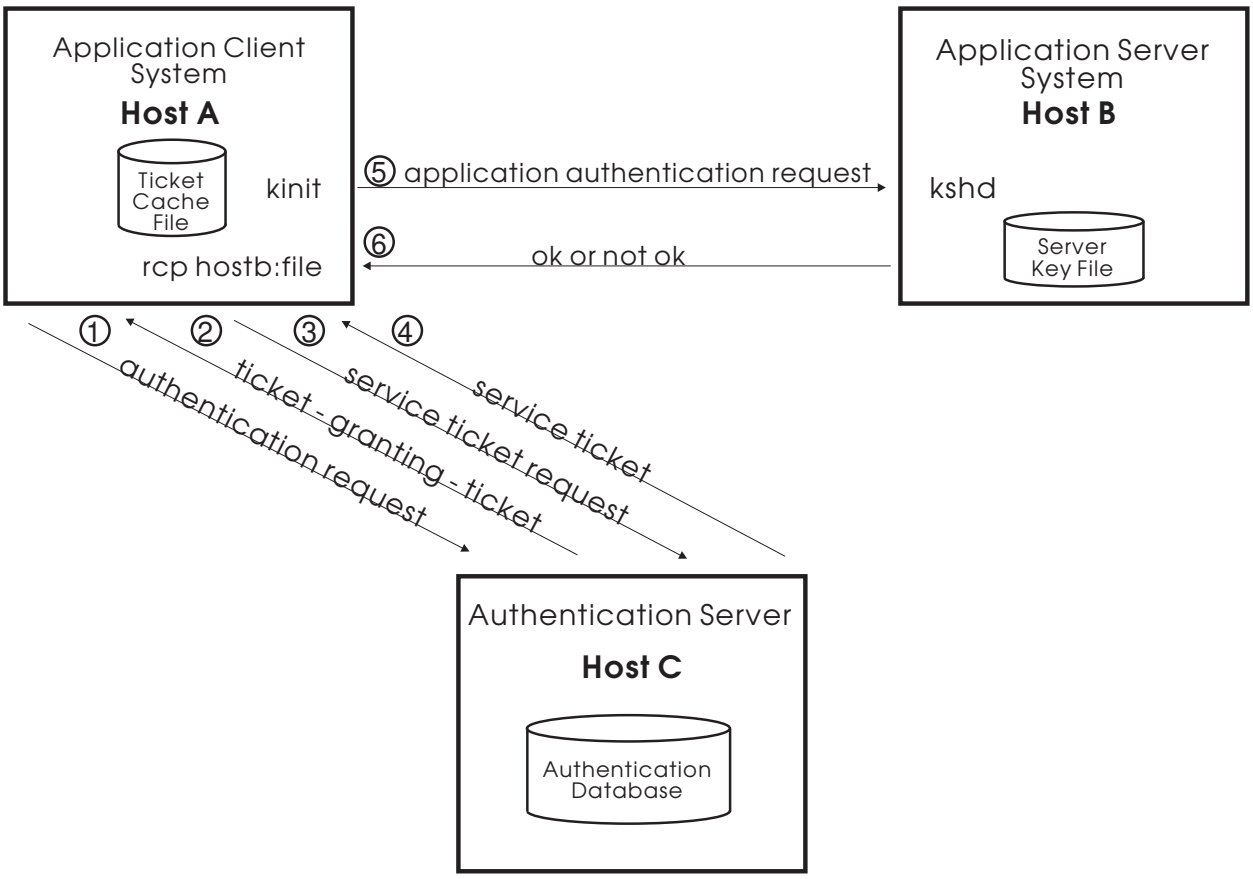


Figure 19. An Overview of Kerberos Authentication

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 Parallel System Support Programs for AIX: 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 RS/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

By convention, a Kerberos Version 4 server runs as a Kerberos principal whose identifier is formed from the service name and the short hostname of the node on which the server runs, converted to lowercase. 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 RS/6000 workstations. Therefore, for each (short) network interface name on these workstations, a service principal is created with the name **hardmon** and the network name as the instance.

The remote commands can be run from, or to, any IBM RS/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 **rcmd** 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 RS/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 Parallel System Support Programs for AIX: 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:

-1	: 5	12	1:	24	2:	36	3:	48	4:
6	5:	72	6:	84	7:	96	8:	1 8	9:
12	1 :	125	1 :25	126	1 :3	127	1 :35	128	1 :4

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

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

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. Another alternative for frequent users of Kerberos authenticated management tools is to have the System Administrator set up Kerberos authentication as an authentication method to be invoked automatically at AIX login. For more information on this option, see "Setting Up a User Account to Obtain Kerberos Credentials at AIX Login" on page 170.

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**
25 2- 3 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 Parallel System Support Programs for AIX: 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 RS/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:
```

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  9: 2:42   Sep 14  9: 2:42   rcmd.spnode3@XYZ.ABC.COM
Aug 15  9: 2:43   Sep 14  9: 2:43   rcmd.spnode5@XYZ.ABC.COM
Aug 15  9: 2:43   Sep 14  9: 2:43   rcmd.spnode6@XYZ.ABC.COM
```

Refer to the **klist** command in the *IBM Parallel System Support Programs for AIX: 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  3:24]
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 19:5 ]
    --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 Parallel System Support Programs for AIX: 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 167 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 Parallel System Support Programs for AIX: 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 5, “Managing File Collections” on page 41. Refer to the *IBM Parallel System Support Programs for AIX: 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 later.

Once you have installed and initialized authentication on your SP system, you might want to add other IBM RS/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. If you are not using AFS Version 3.4 authentication servers, you should only have to modify **/etc/services** if you are using some other service that uses one or more of the traditionally used (but not formally reserved) Kerberos 4 ports. They are:

Service name: kerberos4 Port: 750/udp

Service name: kerberos_admin Port: 751/tcp

Service name: krb_prop Port: 754/tcp

You will also have to modify /**etc**/**services** if you are using AFS Version 3.4 authentication servers. The **kaserver** in AFS Version 3.4 for AIX 4.1 accepts Kerberos Version 4 protocol requests using the well-defined udp port assigned to the “kerberos” service assigned to port 88 in the file distributed with the base operating system. MIT Kerberos Version 4, on which PSSP authentication services are based, uses a default port number of 750. PSSP authentication commands use the service name “kerberos4” to avoid this conflict with the Kerberos 5 service name, which is also used by DCE. For PSSP authentication commands to communicate with an AFS 3.4 **kaserver**, you must do either of the following:

1. Stop the **kaserver**, redefine the udp port number for the “kerberos” service to 750 on the server system, then restart the **kaserver**.
2. Add a statement to /**etc**/**services** that defines the udp port for the “kerberos4” service as 88 on the SP control workstation and on any other independent workstation that will be a client system for PSSP authenticated services.

/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 Parallel System Support Programs for AIX: 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 a User Account to Obtain Kerberos Credentials at AIX Login

As System Administrator, you can make using the RS/6000 SP authenticated services easier for nonroot users by setting up their user accounts so that Kerberos credentials are obtained whenever they login to the system. To do this, you must:

1. Add a stanza to the **/etc/security/login.cfg** file that names a program that will perform the required **kinit** processing on the user's behalf. The name of the method (shown in the following example as KERBEROS) can be anything you want. The rest of the stanza should be entered exactly as shown:

   ```
   KERBEROS:
       program = "/usr/bin/ksh /usr/kerberos/etc/Kinit"
   ```

2. Use SMIT to update either the **auth1** or **auth2** attribute in the user account for each user you want to use this capability. Add this method after the default **SYSTEM** method in the **auth1** attribute, if you want to require the user to obtain Kerberos credentials in order to successfully login to the system. Specify this method separately in the **auth2** attribute if you want to allow (but not require) the user to obtain Kerberos credentials at AIX login. The *Kerberos-principal* must be the user's AIX login name, optionally followed by a period and an instance (for example, **foo** or **foo.admin**). Note the placement of the comma (,) and the semicolon (;) in this rather unusual syntax.

   The following is an example of an auth1 attribute for AIX user **foo**. When **foo** logs in, the **kinit foo** command will be executed after the AIX password for principal **foo**. If **kinit** fails, **foo** is not logged into the system.

   ```
   auth1 = "SYSTEM,KERBEROS;foo"
   ```

   The following is an example of an **auth2** attribute for AIX user **bar**. When **bar** logs in, the **kinit bar.admin** command will be executed after the AIX password is validated. The user is prompted for the Kerberos password for **bar.admin**. In this case, if **kinit** fails, **bar** remains logged into the AIX system, but has no Kerberos credentials.

   ```
   auth2 = "KERBEROS;bar.admin"
   ```

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 RS/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 Parallel System Support Programs for AIX: Installation and Migration 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 Parallel System Support Programs for AIX: Installation and Migration 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:   1999

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

Using the database backup file:

```
grep "^rcmd node3sw" /var/kerberos/database/slavesave
rcmd node3sw 255 1 2   a49bf286 d45c656  2    1 1 459 1995 33 15 2 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 ( ) cksum is 1  4957312, last cpw: Mon May 22 1 :12:5  1995
  password will never expire.
  An unlimited number of unsuccessful authentications is permitted.
  entry never expires.    Max ticket lifetime 25.   hours.
  last mod on Mon May 22 1 :12:5  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 Parallel System Support Programs for AIX: 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=$(umask); umask  77; INITIALPW=changeme$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 Parallel System Support Programs for AIX: 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 Parallel System Support Programs for AIX: 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 Parallel System Support Programs for AIX: 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- 9- 1 ] ?
Max ticket lifetime [ 255 ] ? 171
Attributes [   ] ?
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:   -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.cwken @XYZ.ABC.COM
   1        hardmon.cwken @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             cwken             XYZ.ABC.COM  1
hardmon          cwken             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.n 8tr @XYZ.ABC.COM
1       rcmd.n 8tr1@XYZ.ABC.COM
1       rcmd.n 8fddi@XYZ.ABC.COM
1       rcmd.n 8css @XYZ.ABC.COM
1       rcmd.n 8en @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 Parallel System Support Programs for AIX: Command and Technical Reference* for more information. Also, refer to the *IBM Parallel System Support Programs for AIX: 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 179 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 15. 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. 1996

Figure 20 on page 186 illustrates the way in which the Sysctl elements work together.

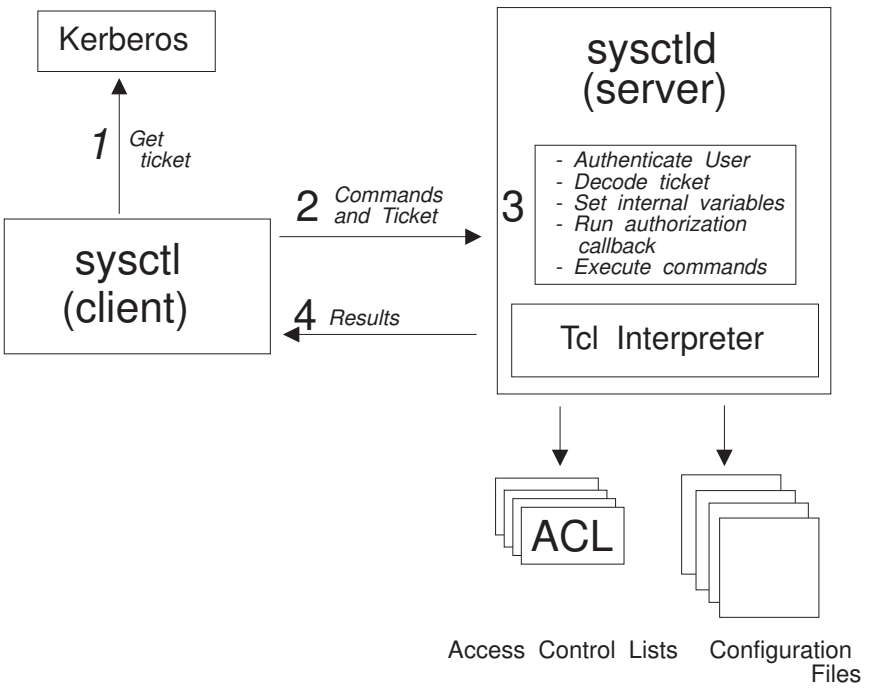


Figure 20. Sysctl Elements

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 14, "Security Features of the SP System" on page 151 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 206.

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 14, "Security Features of the SP System" on page 151).

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 190 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 5, “Managing File Collections” on page 41.

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 Parallel System Support Programs for AIX: Installation and Migration 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 Parallel System Support Programs for AIX: Installation and Migration Guide*).

Note: Some of these steps require administrative access to the authentication database. See Chapter 14, “Security Features of the SP System” on page 151 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      668 /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 7 on page 206.

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 198 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 Parallel System Support Programs for AIX: 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 Parallel System Support Programs for AIX: 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 198 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 190).

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    3 28   37%     815    1233    61%
/var                      16384    4176   122 8   75%     343    3753    92%
/usr                      55296  467888   85 72   16%  232 3  116 61     84%
/tmp                      49152    2386   25292   52%     235   12 53    99%
/home                     65536   63556     198    4%    5375   11  9    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:

```
>> r 5n13
r 5n13.hpssl.kgn.ibm----------------------------------------------------------
Filesystem                Size-KB Used-KB Free-KB %Free   iUsed   iFree %iFree
/                           4 96     3372      724   18%     792     232    23%
/var                        8192      632     1872   23%     297    1751    86%
/usr                      335872   331632      424    2%    1782   66148    79%
/tmp                        8192       52     7672   94%      67    1981    97%
/home                       4 96      168     3928   96%      18    1  6    99%
<<
>> r 5n15
r 5n15.hpssl.kgn.ibm----------------------------------------------------------
Filesystem                Size-KB Used-KB Free-KB %Free   iUsed   iFree %iFree
/                           4 96     3364      732   18%     796     228    23%
/var                        8192     3 28     5164   64%     281    1767    87%
/usr                      335872   331412      446    2%   17825   66143    79%
/tmp                        8192      464     7728   95%      57    1991    98%
/home                       4 96      168     3928   96%      18    1  6    99%
<<
>> r 5n 9
r 5n 9.hpssl.kgn.ibm----------------------------------------------------------
Filesystem                Size-KB Used-KB Free-KB %Free   iUsed   iFree %iFree
/                           4 96      334      756   19%      79     234    23%
/var                        8192     3336     4856   6 %     285    1763    87%
/usr                      331776   331 12      764    1%   17742   66226    79%
/tmp                        8192     1212      698   86%      58     199    98%
/home                       4 96      164     3932   96%      17    1  7    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 201 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[ ]";
```

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)
```

```
| doadminscript 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 4 lists the authorization variables.

Table 4. 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 r 5n11 -r /examples/sysctl/script1
```

sarah receives the following output:

```
I am r 5n11.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    3 28   37%     815    1233    61%
/var                        16384    4184     122   75%     343    3753    92%
/usr                        55296  467888   85 72   16%  232 3  116 61     84%
/tmp                        49152   23868   25284   52%     239   12 49     99%
/home                       65536   63556     198    4%    5375   11  9    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 $sbuf(used). /$sbuf(size).  1  ]
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 8
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 8
<<
>> 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 r 5n11 -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 $argc
for {set i  } {$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 #  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 5 shows the commands available in the Tcl language interpreter and the Sysctl base authorizations required to use them.

Table 5. 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 6 shows the commands available in the TclX language interpreter and the Sysctl base authorizations required to use them.

Table 6. 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 7. 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 8. 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 9. 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 Sets of Nodes

© Copyright IBM Corp. 1996

Chapter 16. Using a Switch

The optional Scalable POWERparallel Switch (SP Switch) and the High Performance Switch offered with the SP system provides high bandwidth, low latency, location-independent communication between nodes. The Scalable POWERparallel Switch adapter is the SP Switch Adapter. The High Performance Switch adapter is the High Performance Switch Adapter-2. This chapter presents information on:

- Switch communication modes
- Primary node takeover
- System partitioning and the switch
- Selecting switch clocks
- Selecting the primary and primary backup node
- Managing the switch topology file
- Starting the switch
- Fencing and unfencing nodes from the switch
- Preparing for system repartitioning
- Monitoring nodes on a switch
- Global shutdowns and reboots of nodes with a switch
- NFS mounts over a switch

Usage Note

Throughout this chapter, use of the term SP Switch also applies to the SP Switch-8 unless specifically stated otherwise. Differences will be noticed in discussions of system partitioning, switch node numbering, and cabling.

Switch Communication Modes

Two communication protocols are offered for the adapter:

1. AIX sockets using standard IP protocol
2. Dedicated user space access (via message passing libraries) and multiuser environments (via IP)

Both modes can be used concurrently, as if the adapter was 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 Parallel System Support Programs for AIX: Installation and Migration Guide*). When you assign the IP address range dynamically using ARP, there are no restrictions.

Any application can communicate over the switch by opening a standard 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 switch. With the possible exception of applications that depend on

© Copyright IBM Corp. 1996

network-specific functions (LAN broadcasts, for example), most applications work over the High Performance Switch without modification.

Notes:

1. The **iptrace** command supports both switch adapters and provides interface-level packet tracing for incoming and outgoing packets.
2. The standard AIX commands **ifconfig** and **arp** have been modified to support the switch adapters. For further information, refer to the *IBM Parallel System Support Programs for AIX: Command and Technical Reference*.
3. Static routes associated with the 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 switch is not available during the phase of system initialization 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.

Both the SP Switch adapter and the High Performance Switch Adapter-2 provide 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).

Primary Node Takeover

Error recovery on the SP Switch is done locally instead of globally as on the High Performance Switch. On the High Performance Switch, a fault on a link would cause the entire fabric to be interrupted while recovery was performed. On the SP Switch, detection and fault isolation is done at the link level while normal traffic flows across the rest of the fabric. Detected faults are forwarded to the active primary node for analysis and handling. When the primary node completes assessing the fault, the remaining nodes on the fabric are nondisruptively informed of status changes.

The primary node initializes the switch fabric and processes switch operations and errors reported from the switch fabric. The primary node is no longer a single point of failure on an SP system containing an SP Switch.

A new node type has been developed called the primary backup node. The primary backup node passively listens for activity from the primary node. When the primary backup node detects that it has not been contacted by the primary node for approximately 7 minutes, it assumes the role of the primary node. This takeover involves nondisruptively reinitializing the switch fabric, selecting another primary backup, and updating the System Data Repository (SDR).

The primary node also watches over the primary backup node. If the primary node detects that the primary backup node can no longer be contacted on the switch fabric, it selects a new primary backup node. The criteria to select a new primary backup node follows:

1. First, select a node on a switch assembly other than the switch assembly to which the primary node is attached.
2. Second, if no other switch assembly exists, select a node attached to a switch chip other than the one to which the primary node is attached.
3. If no other switch chip exists, select any available node on the switch chip to which the primary node is attached.

During the period of time between the failure or loss of the primary node and the takeover by the primary backup node, the switch fabric continues to function. Secondary failures are detected during the reinitialization of the switch fabric.

System Partitioning and the Switch

As discussed in Chapter 17, “Managing System Partitions” on page 229, the SP system can be organized into groups of nodes for various purposes. System partitioning has the following implications for the switch:

Commands related to the 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
- **Equiesce** - Operates only within the current system paritition
- **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
- **Eunpartition** - Operates only within the current system partition

Switch faults and message traffic are contained *within a system partition*.

The subset of the switch defined by the topology file for the system partition is viewed as a logical switch.

There is no connectivity over the switch *between system partitions*. (A gateway node with routing set up to the 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 switch clock source, regardless of system partitioning.

There is one primary node per system partition.

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 RS/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 Parallel System Support Programs for AIX: Command and Technical Reference* for more information.

Selecting the Switch Clock Source

To set the switch clock source using SMIT:

TYPE **smit**

The System Management menu appears

SELECT RS/6000 SP Systems Management

The RS/6000 SP Systems Management menu appears.

SELECT RS/6000 SP Cluster Management

The RS/6000 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
  Automatic selection of Switch Clock Settings
```

Figure 21. 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

Selecting the Primary and Primary Backup Nodes

High Performance Switch Systems:

On High Performance Switch systems, select one node to be the switch primary node. This node will be used as the primary node on the next **Estart**.

If the primary node is unavailable (powered-down, rebooted, and so forth), run **Eprimary** 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.)

Note: Whenever a new switch primary is assigned, the old primary must be rebooted or powered off.

The following scenario describes how to select a primary node on a High Performance Switch system.

The initial value for the primary node defaults to 1 upon powering up the system. This value is placed in the SDR, but there is no active primary node until **Estart** is executed.

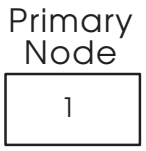


Figure 22. Initial Value

You can change the primary node value in the SDR by executing **Eprimary**. The specified node value is placed in the SDR, but there is still no active primary node.

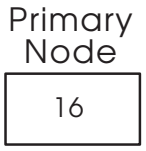


Figure 23. Eprimary Value

When **Estart** is executed, the primary node specified in the SDR becomes the active primary node.

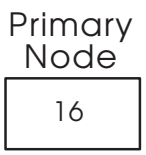


Figure 24. Estart Value

SP Switch Systems:

On SP Switch systems, select two nodes, one to be used as the primary node and the other to be used as the primary backup node. Until the next **Estart** is issued, these nodes are reflected as the oncoming primary and oncoming primary backup nodes (**Eprimary**). Once **Estart** completes, the primary field is updated based on the oncoming primary field and the backup field is updated based on the oncoming backup field.

The criteria to select a new primary backup node follows:

1. First, select a node on a switch assembly other than the switch assembly to which the primary node is attached.
2. Second, if no other switch assembly exists, select a node attached to a switch chip other than the one to which the primary node is attached.
3. If no other switch chip exists, select any available node on the switch chip to which the primary node is attached.

The primary and primary backup fields are updated automatically when primary node and primary backup node failures occur. Therefore, the primary and primary backup fields reflect the state of the current system.

The following scenario describes how to select a primary backup node on an SP Switch system.

Initially, the primary node and backup primary node fields are blank. In a 16-node system, the oncoming primary node field has the default value of 1 and the oncoming primary backup has the default value of 16.

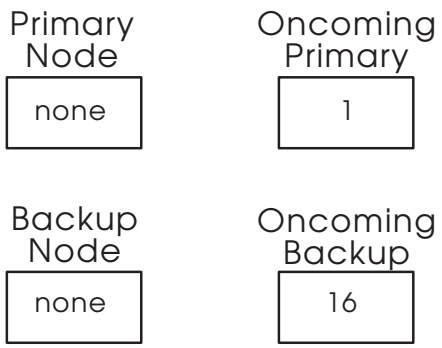


Figure 25. Initial Values

When **Estart** is executed, the node specified in the oncoming primary field becomes the primary node. The node specified in the oncoming primary backup becomes the primary backup.

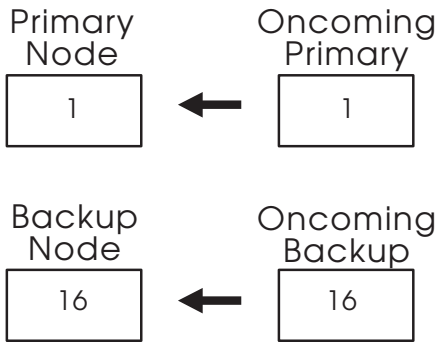


Figure 26. Estart Values

If the primary backup node fails, the primary node automatically selects a replacement.

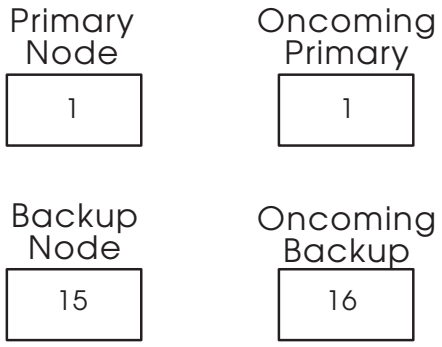


Figure 27. Backup Takeover Values

If the primary node fails, the primary backup node automatically becomes the primary node and a new primary backup is selected.

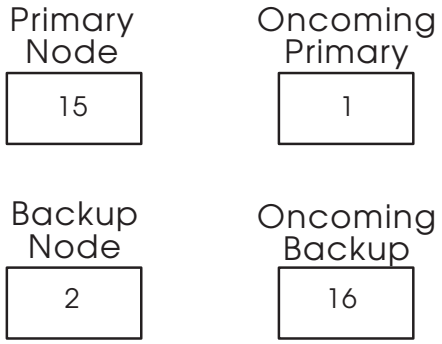


Figure 28. Primary Takeover Values

In summary, primary and primary backup fields reflect the current state of the system and the oncoming fields are not applicable until the next invocation of the **Estart** command.

To select the primary and primary backup nodes using SMIT:

TYPE **smit**

The System Management menu appears

SELECT RS/6000 SP Systems Management

The RS/6000 SP Systems Management menu appears.

SELECT RS/6000 SP Cluster Management

The RS/6000 SP Cluster Management menu appears.

SELECT Perform Switch Operations

The Perform Switch Operations menu appears.

SELECT Set Primary/Primary backup Node

The Set Primary/Primary backup Node menu appears.

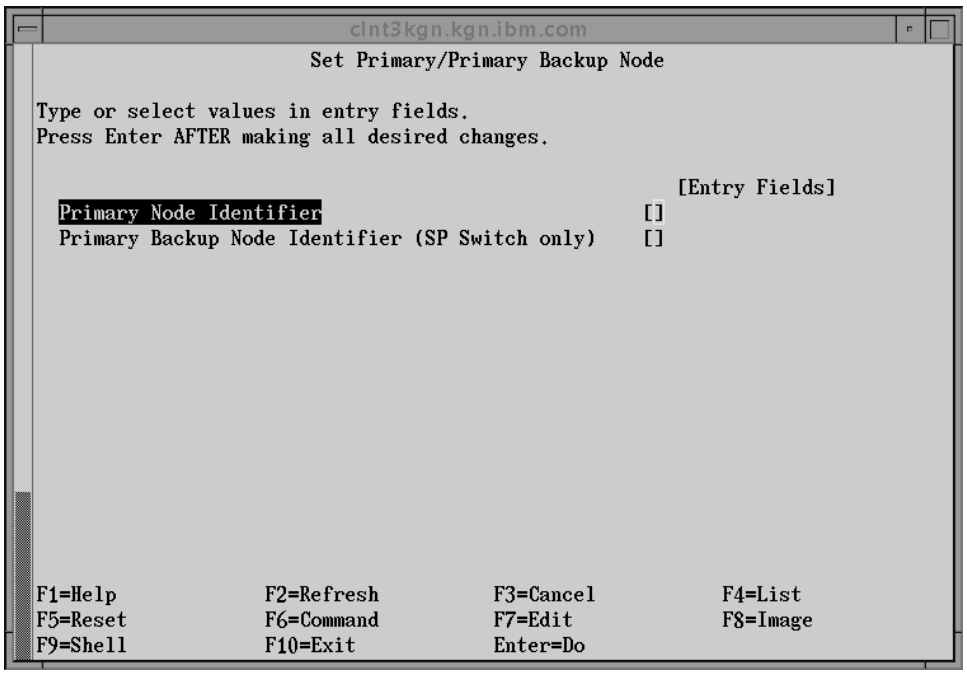


Figure 29. The Set Primary/Primary Backup Node SMIT Menu

Using the Command Line: On High Performance Switch systems, to select a switch primary node by IP address, use the command:

```
Eprimary 129.33.34.1
```

On SP Switch systems, to select an oncoming switch primary node and an oncoming switch primary backup node by IP address, use the command:

```
Eprimary 129.33.34.1 -backup 129.33.34.56
```

Managing the Switch Topology File

The switch topology file describes the wiring configuration for the 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 Parallel System Support Programs for AIX: Installation and Migration 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 Parallel System Support Programs for AIX: 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 switch. Refer to the *IBM RS/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 only contain logical locations. **Eannotator** is not a necessary step for switch operations. On SP Switch systems, however, **Eannotator** is required to update jack information for switch diagnostics. For more details on **Eannotator**, refer to the *IBM Parallel System Support Programs for AIX: Command and Technical Reference*.

Using SMIT:

TYPE	**smit**
	The System Management menu appears
SELECT	RS/6000 SP Systems Management
	The RS/6000 SP Systems Management menu appears.
SELECT	RS/6000 SP Cluster Management
	The RS/6000 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.

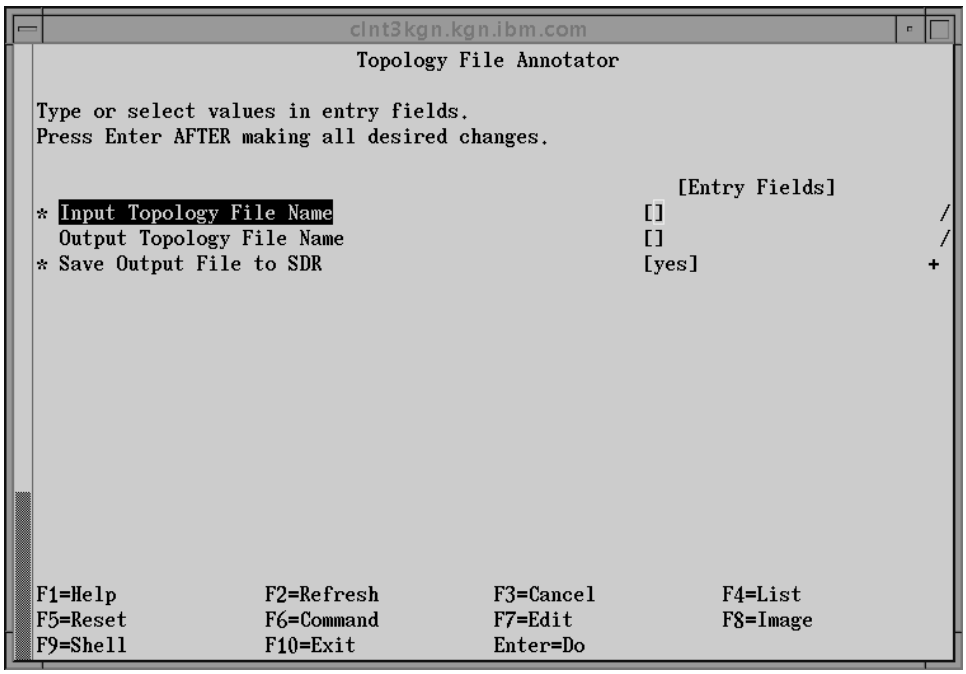


Figure 30. The Topology File Annotator SMIT Menu

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. isb. -f expected.top.2nsb. isb. -O yes

Storing the Switch Topology File in the SDR

Use **Etopology** to store the topology file in the SDR in the current system partition:

Using SMIT:

TYPE **smit**

The System Management menu appears

SELECT RS/6000 SP Systems Management

The RS/6000 SP Systems Management menu appears.

SELECT RS/6000 SP Cluster Management

The RS/6000 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.

Figure 31. The Fetch / Store Topology File SMIT Menu

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. isb.

Starting the Switch

Once all the processor nodes are running, you can start the switch.

1. Check that all processor nodes are running (hostResponds indicator). Refer to "Determining If Nodes Are Operating: All Node Summary" on page 306 for a discussion of the ways you can use the System Monitor to do this.
2. Start the switch. On the control workstation, enter:

   Estart

   (**fault_service_Worm_RTG**, the fault-handling daemon for the High Performance Switch, and **fault_service_Worm_RTG_SP**, the fault-handling daemon for the SP Switch, that check, initialize, and prepare the switch for operation, must be running on the primary node.)

   Note: 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 tserv1 .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 Parallel System Support Programs for AIX: Diagnosis and Messages Guide* and resolve any errors. See the *IBM Parallel System Support Programs for AIX: Command and Technical Reference* for more information on **Estart**.

Determining Switch Connectivity

You can monitor switch connectivity to the nodes by using the "All Node Summary" for the **switchResponds** SP System Monitor variable. (See "Determining If Nodes Are Operating: All Node Summary" on page 306 for additional information.)

On SP Switch systems, the node's status is represented by the following three colors with in the "All Node Summary":

green Indicates that a node is active on the switch.

red Indicates that a node is not active on the switch for one of the following reasons:

- It was "fenced" without the **-autojoin** option
- It faulted and was removed from the switch
- The node's adapter failed to configure properly
- The node's SP Switch adapter was recently reset by the **/usr/lpp/ssp/css/rc.switch** script

To determine the reason why the error occurred, query the **switch_responds** object in the SDR as follows:

```
SDRGetObjects switch_responds
```

If the **adapter_config_status** attribute that is displayed is anything other than **css_ready**, the node's adapter failed to configure for the reason indicated.

If all the attributes displayed are zero (except for the **adapter_config_status = css_ready**), your SP Switch adapter was recently reset. A node in this state will join the switch at the next initialization, by either an explicit issuance of the **Estart** command or a recovery initialization driven by the switch manager within the Communication Subsystem.

If only the **isolated** attribute is set to 1, the node was removed from the switch because it was either "fenced" without the **-autojoin** option by the system administrator or because it faulted.

To determine if a node was "fenced" by the system administrator, examine the **/var/adm/SPlogs/css/Ecommands.log** log of switch-related commands on the control workstation for a recent entry of an **Efence** command specifying this node. If no entry references the node in question, refer to the "Diagnosing Switch Problems" chapter in *IBM Parallel System Support Programs for*

AIX: Diagnosis and Messages Guide to determine why the node faulted.

yellow Indicates that a node is not active on switch because it was "fenced" with the **-autojoin** option.

To determine if a node was "fenced" with the **-autojoin** option by the system administrator, examine the **/var/adm/SPlogs/css/Ecommands.log** log of switch-related commands on the control workstation for a recent entry of an **Efence** command specifying the node in question.

For High Performance Switch systems, the node's status is represented by the following three colors within the "All Node Summary":

green Indicates that a node is active on the switch.

red Indicates that a node is not active on the switch. Refer to the "Diagnosing Switch Problems" chapter in *IBM Parallel System Support Programs for AIX: Diagnosis and Messages Guide* to determine why this node is not active on the switch.

yellow Indicates that a node is not active on switch because it was "fenced."

To determine if a node was "fenced" by the system administrator, examine the **/var/adm/SPlogs/css/Ecommands.log** log of switch-related commands on the control workstation for a recent entry of an **Efence** command specifying the node in question.

Fencing and Unfencing Nodes from the Switch

The node fencing function provides for isolating nodes that will be rebooted or powered-off from the switch in order to prevent such nodes from impacting other nodes. The capability of allowing the node to rejoin the switch once the node is again operational is also provided.

Notes:

1. The primary node and primary backup node for an SP Switch cannot be fenced.
2. The primary node cannot be fenced on High Performance Switch systems.
3. If you specify the **-autojoin** option, a node cannot return via **Eunfence** to the switch network if a switch initialization (**Estart**) or switch fault occurred while the node was fenced. For the SP Switch, a node fenced with the **-autojoin** option will rejoin the switch on a switch fault, node reboot, or **Estart**. If this condition occurs, you must perform another switch initialization to return the node to the High Performance Switch.
4. A node cannot return to the High Performance Switch network via **Eunfence** 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. For the SP Switch, a fenced node can only be returned to the switch via **Eunfence** regardless of whether a fault or **Estart** has occurred.

Using SMIT:

TYPE **smit**

The System Management menu appears

SELECT RS/6000 SP Systems Management

The RS/6000 SP Systems Management menu appears.

SELECT RS/6000 SP Cluster Management

The RS/6000 SP Cluster Management menu appears.

SELECT Perform Switch Operations

The Perform Switch Operations menu appears.

SELECT Fence / Unfence SP Nodes

The Fence / Unfence SP Nodes menu appears.

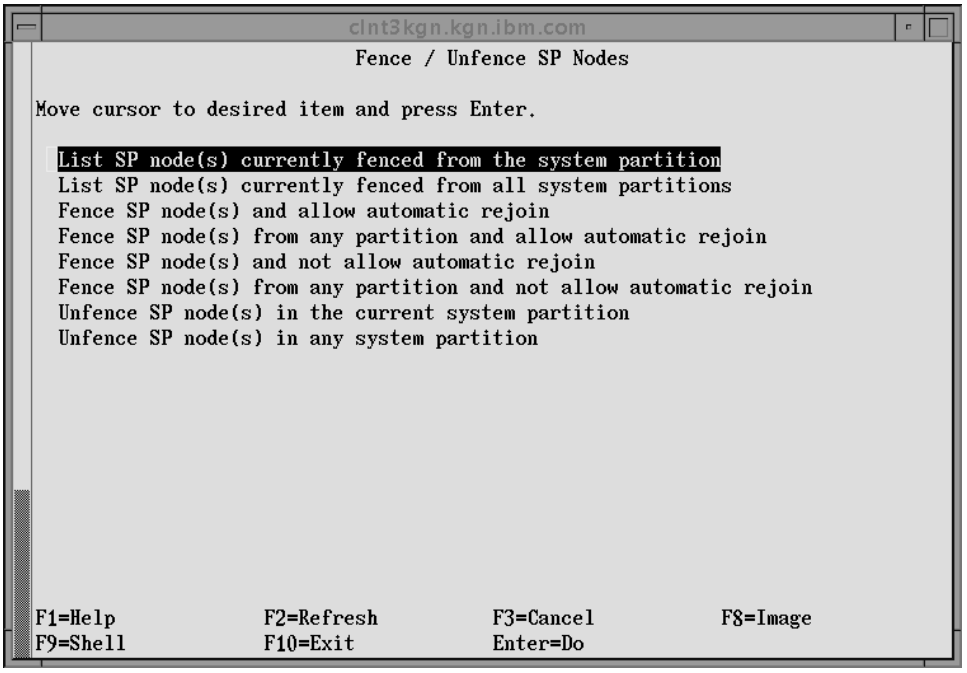


Figure 32. The Fence / Unfence SP Nodes SMIT Menu

Step 1: Fence the Node from the Switch

On the control workstation, use the **Efence** command to fence 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 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.

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 Switch

If you did not specify that the node automatically rejoin the switch in "Step 1: Fence the Node from the Switch" on page 224, use the **Eunfence** command to return the node to the 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.

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 RS/6000 SP Systems Management

The RS/6000 SP Systems Management menu appears.

SELECT RS/6000 SP Cluster Management

The RS/6000 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.

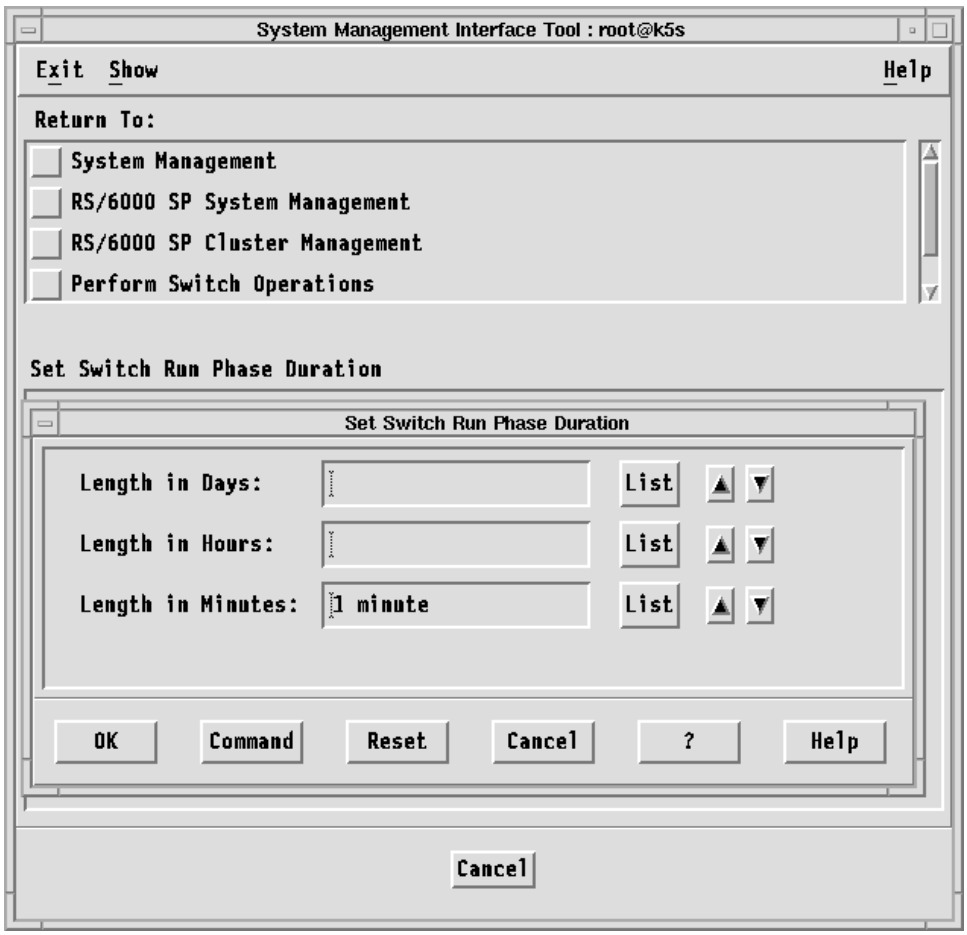


Figure 33. The Set Switch Run Phase Duration SMIT Menu

See the *IBM Parallel System Support Programs for AIX: Command and Technical Reference* for more information on the **Efence**, **Eunfence**, and **Eduration** commands.

Preparing an SP Switch for Repartitioning

Use the **Eunpartition** command to prepare the current system partition for repartitioning. For example:

```
Eunpartition
```

Optionally, you can use a SMIT panel to prepare a switch for repartitioning.

Using SMIT

TYPE **smit**

The System Management menu appears

SELECT RS/6000 SP Systems Management

The RS/6000 SP Systems Management menu appears.

SELECT RS/6000 SP Cluster Management

The RS/6000 SP Cluster Management menu appears.

SELECT Perform Switch Operations

The Perform Switch Operations menu appears.

SELECT Prepare a switch network for repartitioning

The Prepare a switch network for repartitioning menu appears.

See the *IBM Parallel System Support Programs for AIX: Command and Technical Reference* for more information on the **Eunpartition** command.

Monitoring Nodes on a Switch

You can monitor and attempt to maximize the availability of nodes on a switch with the **Emonitor** daemon. **Emonitor** is controlled by the System Resource Controller (SRC). One instance of the daemon exists for each system partition and is named **Emonitor**.*partition_name*. A system-wide configuration file, **/etc/SP/Emonitor.cfg** lists all node numbers (one per line) on the system to be monitored.

Emonitor is invoked with **Estart -m**. Once invoked, it is SRC controlled and so will restart if halted abnormally. To end monitoring, run **/usr/lpp/ssp/bin/emonctrl -s** to stop the daemon in the system partition.

For more information on **Emonitor**, refer to *IBM Parallel System Support Programs for AIX: Command and Technical Reference*.

Global Shutdowns and Reboots of Nodes with a Switch

The **Equiesce** command disables switch error recovery and primary node takeover. It is used to shut off the normal error actions when global activities are performed. For example, when nodes are shut down or rebooted, they are fenced from the switch.

It is possible that the primary node may fence all the other nodes, including the primary backup node, if it is not the first node to shut down during a global shutdown/reboot of the entire system. Primary node takeover can also occur if the primary node is shut down and the primary backup node remains up. Issuing the **Equiesce** command before the shutdown will prevent these situations from occurring.

The **Equiesce** command causes the primary and primary backup nodes to shut down their recovery actions. Data still flows over the switch, but no faults are serviced and primary node takeover is disabled. Only the **Eannotator**, **Eclock**, **Eprimary**, **Estart**, and **Etopology** commands are functional after the **Equiesce** command is issued.

Estart must be issued when the global activity is complete to reestablish switch recovery and primary node takeover.

Use the **Equiesce** command to disable switch error recovery and primary node takeover. For example:

```
Equiesce
```

Optionally, you can use a SMIT panel to quiesce the SP Switch.

Using SMIT

TYPE **smit**

The System Management menu appears.

SELECT RS/6000 SP Systems Management

The RS/6000 SP Systems Management menu appears.

SELECT RS/6000 SP Cluster Management

The RS/6000 SP Cluster Management menu appears.

SELECT Perform Switch Operations

The Perform Switch Operations menu appears.

SELECT Switch Quiesce

The **Equiesce** command is run.

See *IBM Parallel System Support Programs for AIX: Command and Technical Reference* for more information on the **Equiesce** command.

NFS Mounts Over a Switch

We strongly discourage NFS mounts over the switch in the root (/) directory. If a mount is necessary in the root (/) directory, place the NFS mount elsewhere in the file system with a symbolic link into the root (/) directory. For example:

```
mount -o intr,hard server:/myfs /otherdir/myfs
ln -s /otherdir/myfs /myfs
```

Chapter 17. Managing System Partitions

System partitioning is a method for organizing the SP system into non-overlapping 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
- Managing system partition-sensitive subsystems using the **syspar_ctrl** command

You will also find information on system partitioning in the *IBM Parallel System Support Programs for AIX: System Planning* and the *IBM Parallel System Support Programs for AIX: Installation and Migration Guide*. For information on accounting and system partitioning, refer to “Accounting and System Partitioning” on page 136.

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 non-overlapping 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 system partition to use for non-disruptive testing when you migrate to a new release of software. Or you may wish to configure system partitions for multiple isolated production environments so that the workload in one environment will not interfere with the workload in another environment. Isolation of production environments is achieved by partitioning the switch in such a way that communication paths for different system partitions do not cross each other.

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. 1996

```
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 **-G** flag allows specification of target nodes outside of the current system partition.
cstartup -G	The **-G** flag allows specification of target nodes outside of the current system partition.

Command	Effect
dsh **dsh -w** {*hostname* | -}	Hosts added to the working collective by host name need not be in the current system partition.
dsh -aG	The **-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.
emonctrl -c	The system partition-sensitive control script for the **emon** subsystem supports the **-c** option, which crosses system partitions.
Eunfence -G	The **-G** flag allows specification of nodes outside of the current system partition.
haemctrl -c	The system partition-sensitive control script for the **haem** subsystem supports the **-c** option, which crosses system partitions.
hagsctrl -c	The system partition-sensitive control script for the **hags** subsystem supports the **-c** option, which crosses system partitions.
hatsctrl -c	The system partition-sensitive control script for the **hats** subsystem supports the **-c** option, which crosses system partitions.
hbctrl -c	The system partition-sensitive control script for the **hb** subsystem supports the **-c** option, which crosses system partitions.
hmcmds -G	The **-G** flag allows the **hmcmds** commands to be sent to any hardware on the SP system.
hmmon -G	The **-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 **-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.
hrctrl -c	The system partition-sensitive control script for the **hr** subsystem supports the **-c** option, which crosses system partitions.
lppdiff -aG	The **-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.
nodecond -G	The **-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).
pmanctrl -c	The system partition-sensitive control script for the **pman** subsystem supports the **-c** option, which crosses system partitions.

Command	Effect	
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 **-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.	
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.	
sp_configdctrl -c	The system partition-sensitive control script for the **sp_configd** subsystem supports the **-c** option, which crosses system partitions.	
spdelfram -G	The **-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.	
splst_versions -G	The **-G** flag allows for retrieval of PSSP version information from nodes outside the current system partition.	

Command	Effect
splm	The target nodes defined in the input table can include nodes from any system partition.
splstdata -G	The **-G** flag allows display of information on nodes and adapters outside of the current system partition.
spmon -G	The **-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.
syspar_ctrl -c -G	The **-c** and **-G** flags allow for the crossing of system partitions in providing a single interface to the control scripts for the system partition-sensitive subsystems.
s1term -G	The **-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.
vsdsklst -G	The **-G** flag specifies the display of information for disks outside the current system partition.

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 SP Perspectives GUI provides the capability to control and monitor the entire system or a system partition
- 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.X boot/install server is required for all AIX 4.X system partitions.

System partitioning can be achieved either by applying one of a fixed set of supplied configurations (refer to the *IBM Parallel System Support Programs for AIX: System Planning* for specific information on supplied configurations) or by creating a different system partition configuration and applying it. The **ssp.top** option in the **pssp** install package contains a directory structure with all relevant files for partitioning the system with supplied system partition configurations. In addition, the **ssp.top** option contains the System Partitioning Aid, a tool for creating customized system partitions to suit your needs. This option must be installed on

the control workstation if you are partitioning your system into more than one system partition.

SP Switch Usage Note

If you are repartitioning your system and it has an SP Switch, you must run **Eunpartition**. Failure to do this will produce unpredictable **Estart** results in the new system partitions. The **Estart** command results can range from only the old system partition portion of the new system partition being accessible on the switch, to a complete **Estart** timeout failure. Various recovery scenarios are available. A guaranteed recovery can be accomplished by:

1. Issuing **Eclock** to reset the switch
2. Rebooting all the nodes or issuing a **css_restart_node** on all nodes
3. Issuing **SDRChangeAttrValues switch_responds isolated=0** from the global view of the system monitor (or in each system partition)
4. Issuing **Estart** in each of the system partitions

This recovery scenario is disruptive to all system partitions, even those unaffected by the repartition.

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 tr  alias 129.4 .127.1 1 netmask 255.255.255.  up >>$LOGFILE 2>&1
   ```

2. Execute the **/etc/rc.net** script. Alternatively, you could enter the **ifconfig** command directly. For example:

   ifconfig tr alias 129.4 .127.1 1 netmask 255.255.255. 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
   tr 129.4 .127 129.4 .127.43
   tr 129.4 .127 129.4 .127.1 1

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 tr delete 129.4 .127.1 1

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 configuration for your size system. The steps involved in partitioning an SP system are:

1. Check if the desired system partition configuration is supplied.
2. If the desired system partition configuration is not supplied, use the System Partitioning Aid to generate and save a system partition configuration.
3. Archive the System Data Repository.
4. Select a system partition configuration and layout to apply.
5. Customize (name and specify attributes) the system partitions.

   Note: If you are repartitioning your system, and your system contains an SP Switch, you must run **Eunpartition** in each of the partitions that are going to change. This prepares the switch in those partitions for repartitioning.

6. Shut down all nodes in changing system partitions.
7. Apply the configuration.
8. 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 modify an existing system partition or create a new partition for a new production environment, you would follow these steps to repartition the system.

How to Start

If you know that the desired system partition configuration exists, start with "Step 3: Archive the System Data Repository" on page 238; otherwise, start with "Step 1: Check if the Desired System Partition Configuration is Supplied" on page 237.

For partitioning or repartitioning the system, you have the option of using SMIT panels or commands. SMIT panels can be accessed through the **smit** command or through the SP Perspectives GUI. (For more information, and further examples, of the commands used in this chapter, refer to the *IBM Parallel System Support Programs for AIX: 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 RS/6000 SP System Management

The RS/6000 SP System Management menu appears.

SELECT RS/6000 SP Configuration Database Management

The RS/6000 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.

Using the SP Perspectives GUI to Access SMIT:

TYPE **perspectives &**

The SP Perspectives Launch Pad appears.

SELECT smit config_data on CWS

The RS/6000 SP Configuration Database Management menu appears in a separate AIX window.

SELECT Enter Database Information

The Enter Database Information menu appears.

SELECT System Partition Configuration

The System Partition Configuration menu appears.

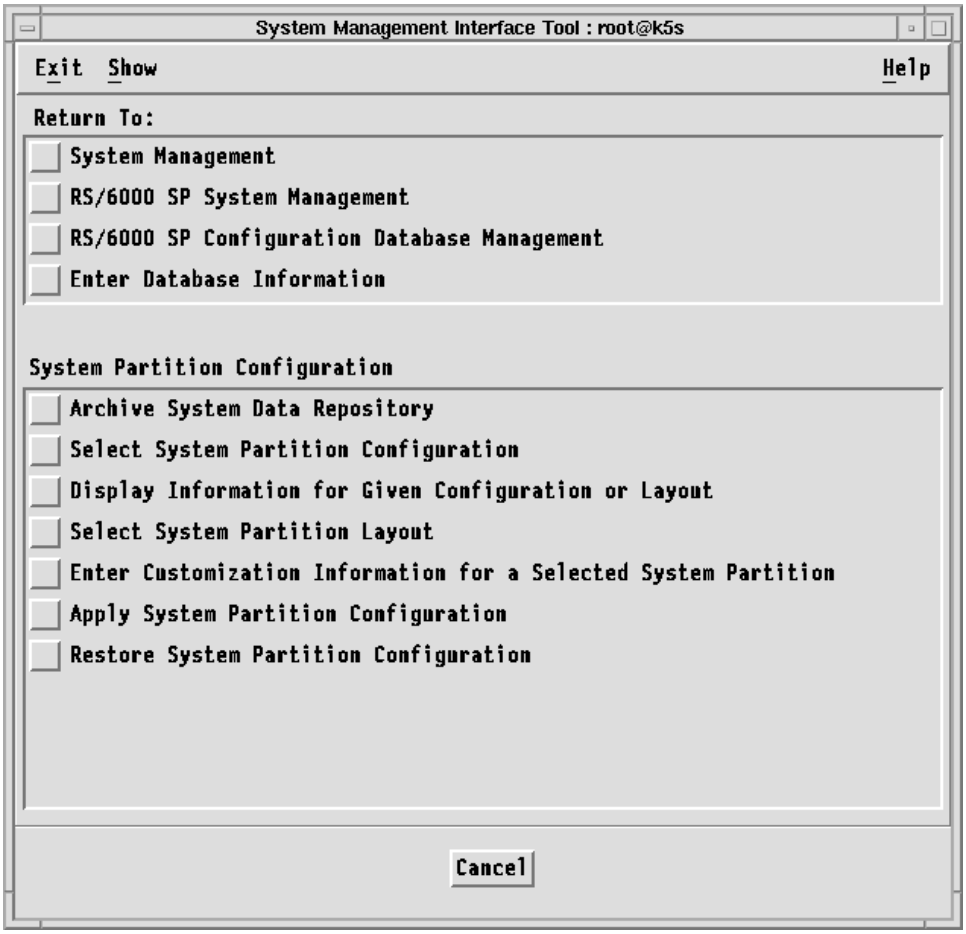


Figure 34. System Partition Configuration SMIT Panel

Step 1: Check if the Desired System Partition Configuration is Supplied

Use the System Partitioning Aid to check if the desired system partition configuration is supplied. The System Partitioning Aid can be invoked from the command line or from the SP Perspectives GUI. In this section, we will provide examples of using the command line interface. (For information on the SP Perspectives GUI to the System Partitioning Aid, invoke the Aid with the **spsyspar** command and view the online help that is available.) For additional information on using the System Partitioning Aid, refer to *IBM Parallel System Support Programs for AIX: System Planning*.

To check if the desired system partition configuration is supplied:

1. Enter information for your desired system partition configuration into a file. The template for this file is **/spdata/sys1/syspar_configs/bin/inpfile.template**.
2. Once you have created the input file, enter:

   sysparaid *input_file*

3. If the desired system partition configuration is found, layout directory information is displayed. Go to “Step 3: Archive the System Data Repository” on page 238.
4. If the desired system partition configuration is not found, a message is displayed indicating so and stating whether the desired system partition configuration is possible or not possible. If the system cannot be partitioned as desired, an error message is output indicating the reason for the failure. If the desired system partition configuration is possible, a snapshot of the switch chips in the system and performance data for each system partition is output to a set of files.
   a. If the system cannot be partitioned as desired, analyze the error message and create another input file specifying an alternative system partition configuration and rerun the **sysparaid** command.
   b. If the desired system partition configuration is possible, review the performance data that is output and, if it is acceptable, proceed to “Step 2: Generate and Save the Desired System Partition Configuration.”

For more information on the **sysparaid** command, refer to the *IBM Parallel System Support Programs for AIX: Command and Technical Reference*.

Step 2: Generate and Save the Desired System Partition Configuration

Once you have determined that the system partition configuration you desire is possible, generate and save it:

sysparaid -s *layout_name input_file*

The newly generated system partition configuration layout will be saved to the **spdata/sys1/syspar_configs** directory.

The fastpath invocation for the System Partition Configuration menu is:

smit syspar

Step 3: 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

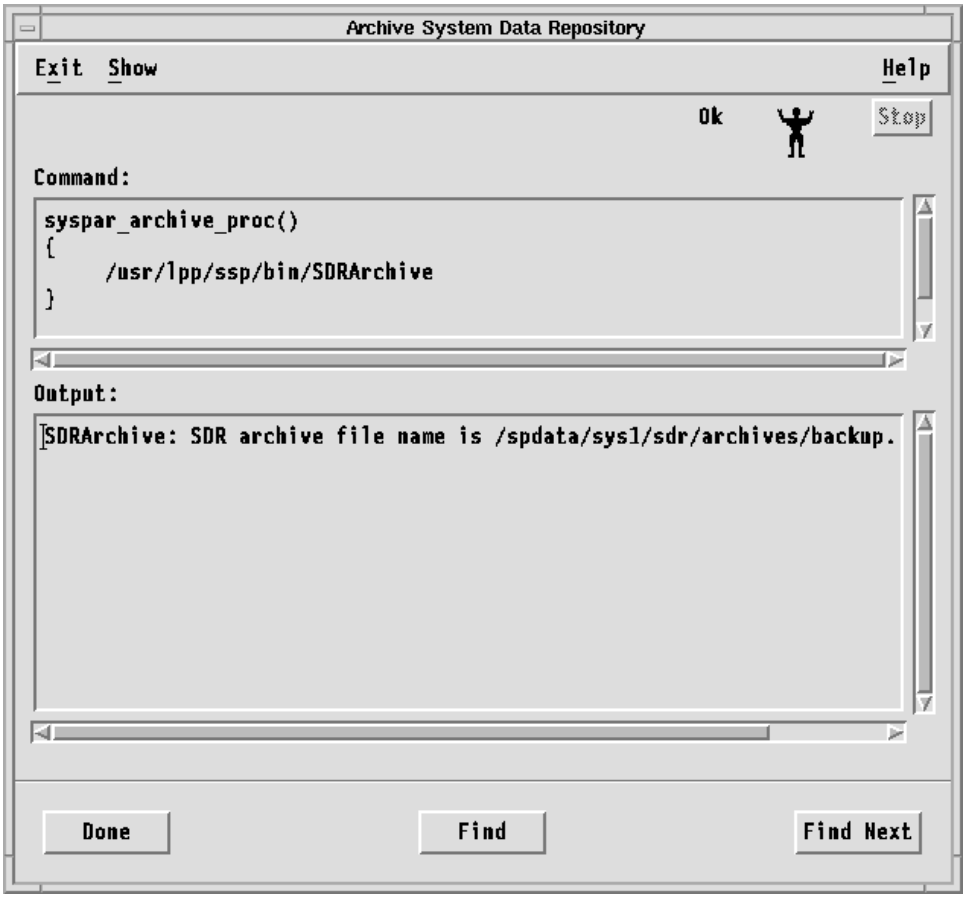


Figure 35. Archive System Data Repository SMIT Panel

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 4: 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 Parallel System Support Programs for AIX: 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. (If you have created layouts using the System Partitioning Aid, such layouts will also be listed on the SMIT panel.)

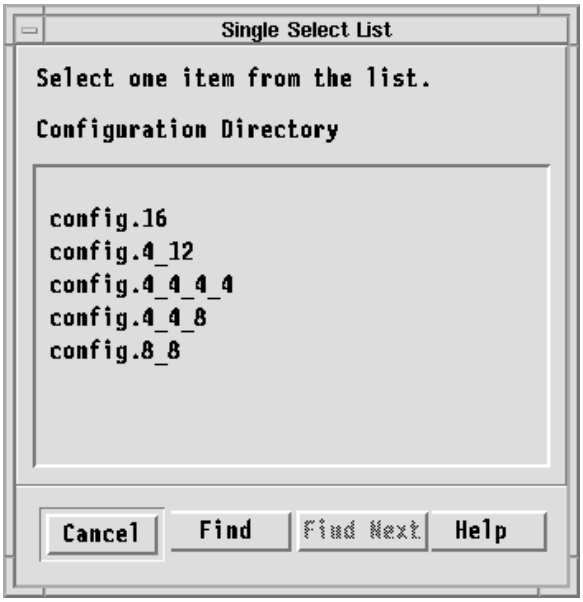


Figure 36. System Partition Configuration List SMIT Panel

SELECT A system partition configuration

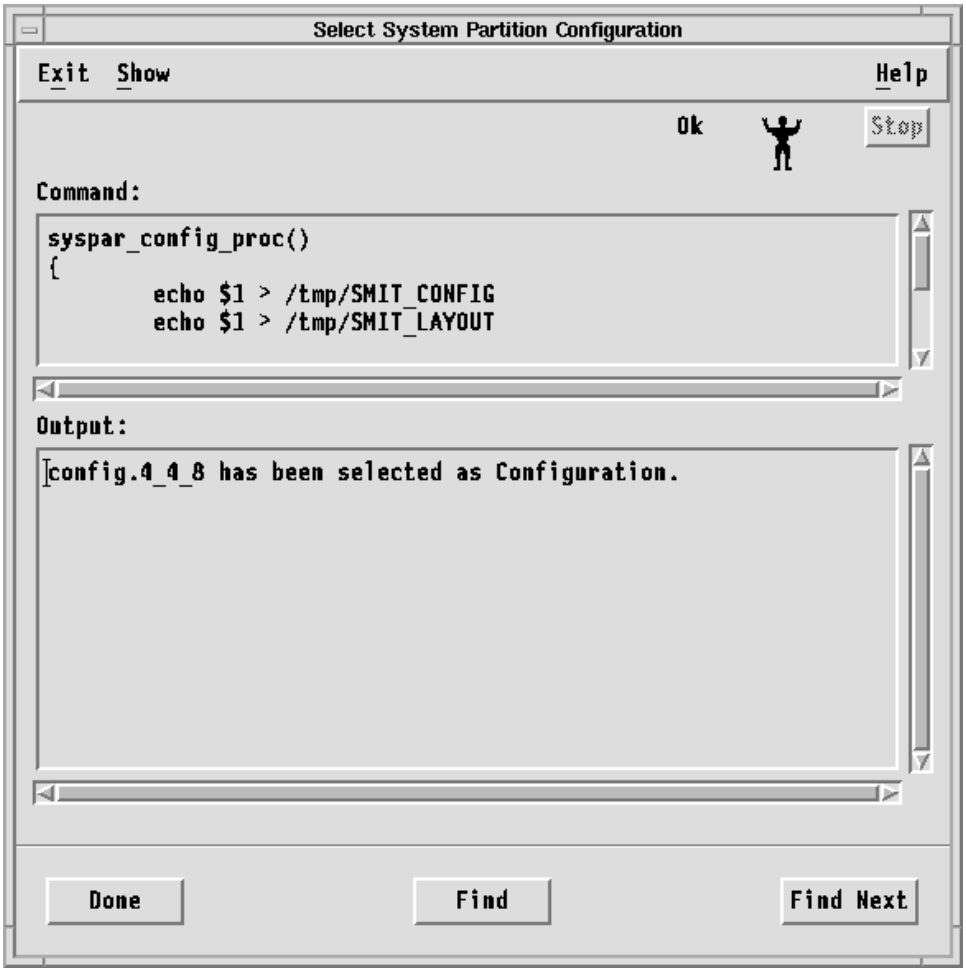


Figure 37. Select System Partition Configuration SMIT Panel

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

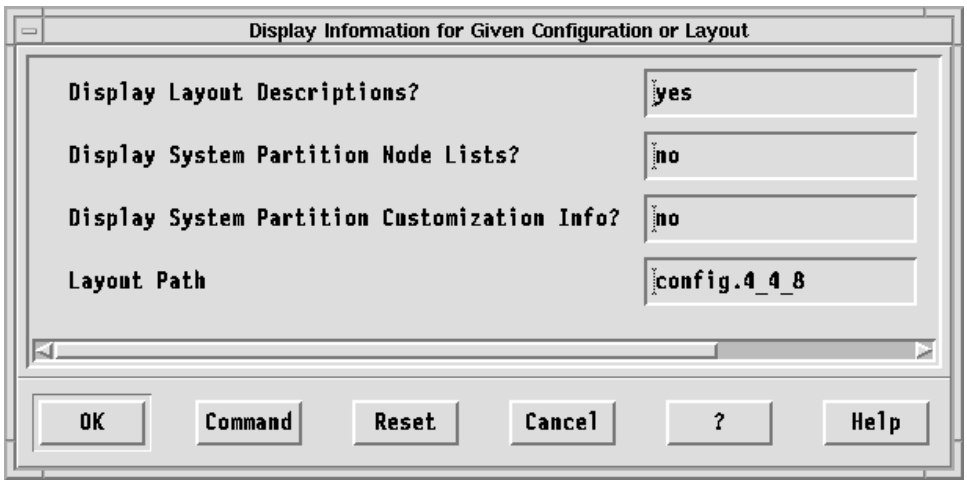


Figure 38. Display Information for Given Partition or Layout

SELECT OK

System partition layout information for the chosen system partition configuration appears.

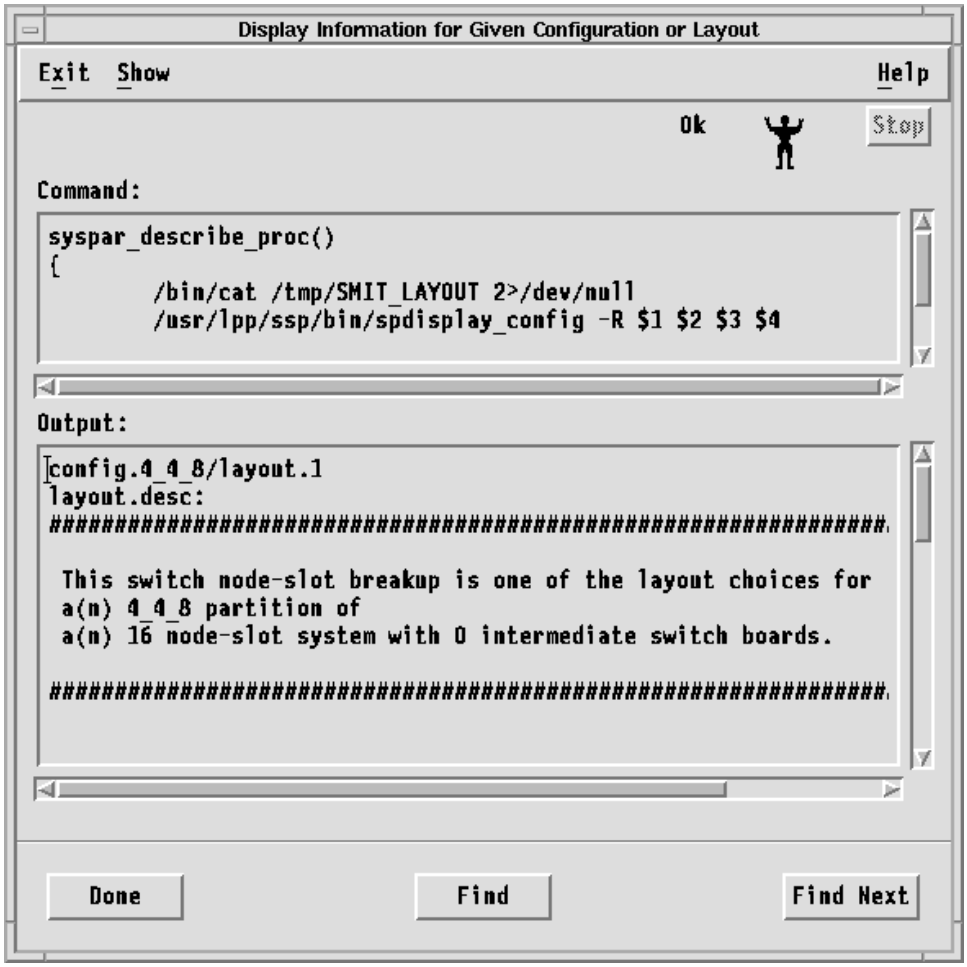


Figure 39. Display Information for Given Partition or Layout Output

(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. (If you have created layouts using the System Partitioning Aid, such layouts will also be listed on the SMIT panel.)

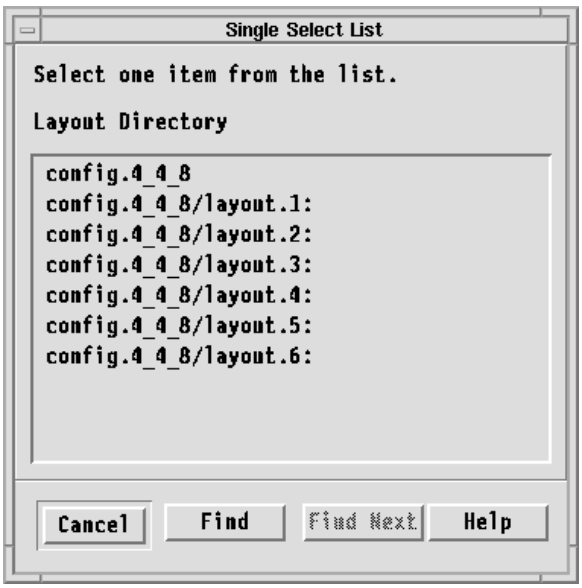


Figure 40. Select System Partition Layout SMIT Panel

SELECT A system partition layout

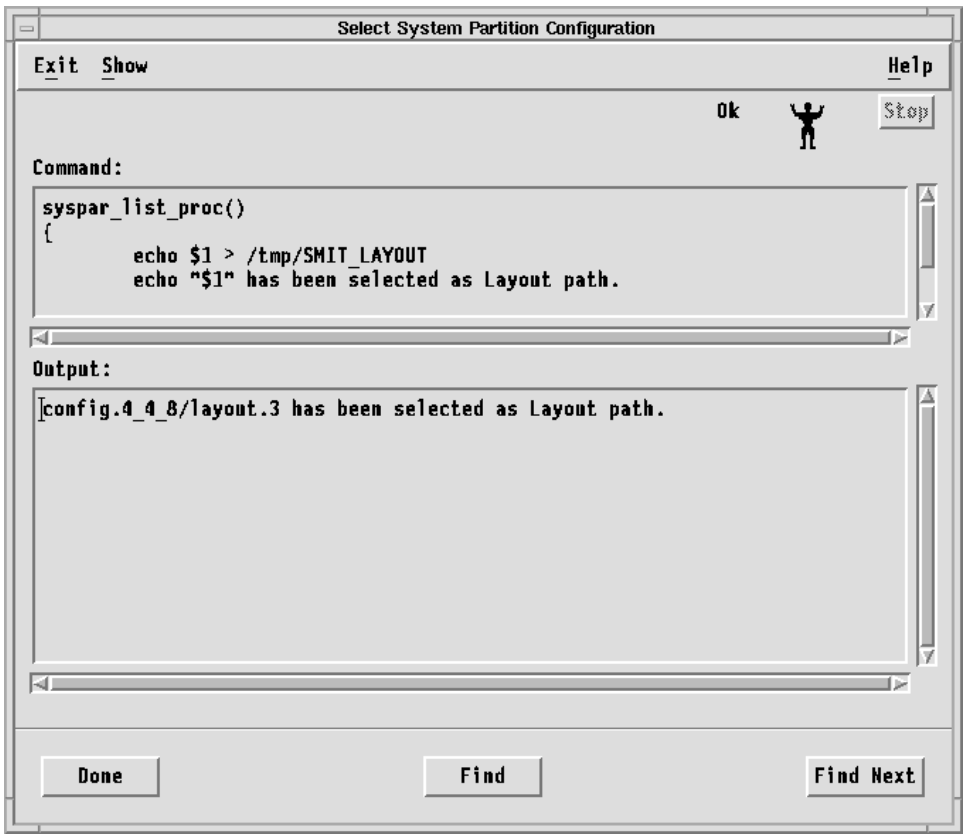


Figure 41. Select System Partition Layout Output SMIT Panel

In this example, we have chosen layout 3 for our selected 4-4-8 system partition configuration.

Step 5: 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. It is imperative that all new IP addresses and host names being used will be resolvable on your SP system.

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.

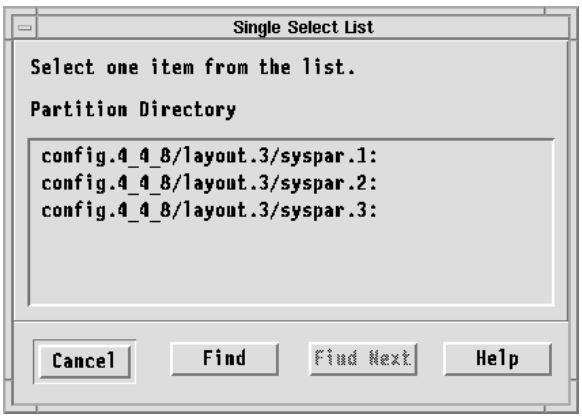


Figure 42. Partition Directory SMIT Panel

SELECT A system partition and enter customization arguments

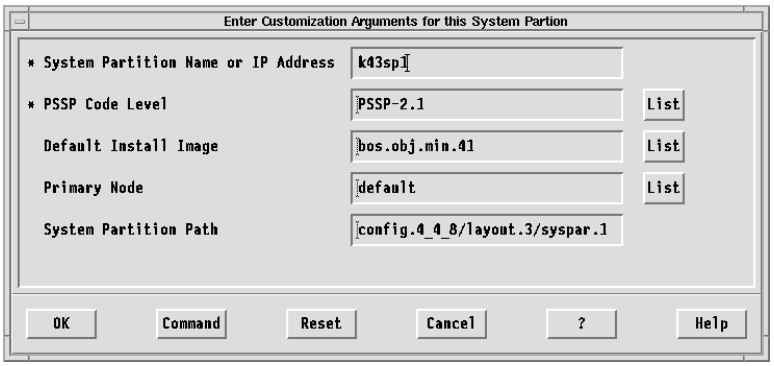


Figure 43. Enter Customization Arguments for this System Partition SMIT Panel

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) or **PSSP-2.2**. For system partitions that contain mixed levels of PSSP, this should be set to the minimum, or earliest, level installed in this system partition. For more information, refer to the technical bulletin *Implementing Coexistence with PSSP 2.2*, available through your IBM marketing representative or at **http//:www.rs6000.ibm.com/tech**.

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). For system partitions that contain mixed levels of PSSP, this default install image should be associated with the PSSP code level specified above. For more information, refer to the technical bulletin *Implementing Coexistence with PSSP 2.2*, available through your IBM marketing representative or at **http//:www.rs6000.ibm.com/tech**.

Primary Node The node on the switch that initializes, recovers, provides diagnosis, and performs other operations to the switch network. The default is the first node in the **nodelist** file.

Backup Primary Node
Available only on SP Switch systems, the node designated as the oncoming switch primary node.

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 6: 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. If your system contains an SP Switch, you must invoke **Eunpartition** in each system partition that is being changed. See the **Eunpartition** command in the *IBM Parallel System Support Programs for AIX: Command and Technical Reference* for more information.

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 Parallel System Support Programs for AIX: 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

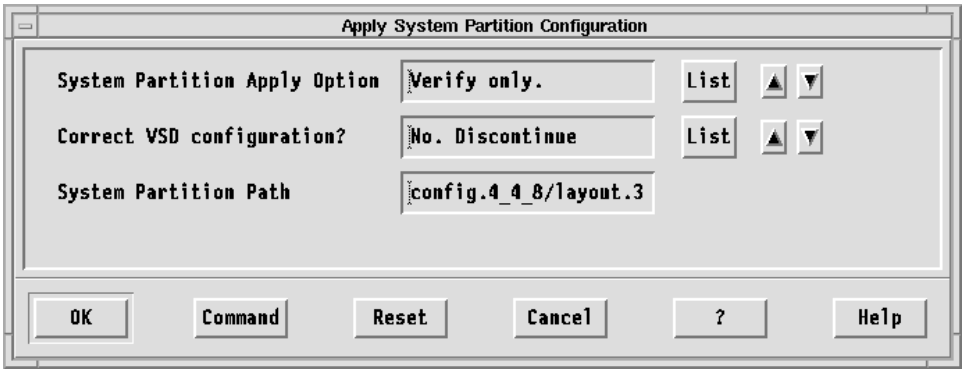


Figure 44. Verify System Partition Configuration SMIT Panel

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 7: 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 **sdrd** daemons are created and started.

New system partition-sensitive subsystems are created and started via the **syspar_ctrl** command.

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 Parallel System Support Programs for AIX: Command and Technical Reference* for more details.)

Old unused system partitions on the control workstation, along with their **sdrd** daemons, are deleted and old system partition-sensitive subsystems 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

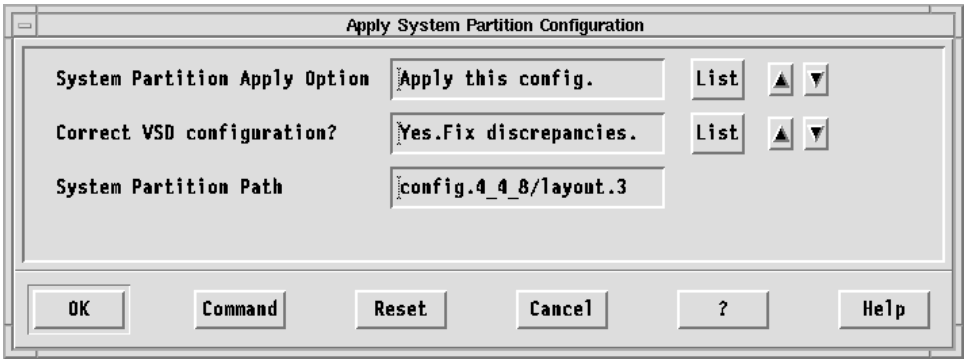


Figure 45. Apply System Partition Configuration SMIT panel

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 system partition-sensitive subsystems.

Note: Due to system partitioning changes, your SP_NAME environment variable may no longer be set to a valid system partition name. To get a list of valid system partition names, enter the **splst_syspars -n** command. Then verify that your SP_NAME environment variable is either unset or set to one of the partition names in the list.

Step 8: Reboot All Nodes in Changed Partitions

Reboot any nodes that were shut down in "Step 6: Shut Down All Nodes in Changing Partitions" on page 246.

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 RS/6000 SP System Management

The RS/6000 SP System Management menu appears.

SELECT RS/6000 SP Configuration Database Management

The RS/6000 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.

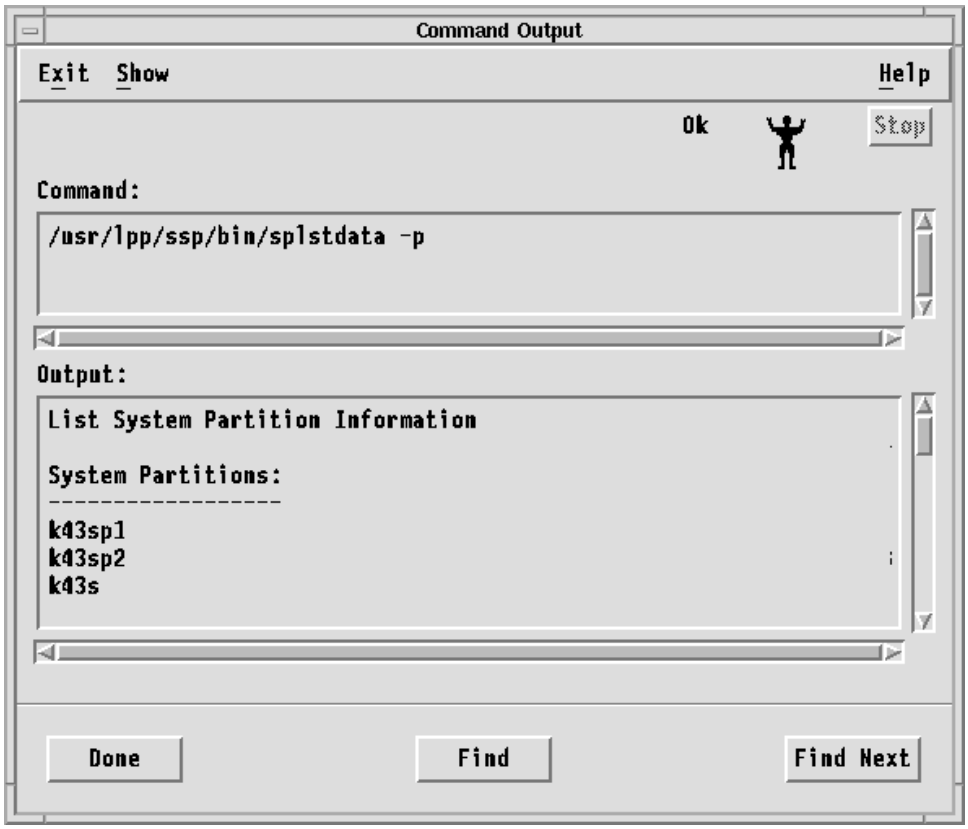


Figure 46. List System Partition Database Information Output

(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 RS/6000 SP System Management

The RS/6000 SP System Management menu appears.

SELECT RS/6000 SP Installation/Configuration Verification

The RS/6000 SP Installation/Configuration Verification menu appears.

SELECT System Partition Configuration

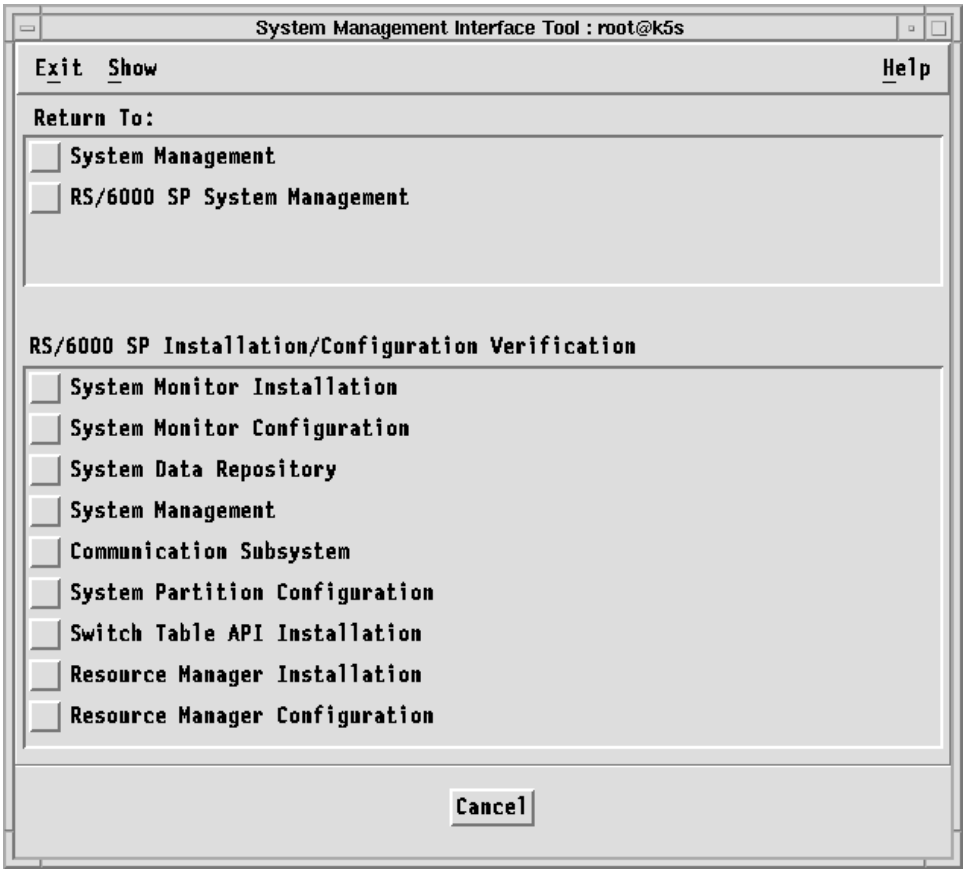


Figure 47. Verify System Partition Configuration SMIT Panel

The fastpath invocation for the Verify 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**.

Managing System Partition-Sensitive Subsystems Using syspar_ctrl

Some of the subsystems that run in the SP environment operate within the domain of a system partition, rather than the domain of the system as a whole. These subsystems are said to be **system partition-sensitive**.

To help you manage these subsystems, PSSP includes the Syspar Controller, which operates through the **syspar_ctrl** command. With this command you can add, start, stop, refresh, and delete system partition-sensitive subsystems, as well as control other aspects of their operation, such as tracing.

The **syspar_ctrl** command provides a single interface to the control scripts for the system partition-sensitive subsystems, which perform the real work of configuring and operating the subsystems.

Listing the System Partition-Sensitive Subsystems

To list the system partition-sensitive subsystems that are managed by the Syspar Controller and their control scripts, enter:

```
syspar_ctrl -E
```

In response, the system displays information similar to the following:

```
Syspar Controller managed subsystems and control scripts:
  hats          /usr/lpp/ssp/bin/hatsctrl
  hb            /usr/lpp/ssp/bin/hbctrl
  hags          /usr/lpp/ssp/bin/hagsctrl
  haem          /usr/lpp/ssp/bin/haemctrl
  hr            /usr/lpp/ssp/bin/hrctrl
  pman          /usr/lpp/ssp/bin/pmanctrl
  emon          /usr/lpp/ssp/bin/emonctrl
  sp_configd    /usr/lpp/ssp/bin/sp_configdctrl
  emcond        /usr/lpp/ssp/bin/emconditionctrl
```

The order in which the subsystems are listed (from top to bottom) is the order in which the subsystems are added or started by the **syspar_ctrl** command. The command deletes or stops the subsystems in the reverse order (from bottom to top). Other optionally installed subsystems may be listed. Refer to the documentation for those subsystems for more information.

The system partition-sensitive subsystems provide the following services:

hats – the Topology Services subsystem, which provides information to other PSSP subsystems about the state of the nodes and adapters on the SP system.

hb – the Heartbeat subsystem, which communicates with several PSSP subsystems as part of providing information about the state of the nodes and adapters on the SP system.

hags – the Group Services subsystem, which provides a general purpose facility for coordinating and monitoring changes to the state of a client program that is running on a set of nodes in an SP system.

haem – the Event Management subsystem, which compares information about the state of system resources provided by resource monitors with requests for information by client programs, and creates events that notify the clients when a match occurs

hr – the Host_Responds subsystem, which provides information about the state of the nodes and adapters on the SP system.

pman – the Problem Management subsystem, which allows the installation to be notified when events related to problem management occur

emon – the Emonitor subsystem, which monitors the status of nodes in a switch network, bringing a node back into the switch network when necessary. This subsystem is not started by the **syspar_ctrl** command. It is started using the **Estart -m** command.

sp_configd – the SNMP Support portion of the Problem management subsystem, which creates and sends SNMP traps when selected types of errors are recorded.

emcond – the Event Management Conditions subsystem loads the SDR with pre-defined events or conditions of interest. These conditions are used by the SP Perspectives GUI.

Understanding System Partition-Sensitive Subsystems

Except where noted, the PSSP system partition-sensitive subsystems operate under the control of the System Resource Controller (SRC) and have associated SRC group and subsystem names. Each subsystem has a control script. Except where noted, each subsystem is associated with one or more daemons.

Except where noted, for each system partition, one instance of a subsystem executes on the control workstation and one instance executes on each of the system partition's nodes.

On the control workstation, there are multiple instances of each subsystem, one for each system partition that is defined. Accordingly, the subsystem names on the control workstation have the system partition name appended to them.

On a node, only one instance of a subsystem executes. Hence, the system partition name is not required to differentiate instances of a subsystem.

Several of the subsystems that require configuration have an associated script that configures the daemon before starting it.

Table 10 on page 254 summarizes key information about the system partition-sensitive subsystems. It uses the following notation:

syspar_name is the name of a system partition.

CWS indicates the control workstation

For details about the operation of each subsystem, see the man page for its control script. For information about the SRC, see *AIX Version 4 General Programming Concepts: Writing and Debugging Programs*.

Table 10. Summary of System Partition-Sensitive Subsystems

Subsystem	*SRC Group*	*Control Script*	*Daemons*	*Subsystem Names*
Topology Services	**hats**	**hatsctrl**	**hatsd**	CWS: **hats.***syspar_name* Nodes: **hats**
Heartbeat	**hb**	**hbctrl**	**hbd**	CWS: **hb.***syspar_name* Nodes: **hb**
Group Services	**hags**	**hagsctrl**	**hagsd** **hagsglsmd**	CWS: **hags.***syspar_name* Nodes: **hags** CWS: **hagsglsm.***syspar_name* Nodes: **hagsglsm**
Event Management	**haem**	**haemctrl**	**haemd**	CWS: **haem.***syspar_name* Nodes: **haem**
Host_ Responds	**hr**	**hrctrl**	**hrd**	CWS: **hr.***syspar_name* Nodes: None
Problem Management (Event Management)	**pman**	**pmanctrl**	**pmand** **pmanrmd**	CWS: **pman.***syspar_name* Nodes: **pman** CWS: **pmanrm.***syspar_name* Nodes: **pmanrm**
Emonitor	**emon**	**emonctrl**	**Emonitor**	CWS: **Emonitor.***syspar_name* (started only by **Estart -m**) Nodes: None
Problem Management (SNMP Support)	None	**sp_configdctrl**	**sp_configd**	CWS: **sp_configd** (only one instance per system) Nodes: **sp_configd**
Event Management Conditions	None	**emconditionctrl**	None	None

Checking the Status of System Partition-Sensitive Subsystems

Normally, you do not have to do anything to manage the system partition-sensitive subsystems. In general, they are all added and started during installation and use standard mechanisms to restart themselves when they are not explicitly stopped.

To make sure that all of the system partition-sensitive subsystems for a system partition are running, login to the control workstation and use the **lssrc -a** command, grepping for the system partition name. Issue another command to check the **sp_configd** subsystem, which does not have the system partition name appended to it.

For example, if the system partition is named **k22sp1**, enter:

```
lssrc -a | grep k22sp1
lssrc -a | grep sp_configd
```

In response, you should receive output that indicates that all of the system-partition subsystems except Emonitor are active, similar to the following:

```
hats.k22sp1          hats                 34126      active
hb.k22sp1             hb                   48664      active
hags.k22sp1          hags                 34446      active
hagsglsm.k22sp1   hags                 34464      active
haem.k22sp1          haem                  26 88      active
hr.k22sp1             hr                   16566      active
pman.k22sp1          pman                  328 6      active
pmanrm.k22sp1       pman                  4 496      active
Emonitor.k22sp1   emon                                inoperative
sdr.k22sp1           sdr                  33912      active

sp_configd                                39568      active
```

Note that, although its subsystem appears in the output for the system partition, the SDR is not managed by the Syspar Controller.

Also, the Emonitor subsystem is not started by the **syspar_ctrl** command. Therefore, its status is inoperative until it is started by the **Estart -m** command.

Chapter 18. Managing Node Groups

Node grouping is a persistently saved naming facility for sets of nodes: nodes can be grouped together so that they can be acted on as an individual entity. Many system management commands can accept a node group as an argument, causing the action to be performed on each node in the node group. Node groups can be created and managed through a command line interface, SMIT panels, and the SP Perspectives GUI.

This chapter provides information on:

- Understanding node groups
- Working with node groups
- Using node groups with commands

Understanding Node Groups

A node group is a set of nodes whose members may be individual nodes and other node groups. Individual nodes that belong to a node group are *member nodes*; node groups that belong to a node group are *member node groups*. Node groups containing only nodes are *flat node groups*; node groups containing node groups are *hierarchical node groups*. Any node group can be resolved into a flat version that contains only valid member nodes.

Commands, except those that directly manage the node groups, issued against a node group have the same effect as issuing the commands against the set of nodes that represent the flat version of the node group.

Node Group Information in the SDR

Node groups are stored persistently in the SDR. A node group in the SDR is made up of two components: a name and a definition list. A node group name must start with an alphabetic character (A-Z, a-z), but can then contain a sequence of alphanumeric characters (A-Z, a-z, 0-9), a period (.) or an underbar (_). (Thus named, node groups can be easily differentiated from node objects, whose names are a string of numeric characters representing the node number.) A node group name may not contain spaces. The definition list of a node group is made up of a list of nodes and node groups that belong to the node group directly, (not through member node groups).

Node Group Modes

Some node groups exist within a system partition (*partition-bound* node groups or *local* node groups) and some node groups ignore system partition boundaries (*global* node groups). Partition-bound or local node groups can be accessed only within the context of the system partition they belong to. Global node groups are not bound to any system partition, but can be acted on only by commands that accept the **-G** flag or when using the SP Perspectives GUI with a Global view of the SP system. Partition-bound and global refer to the *mode* of the node group.

Partition-bound node groups (local node groups) are stored in the partitioned SDR class **NodeGroup**. Global node groups are stored in the system SDR class **SysNodeGroup**. Valid member nodes of a partition-bound node group must be

© Copyright IBM Corp. 1996

node numbers that are defined in that particular system partition. Valid member nodes of a global node group are node numbers that are defined in that particular SP system (the union of all nodes in each system partition). Any node number or node group can be added to the definition list of a node group, but only valid member nodes or valid member node groups will be used when resolving (flattening) the node group.

Note: The global name space and the partition name spaces are mutually exclusive for node groups.

Most SP system end users work within system partitions and, therefore, will use partition-bound node groups.

System Administrators, however, will often find it necessary, and advantageous, to work with global node groups. The use of global node groups allows a System Administrator to perform certain tasks, such as installing a daemon on specific nodes or powering off and reconfiguring all nodes on which a given application is running, without regard to system partitions.

Partition-bound node groups and global node groups are different, but they both function to provide a naming facility for sets of nodes.

Considerations for Creating and Resolving Node Groups

When creating a node group or adding to an existing node group's definition list, you can use the command line interface or SMIT panels to add any node number or node group. The member node group does not need to have been defined at the time it is added to another node group's definition list. The member node does not have to be a valid node number for a system partition or even the SP system at the time it is added to another node group's definition list. In this way, you can create node groups to accommodate future growth or changes, such as adding nodes or repartitioning your SP system.

Only users that have the appropriate authority to modify objects in the SDR can create, modify or delete node groups. Any user that has the appropriate authority to read objects from the SDR can reference existing node groups. Any user can modify or reference node groups created by other users, provided they have the appropriate SDR authority. For more information on the authority needed to access the SDR, refer to “Authorization” on page 543.

When a node group is resolved into its flat version, only valid member nodes will appear in the flat version and only valid member node groups will be resolved recursively to produce the flat version. If a node group is a partition-bound node group, when it is flattened all of its member nodes and member node groups must be defined in the same system partition; if they are not, they will not be used to produce the flat version of this node group. If the node group is a global node group, when it is flattened all of its member nodes must be node numbers for the system and member node groups must also be defined as global node groups. Member nodes or member node groups that do not correspond to entities in persistent storage will be ignored when producing a flat version of the node group. A node group's flat version can be empty.

The following is a simple example of how a partition-bound node group will be resolved into a flat list of node numbers.

system partition P1 contains nodes 1, 2, 5, 6, 9, 1 , 13, 14
system partition P1 contains node groups A, B, C
Node Group A name = A; definition list = 1, 2, B, C, D
Node Group B name = B; definition list = 1, 2, 3, 4, 5, 6
Node Group C name = C; definition list = 13, 14
flat(A) = nodes 1, 2, 5, 6, 13, 14
flat(B) = nodes 1, 2, 5, 6
flat(C) = nodes 13, 14

Note: Nodes 3,4 are not in flat(A) or flat(B) because they are not part of partition P1.

Note: Node group D has no affect on the flat(A) because node group D is not defined in partition P1.

Working with Node Groups

A command line interface, SMIT panels, and the SP Perspectives GUI are provided for working with node groups. For a complete description of commands, refer to *IBM Parallel System Support Programs for AIX: Command and Technical Reference*. For information on using the SP Perspectives GUI to create and manage node groups, refer to the online help.

The following commands are provided for creating, modifying, and deleting node groups:

ngcreate Creates and optionally populates a named node group.

ngnew Creates new node groups in persistent storage.

ngaddto Adds specified nodes and node groups to named node group.

ngdelfrom Removes specified nodes and node groups from named node group.

ngdelete Removes node groups from persistent storage.

ngclean Removes nodes and node groups that do not currently exist from specified node groups or every node group in the current system partition.

The following commands are provided to display node group information to standard output:

ngresolve
Displays a list of nodes in the node group.

nglist Writes a list of all persistent node groups.

ngfind Writes a list of all node groups containing the supplied node or node group.

To access SMIT panels (using the fastpath invocation) for creating, modifying, and deleting node groups:

TYPE **smit nodegroups**

The Node Group Informaton menu appears.

SELECT Partition Node Groups OR System Node Groups

The Partition Node Groups or System Node Groups panel appears.

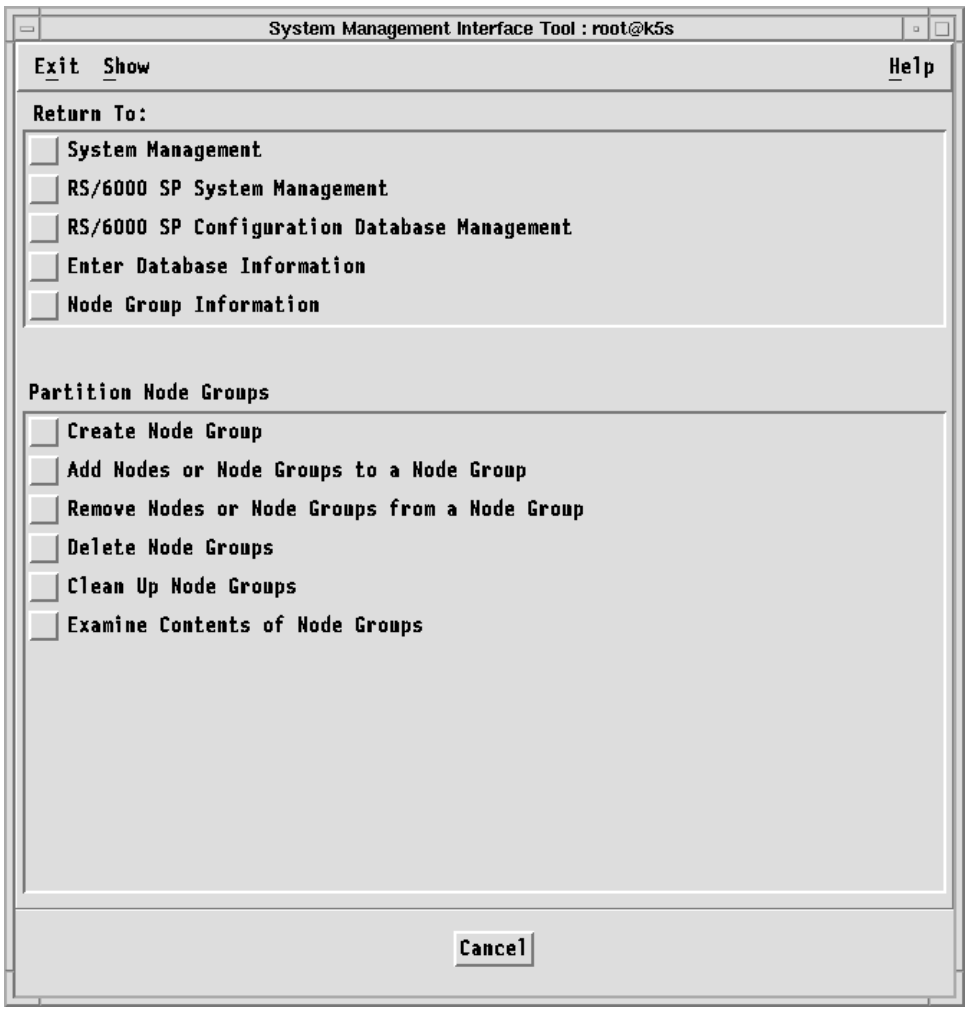


Figure 48. The Partition Node Groups SMIT Panel

To access SMIT panels (using the fastpath invocation) for examining existing node groups:

TYPE **smit list_nodegroup**

The List Node Group Database Information menu appears.

SELECT Examine Partition Node Groups OR Examine System Node Groups

The Examine Partition Node Groups or Examine System Node Groups panel appears.

To access SMIT panels (using the fastpath invocation) for deleting node groups:

TYPE **smit delete_nodegroup**

The Delete Node Group Information menu appears.

SELECT Delete Partition Node Groups OR Delete System Node Groups

The Delete Partition Node Groups or Delete System Node Groups panel appears.

Using Node Groups with Commands

The following commands will accept a node group as an argument. The node group may be supplied as an operand, if this is what is most consistent with the existing syntax. Some commands use the **-N** flag to indicate a node group. Refer to *IBM Parallel System Support Programs for AIX: Command and Technical Reference* for actual syntax for each command.

cshutdown
cstartup
dsh (can be used to run a command on every node in a flattened node group)
Efence
Eunfence
hostlist
pmandef
p-commands

The following commands all pass arguments to **hostlist**. **hostlist** accepts a node group as an argument, so these commands get node group support through **hostlist**:

- **p_cat**
- **pcp**
- **pdf**
- **pexec**
- **pexscr**
- **pfck**
- **pfind**
- **pfps**
- **pls**
- **pmv**
- **ppred**
- **pps**
- **prm**

spacctnd
spadaptrs
spbootins
spdeladap
spdelnode
spethernt
sphostnam
sphrdwrad
splst_versions
splstdata

System Monitoring

© Copyright IBM Corp. 1996

Chapter 19. Using the SP Perspectives GUI

Scalable POWERparallel Perspectives for AIX (SP Perspectives) is a graphical user interface (GUI) that enables you to perform system management tasks for your SP system through direct manipulation of system objects represented by icons.

You simply select an SP system object (a managed SP system resource such as a frame, node, or switch) by clicking on it with a mouse and then select an action to perform on that object from the menu or tool bar. Use this pattern of selecting an object and then selecting an action to perform on that object to accomplish numerous system management tasks with the SP Perspectives GUI.

This release of SP Perspectives provides function for hardware control and object monitoring. For example, you can use the SP Perspectives GUI to power on and off or reset hardware. Some configuration function is provided by the SP Perspectives GUI (for example, the ability to group nodes into node groups).

You can use the SP Perspectives GUI to perform the following system management tasks:

- Monitor and control hardware
- Create and monitor system events
- Define and manage IBM Virtual Shared Disks
- Generate and save system partition configurations
- Set up performance monitoring hierarchies and archiving

The SP Perspectives GUI also provides access to system management tools such as SMIT panels and AIX 4.1 Visual System Management (VSM) windows. In addition, you can create your own applications and make them accessible from the SP Perspectives GUI Launch Pad.

Because the SP Perspectives GUI provides access to a wider range of system management tasks, we recommend its use over the SP System Monitor.

This chapter presents overview information for the SP Perspectives GUI; it covers the following topics:

- Getting started
- Understanding the SP Perspectives GUI interface
- Accessing help for the SP Perspectives GUI

Information for the tasks you can perform using the various SP Perspectives applications is available through an extensive online help system (see "Accessing Help" on page 277) and is not repeated in this chapter.

Getting Started

This section provides information on authorizing users for the SP Perspectives GUI and starting the SP Perspectives GUI.

© Copyright IBM Corp. 1996

Authorizing Users for the SP Perspectives GUI

No special authorization is required to use the SP Perspectives GUI. However, the applications available from the SP Perspectives GUI may require special authorization.

In order to be able to control the SP system hardware or open a serial link from the Hardware perspective, an SP Perspectives client needs the appropriate **hardmon** ACLs (access control lists) set up on the control workstation in **/spdata/sys1/spmon/hmacls** (refer to the step on initializing authentication services in *IBM Parallel System Support Programs for AIX: Installation and Migration Guide*). Then the client must have a Kerberos token that has the desired **hardmon** authority in order to control the SP system hardware.

There are four commands that root can issue on the control workstation that a non-root user or a user on another node can be given access to:

cshutdown
cstartup
Efence
Eunfence
Estart

These commands must be run by root on the control workstation unless **sysctl** acls have been set up on the control workstation for the principal that the SP Perspectives client is using. Also, in order to be able to issue these commands through **sysctl**, the principal must have authority for **hardmon** for the frames that the command is issued for. To set up the **sysctl** acl for these commands, edit the file **/etc/sysctl.rootcmds.acl** and add the principal that you want to be authorized to issue these commands.

To create, modify, or delete node groups from the Hardware perspective, you must be root on the control workstation or an SP node. If you are root on an SP node, you can write to the SDR only for the system partition the node is in.

No special authorization is required for the Event perspective. However, to use the Problem Management subsystem from the SP Perspectives GUI (available under the Event perspective), your Kerberos principal must be added to the **/etc/sysctl.pman.acl** file on the control workstation and nodes. Add the principal to the file on the control workstation and copy it to the nodes by issuing:

```
pcp -a /etc/sysctl.pman.acl /etc
```

If you wish, you can use another way to distribute the ACL file, such as file collections. You may also wish to be more selective about what nodes will be authorized for Problem Management. For more information on security in Problem Management, refer to “Security Considerations” on page 372.

To work with the IBM Virtual Shared Disk and System Performance Tool perspectives, the same authorization as is described for node groups is required.

Starting the SP Perspectives GUI

To start the SP Perspectives GUI running as a background process:

1. Enter **perspectives &** (presuming that you have placed **/usr/lpp/ssp/bin** in your PATH) at the command line to bring up the Launch Pad.
2. Double-click on the icon for the application you want to run. The primary window (also known as the "perspective") for the application will open.

Alternatively:

1. Enter **perspectives &** (presuming that you have placed **/usr/lpp/ssp/bin** in your PATH) at the command line to bring up the Launch Pad.
2. Click on the icon for the application you want to run.
3. Click on the **Actions** menu bar item.
4. Click on **Launch**. The primary window (or "perspective") for the application will open.

   Note: This may take up to one minute, depending on the configuration of your control workstation.

To start SP Perspectives GUI applications (running as background processes) without using the Launch Pad, enter the following commands at the command line:

sphardware & Hardware

spevent & Event

spvsd & IBM Virtual Shared Disk

spsyspar & System Partitioning Aid

spperfmon & System Performance Tool

You can also start SP Perspectives GUI applications from the Common Desktop Environment (CDE). (Double-click on the SP Perspectives icon in the CDE Application Manager window.)

For a general overview of how to perform common SP system administration tasks with the SP Perspectives GUI, refer to "Performing Common Administration Tasks with the SP Perspectives GUI" on page 278.

Understanding the SP Perspectives GUI Interface

The SP Perspectives GUI consists of the following graphical elements:

- Launch Pad
- Primary windows (also known as "perspectives")
- Dialog boxes
- Notebooks
- Message boxes

The Launch Pad

The Launch Pad starts applications associated with managing an SP system. It also provides options for customizing the appearance of the Launch Pad.

System management applications provided by SP Perspectives are represented as icons on the Launch Pad. Other system management tools, such as SMIT panels and VSM windows, are also accessible from the Launch Pad. In addition, you can add your own applications to the Launch Pad.

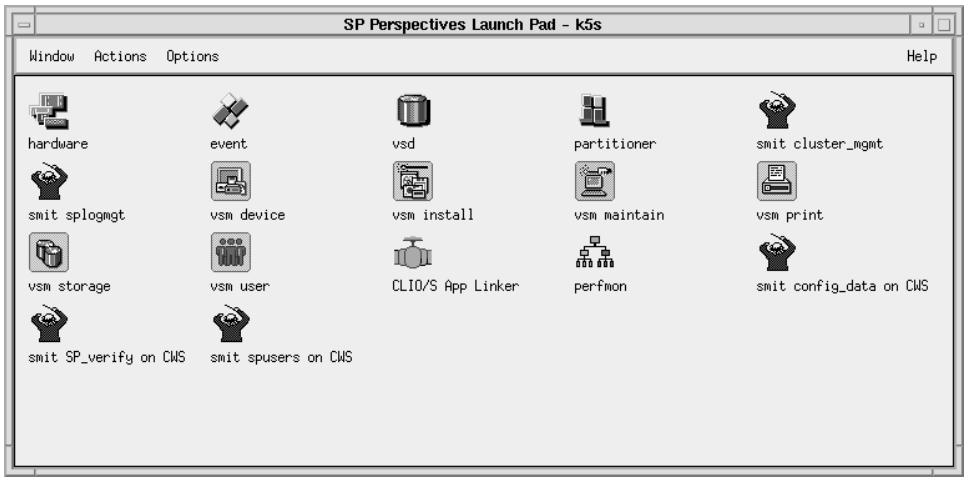


Figure 49. The SP Perspectives GUI Launch Pad

The Launch Pad window contains the following elements:

The title bar presents the name of the window and an identifier for the control workstation: **SP Perspectives Launch Pad** - *controlworkstation_name*

The menu bar contains menu items that provide for various actions:

- **Window**

  Close provides for closing the Launch Pad window. Closing the Launch Pad window does not affect application primary windows that are open.

- **Actions**

  - **Launch** starts the selected application (represented by an icon)

- **Options** (actions that operate on the presentation or appearance of the Launch Pad only)

  - **Customize Applications** extends the Launch Pad window to include a panel that provides for adding, deleting, or modifying applications on the Launch Pad. You specify a unique application name, a one-line application description, executable file name, and icon file name (a pixmap in xmp format) and select whether you want to add, delete, or modify the icon representing the application. The Launch Pad is updated immediately.

  - **Details View** presents the launch area in a way that displays icons representing applications, along with the application name, application description, and executable path.

  - **Load Preferences...** presents a dialog box that provides for loading files of preferences. You specify the system or user file name from

which to load the preferences and select aspects of preferences to load. Aspects of preferences include foreground and background colors, type font, window position and size, and applications.

- **Save Preferences...** presents a dialog box that provides for saving files of preferences. You specify the system or user file name to save the preferences to, and select aspects of preferences to save. Aspects of preferences include foreground and background colors, type font, window position and size, and applications.
- **Set Colors...** presents a dialog box that provides for setting colors. You choose to set the colors for the background and foreground and select the colors from sliders, a palette, or a list. A sample of the colors you select is dynamically displayed in the dialog box.
- **Set Font...** presents a dialog box that provides for specifying typeface style and size. You select a typeface family and point size and specify bold or italic. A sample of fonts you select is dynamically displayed in the dialog box.

– **Help**

The **Help** button provides options for information about windows, actions (tasks), and objects.

Primary Windows (Perspectives)

The primary window for each SP Perspectives application is called a “perspective” because it provides a unique view of your SP system. The Hardware perspective, for example, allows you to view the system as a set of hardware objects (for example, nodes, frames, switches). The Event perspective allows you to view the system in terms of conditions of interest related to system resources.

Primary windows contain work areas known as “panes”. Panes contain icons representing system objects (a managed SP system resource such as a node, frame, or switch).

To perform actions on the *display* of objects in panes:

1. Click on the pane to give it focus.
2. Select **View** from the menu bar.
3. Select the desired action from the pulldown menu. (Alternatively, some of the **View** options are available from the tool bar.)

To perform actions on the *objects themselves*:

1. Select one or more objects.
   – To select multiple objects that are next to each other in a pane, click on the icon of the first desired object, hold the left mouse button down, and move the cursor over the icons of other desired objects in the pane.
   – To select multiple objects that are not next to each other in a pane, hold the **Ctrl** key on your keyboard down while you click on each desired object in the pane.
2. Select **Actions** from the menu bar.
3. Select the name of the pane from the pulldown menu.

4. Select the desired action from the slide-right menu. (Alternatively, some of the **Actions** options are available from the tool bar.)

Each perspective consists of the same basic format as illustrated in Figure 50.

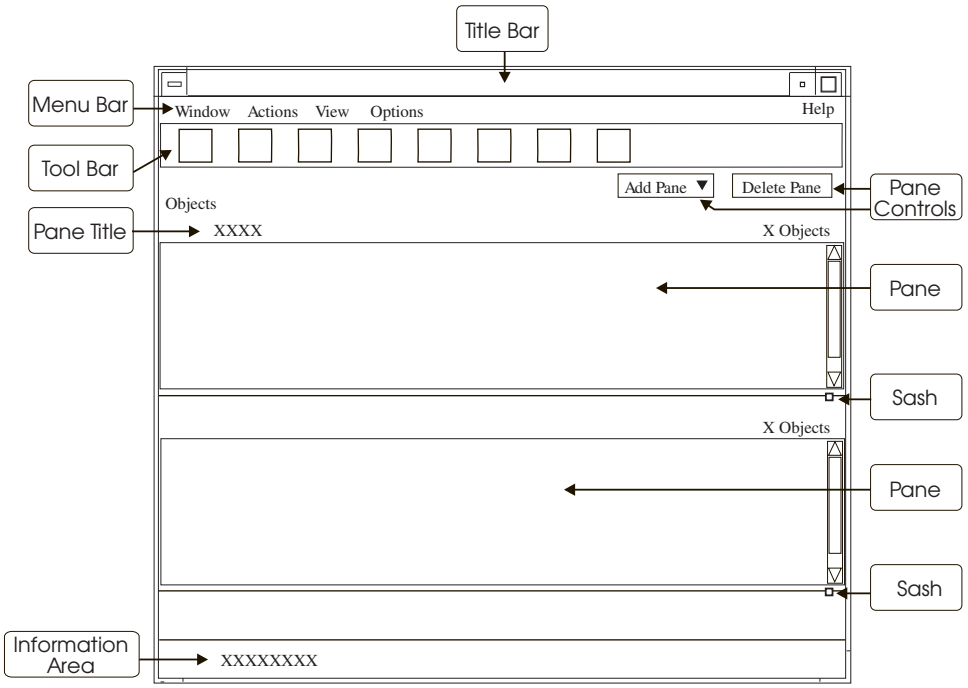


Figure 50. Elements of an SP Perspectives GUI Primary Window

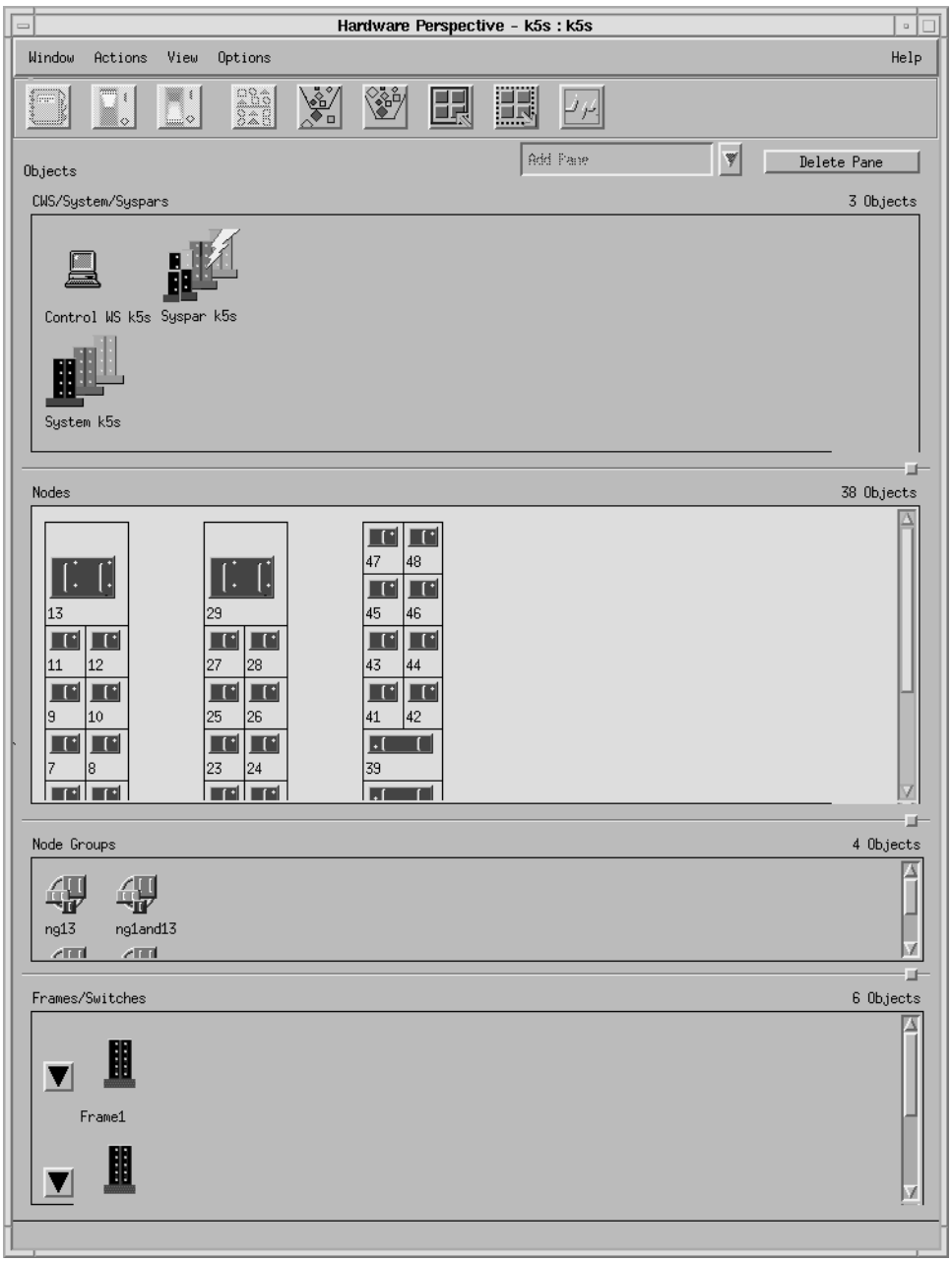


Figure 51. An SP Perspectives GUI Primary Window

The elements of this format are:

The title bar displays the name of the application, the name of the control workstation running SP Perspectives, and an identifier for the system. If the selected object is the **System** icon, the system is identified as “Global”. If the selected object is a **Syspar** icon, the system is identified by the system partition name. The format for title bar information is:

*application_name - controlworkstation_name:***Global** or *partition_name*

The menu bar contains menu items that provide for various actions:

- **Window** (actions that operate on a window as a whole)

Close provides for closing the window. Closing a primary window also closes any subordinate windows that have been opened for that perspective.

– **Actions** (actions that operate on SP system objects)

  Actions provides a list of panes against which actions are performed. For each pane in the list there is a slide-right menu that lists the actions available for objects within the pane. Actions that are unavailable are grayed out.

– **View** (actions that operate on the display of SP system objects in a selected pane)

  The following **View** actions are common to most of the SP Perspectives applications. The application determines what **View** options are available.

  - **Display/Hide Info Area** provides for making the Information Area viewable or not viewable.
  - **Sort...** presents a dialog box that provides for sorting objects by name in either ascending or descending order.
  - **Filter...** presents a dialog box that provides for specifying which objects are displayed in a pane. You can include or exclude objects by object name or by selecting objects.
  - **Reset** undoes a **Filter** action and redisplays all the objects in a pane.
  - **Select All** selects all objects in the selected pane.
  - **Deselect All** deselects all objects in the selected pane.
  - **Monitor...** provides a dynamic display of selected object status.
  - **Enable/Disable Tree View** displays objects in the selected pane in a 'tree' structure or free form.
  - **Expand Tree** extends the 'tree' structure to display 'children' of all the objects in the currently selected pane.
  - **Collapse Tree** collapses the expanded 'tree' structure of all the objects in the currently selected pane.
  - **Object Label...** sets the text label for objects in the currently selected pane.
  - **Enable Icon View** provides a view of node objects as individual icons as opposed to nodes in frames.

– **Options** (actions that operate on the presentation or appearance of individual primary windows)

  - **Load Preferences...** presents a dialog box that provides for loading files of preferences. You specify the system or user file name from which to load the preferences and select aspects of preferences to load. Aspects of preferences include foreground and background colors, type font, and window position and size.
  - **Save Preferences...** presents a dialog box that provides for saving files of preferences. You specify the system or user file name to save the preferences to, and select aspects of preferences to save. Aspects of preferences include foreground and background colors, type font, and window position and size.

- **Set Colors...** presents a dialog box that provides for setting colors. You choose to set the colors for the background and foreground and select the colors from sliders, a palette, or a list. A sample of the colors you select is dynamically displayed in the dialog box.
- **Set Font...** presents a dialog box that provides for specifying typeface style and size. You select a typeface family and point size and specify bold or italic. A sample of fonts you select is dynamically displayed in the dialog box.

– **Help**

The **Help** button provides options for information about windows, areas, actions (tasks), and objects.

The tool bar displays icons for often-used actions against selected objects and against the display of objects. These actions are a subset of those available from the menu bar.

Two pane controls enable you to add and delete panes.

Panes are sections of the window work area that contain icons that represent system objects. Typically, but not always, there is one type of system object per pane. System objects in a pane can be manipulated using actions available in the menu bar or tool bar.

A sash (a small square and a separator line) provides for controlling the relative size of the panes above and below it.

The Information Area, the bottom line of the window, displays information about the object or area at the current cursor location. This display can be toggled on and off from the **View** menu.

Dialog Boxes

When you use an SP Perspectives GUI primary window to perform certain actions, a dialog box is presented. The dialog box solicits or displays additional information needed to complete the action. The specific information or actions contained in the dialog box depend on the application.

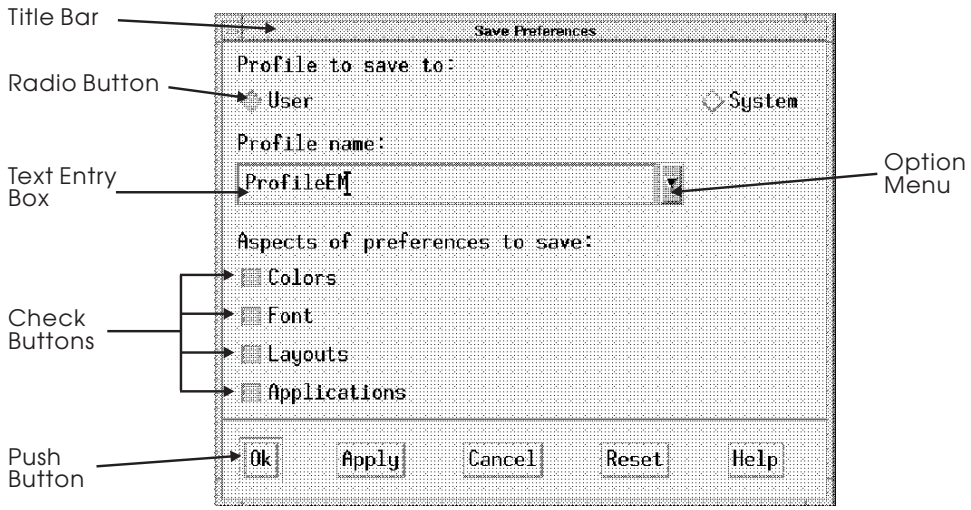


Figure 52. An SP Perspectives GUI Dialog Box

SP Perspectives GUI dialog boxes may contain the following elements (not every dialog box contains all of these elements):

The title bar displays the name of the action being performed, the object name or type, and, optionally, the system partition name.

Text entry boxes provide for entering or viewing text information.

Check buttons provide for selecting one or more choices from a list of options that are not mutually exclusive.

Radio buttons provide for making a single choice from a list of mutually exclusive options.

Option menus provide for entering information and selecting a single choice from a list of mutually exclusive options.

Push buttons provide for selecting an action:

- **OK** performs the action based on information entered or selected in the dialog box and closes the dialog box.
- **Apply** performs any changes made to information entered or selected in the dialog box and leaves the dialog box open.
- **Cancel** resets any changes made to information in the dialog box to the original values displayed when the dialog box was opened and closes the dialog box.
- **Reset** returns any changes made to information in the dialog box to the last applied values and leaves the dialog box open.
- **Help** displays general help about the purpose of and items in the dialog box.

Notebooks

An SP Perspectives GUI notebook is a type of dialog box that is in a notebook format. Each page in the notebook contains attribute information (defining or operational characteristics) about system objects. Like attributes are grouped on the same page and “Tabs” provide a visual identification of the category of attributes on the page. The specific attribute information contained on a notebook page depends on what the selected system object is (for example, the hardware or a system event).

Notebooks are provided as a mechanism for viewing or altering object attributes.

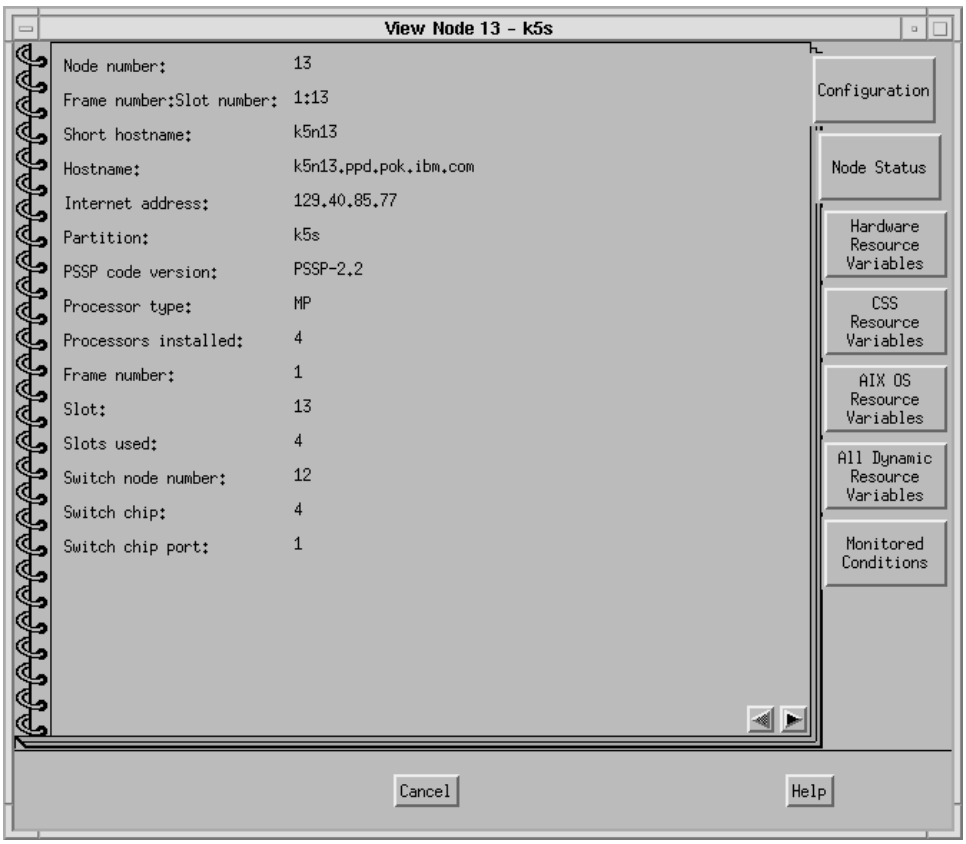


Figure 53. An SP Perspectives GUI Notebook

SP Perspectives GUI notebooks may contain the following elements (not every notebook contains all of these elements):

The title bar displays the action (view, modify), the object name or type, and, optionally, the system partition name.

Text entry boxes provide for entering text information for object attributes that can be modified.

Static text entry fields provide for displaying object attribute information that cannot be modified (view only).

Check buttons provide for selecting one or more choices from a list of options that are not mutually exclusive.

Radio buttons provide for making a single choice from a list of mutually exclusive options.

Option menus provide for entering information and selecting a single choice from a list of mutually exclusive options.

Push buttons provide for selecting an action:

- **OK** updates attribute information based on data entered in the notebook page and closes the notebook.
- **Apply** performs any changes made to information entered in the notebook and leaves the notebook open.

- **Cancel** resets any changes made to information in the notebook to the original values displayed when the notebook was opened and closes the notebook.
- **Reset** returns any changes made to information in the notebook to the last applied values and leaves the notebook open.
- **Help** displays general help about the purpose of items in the notebook.

Message Boxes

A message box is a type of dialog box used to convey a message to you. There are four types of messages:

Informational

Error

Fatal

Warning

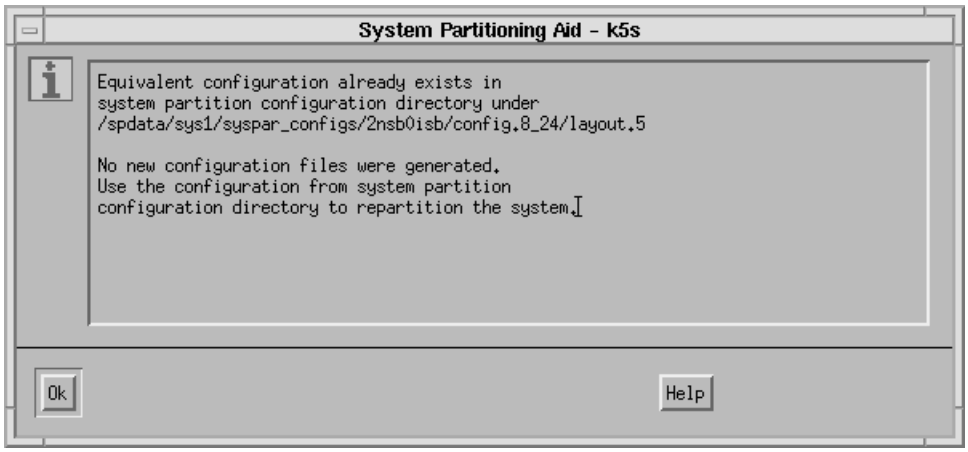


Figure 54. An SP Perspectives GUI Message Box

Elements common to SP Perspectives GUI message boxes are:

The title bar displays the name of the perspective.

A symbol represents the message type:

- Informational message

- Error message / Fatal message

- Warning message

The text of the message.

Push buttons provide for selecting an action, depending on the type of message:

- Informational, Error, Fatal
  - **OK** closes the message box.
  - **Help** displays extended information on the contents of the message and provides a way to view internal diagnostics information.
- Warning
  - **Enter**/**OK** confirms the action.
  - **Cancel** stops the action.
  - **Help** displays extended information on the contents of the message and provides a way to view internal diagnostics information.

Accessing Help

You can access help for the SP Perspectives GUI from:

The menu bar **Help** button

The Information Area

The dialog box, notebook, and message box **Help** buttons

The Menu Bar Help Button

The menu bar **Help** button contains items that provide information about windows, actions (tasks), and objects. To display the menu items, click on **Help**.

Getting Started provides information on starting the SP Perspectives GUI.

Overview activates a standalone help volume that contains overview and task-oriented information for the SP Perspectives GUI.

Contents displays a list of hyper-linked help topics for the SP Perspectives GUI.

Tasks accesses task-oriented help information for the perspective that you are in when you click on **Tasks**.

Using Help provides information on how to access help for the SP Perspectives GUI.

About SP Perspectives displays the name and version of the application.

Give Feedback to IBM provides information about how to contact the SP Perspectives team with your comments.

The Information Area

The Information Area is present on primary windows and displays information about the object or area at the current cursor location. This display can be toggled on and off from the **View** menu.

The Dialog Box, Notebook, and Message Box Help Buttons

The dialog box and notebook **Help** buttons display general information about the purpose of and items in the dialog box or notebook. The message box **Help** button displays extended information on the contents of the message and provides a way to view internal diagnostics information.

Performing Common Administration Tasks with the SP Perspectives GUI

The following table provides an overview of how to perform common SP system administration tasks using the SP Perspectives GUI. **SELECT** indicates clicking on an icon or a menu choice; **PRESS** indicates a button press; → indicates a menu item sequence.

Task	Perspectives
Bringing up the Launch Pad	**TYPE** perspectives
Launching applications	Double click on application icon in Launch Pad or 1. **SELECT** application icon 2. **SELECT** Actions → Launch
Closing the Launch Pad / Closing a perspective	**SELECT** Window→ Close
Powering off one node	From hardware perspective menu bar 1. **SELECT** node to be powered off 2. **SELECT** Actions → Nodes → Power Off... 3. **SELECT** power off options 4. **PRESS** Apply or From hardware perspective tool bar 1. **SELECT** node to be powered off 2. **PRESS** Power off icon from Tool bar 3. **SELECT** power off options 4. **PRESS** Apply

Task	Perspectives
Powering off more than one node	From hardware perspective menu bar 1. **SELECT** nodes to be powered off 2. **SELECT** Actions → Nodes → Power Off... 3. **SELECT** power off options 4. **PRESS** Apply or From hardware perspective menu bar with Node Groups 1. **SELECT** node group to be powered off 2. **SELECT** Actions → Node Group → Power Off... 3. **SELECT** power off options 4. **PRESS** Apply or From hardware perspective tool bar 1. **SELECT** nodes to be powered off 2. **PRESS** Power off icon from Tool bar 3. **SELECT** power off options 4. **PRESS** Apply
Powering on one node	From hardware perspective menu bar 1. **SELECT** node to be powered on 2. **SELECT** Actions → Nodes → Power On... 3. **SELECT** power on options 4. **PRESS** Apply or From hardware perspective tool bar 1. **SELECT** node to be powered on 2. **PRESS** Power on icon from Tool bar 3. **SELECT** power on options 4. **PRESS** Apply
Powering on more than one node	From hardware perspective menu bar 1. **SELECT** nodes to be powered on 2. **SELECT** Actions → Nodes → Power On... 3. **SELECT** power on options 4. **PRESS** Apply or From hardware perspective menu bar with Node Groups 1. **SELECT** node groups to be powered on 2. **SELECT** Actions → Node Groups → Power On... 3. **SELECT** power on options 4. **PRESS** Apply or From hardware perspective tool bar 1. **SELECT** nodes to be powered on 2. **PRESS** Power on icon from Tool bar 3. **SELECT** power on options 4. **PRESS** Apply

Task	Perspectives
Verifying node status for many nodes: hostResponds, powerLED, switchResponds, and 3 digit display	From hardware perspective menu bar 1. Click on the Nodes pane 2. **SELECT** View→ Monitor 3. **SELECT** hostResponds 4. **PRESS** Apply Repeat these steps for individual viewing of powerLED and switchResponds status, or select all three. Check to see if any node icons do not turn green. If a red X is showing, then there is a problem with that node. Three digit display is always displayed in it's own window. Bring up the 3 digit display: 1. Click on the Nodes pane 2. **SELECT** Actions→ Nodes→ 3 digit display
Verifying node status for one node: hostResponds, powerLED, switchResponds, and 3 digit display	From hardware perspective node notebook 1. Double click on a node 2. **SELECT** Node Status tab The Node Status page does not display the value of the 3 digit display for the node. To bring up the 3 digit display: 1. Click on the Nodes pane 2. **SELECT** Actions→ Nodes→ 3 digit display
Viewing node information	1. Double click on a node 2. **SELECT** Tab corresponding to desired page
Changing to the system (global) view	From hardware perspective menu bar 1. **SELECT** System icon 2. **SELECT** Actions→CWS/System/Syspars→Set Current Partition
Changing to a partition (syspar) view	From hardware perspective menu bar 1. **SELECT** Syspar icon 2. **SELECT** Actions→CWS/System/Syspars→ Set Current Partition

Chapter 20. Using the SP System Monitor

The SP System Monitor provides a way for authorized operators and administrators to operate and monitor the system hardware. Because the SP Perspectives GUI provides access to a wider range of system management tasks, we recommend its use over the SP System Monitor. For more information on the SP Perspectives GUI, refer to Chapter 19, “Using the SP Perspectives GUI” on page 265.

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 55 on page 282 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. 1996

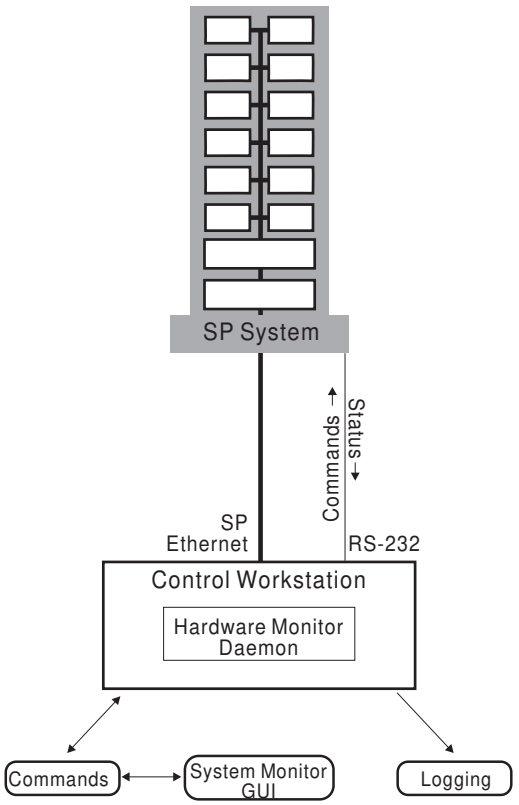


Figure 55. The SP System Monitor

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 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 513. 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

Note: Some environment information, such as fan speed and voltage, is not available for a 604 High Node.

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

Note: On SP Switch systems, you must use the **Eclock** command to change the 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 14, "Security Features of the SP System" on page 151 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**.

Invoke the **hmadm setacls** command after the ACL configuration file has been modified to update the hardware monitor daemon's internal ACL tables.

Refer to Chapter 14, "Security Features of the SP System" on page 151 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 512 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 14, "Security Features of the SP System" on page 151 for more information on SP authentication services.)

See the *IBM Parallel System Support Programs for AIX: 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 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 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 all three types of 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 284 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 RS/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:
```

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 14, “Security Features of the SP System” on page 151 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.

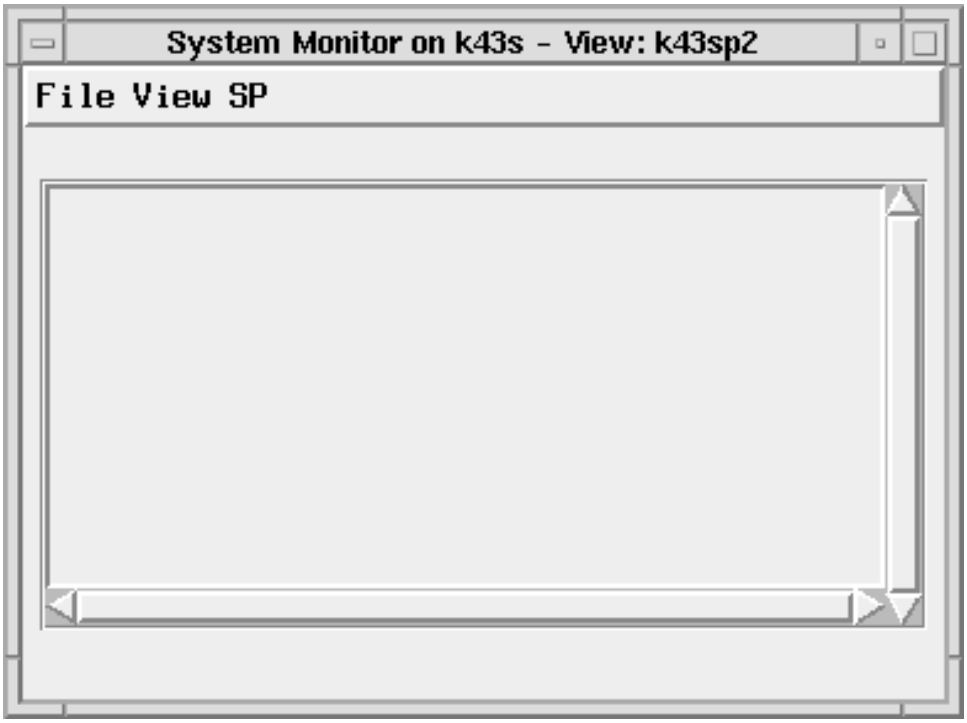


Figure 56. SP System Monitor GUI Main Window

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 57 on page 290.

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.

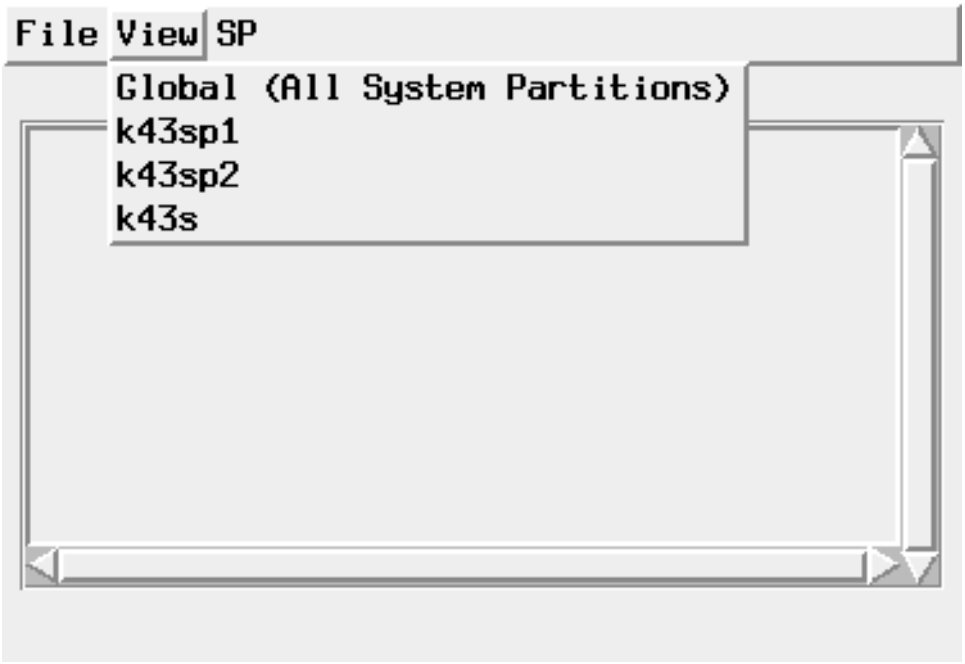


Figure 57. View Pulldown Menu

Step 4: Select SP

SELECT **SP**,

A pulldown menu appears, shown in Figure 58.

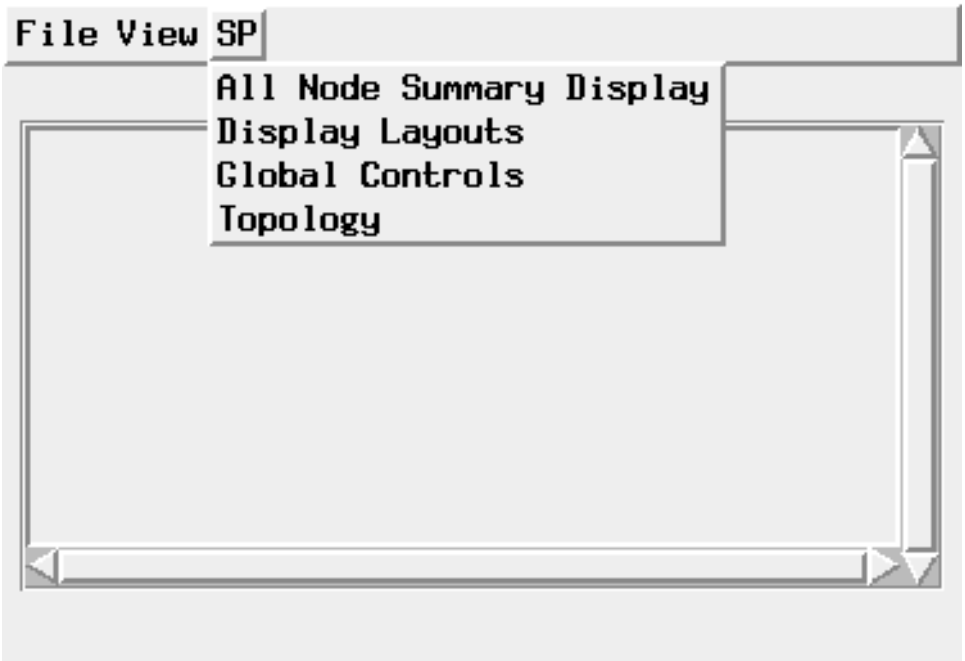


Figure 58. SP Pulldown Menu

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.

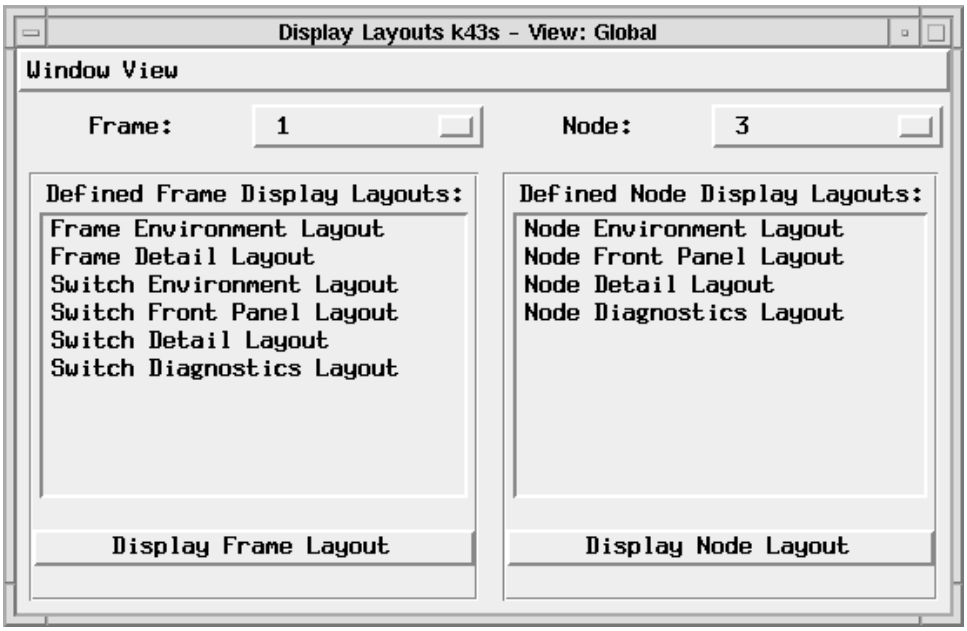


Figure 59. SP System Monitor GUI Display Layouts Selection Window

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 11 summarizes the information available on the layouts that come with the SP system.

Defined Layouts

Table 11. 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 (The 604 High Node does not have a Node Diagnostics layout.)

Monitoring and Controlling the Switch

You can use the Switch Front Panel layout to view the switch indicators and monitor and control the switch.

Note: On SP systems with more than one system partition, you must use the 'Global' view of the system to control the switch. A 'Global' view is required to 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

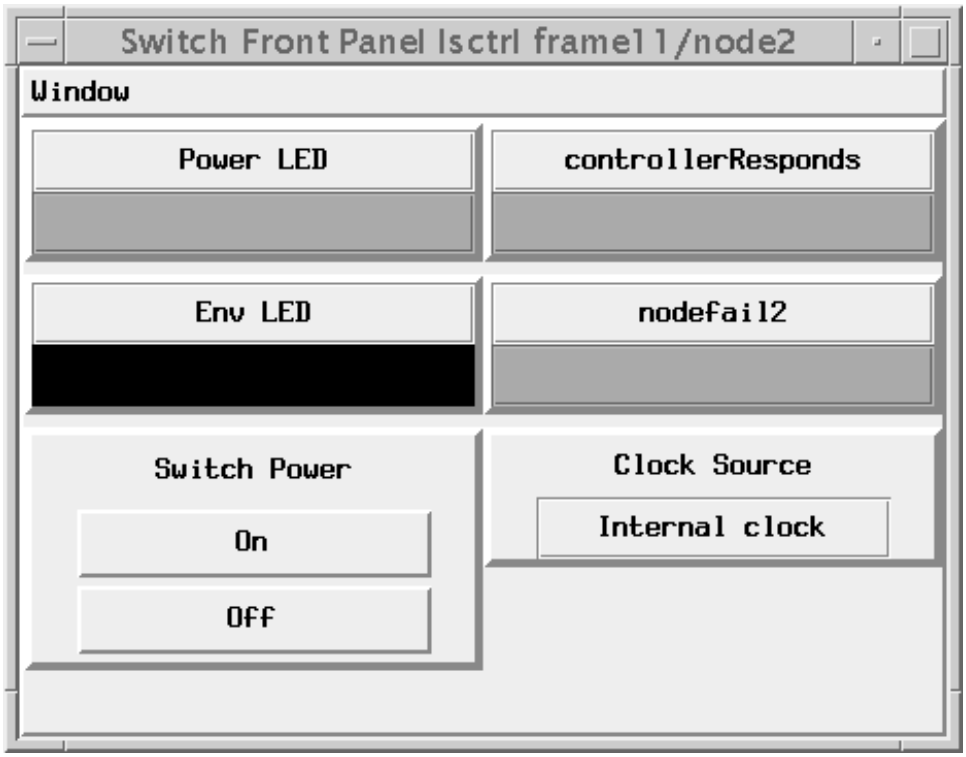


Figure 60. SP System Monitor GUI Switch Front Panel Layout Window

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.
Clock Source	Reflects the current clock source selected.

Notes:

1. We recommend terminating all applications that use the switch before powering off the switch.
2. You must use the **Eclock** command to change the clock source. See the *IBM Parallel System Support Programs for AIX: Command and Technical Reference* for details on command syntax.

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 304.

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 291.
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 RS/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 RS/6000 Scalable POWERparallel Systems: Site Planning*, SC23-3869, for more information on PCI.

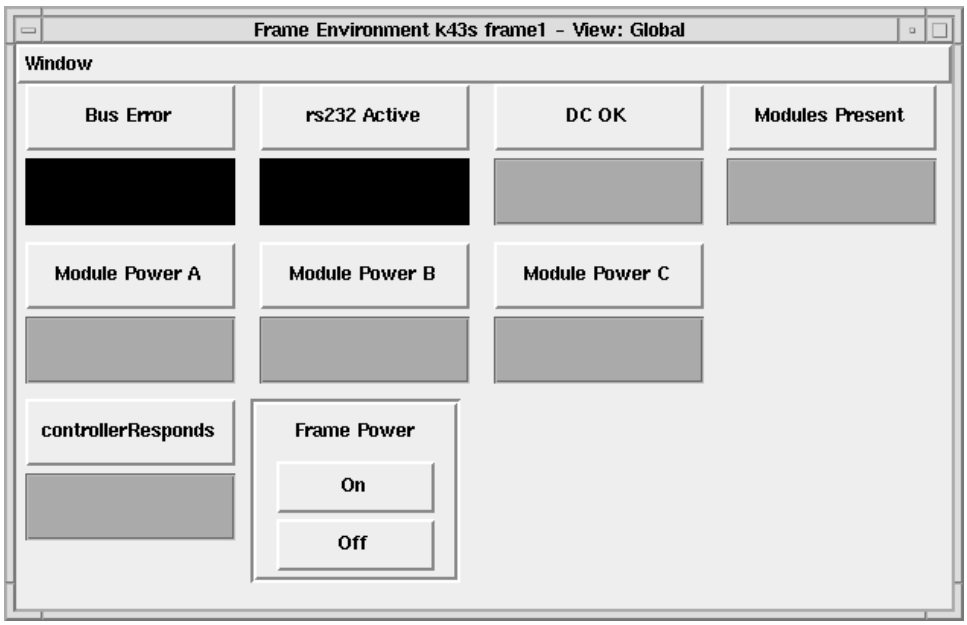


Figure 61. Frame Environment Layout for Type 18 Frame Supervisor

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).

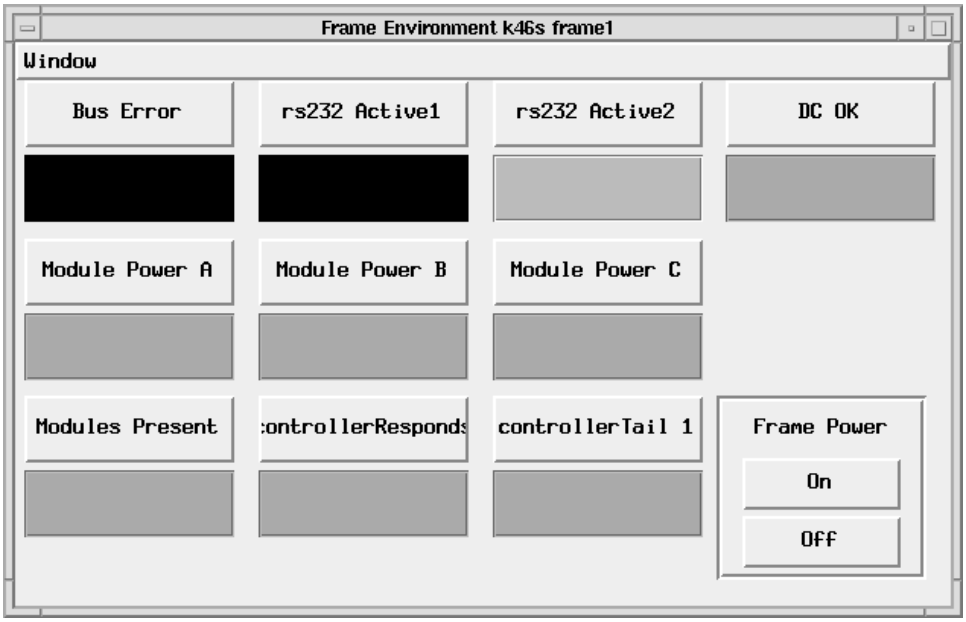


Figure 62. Frame Environment Layout for Type 22 Frame Supervisor

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 RS/6000 front control panel.

You can also use Front Panel controls to isolate nodes from the 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 291.

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.

Note: If you do not have a switch installed on your system, Fence and Unfence will not appear on the Node Front Panel Display.

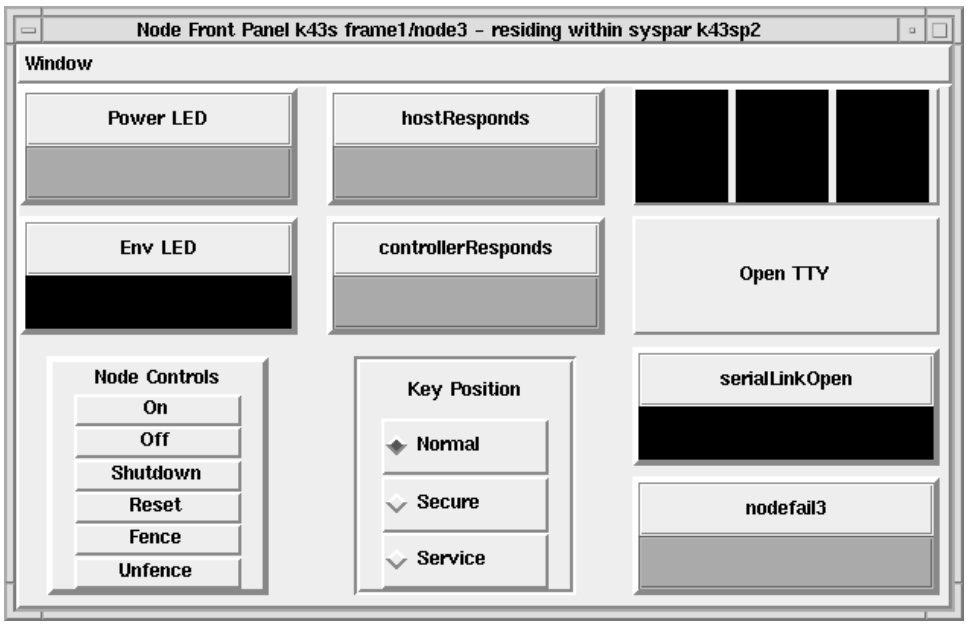


Figure 63. Node Front Panel Display

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	Fences the node from the switch
Unfence	Returns the node to the 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

Note: Powering on the 604 High Node takes a longer time than powering on a thin or wide node. It may take 15 minutes to power on a 604 High Node.

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 297 or refer to "Using Global Controls" on page 304 for instructions on accessing these displays.)

Chapter 3, "Starting Up and Shutting Down the SP System" on page 19 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.

Note: If you do not have a switch installed on your system, Fence and Enable AutoJoin will not appear on the Power Off Menu.

Figure 64. Power Off Menu

PRESS Options:

Fence
: Fences the node from the switch

Enable AutoJoin (can be selected only in conjunction with Fence)
: Enables node to automatically rejoin the 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 65. 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 Parallel System Support Programs for AIX: Command and Technical Reference* for details on command syntax. Also, see the *IBM Parallel System Support Programs for AIX: Installation and Migration 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 Parallel System Support Programs for AIX: Diagnosis and Messages Guide* and the *IBM RS/6000 Problem Solving Guide* or the *IBM RS/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.

Note: If you do not have a switch installed on your system, Fence and Enable AutoJoin will not appear on the Reset Menu.

Figure 66. Reset Menu

PRESS Options:

Fence
Fences node from the switch

Enable AutoJoin (can be selected only in conjunction with Fence)
Enables node to automatically rejoin the 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 Parallel System Support Programs for AIX: Diagnosis and Messages Guide* and the *IBM RS/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.

Note: If you do not have a switch installed on your system, Fence and Enable AutoJoin will not appear on the Shutdown Menu.

Figure 67. 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

Fencing Nodes from the Switch

To use the Node Front Panel Display to fence nodes from the switch, take the following steps:

1. Select Fence

   Note: If you do not have a switch installed on your system, the Fence option will not appear on the Node Front Panel Display.

2. Select Fence

Figure 68. Fence Menu

Unfencing Nodes from the Switch

To use the Node Front Panel Display to unfence nodes from the switch, select Unfence.

Note: If you do not have a switch installed on your system, the Unfence option will not appear on the Node Front Panel Display.

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 306) to display changing LED status. Refer to "Node Front Panel Controls" on page 299, which describes how each of these options works, with the exception of Network Boot, or to "Monitoring and Controlling the Switch" on page 292 as appropriate.

You can issue commands from the Global Controls window to isolate nodes from the switch. Refer to "Fencing Nodes from the Switch" on page 306 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 Parallel System Support Programs for AIX: Installation and Migration 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.

Note: If you do not have a switch installed on your system, Fence and Unfence will not appear on the Global Controls Display menu.

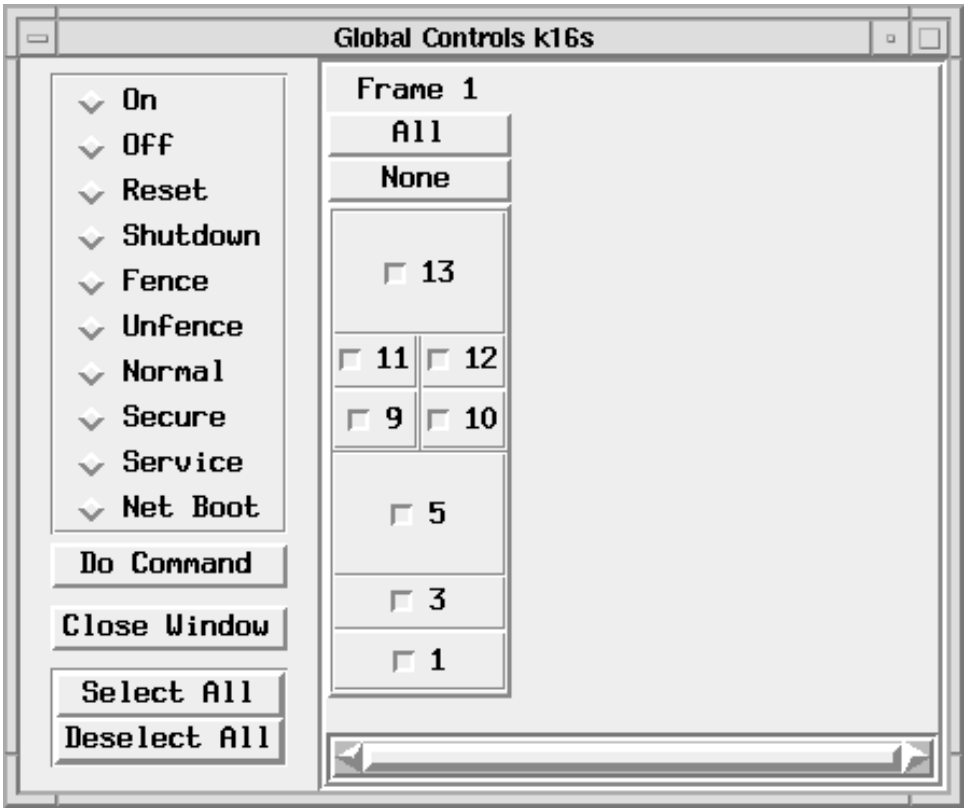


Figure 69. Global Controls Display

Frame 1 in this display contains two wide nodes, two 604 High Nodes, and four thin nodes.

Note: If a system is multipartitioned, a 'View' selection is available below the 'Deselect All' option; 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.

Fencing Nodes from the Switch

To use the Global Controls Display to isolate nodes from the 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 switch once nodes have restarted.

For additional information on isolating nodes from the switch, refer to Chapter 16, "Using a Switch" on page 211.

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 513 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.

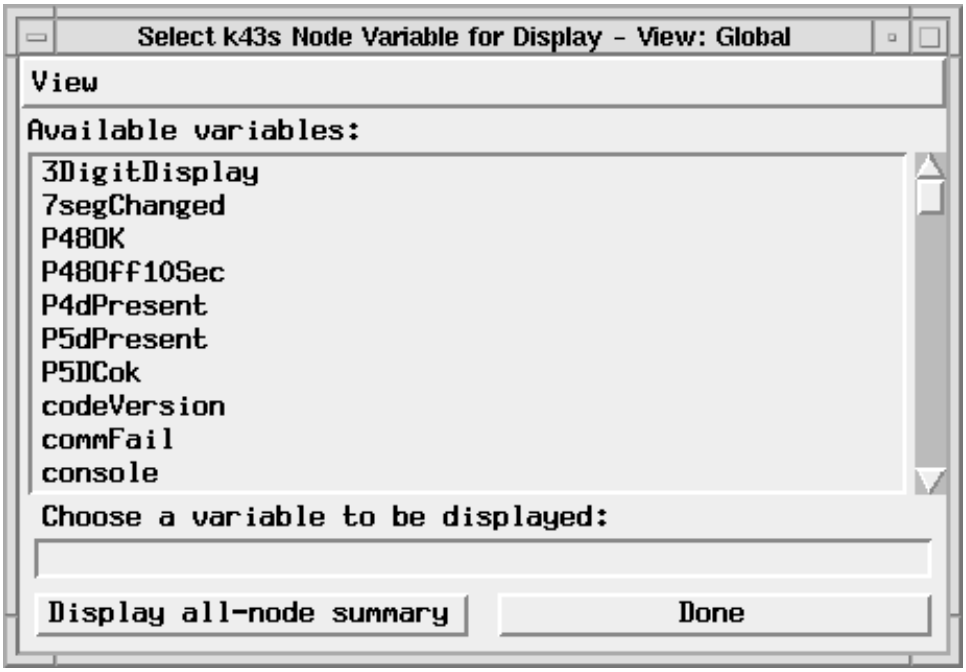


Figure 70. Node Variable Selection Window

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.

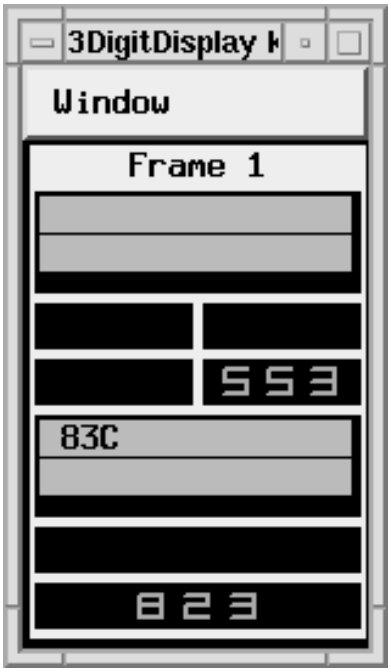


Figure 71. 3-Digit Display

This 3-Digit Display show a frame with four thin nodes, two wide nodes, and two 604 High Nodes. Each node shows the LED information. While thin and wide nodes support only 3-digit LED information, the 604 High Node supports a 16 character, 2 line LCD 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 12 on page 309 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 12. 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 RS/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.

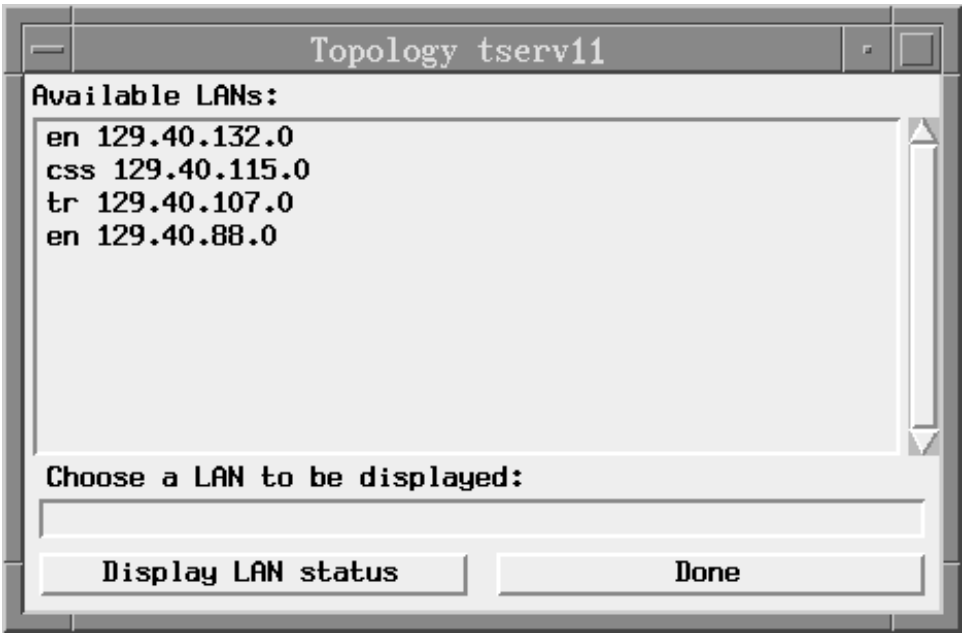


Figure 72. LAN Topology Display

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.

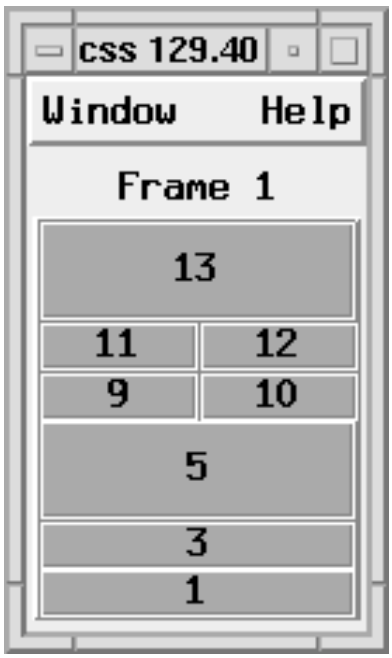


Figure 73. LAN Status Display

This display shows that wide nodes residing in slots 1 and 3, the 604 High Nodes residing in slots 5 and 13, and the thin nodes residing in slots 9, 10, 11, and 12 are configured for the chosen LAN.

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 13. 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 Parallel System Support Programs for AIX: Installation and Migration Guide*.

Step 2: Edit the /**spdata**/**sys1**/**spmon**/**hwevents** file to configure the logging options you want. Refer to **splogd** in the *IBM Parallel System Support Programs for AIX: 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 Parallel System Support Programs for AIX: 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 Parallel System Support Programs for AIX: Installation and Migration 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.

Availability and Problem Management

© Copyright IBM Corp. 1996

Chapter 21. Managing a High Availability Control Workstation Configuration

Notice

The High Availability Control Workstation hardware and software functions are separately priced features.

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 RS/6000 Scalable POWERparallel Systems: Site Planning* and *IBM Parallel System Support Programs for AIX: System Planning*.

We also recommend that you refer to the documentation for AIX High Availability Cluster Multi-Processing/6000 (HACMP/6000) Version 4.2 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 Parallel System Support Programs for AIX: 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. 1996

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 RS/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:

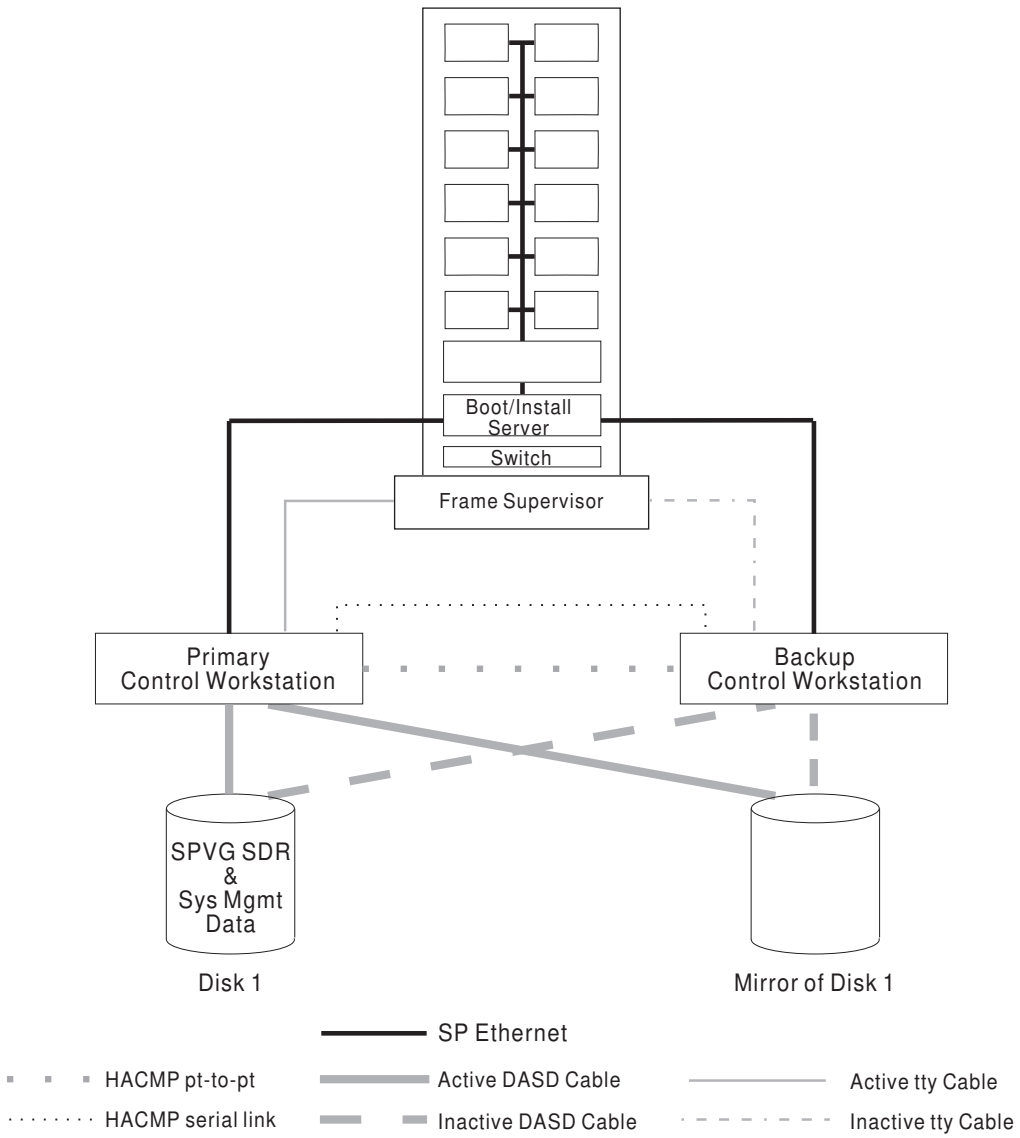


Figure 74. An HACWS Configuration

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 RS/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.

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 4.2. Only the high availability subsystem is required. The concurrent Resource Manager 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 4.2 is required for the control workstation.

Note: In an HACWS configuration, the nodes may be running any level of AIX and PSSP.

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. Remounts file systems
5. Restarts the control workstation applications
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 75 on page 322 for an illustration of addressing in an HACWS configuration.

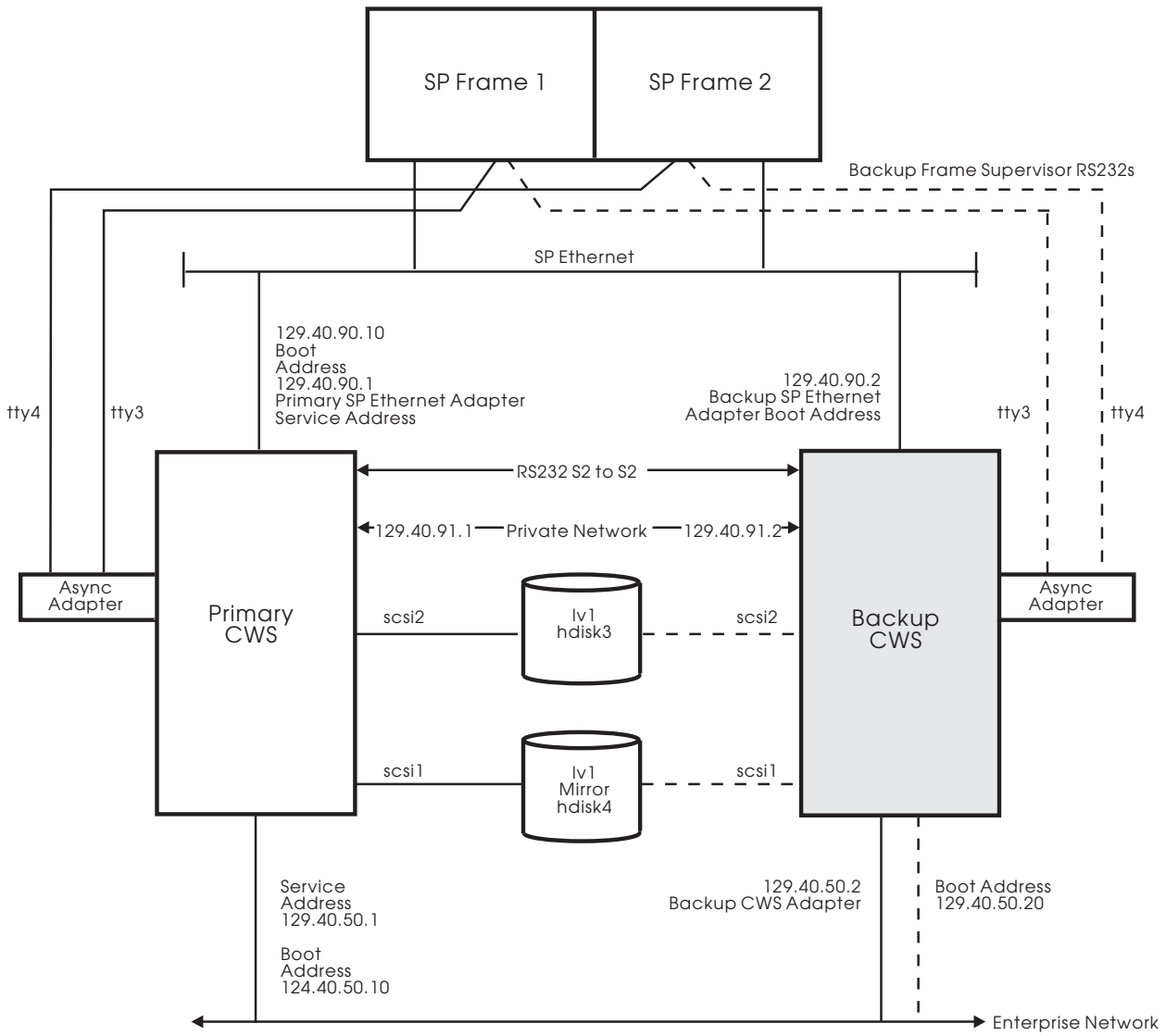


Figure 75. Addressing in an HACWS Configuration

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 RS/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 Configurations Without Standby Adapters

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:

```
18 .146.181.119
steamer
chowder
18 .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 14 (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 14 (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.

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 or SP Perspectives GUI

If you have installed **ssp.clients** or **ssp.gui** on a RS/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** or **perspectives** commands 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 or SP Perspectives GUI until you take another action. Once you take another action, the **spmon** or **perspectives** commands will exit. You will have to reexecute the **spmon** or **perspectives** commands 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** or **perspectives** commands 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.

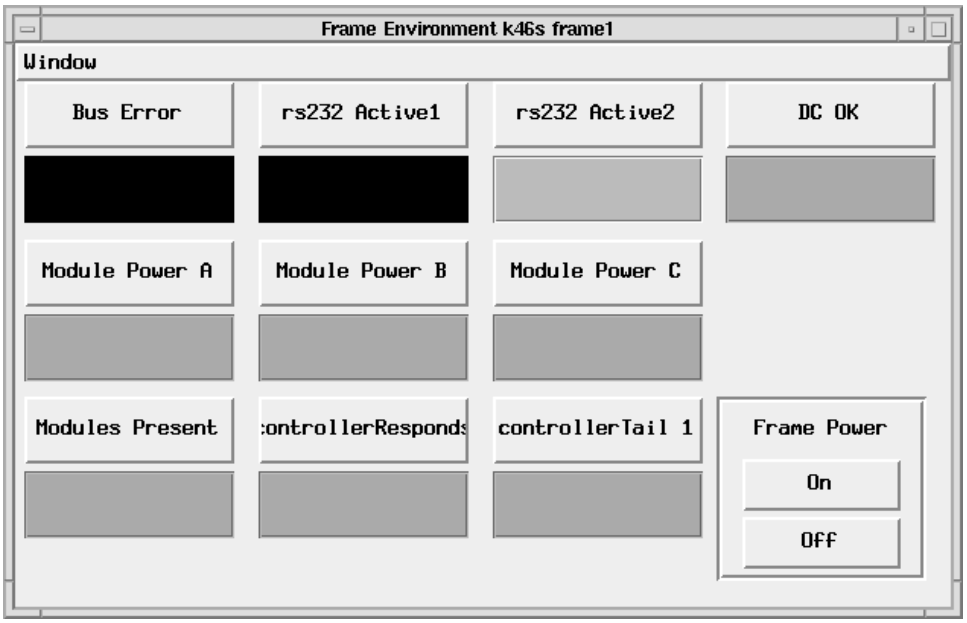


Figure 76. Frame Environment Layout Showing Active Control Workstation

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.

The SP Switch

When an SP Switch event occurs during a control workstation failover, the communication subsystem continues to handle switch faults, primary node failures, and primary backup node failures. System data about the switch is inaccessible during the control workstation failover and can become out-of-date. When the backup control workstation assumes the duties of the active control workstation, it must determine if a double failure has occurred (a switch event that occurred during a control workstation failover).

Perform the following to determine if a switch event has occurred and to resynchronize the communication subsystem with the system data on the new control workstation:

1. To determine if a primary node takeover (failure) has occurred, check all nodes in each system partition for a file named **act.top.1** in the **/var/adm/SPlogs/css** directory that contain a timestamp during the control workstation failover. If you locate an **act.top.1** file containing a timestamp during the control workstation failover:
   a. Run **/usr/lpp/ssp/css/rc.switch** on all the nodes in each system partition that has a node with an **act.top.1** file.
   b. Run **Estart** for each system partition that has a node with an **act.top.1** file.
2. To determine if a primary backup node failed or a switch fault occurred, examine the **flt** file in the **/var/adm/SPlogs/css** directory on the primary node for entries with a timestamp during the control workstation failover. If you locate entries that contain a timestamp during the control workstation failover, run **Estart** for each system partition having entries on its primary node.

   We recommend that you use an Error Notification Object for switch faults, primary node takeovers, and control workstation failures to send mail or a message to alert you about such double failures. You can then examine the communication subsystem as described previously. Because the timeframe and probability for these double failures are very small, we do not recommend that you run **rc.switch** or **Estart** whenever a control workstation 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
else

# Go run the command
$1
fi
exit
```

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 4.2 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.

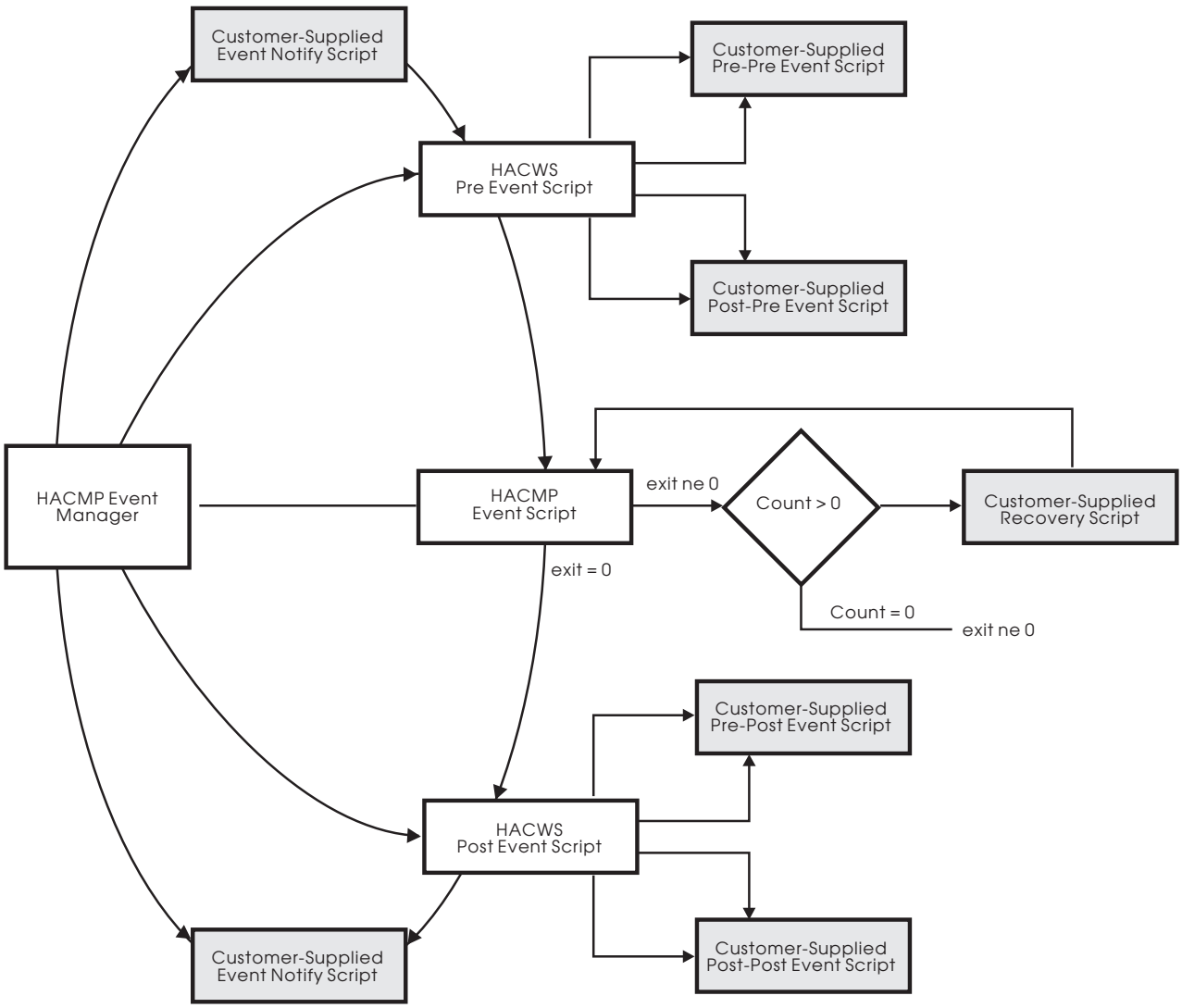


Figure 77. HACWS Pre- and Post-Events

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 22. The Group Services Subsystem

This chapter introduces you to the Group Services subsystem. It includes information about the subsystem's components, its configuration, other components on which it depends, and its operation. It also discusses the relationship of the Group Services subsystem to the other PSSP high availability subsystems. Finally, it describes a procedure you can use to check the status of the subsystem.

After reading this chapter, you will be able to manage the Group Services subsystem and, if necessary, perform problem determination.

Introducing Group Services

Group Services is a distributed subsystem of the IBM Parallel System Support Programs (PSSP) on the RS/6000 Scalable POWERparallel System (SP). It is one of several subsystems in PSSP that provide a set of high availability services. Other subsystems that provide high availability services include the Event Management and Topology Services (heartbeat) subsystems.

The function of the Group Services subsystem is to provide other subsystems with a distributed coordination and synchronization service. These other subsystems that depend upon Group Services are called **client subsystems**. Each client subsystem forms one or more **groups** by having its processes connect to the Group Services subsystem and use the various Group Services interfaces. A process of a client subsystem is called a **GS client**.

A group consists of two pieces of information:

- The list of processes that have joined the group, called the **group membership list**
- A client-specified **group state value**.

Group Services guarantees that all processes that are joined to a group see the same values for the group information, and that they see all changes to the group information in the same order. In addition, the processes may initiate changes to the group information via **protocols** that are controlled by Group Services.

A GS client that has joined a group is called a **provider**. A GS client that wishes only to monitor a group, without being able to initiate changes in the group, is called a **subscriber**.

Once a GS client has initialized its connection to Group Services, it may join a group and become a provider. All other GS clients that have already joined the group (those that have already become providers) are told as part of a join protocol about the new providers that wish to join. The existing providers may either accept new joiners unconditionally (by establishing a one-phase join protocol) or vote on the protocol (by establishing an n-phase protocol). During a vote, they can choose to approve the protocol and accept the new providers into the group, or reject the protocol and refuse to allow the new providers to join.

Group Services monitors the status of all of the processes that are joined to a group. If either the process or the node on which a process is executing fails,

© Copyright IBM Corp. 1996

Group Services initiates a failure protocol that informs the remaining providers in the group that one or more providers have been lost.

Join and failure protocols are used to modify the group's membership list. Any provider in the group may also propose protocols to modify the group's state value. All protocols may be either unconditional (one-phase) protocols, which are automatically approved and not voted on, or conditional (n-phase) protocols, which are voted on by the providers.

During each phase of an n-phase protocol, each provider is expected to establish or recover its resources. Then, it votes to approve, reject, or continue the protocol. The protocol completes when it is either approved (the proposed changes become established in the group), or rejected (the proposed changes are dropped).

For more conceptual information about the Group Services subsystem, see *PSSP Group Services Programming Guide and Reference*.

Group Services Components

The Group Services subsystem consists of the following components:

Group Services daemon
: The central component of the Group Services subsystem.

Group Services API (GSAPI)
: The application programming interface that GS clients use to obtain the services of the Group Services subsystem.

Port numbers
: TCP/IP port numbers that the Group Services subsystem uses for communications. The Group Services subsystem also uses Unix domain sockets.

Control script
: A shell script that is used to add, start, stop, and delete the Group Services subsystem, which operates under System Resource Controller (SRC) control.

Files and directories
: Various files and directories that are used by the Group Services subsystem to maintain run-time data.

The sections that follow contain more details about each of these components.

The Group Services Daemon (hagsd)

The Group Services daemon is contained in the executable file **/usr/lpp/ssp/bin/hagsd**. This daemon runs on each node of a system partition and on the control workstation. If there is more than one system partition, then multiple daemons run on the control workstation, one per system partition.

Note that the operational domain of the Group Services subsystem is a system partition. The control workstation is considered to be part of each domain. Unless otherwise stated, a reference to "the Group Services subsystem" is a reference to the Group Services subsystem in a single system partition.

A GS client communicates with the single Group Services daemon that is running on the same node as the GS client. A GS client communicates with the Group Services daemon, through the GSAPI, using a Unix domain socket. Before a GS

client registers with Group Services, it must set the **HA_SYSPAR_NAME** environment variable to the name of the system partition in which it is executing.

The Group Services API (GSAPI)

The Group Services Application Programming Interface (GSAPI) is a shared library that a GS client uses to obtain the services of the Group Services subsystem. This shared library is supplied in two versions: one for non-thread safe programs and one for thread-safe programs. These libraries are referenced by the following path names:

/usr/lib/libha_gs.a (non-thread safe version)

/usr/lib/libha_gs_r.a (thread-safe version)

These path names are actually symbolic links to **/usr/lpp/ssp/lib/libha_gs.a** and **/usr/lpp/ssp/lib/libha_gs_r.a**, respectively. The symbolic links are placed in **/usr/lib** for ease of use. For serviceability, the actual libraries are placed in the **/usr/lpp/ssp/lib** directory. These libraries are supplied as shared libraries, also for serviceability.

For details on the GSAPI, see *Group Services Programming Guide and Reference*.

Port Numbers and Sockets

The Group Services subsystem uses several types of communications:

UDP port numbers for intra-domain communications, that is, communications between Group Services daemons within an operational domain.

Unix domain sockets for communication between GS clients and the local Group Services daemon (via the GSAPI).

Intra-Domain Port Numbers

For communication between Group Services daemons within an operational domain, the Group Services subsystem uses a single UDP port number. This port number is recorded in the **Syspar_ports** SDR class. The class contains two attributes: **subsystem** and **port**. The class object whose **subsystem** attribute has a value of **hags** contains the port number for the Group Services subsystem. Note that the **Syspar_ports** class is a partitioned SDR class, that is, there is a set of objects for each system partition. When you configure the Group Services subsystem for a particular system partition, you use the SDR data that corresponds to that system partition.

The Group Services port number is stored in the SDR so that, when the Group Services subsystem is configured on each node, the port number is fetched from the SDR. This ensures that the same port number is used by all Group Services daemons in the same operational domain. Note that because a Group Services daemon from each operational domain runs on the control workstation, the Group Services subsystem in each system partition must have a unique port number.

This intra-domain port number is also set in the **/etc/services** file, using the service name **hags.***syspar_name*, where *syspar_name* is the name of the system partition. The **/etc/services** file is updated on all nodes in the system partition and on the control workstation. The Group Services daemon obtains the port number from the **/etc/services** file during initialization.

Unix Domain Sockets

Unix domain sockets are used for communication between GS clients and the local Group Services daemon (via the GSAPI). These are connection-oriented sockets. The following socket name is used (*syspar_name* is the name of the system partition):

/var/ha/soc/hagsdsocket.*syspar_name*
: Used by the GSAPI to connect to the Group Services daemon.

The Control Script (hagsctrl)

The Group Services control script is contained in the executable file **/usr/lpp/ssp/bin/hagsctrl**. This script is normally invoked by the **syspar_ctrl** script, which provides an interface to all of the system partition-sensitive subsystems.

If necessary, you can invoke **hagsctrl** directly from the command line. Note that before you invoke **hagsctrl**, you must ensure that the **SP_NAME** environment variable is set to the appropriate system partition name.

For more information about **hagsctrl** and **syspar_ctrl**, see *PSSP Command and Technical Reference*.

The purpose of the **hagsctrl** command is to add (configure) the Group Services subsystem to a system partition. It can also be used to remove the subsystem from a system partition, start the subsystem, stop it, and clean the subsystem from all system partitions.

For more information, see “Group Services Configuration” on page 342.

Files and Directories

The Group Services subsystem uses the following directories:

/var/ha/lck, for lock files

/var/ha/log, for log files

/var/ha/run, for Group Services daemon current working directories

/var/ha/soc, for socket files.

The /var/ha/lck Directory (Lock Files)

In the **/var/ha/lck** directory, the **hags.tid.***syspar_name* file is used to ensure a single running instance of the Group Services daemon, and to establish an instance number for each invocation of the daemon. In the file name, *syspar_name* is the name of the system partition. On the control workstation, there may be several instances of the Group Services daemon running, but they are in different operational domains.

The /var/ha/log Directory (Log Files)

The **/var/ha/log** directory contains trace output from the Group Services daemon.

On the nodes, the file is called **hags_***nodenum_instnum.syspar_name*, where:

nodenum is the node number on which the daemon is running

instnum is the instance number of the daemon.

syspar_name is the name of the system partition to which the node belongs.

On the control workstation, there is a file for each daemon that is running on the control workstation. Each file is called **hags.***syspar_name_nodenum_instnum.syspar_name*, where:

syspar_name is the name of the system partition in which the daemon is running.

nodenum is the node number on which the daemon is running

instnum is the instance number of the daemon.

The Group Services daemon limits the log size to a pre-established number of lines (by default, 10,000 lines). When the limit is reached, the daemon appends the string **.bak** to the name of the current log file and begins a new log. If a **.bak** version already exists, it is removed before the current log is renamed.

The /var/ha/run Directory (Daemon Working Files)

In the /**var**/**ha**/**run** directory, a directory called **hags.***syspar_name* is created, where *syspar_name* is the system partition name. This directory is the current working directory for the Group Services daemon. If the Group Services daemon abnormally terminates, the core dump file is placed in this directory. Whenever the Group Services daemon starts, it renames any core file to **core**_*nodenum.instnum*, where *nodenum* is the node number on which the daemon is running and *instnum* is the instance number of the daemon.

Components on Which Group Services Depends

The Group Services subsystem depends on the following components:

System Resource Controller (SRC)
: An AIX feature that can be used to define and control subsystems. The Group Services subsystem is called **hags** on SP nodes and is called **hags.***syspar_name* on the control workstation, where *syspar_name* is the name of a system partition. The subsystem name is used with the SRC commands (for example, **startsrc** and **lssrc**).

System Data Repository (SDR)
: A repository of SP system configuration information.

Topology Services
: A PSSP subsystem that is used to determine which nodes in a system can be reached (that is, are running) at any given time. It is often referred to as **heartbeat**. The Topology Services subsystem is SRC-controlled. It is called **hats** on SP nodes and is called **hats.***syspar_name* on the control workstation, where *syspar_name* is the name of a system partition.

Group Services Configuration and Operation

The following sections describe how the components of the Group Services subsystem work together to provide group services. Included are discussions of Group Services:

- Configuration
- Daemon initialization and errors
- Operation.

Group Services Configuration

The Group Services subsystem is installed as part of the installation of the PSSP product. The Group Services subsystem is contained in the **ssp.ha** fileset.

After the components are installed, the subsystem must be configured for operation. Group Services configuration is performed by the **hagsctrl** command, which is invoked by the **syspar_ctrl** command.

The **syspar_ctrl** command configures all of the system partition-sensitive subsystems. The person who installs PSSP issues the **syspar_ctrl** command during installation of the control workstation. The **syspar_ctrl** command is executed automatically on the nodes when the nodes are installed. The **syspar_ctrl** command is also executed automatically when system partitions are created or destroyed. For more information on using the **syspar_ctrl** command, see *PSSP Installation and Migration Guide* and *PSSP Command and Technical Reference*.

The **hagsctrl** command provides a number of functions for controlling the operation of the Group Services system. You can use it to:

- Add (configure) the Group Services subsystem
- Start the subsystem
- Stop the subsystem
- Delete (unconfigure) the subsystem
- "Clean" all Group Services subsystems
- Turn tracing of the Group Services daemon on or off

Except for the clean function, **hagsctrl** affects the Group Services subsystem in the current system partition, that is, the system partition that is specified by the **SP_NAME** environment variable.

Adding the Subsystem

If the **hagsctrl** command is running on the control workstation, the first step in the add function is to select a Group Services daemon communications port number and save it in the **Syspar_ports** SDR class. This port number is then placed in the **/etc/services** file. If the **hagsctrl** command is running on a node, the port number is fetched from the SDR and placed in the **/etc/services** file. Port numbers are selected from the range 10000 through 10100.

The second step is to add the Group Services daemon to the System Resource Controller (SRC) using the **mkssys** command. On the control workstation, the IP

address of the system partition is an argument to the **hagsd** program in the SRC subsystem specification.

The third step is to add an entry to the /**etc**/**inittab** file so that the Group Services daemon will be started during boot. However, if **hagsctrl** is running on a High Availability Control Workstation (HACWS), no entry is made in the /**etc**/**inittab** file. Instead, HACWS manages starting and stopping the Group Services daemon.

Note that if the **hagsctrl** add function terminates with an error, the command can be rerun after the problem is fixed. The command takes into account any steps that already completed successfully.

Starting and Stopping the Subsystem

The start and stop functions of the **hagsctrl** command simply run the **startsrc** and **stopsrc** commands, respectively. However, **hagsctrl** automatically specifies the subsystem argument to these SRC commands.

Deleting the Subsystem

The delete function of the **hagsctrl** command removes the subsystem from the SRC, removes the entry from /**etc**/**inittab**, and removes the Group Services daemon communications port number from /**etc**/**services**. It does **not** remove anything from the SDR, because the Group Services subsystem may still be configured on other nodes in the operational domain.

Cleaning the Subsystem

The clean function of the **hagsctrl** command performs the same function as the delete function, except in all system partitions. In addition, it removes the Group Services daemon remote client communications port number from the /**etc**/**services** file.

The clean function does **not** remove anything from the SDR. This function is provided to support restoring the system to a known state, where the known state is the (possibly restored copy of the) SDR database.

Tracing the Subsystem

The tracing function of the **hagsctrl** command is provided to supply additional problem determination information when it is requested by the IBM Support Center. Normally, tracing should **not** be turned on, because it may slightly degrade Group Services subsystem performance and can consume large amounts of disk space in the /**var** file system.

Group Services Daemon Initialization

Normally, the Group Services daemon is started by an entry in the /**etc**/**inittab** file, via the **startsrc** command. If necessary, the Group Services daemon can be started using the **hagsctrl** command or the **startsrc** command directly.

During its initialization, the Group Services daemon performs the following steps:

1. It gets the number of the node on which it is running using the /**usr**/**lpp**/**ssp**/**install**/**bin**/**node_number** command. Node 0 is the control workstation.
2. It fetches the name of the system partition from the **Syspar** SDR class. (Recall that one instance of the Group Services daemon runs on the control

workstation for each system partition to which the Group Services subsystem was added.)

3. It tries to connect to the Topology Services subsystem. If the connection cannot be established because the Topology Services subsystem is not running, it is scheduled to be retried in 1 second. This continues until the connection to Topology Services is established. Until the connection is established, the Group Services daemon writes an AIX error log entry every minute and no clients may connect to the Group Services subsystem.
4. It performs actions that are necessary to become a daemon. This includes establishing communications with the SRC subsystem so that it can return status in response to SRC commands.
5. It establishes the Group Services domain, which is the set of nodes within the SP system partition in which a Group Services daemon is executing.

   At this point, one of the GS daemons establishes itself as the GS nameserver. For details, see "Establishing the GS Nameserver."

   Until the domain is established, no GS client requests to join or subscribe to groups are processed.
6. It enters the main control loop.

   In this loop, the Group Services daemon waits for requests from GS clients, messages from other Group Services daemons, messages from the Topology Services subsystem, and requests from the SRC for status.

Establishing the GS Nameserver

The Group Services subsystem must be able to keep track of the groups that its clients want to form. To do this, it establishes a GS nameserver within each domain (a domain is the set of running nodes within each SP system partition). The GS nameserver is responsible for keeping track of all client groups that are created in that domain.

To ensure that only one node becomes a GS nameserver, Group Services uses the following protocol:

1. When each daemon is connected to the Topology Services subsystem, it waits for Topology Services to tell it which nodes are currently running in this system partition.
2. Based on the input from Topology Services, each daemon finds the lowest-numbered running node in the partition. The daemon compares its own node number to the lowest-numbered node and performs one of the following:

   If the daemon's node is the lowest-numbered node, the daemon waits for all other running nodes to nominate it as the GS nameserver.

   If the daemon's node is not the lowest-numbered node, it starts sending nomination messages to the lowest-numbered node every 5 seconds.
3. Once all running nodes have nominated the GS nameserver-to-be and a 20-second timer has expired, the nominee sends an insert message to the nodes. All nodes must acknowledge this message. When they do, the nominee becomes the established GS nameserver, and it sends a commit message to all of the nodes.
4. At this point, the Group Services domain is established, and requests by clients to join or subscribe to groups are processed.

Note that this description is in effect when all nodes are being booted simultaneously, such as at initial system power-on. It is often the case, however, that a Group Services daemon is already running on at least one node (for example, the control workstation) and is already established as the domain's GS nameserver. In that case, the GS nameserver does not have wait for the daemons on all other running nodes to nominate it, but only those on newly-running nodes. Newly-running nodes are nodes that input from Topology Services indicates have just "recently" started running, and those nodes are not inserted into the Group Services domain established by the GS nameserver. Once those nodes nominate the GS nameserver, the GS nameserver simply executes one or more insert protocols to insert the newly-running nodes into the domain.

Group Services Initialization Errors

The Group Services subsystem creates AIX error log entries to indicate severe internal problems. For most of these, the best response is to contact the IBM Support Center.

However, if you get a message that there has been no heartbeat connection for some time, it indicates that the Topology Services subsystem is not running.

To check the status of the Topology Services subsystem, issue the **lssrc -g hats** command. If the response indicates that the Topology Services subsystem is inoperative, you will need to restart it using the **hatsctrl -s** command. If you are unable to restart it, call the IBM Support Center.

Group Services Daemon Operation

Normal operation of the Group Services subsystem requires no administrative intervention. The subsystem normally recovers from temporary failures, such as node failures or failures of Group Services daemons, automatically. However, there are some operational characteristics that may be of interest to administrators.

Due to AIX per-process file descriptor limits, the Group Services subsystem can support a maximum of approximately 2000 GS clients on each node.

The maximum number of nodes that can be contained within a domain is 2048, although Group Services can support node numbers in the range -32,768 to 32,767.

The maximum number of groups to which a GS client may subscribe or that a GS client may join is equivalent to the largest value containable in a signed integer variable.

The maximum number of groups allowed within a domain is 65,536.

These limits are the theoretical maximum limits. In practice, the amount of memory available to the Group Services daemon and its clients will reduce the limits to smaller values.

Group Services Procedures

For the most part the Group Services subsystem runs itself without requiring administrator intervention. However, on occasion, you may need to check the status of the subsystem.

Displaying the Status of the Group Services Daemon

You can display the operational status of the Group Services daemon by issuing the **lssrc** command.

On the control workstation, enter:

lssrc -l -s hags.*syspar_name*

where *syspar_name* is the name of the system partition of interest.

On a node, enter:

lssrc -l -s hags

In response, the **lssrc** command writes the status information to standard output. The information includes:

- The information provided by the **lssrc -s hags** command (short form)
- The number of currently connected clients and their process IDs
- The status of the Group Services domain
- The node number on which the GS nameserver is running
- Statistics for client groups with providers or subscribers on this node.

Note that if the **lssrc** command times out, the Group Services daemon is probably unable to connect to the Topology Services subsystem. For more information, see "Group Services Initialization Errors" on page 345.

Here is a sample of the output from the **lssrc -l -s hags** command on a node for the **k21sp2** system partition:

```
Subsystem         Group            PID     Status
 hags             hags             11938   active
4 locally-connected clients.  Their PIDs:
21344 17   18852 23486
HA Group Services domain information:
Domain established by node 9.
Number of groups known locally: 3
                           Number of     Number of local
Group name                 providers     providers/subscribers
 cssMembership                 5               1
 WomSchg_1                     5               1               1
 ha_em_peers                   7               1
```

Here is a sample of the output of the **lssrc -l -s hags.k21sp2** command on the control workstation for the same domain:

```
Subsystem         Group          PID        Status
 hags.k21sp2      hags           248 4      active
 1 locally-connected clients.  Their PIDs:
 44146
 HA Group Services domain information:
 Domain established by node 9.
 Number of groups known locally: 1
                                 Number of     Number of local
 Group name                      providers     providers/subscribers
  ha_em_peers                        7              1
```

In this domain, the GS nameserver is on node 9 of the system partition.

If a GS nameserver has not yet been established, the status indicates that the domain is not established. Similarly, if the GS nameserver fails, the status shows that the domain is recovering. Both of these conditions should clear in a short time. If they do not and the Topology Services subsystem is active, call the IBM Support Center.

Chapter 23. The Event Management Subsystem

This chapter introduces you to the Event Management subsystem. It includes information about the subsystem's components, its configuration, other components on which it depends, and its operation. It also discusses the relationship of the Event Management subsystem to the other PSSP high availability subsystems. Finally, it contains procedures you can use to check the status of the subsystem, or to add or change some of the Event Management configuration data.

After reading this chapter, you will be able to manage the Event Management subsystem and, if necessary, perform problem determination.

Introducing Event Management

Event Management is a distributed subsystem of the IBM Parallel System Support Programs (PSSP) on the RS/6000 Scalable POWERparallel System (SP). It is one of several subsystems in PSSP that provide a set of high availability services. Other subsystems that provide high availability services include the Group Services and Topology Services (heartbeat) subsystems.

The function of the Event Management subsystem is to match information about the state of system resources with information about resource conditions that are of interest to client programs, which may include applications, subsystems, and other programs. In this chapter, these client programs are referred to as **EM clients**.

Resource states are represented by resource variables. Resource conditions are represented as expressions, called **predicates**, that have a syntax that is a subset of the expression syntax of the C programming language.

Resource monitors are programs that observe the state of specific system resources and transform this state into several resource variables. The resource monitors periodically pass these variables to the Event Manager daemon. The Event Manager daemon applies predicates, which have been specified by EM clients, to each resource variable. If the predicate is true, an event is generated and sent to the appropriate EM client. EM clients may also query the Event Manager daemon for the current values of resource variables.

Resource variables, resource monitors, and other related information are specified in several System Data Repository (SDR) object classes. Information stored in these SDR classes is then translated into a form that can be easily used by the Event Management subsystem.

For more conceptual information about the Event Management subsystem, see *PSSP Event Management Programming Guide and Reference*.

Event Management Components

The Event Management subsystem consists of the following components:

Event Manager daemon
: The central component of the Event Management subsystem.

© Copyright IBM Corp. 1996

Event Management API (EMAPI)
The application programming interface that EM clients use to obtain the services of the Event Management subsystem.

Resource Monitor API (RMAPI)
The application programming interface that resource monitors use to supply resource variables to the Event Manager daemon.

Resource monitors Programs that monitor the state of system resources. Several resource monitors are shipped with PSSP.

Port numbers TCP/IP port numbers that the Event Management subsystem uses for communications. The Event Management subsystem also uses Unix domain sockets.

Configuration data Information that defines resource variables, resource monitors, and related data to the Event Management subsystem. This information is stored in the SDR.

Configuration utility
A program that translates the Event Management configuration data that is stored in the SDR to a binary format that is used by the Event Manager daemon and the RMAPI.

Configuration database
The Event Management Configuration Database (EMCDB) is the output of the configuration utility.

Control script A shell script that is used to add, start, stop, and delete the Event Management subsystem, which operates under System Resource Controller (SRC) control.

Configuration load utility
A program that loads the default configuration data that is shipped with PSSP into the SDR.

Default configuration data
A file that is shipped with PSSP that contains the default configuration data for the Event Management subsystem.

Files and directories
Various files and directories that are used by the Event Management subsystem to maintain run-time data.

The sections that follow contain more details about each of these components.

The Event Manager Daemon (haemd)

The Event Manager daemon is contained in the executable file **/usr/lpp/ssp/bin/haemd**. This daemon runs on each node of a system partition and on the control workstation. If there is more than one system partition, then multiple daemons run on the control workstation, one per system partition.

Note that the operational domain of the Event Management subsystem is a system partition. The control workstation is considered to be part of each domain. Unless otherwise stated, a reference to "the Event Management subsystem" is a reference to the Event Management subsystem in a single system partition.

EM clients communicate with a single Event Manager daemon in the operational domain that contains the resources in which they are interested, as follows:

If the operational domain contains the SP node on which the EM client is running, the client communicates with the local Event Manager daemon, through the EMAPI, using a Unix domain socket.

If an EM client is running on the control workstation, it must specify, explicitly or implicitly, a system partition. The EM client then communicates with the Event Manager daemon on the control workstation, in that system partition, using a Unix domain socket.

An EM client that is running on a node that is outside of the operational domain that contains the resources in which it is interested uses a TCP socket to communicate with the Event Manager daemon on the control workstation that is in that operational domain. An EM client that is running on any RS/6000 is considered to be on a node outside of any operational domain. In both of these cases, the EM client must specify, explicitly or implicitly, the name of a system partition that is in the operational domain of interest. For an EM client to run on an RS/6000, the PSSP fileset **ssp.clients** must be installed on the RS/6000.

Once an EM client has established communications with an Event Manager daemon, the EM client has access to all of the resources in that operational domain.

Resource monitors communicate with the Event Manager daemon, through the RMAPI, using Unix domain sockets. Resource monitors are always on the same node as the Event Manager daemon. Note that the control workstation is node 0.

The Event Management API (EMAPI)

The Event Management Application Programming Interface (EMAPI) is a shared library that an EM client uses to obtain the services of the Event Management subsystem. This shared library is supplied in two versions: one for non-thread safe programs and one for thread-safe programs. These libraries are referenced by the following path names:

/usr/lib/libha_em.a (non-thread safe version)

/usr/lib/libha_em_r.a (thread-safe version)

These path names are actually symbolic links to **/usr/lpp/ssp/lib/libha_em.a** and **/usr/lpp/ssp/lib/libha_em_r.a**, respectively. The symbolic links are placed in **/usr/lib** for ease of use. For serviceability, the actual libraries are placed in the **/usr/lpp/ssp/lib** directory. These libraries are supplied as shared libraries, also for serviceability.

For details on the EMAPI, see *Event Management Programming Guide and Reference*.

The Resource Monitor API (RMAPI)

The Resource Monitor Application Programming Interface (RMAPI) is a shared library that a resource monitor uses to supply resource variables to the Event Manager daemon. This shared library is supplied only in a non-thread safe version. This library is referenced through the path name

/usr/lib/libha_rr.a

which is a symbolic link to **/usr/lpp/ssp/lib/libha_rr.a**. The symbolic link is placed in **/usr/lib** for ease of use. For serviceability, the actual library is placed in the **/usr/lpp/ssp/lib** directory. This library is supplied as a shared library, also for serviceability.

For details on the RMAPI, see *Event Management Programming Guide and Reference*.

Resource Monitors

A resource monitor conforms to one of the following programming models:

The resource monitor is a daemon. When necessary, it is started by the Event Manager daemon. The Event Manager daemon connects to the resource monitor to establish communications. The resource monitor has a connection type of **server**.

The resource monitor logic is incorporated in a subsystem that manages the resources. The Event Manager daemon connects to the resource monitor to establish communications, but the Event Manager daemon does not start the resource monitor (because it is actually another subsystem). The resource monitor has a connection type of **server**.

The resource monitor logic is implemented in a command. The command can be used by scripts to supply resource variables to the Event Manager daemon. A command-based resource monitor connects to the Event Manager daemon to establish communications. The resource monitor has a connection type of **client**.

In addition, the Event Manager daemon itself performs some resource monitoring function. This function is considered to be a resource monitor with a connection type of **internal**.

PSSP supplies the following resource monitors:

IBM.PSSP.harmld Supplies resource variables for the CSS, VSD, and LoadLeveler subsystems. This data is also furnished to the Performance Monitor subsystem. This is a daemon with a connection type of **server**.

IBM.PSSP.harmpd Supplies resource variables that represent the number of processes executing a particular program. These variables can be used to determine whether a particular system daemon is running. This is a daemon with a connection type of **server**.

IBM.PSSP.hmrmd Supplies resource variables that represent the hardware state of the SP system. The resource information is obtained from the PSSP hardware monitoring subsystem (**hardmon**). This is a daemon with a connection type of **server**.

IBM.PSSP.pmanrmd
Supplies resource variables provided by the PSSP Problem Management subsystem. This is a command-based resource monitor with a connection type of **client**.

There are also several internal resource monitors incorporated in the Event Manager daemon itself:

Membership	Supplies resource variables that represent the Host Membership and Adapter Membership states. The Event Manager daemon obtains this information directly from the Group Services subsystem by subscribing to the **HostMembership**, **enMembership**, and **cssMembership** system groups.
Response	Supplies resource variables that represent the information in the **host_responds** and **switch_responds** SDR classes. These resource variables are provided for compatibility with earlier releases of PSSP.
aixos	Supplies resource variables that represent AIX operating system resources.

Port Numbers and Sockets

The Event Management subsystem uses several types of communications:

UDP port numbers for intra-domain communications, that is, communications between Event Manager daemons within an operational domain.

TCP port numbers for remote communications, that is, communications between Event Manager daemons and EM clients that are running outside their operational domains.

Unix domain sockets for communication between EM clients and the local Event Manager daemon (via the EMAPI) and between resource monitors and the Event Manager daemon (via the RMAPI).

Intra-Domain Port Numbers

For communication between Event Manager daemons within an operational domain, the Event Management subsystem uses a single UDP port number. This port number is recorded in the **Syspar_ports** SDR class. The class contains two attributes: **subsystem** and **port**. The class object whose **subsystem** attribute has a value of **haem** contains the port number for the Event Management subsystem. Note that the **Syspar_ports** class is a partitioned SDR class, that is, there is a set of objects for each system partition. When you configure the Event Management subsystem for a particular system partition, you use the SDR data that corresponds to that system partition.

The Event Management port number is stored in the SDR so that, when the Event Management subsystem is configured on each node, the port number is fetched from the SDR. This ensures that the same port number is used by all Event Manager daemons in the same operational domain. Note that because an Event Manager daemon from each operational domain runs on the control workstation, the Event Management subsystem in each system partition must have a unique port number.

This intra-domain port number is also set in the /**etc**/**services** file, using the service name **haem.***syspar_name*, where *syspar_name* is the name of the system partition. The /**etc**/**services** file is updated on all nodes in the system partition and on the control workstation. The Event Manager daemon obtains the port number from the /**etc**/**services** file during initialization.

Remote Port Numbers

The Event Management subsystem also uses a single TCP port number to accept communications from clients outside its operational domain. All Event Manager daemons that run on the control workstation use this port number to accept connections from EM clients outside their respective operational domains. Each daemon binds to a distinct IP address that is the IP address of its system partition.

The TCP port number is stored in the **SP_ports** SDR system class. The class contains the **daemon**, **hostname**, and **port** attributes. The class object whose **daemon** attribute has a value of **haemd** contains the port number for the Event Management subsystem(s). The **hostname** attribute is not used in this object.

This remote port number is also set in the /**etc**/**services** file on the control workstation, using the service name **haemd**.

Unix Domain Sockets

Unix domain sockets are used for communication:

- Between EM clients and the local Event Manager daemon (via the EMAPI)
- Between resource monitors and the Event Manager daemon (via the RMAPI).

These are connection-oriented sockets. The following socket names are used (*syspar_name* is the name of the system partition):

/**var**/**ha**/**soc**/**em.clsrv.***syspar_name*
: Used by the EMAPI to connect to the Event Manager daemon.

/**var**/**ha**/**soc**/**em.rmsrv.***syspar_name*
: Used by resource monitors with a connection type of **client** to connect to the Event Manager daemon.

/**var**/**ha**/**soc**/**em.RM***rmname***.***syspar_name*
: Used by the Event Manager daemon to connect to the resource monitor that is specified by *rmname*. This resource monitor has a connection type of **server**.

Configuration Data

Configuration information for the Event Management subsystem is stored in the SDR as objects in the following classes:

EM_Resource_Variable
: Contains the definitions of all resource variables. The resources with which these variables are associated are implied in the definitions.

EM_Instance_Vector
: Contains the definitions of all of the instance vectors that are associated with the resources that are specified in the **EM_Resource_Variable** class.

EM_Structured_Byte_String
: Contains the definitions of the structured fields for all resource variables that have a data type of structured byte string (SBS).

EM_Resource_Class
Contains the definitions for the resource classes with which each resource variable is associated.

EM_Resource_Monitor
Contains the definitions for all resource monitors.

For details about the Event Management SDR classes and attributes, see *Event Management Programming Guide and Reference*.

These SDR classes are partitioned classes. Therefore, each Event Management subsystem operational domain has its own configuration data. This permits a different set of resources to be monitored in each operational domain, as appropriate for the workload within the system partition.

The Event Management SDR objects are loaded from the default configuration data when the Event Management subsystem is added to a system partition using the control script.

There are several attributes that you, as the system administrator, may wish to change. They are:

The **rvPredicate** attribute of the **EM_Resource_Variable** class

Change this attribute to add or replace a default predicate for a resource variable.

The **rcObservation_interval** attribute of the **EM_Resource_Class**.

Change this attribute to increase or decrease the time between observations of a resource variable value by the Event Manager daemon.

For more information, see “Changing Configuration Data in the SDR” on page 367.

If resource monitors supplied by other IBM products or by third parties are installed on the SP system, you must load the configuration data supplied with the resource monitors into the SDR. For more information, see “Loading Non-PSSP Configuration Data into the SDR” on page 367.

The Configuration Utility (haemcfg)

The Event Management configuration utility is contained in the executable file **/usr/lpp/ssp/bin/haemcfg**. This utility is normally run on the control workstation in the environment of a system partition, that is, the value of the **SP_NAME** environment variable determines the partitioned SDR classes that are referenced. If **SP_NAME** is not set, the default system partition is assumed.

The **haemcfg** command converts the data stored in the Event Management SDR classes to a binary format called the Event Management Configuration Database (EMCDB). The EMCDB is placed in the staging file called **/spdata/sys1/ha/cfg/em.***syspar_name***.cdb**, where *syspar_name* is the name of the system partition in which the utility is run.

A version string that uniquely identifies the EMCDB is placed both in the EMCDB itself and in the **Syspar** partitioned SDR class. The version string is set as the value of the **haem_cdb_version** attribute in the object whose **syspar_name** attribute value matches the current system partition. The version string is used to

ensure that all of the Event Manager daemons and resource monitors that are running within the operational domain are using the same copy of the EMCDB.

If a previous copy of the **em.***syspar_name***.cdb** file exists, it is renamed to **/spdata/sys1/ha/cfg/em.***syspar_name***.cdb.***ssss,nnnn,v*, where *ssss,nnnn,v* is the version string of the previous copy.

The **haemcfg** command is automatically invoked by the control script when the Event Management subsystem is added to a system partition.

For details on the **haemcfg** command, see *PSSP Command and Technical Reference*.

The Configuration Database (EMCDB)

The Event Management Configuration Database (EMCDB) is produced by the **haemcfg** command from the information in the SDR. The format of the EMCDB is designed to permit quick loading of the database by the Event Manager daemon and the RMAPI. It also contains configuration data in an optimized format to minimize the amount of data that must be sent between Event Manager daemons and between an Event Manager daemon and its resource monitors.

When the SDR data is compiled, the EMCDB is placed in a staging file. When the Event Manager daemon on a node or the control workstation initializes, it automatically copies the EMCDB from the staging file to a run-time file on the node or the control workstation. The run-time file is called **/etc/ha/cfg/em.***syspar_name***.cdb**, where *syspar_name* is the name of the system partition.

For more information, see “Reading the EMCDB” on page 364.

The Control Script (haemctrl)

The Event Management control script is contained in the executable file **/usr/lpp/ssp/bin/haemctrl**. This script is normally invoked by the **syspar_ctrl** script, which provides an interface to all of the system partition-sensitive subsystems.

If necessary, you can invoke **haemctrl** directly from the command line. Note that before you invoke **haemctrl**, you must ensure that the **SP_NAME** environment variable is set to the appropriate system partition name.

For more information about **haemctrl** and **syspar_ctrl**, see *PSSP Command and Technical Reference*.

The purpose of the **haemctrl** command is to add (configure) the Event Management subsystem to a system partition. It can also be used to remove the subsystem from a system partition, start the subsystem, stop it, and clean the subsystem from all system partitions.

For more information, see “Event Management Configuration” on page 361.

The Configuration Load Utility (haemloadcfg)

The configuration load utility is contained in the executable file **/usr/lpp/ssp/install/bin/haemloadcfg**. The purpose of this program is to load the default configuration data into the SDR. Normally, it is invoked by the **haemctrl** script, but you can also use it to load additional configuration data into the SDR. Note that before you invoke **haemloadcfg**, you must ensure that the **SP_NAME** environment variable is set to the appropriate system partition name.

This utility does not replace existing objects in the SDR. Input data is matched with existing objects based on key attributes. The key attributes for each of the Event Management SDR classes are:

SDR Class	Key Attributes
EM_Resource_Variable	**rvName**
EM_Instance_Vector	**ivResource_name**, **ivElement_name**
EM_Structured_Byte_String	**sbsVariable_name**, **sbsField_name**
EM_Resource_Class	**rcClass**
EM_Resource_Monitor	**rmName**

Note that the way in which the **haemloadcfg** command handles existing SDR objects is different from the way in which the **SDRCreateObjects** command handles them. The **SDRCreateObjects** command creates a new object as long as the attributes, taken as a group, are unique.

The **haemloadcfg** command is actually a shell script that invokes the **/usr/lpp/ssp/bin/loadsdr** program. The purpose of this program is to batch load any number of objects into SDR classes. However, the **loadsdr** command cannot load arbitrary SDR classes. It can load only the Event Management SDR classes.

The Default Configuration Data (haemloadlist)

The default Event Management configuration data for PSSP is supplied in the **/usr/lpp/ssp/install/config/haemloadlist** file. The data in this file is loaded into each system partition to which the Event Management subsystem is added by the **haemctrl** script, using the **haemloadcfg** command.

Files and Directories

The Event Management subsystem uses the following directories:

/var/ha/lck, for lock files

/var/ha/log, for log files

/var/ha/run, for Event Manager daemon current working directories

/var/ha/soc, for socket files

/etc/ha/cfg, for run-time EMCDB and related files

/spdata/sys1/ha/cfg, as the staging area for EMCDB files.

The /var/ha/lck Directory (Lock Files)

In the **/var/ha/lck** directory, the **em.RM***rmname***.***syspar_name* file is used to ensure a single running instance of a resource monitor. In the file name, *rmname* is the name of the resource monitor and *syspar_name* is the name of the system partition. On the control workstation, there may be several instances of the same resource monitor running, but they are in different operational domains.

The **em.RM***rmname***SHM** file is used by the RMAPI to create a shared memory key when the resource monitor attaches to the shared memory that is managed by the SPMI library.

The **em.haemd.***syspar_name* file is used to ensure a single running instance of an Event Manager daemon in an operational domain.

The /var/ha/log Directory (Log Files)

In the **/var/ha/log** directory, the **em.trace.***syspar_name* file contains trace output from the Event Manager daemon. In the file name, *syspar_name* is the name of the system partition.

The **em.msgtrace.***syspar_name* file contains message trace output from the Event Manager daemon.

The **em.default.***syspar_name* file contains any error messages from the Event Manager daemon that cannot be written to the AIX error log. Normally, all daemon error messages are written to the AIX error log.

The /var/ha/run Directory (Daemon Working Files)

In the **/var/ha/run** directory, a directory called **haem.***syspar_name* is created, where *syspar_name* is the system partition name. This directory is the current working directory for the Event Manager daemon. If the Event Manager daemon abnormally terminates, the core dump file is placed in this directory. Whenever the Event Manager daemon starts, it renames any core file to **core.last**.

This directory also contains the working directories of any resource monitors that are started by the Event Manager daemon. Each directory has the name of its resource monitor.

Finally, this directory contains the **Rcache_local** and **Rcache_remote** directories. These directories contain the registration cache for local and remote client registration requests, respectively.

For each EM client that establishes communications with the Event Manager daemon, a cache subdirectory is created in the appropriate registration cache directory. This subdirectory has a name of the form *p,s,n,q,i*, where:

p is the process ID of the EM client

s is the seconds portion of the timestamp recorded when the EM client established its session with the Event Manager daemon

n is the nanoseconds portion of the timestamp

q is a sequence number for the session

i is the IP address of the host where the EM client is running. For local clients, the IP address is 0.0.0.0.

Within each subdirectory are several files with numeric names. Each file contains a registration request. The file name is the event command group ID of the group of events within the request.

The /etc/ha/cfg Directory (Run-Time EMCDB and Related Files)

The **/etc/ha/cfg** directory contains the run-time EMCDB file for each system partition. The file is called **/etc/ha/cfg/em.***syspar_name***.cdb**, where *syspar_name* is the name of the system partition.

In addition, there is a file called **em.***syspar_name***.cdb_vers**, where *syspar_name* is the name of the system partition. This file is created by the Event Manager daemon and contains the version string of the EMCDB file used by the daemon. This file is also used by the RMAPI to ensure that it is using the same EMCDB as the Event Manager daemon.

Components on Which Event Management Depends

The Event Management subsystem depends on the following components:

SPMI The System Performance Monitor Interface (SPMI), a library that is included with Performance Aide for AIX, which is a component of the IBM Performance Toolbox for AIX 2.2 product. This library is used by the RMAPI and the Event Manager daemon.

Group Services Another PSSP subsystem. It provides high availability membership services for coordinating activities on multiple nodes that are used by the Event Management subsystem.

Host_Responds daemon
A daemon that supplies information to the Event Management subsystem for compatibility with earlier releases of PSSP.

System Data Repository (SDR)
A repository of SP system configuration information that includes the configuration information for the Event Management subsystem.

The sections that follow contain more details about each of these components.

The System Performance Monitor Interface (SPMI) Library

The System Performance Monitor Interface (SPMI) library resides at **/usr/lib/libSpmi.a**. The RMAPI uses the shared memory technology of this library to deliver resource variable values to the Event Manager daemon. In addition, the data that is placed in shared memory can also be used by the Performance Toolbox for AIX product and the Performance Toolbox Parallel Extensions (PTPE) optional feature of PSSP. Thus, a resource monitor supplies data not only for event management, but also for performance monitoring.

Also, the Event Manager daemon uses the SPMI to obtain AIX operating system statistics that are the source of the AIX operating system resource variables that are supplied by PSSP.

The Group Services Subsystem

The Group Services subsystem provides another set of high availability services. The Event Management subsystem primarily uses Group Services to monitor the state of each Event Manager daemon in the operational domain, as follows. Each Event Manager daemon in the domain joins a Group Services group called **ha_em_peers**. Group Services informs each Event Manager daemon, in a synchronized fashion, when a daemon has joined or left the group.

Associated with the **ha_em_peers** group is a group state that contains the version string of the EMCDB. Thus, when an Event Manager daemon joins the **ha_em_peers** group, it can determine the version of the EMCDB that the rest of the group is using.

The Event Manager daemon also subscribes to the **HostMembership**, **enMembership**, and **cssMembership** system groups. Instances of the **IBM.PSSP.Membership.Node.state** resource variable are derived from **HostMembership** group information. Instances of the **IBM.PSSP.Membership.LANAdapter.state** resource variable are derived from **enMembership** and **cssMembership** group information.

The Host_Responds Daemon

On the control workstation, the Event Manager daemon is a client of the Host_Responds daemon, which is configured and controlled by the **/usr/lpp/ssp/bin/hr** script. Instances of the **IBM.PSSP.Response.Host.state** resource variable are created from information from the Host_Responds daemon. This is the same information that the Host_Responds daemon places in the **host_responds** SDR class. You can use the **IBM.PSSP.Response.Host.state** resource variable to obtain the status of any node in the cluster, even if the node is running a level of PSSP that is earlier than PSSP 2.2.

The System Data Repository (SDR)

The Event Manager daemon uses the SDR primarily as a repository of configuration information. In addition, on the control workstation, the Event Manager daemon polls the SDR for information that it uses to create instances of the **IBM.PSSP.Response.Switch.state** resource variable. This information is taken from the **switch_responds** SDR class.

Event Management Configuration and Operation

The following sections describe how the components of the Event Management subsystem work together to provide event management services. Included are discussions of:

- Event Management configuration
- Event Manager daemon initialization
- Event Management operation.

Event Management Configuration

The Event Management subsystem is installed as part of the installation of the PSSP product. The Event Management subsystem is contained in the **ssp.ha** fileset. The EMAPI libraries are contained in the **ssp.clients** fileset.

After the components are installed, the subsystem must be configured for operation. Event Management configuration is performed by the **haemctrl** command, which is invoked by the **syspar_ctrl** command.

The **syspar_ctrl** command configures all of the system partition-sensitive subsystems. The person who installs PSSP issues the **syspar_ctrl** command during installation of the control workstation. The **syspar_ctrl** command is executed automatically on the nodes when the nodes are installed. The **syspar_ctrl** command is also executed automatically when system partitions are created or destroyed. For more information on using the **syspar_ctrl** command, see *PSSP Installation and Migration Guide* and *PSSP Command and Technical Reference*.

The **haemctrl** command provides a number of functions for controlling the operation of the Event Management system. You can use it to:

- Add (configure) the Event Management subsystem
- Start the subsystem
- Stop the subsystem
- Delete (unconfigure) the subsystem
- "Clean" all Event Management subsystems
- Turn tracing of the Event Manager daemon on or off

Except for the clean function, **haemctrl** affects the Event Management subsystem in the current system partition, that is, the system partition that is specified by the **SP_NAME** environment variable.

Adding the Subsystem

If the **haemctrl** command is running on the control workstation, the first step in the add function is to select an Event Manager daemon communications port number and save it in the **Syspar_ports** SDR class. This port number is then placed in the /**etc**/**services** file. If the **haemctrl** command is running on a node, the port number is fetched from the SDR and placed in the /**etc**/**services** file. Port numbers are selected from the range 10000 through 10100.

The second step is to add the Event Manager daemon to the System Resource Controller (SRC) using the **mkssys** command. On the control workstation, the IP address of the system partition is an argument to the **haemd** program in the SRC subsystem specification.

The third step is to add an entry to the /**etc**/**inittab** file so that the Event Manager daemon will be started during boot. However, if **haemctrl** is running on a High Availability Control Workstation (HACWS), no entry is made in the /**etc**/**inittab** file. Instead, HACWS manages starting and stopping the Event Manager daemon.

The remaining steps in the add function are performed only on the control workstation. The **haemloadcfg** program is run to load the default configuration

data into the SDR. The **haemcfg** command is run to create the EMCDB and place it into the staging directory. Finally, if it is not already stored in the SDR, an Event Manager daemon remote client communications port number is selected from the range 10000 through 10100. This port number is then placed in the /**etc**/**services** file. This port number is used by all of the Event Manager daemons on the control workstation.

Note that if the **haemctrl** add function terminates with an error, the command can be rerun after the problem is fixed. The command takes into account any steps that already completed successfully.

Starting and Stopping the Subsystem

The start and stop functions of the **haemctrl** command simply run the **startsrc** and **stopsrc** commands, respectively. However, **haemctrl** automatically specifies the subsystem argument to these SRC commands.

Deleting the Subsystem

The delete function of the **haemctrl** command removes the subsystem from the SRC, removes the entry from /**etc**/**inittab**, and removes the Event Manager daemon communications port number from /**etc**/**services**. It does **not** remove anything from the SDR, because the Event Management subsystem may still be configured on other nodes in the operational domain.

Cleaning the Subsystem

The clean function of the **haemctrl** command performs the same function as the delete function, except in all system partitions. In addition, it removes the Event Manager daemon remote client communications port number from the /**etc**/**services** file.

The clean function does **not** remove anything from the SDR. This function is provided to support restoring the system to a known state, where the known state is the (possibly restored copy of the) SDR database.

Tracing the Subsystem

The tracing function of the **haemctrl** command is provided to supply additional problem determination information when it is requested by the IBM Support Center. Normally, tracing should **not** be turned on, because it may slightly degrade Event Management subsystem performance and can consume large amounts of disk space in the /**var** file system.

Event Manager Daemon Initialization

Normally, the Event Manager daemon is started by an entry in the /**etc**/**inittab** file, via the **startsrc** command. If necessary, the Event Manager daemon can be started using the **haemctrl** command or the **startsrc** command directly.

During its initialization, the Event Manager daemon performs the following steps:

1. It gets the number of the node on which it is running using the /**usr**/**lpp**/**ssp**/**install**/**bin**/**node_number** command. Node 0 is the control workstation.
2. It fetches the name of the system partition and the EMCDB version string from the **Syspar** SDR class. (Recall that one instance of the Event Manager daemon runs on the control workstation for each system partition to which the

Event Management subsystem was added.) It also fetches the Event Manager daemon remote client communications port number from the **SP_ports** SDR class.

3. It performs actions that are necessary to become a daemon. This includes establishing communications with the SRC subsystem so that it can return status in response to SRC commands.
4. It removes from the registration cache all of the subdirectories for local EM clients that no longer exist. That is, if the process ID in the subdirectory name cannot be found, it removes the subdirectory.

   Note that subdirectories for remote clients cannot be removed automatically, because the Event Manager daemon cannot determine if remote processes still exist.
5. It tries to connect to the Group Services subsystem. If the connection cannot be established because the Group Services subsystem is not running, it is scheduled to be retried in 5 seconds. This continues until the connection to Group Services is established. Meanwhile, Event Manager daemon initialization continues.
6. It enters the main control loop.

   In this loop, the Event Manager daemon waits for requests from EM clients, messages from resource monitors and other Event Manager daemons, messages from the Group Services subsystem, and requests from the SRC for status. It also waits for internal signals that indicate a function that was previously scheduled should now be executed, for example, retrying a connection to Group Services.

   However, EM client requests, messages from resource monitors, and messages from other Event Manager daemons (called peers) are refused until the Event Manager daemon has successfully joined the daemon peer group (the **ha_em_peers** group) and has fetched the correct version of the EMCDB.

Joining the Peer Group

After the Event Manager daemon has successfully established a connection with the Group Services subsystem, it tries to join the daemon peer group, a Group Services group called **ha_em_peers**. If this is the first Event Manager daemon to come up in the operational domain, it establishes the peer group. Otherwise, the other daemons in the peer group either accept or reject the daemon's join request. If an existing peer group member is still recovering from a prior termination of the joining daemon, the join request is rejected. If its join request is rejected, the daemon tries to join again in 15 seconds. This continues until the daemon's join request is accepted.

When it joins the daemon peer group, the Event Manager daemon examines the group state. The group state is the EMCDB version string.

If the group state is null, the joining daemon proposes that the group state be set to the version string that the daemon has fetched from the SDR. If several daemons try to join the group at about the same time, and the group state is null, then each daemon proposes the group state. When the group is formed, Group Services selects one of the proposals and sets the group state to it. Note that each daemon is proposing the EMCDB version string that it has fetched from the SDR. Unless the **haemcfg** command has been run at about the same time, the proposed version strings should be identical.

If the group state is not null when it is examined by the joining daemon, a group has already formed and the daemon does **not** propose a new group state.

After the daemon has successfully joined the peer group, it compares the EMCDB version string contained in the group state to the version string it fetched from the SDR. If they are different, the version that was fetched from the SDR is replaced by the version in the group state.

Reading the EMCDB

Once the daemon has joined the peer group and has determined the EMCDB version, it reads the run-time EMCDB file from the **/etc/ha/cfg** directory. If the file does not exist, it is copied from the staging directory on the control workstation.

Once the daemon has read the file, it compares the version string in the EMCDB to the one it fetched (from the SDR or from the group state). If the two version strings do not match, and the daemon has not just copied the EMCDB from the control workstation, then it copies the run-time EMCDB from the control workstation. If the version strings still do not match, the daemon terminates with an error.

After the daemon has read and validated the EMCDB, it enables daemon communications. This permits EM clients to send requests to the daemon, command-based resource monitors to connect to the daemon, and peers to send messages to the daemon. At this point, the initialization of the Event Manager daemon is complete.

To copy the EMCDB from the control workstation to the **/etc/ha/cfg** directory, the Event Manager daemon uses the **/usr/lpp/ssp/install/bin/haemrcpcdb** script. This script uses the **rcp** command to perform the actual copy.

The way in which Event Manager daemons determine the EMCDB version has important implications for the configuration of the subsystem. To place a new version of the EMCDB into production (that is, to make it the run-time version that is used by the Event Management subsystem), you must stop each Event Manager daemon in the operational domain after the **haemcfg** command is run. Stopping the daemons dissolves the existing peer group. Once the existing peer group is dissolved, the daemons can be restarted. As they restart, the daemons form a new peer group. A new EMCDB version string can be submitted as the group state only when a peer group is formed.

EM Daemon Operation

Normal operation of the Event Management subsystem requires no administrative intervention. The subsystem recovers from temporary failures automatically. However, there are some operational characteristics that may be of interest to administrators.

For performance reasons, the Event Manager daemon has an internal limit of 256 open file descriptors. In practice, this limits the number of EM client sessions, either local or remote, to about 225. This file descriptor limit is per daemon; it does not limit the number of EM clients in the operational domain.

Once the Event Manager daemon has successfully connected to a resource monitor that has a connection type of **server**, and that resource monitor terminates or closes its connection to the Event Manager daemon, the Event Manager daemon

tries to reconnect every 60 seconds. If appropriate, the Event Manager daemon creates the resource monitor process first at each reconnection attempt.

The primary function of the Event Manager daemon is to generate events, by observing resource variable values and applying predicates to those values. However, this function is performed for a resource variable only if an EM client has registered to receive events for that resource variable. The Event Manager daemon also observes a resource variable once to satisfy a query request, if the resource variable is not already being observed. When observations are necessary, the Event Manager daemon commands the appropriate resource monitor (if it has a connection type of **server**) to supply resource variable values. When observations are no longer necessary, the Event Manager daemon commands the resource monitor to stop supplying values. In this way, the Event Manager daemon performs no action for resource variables that are not of interest to clients.

Even if a resource monitor that has a connection type of **server** is running, it does not supply data to the Event Manager daemon except by command of the daemon.

The Event Manager daemon either observes a resource variable located in shared memory every X seconds, where X is the observation interval that is specified in the resource variable's resource class definition, or when the resource variable's value is sent to the Event Manager daemon by the resource monitor (transparently, via the RMAPI). The values of resource variables of value type Counter and Quantity are located in shared memory. The values of resource variables of value type State are not.

All resource variables that are located in shared memory with the same observation interval are observed on the same time boundary. This minimizes the observation overhead, no matter when the request for a resource variable is made.

Event Management Procedures

For the most part the Event Management subsystem runs itself without requiring administrator intervention. However, on occasion, you may need to check the status of the subsystem, or add or change some of the configuration data.

This section contains the procedures that you need to do these tasks, which include:

- Displaying the status of the Event Manager daemon
- Loading non-PSSP configuration data into the SDR
- Changing configuration data in the SDR
- Activating the configuration data in the SDR

Displaying the Status of the Event Manager Daemon

You can display the operational status of the Event Manager daemon by issuing the **lssrc** command.

On the control workstation, enter:

lssrc -l -s haem.*syspar_name*

where *syspar_name* is the name of the system partition of interest.

On a node, enter:

lssrc -l -s haem

In response, the **lssrc** command writes the status information to standard output. The information includes:

The information provided by the **lssrc -s haem** command (short form)

The names of any trace flags that are set

For information on these flags, see the **haemtrcon** command in *PSSP Command and Technical Reference*.

The EMCDB version string and an indication as to whether the version string is taken from the SDR or from the peer group state.

The day and time the Event Manager daemon was started.

A report on the daemon's progress through initialization:

```
Daemon connected to group services: TRUE/FALSE
Daemon has joined peer group:       TRUE/FALSE
Daemon communications enabled :     TRUE/FALSE
```

A count of the peer daemons that are currently in the peer group. The count does not include this daemon.

The number and type of EM client connections. Note that when a daemon relays a request to another daemon, the sending daemon is treated as a client by the receiving daemon.

A list of each resource monitor that is defined in the EMCDB, its connection type (**server**, **client**, or **internal**), and its connection status.

The connection status is indicated by a file descriptor (FD). If the file descriptor is greater than or equal to 0, a connection is open. The process ID of the resource monitor is also listed. If the resource monitor has not yet been started, or is not startable, by the Event Manager daemon, the process ID is -1.

The highest file descriptor in use.

The peer daemon status.

This lists the status of peer daemons by node number, starting with node 0, and up to and including the highest node number seen for a peer. Note that this may include node numbers that are not in the operational domain of this Event Manager daemon.

Following the node number are two characters. If both characters are **S**, the specified node is the number of the node on which this daemon is running. Otherwise, the characters can take on values as follows.

The first character is **I**, **O**, or **-**, where:

- **I** indicates that the peer on the specified node is a peer group member.
- **O** indicates that the peer is no longer a peer group member (but was at one time).
- **-** indicates that no peer on the specified node has ever joined the group.

The second character is either **A** or **R**, where:

- **A** indicates that this daemon is accepting join requests from the peer on the specified node.
- **R** means this daemon is rejecting join requests.

A list of internal daemon counters, for use by IBM service personnel.

Loading Non-PSSP Configuration Data into the SDR

The default configuration data supplied with PSSP is normally loaded into the SDR and compiled into its binary format automatically by the **haemctrl** script. However, if resource monitors supplied by other IBM products or by third parties are installed on the SP system, you must load the configuration data supplied with the resource monitors into the SDR and activate it. To do this:

1. Login to the control workstation. Use an ID that has **root** authority.
2. Create a file in load list format with the data to be loaded, or identify the path name of the file that has been supplied.

   If you are creating a new file, use the format specified in the man page for the **haemloadlist** file.
3. Set the **SP_NAME** environment variable to the appropriate system partition name.
4. Load the data into the SDR using the **haemloadcfg** command. Enter:

   haemloadcfg *new_loadlist*

   where *new_loadlist* is the path name of the load list you previously created or identified.
5. Activate all of the Event Management data in the SDR, including your new data, using the procedure in “Activating the Configuration Data in the SDR” on page 368.

Changing Configuration Data in the SDR

The **haemloadcfg** command does not update objects (identified by their key attributes) that already exist in the SDR. Accordingly, if you want to change an object that already exists, you must first delete the object and then recreate the changed object. To do this:

1. Login to the control workstation. Use an ID that has **root** authority.
2. Change the object's attribute in the load list file.

   For PSSP configuration data, this is the default configuration data in the **/usr/lpp/ssp/install/config/haemloadlist** file. If you have non-PSSP or third-party resource monitors, the object will be loaded from another load list file.

   Do this step for as many objects as you are changing.
3. Set the **SP_NAME** environment variable to the appropriate system partition name.
4. Use the **SDRDeleteObjects** command to delete the object from the SDR.

   Do this step for as many objects as you are changing.
5. Run the **haemloadcfg** command, specifying the changed load list file.

In response, the **haemloadcfg** command creates a new object in the SDR for each object that you changed in the load list file.

6. Activate all of the Event Management data in the SDR, including your changed data, using the procedure in "Activating the Configuration Data in the SDR."

Activating the Configuration Data in the SDR

When you have added or changed the Event Management data in the SDR, you must activate it by recompiling the EMCDB and stopping and restarting the Event Manager daemons. To do this:

1. Login to the control workstation. Use an ID that has **root** authority.
2. Set the **SP_NAME** environment variable to the appropriate system partition name.
3. Recompile the EMCDB. Enter:

   haemcfg

   The output is placed in the staging directory in a file whose name indicates the system partition.
4. Stop the Event Manager daemons in this system partition.

   Issue the **haemctrl -k** command on the control workstation and on each of the nodes in the system partition.

   You can use the **dsh** or **Sysctl** commands to run the command on multiple nodes from the control workstation. For more information on using these commands, see Chapter 4, "Parallel Management Commands" on page 31.
5. Verify that all of the Event Manager daemons in this system partition have stopped.

   On the control workstation, issue the **lssrc -s haem.***syspar_name* command. On the nodes, issue the **lssrc -s haem** command.

   You can use the **dsh** or **Sysctl** commands to run the command on multiple nodes from the control workstation. For more information on using these commands, see Chapter 4, "Parallel Management Commands" on page 31.

   The status of each daemon should indicate that it is inactive.
6. Restart the Event Manager daemons in this system partition.

   Issue the **haemctrl -s** command on the control workstation and on each of the nodes in the system partition.

   You can use the **dsh** or **Sysctl** commands to run the command on multiple nodes from the control workstation. For more information on using these commands, see Chapter 4, "Parallel Management Commands" on page 31.

Chapter 24. Using the Problem Management Subsystem

The Problem Management subsystem (**pman**) provides an infrastructure for recognizing and acting on problem events in your SP system. This infrastructure is based on an Event Management application that provides configurable access to Event Management client and resource monitor function without the necessity of writing C programs that use the Event Management APIs.

To understand the information presented in this chapter, you should be familiar with Event Management concepts and terminology. Some important Event Management terms are:

event In Event Management, the notification that a predicate evaluated to true. This evaluation occurs each time an instance of a resource variable is observed.

instance vector In Event Management, a set of elements, where each element is a name/value pair of the form name=value, whose values uniquely identify the copy of the resource (and by extension, the copy of the resource variable) in the system.

predicate In Event Management, the relational expression between a resource variable and other elements (such as constants or the previous value of an instance of the variable) that, when true, generates an event. An example of a predicate is $X < 1$ where X represents the resource variable IBM.PSSP.aixos.PagSp.%totalfree (the percentage of total free paging space). When the predicate is true, that is, when the total free paging space is observed to be less than 10%, the Event Management subsystem generates an event to notify the appropriate application.

rearm predicate In Event Management, a predicate used to generate an event that alternates with an original event predicate in the following way: the event predicate is used until it is true, then the rearm predicate is used until it is true, then the event predicate is used, and so on. The rearm predicate is commonly the inverse of the event predicate (for example, a resource variable is on or off). It can also be used with the event predicate to define an upper and lower boundary for a condition of interest.

resource In Event Management, an entity in the system that provides a set of services. Examples of resources include hardware entities such as processors, disk drives, memory, and adapters, and software entities such as database applications, processes, and file systems. Each resource in the system has one or more attributes that define the state of the resource.

resource monitor A program that supplies information about the resources in the system. It can be a command, a daemon, or part of an application or subsystem that manages any type of system resource.

resource variable In Event Management, the representation of an attribute of a resource. An example of a resource variable is IBM.AIX.PagSp.%totalfree, which represents the percentage of

© Copyright IBM Corp. 1996

total free paging space. IBM.AIX.PagSp specifies the resource name and %totalfree specifies the resource attribute.

For more information on Event Management services, refer to *IBM Parallel System Support Programs for AIX: Event Management Programming Guide and Reference*.

The major components of the Problem Management subsystem are:

- A Problem Management daemon (**pmand**) that provides access to events generated by Event Management on all nodes. In addition, the following function is provided to work in conjunction with **pmand**:
  - A command, **pmandef**, that provides for subscribing to Event Management events and associating actions for those events
  - A command, **pmanquery**, that queries the SDR for a description of a Problem Management subscription
  - A script, **pmandefaults**, that causes **pmand** to register for a set of default events
  - A script, **notify_event**, that mails an event notification when an event occurs
  - A script, **log_event**, that logs a record of an event to a regular wraparound file
- A resource monitor daemon, **pmanrmd**, that provides resource variables to Event Management. The following function is provided to work in conjunction with **pmanrmd**:
  - Sixteen resource variables. The resource variables are named **IBM.PSSP.pm.User_state1** through **IBM.PSSP.pm.User_state16**.
  - A command, **pmanrminput**, that provides for associating values with the sixteen supplied resource variables.
  - A sample file, **pmanrmd.conf**, for configuring the **pmanrmd** daemon.
  - A command, **pmanrmdloadSDR**, for loading the configuration information into the System Data Repository.

This chapter provides information on each of the components of Problem Management.

Understanding the Problem Management Daemon

The **pmand** daemon is a client of Event Management; it can be configured to register for Event Management events and perform actions when those events occur.

Event Management provides access to events throughout an SP system partition; therefore, **pmand** can monitor and react to events on the node on which it is running as well as on all other nodes in the system partition and the control workstation.

When you install your system, a **pmand** daemon is automatically configured on each node in a system partition. Additionally, there is a **pmand** daemon running on the control workstation for each system partition in the SP system that is running Event Management. When running on a node, the **pmand** daemon:

Monitors events occurring on the node on which the daemon is running

Monitors events on all other nodes in the system partition

Monitors events not associated with a node, such as frame events, as supplied by Event Management

There are no restrictions on what a **pmand** daemon can monitor:

Any number of **pmand** daemons can monitor and act on a single event

A single **pmand** daemon can monitor any number of events locally or remotely

A single **pmand** daemon can monitor the same event multiple times and all the actions associated with all the event registrations are taken by the daemon when the event occurs

In Figure 78 **pmand** daemons are running on a 4 node SP system partition.

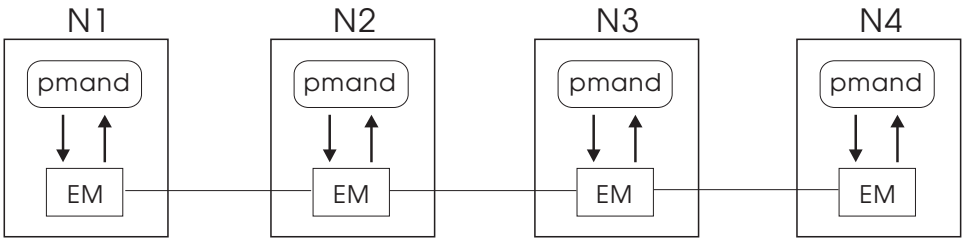


Figure 78. An Example of Problem Management Daemon Configuration

Each **pmand** daemon has access to events on the node on which it is running, as well as to the other nodes in the system partition. The access to the events is provided by Event Management, which is able, due to its distributed nature, to monitor resource variables throughout the system partition and to generate events based on the values of the resource variables. When an event that any of the daemons has subscribed to occurs (whether that event is local or remote), all the **pmand** daemons registered for it will perform the actions they are configured for (if any).

Because **pmand** is a daemon, its subscriptions to events are persistent. That is, the daemon continues to subscribe to events even after the process or user who created the subscription has gone away. A system administrator, for example, can set up automated operations with unattended monitoring and recovery actions.

Controlling pmand

The **pmand** daemon is under the System Resource Controller (SRC) and can be controlled by the following commands:

startsrc -s pman starts **pmand**

stopsrc -s pman stops **pmand**

refresh -s pman causes **pmand** to update its internal configuration from the SDR and start pairing actions with events as specified by the **pmandef** command. If a **pmand** refresh occurs, all currently monitored events will be unregistered from before the configuration information is reread from the SDR. The SDR contains persistent information, so that a refresh results only in configuration changes that have been put into the SDR. If you haven't deleted

or modified a configuration record for a particular event, refreshing the daemon results in reregistering for the same event.

lssrc -ls provides status on the **pmand** daemon. To check the status of **pmand** running on a node, enter:

```
lssrc -ls pman
```

To check the status of **pmand** running on the control workstation, enter:

```
lssrc -ls pman.system_partition_name
```

This command provides the following status information:

- When **pmand** was started.
- When **pmand** was last refreshed.
- Whether tracing (debug mode) is on or off. When debug mode is on, all SRC requests and all events are logged to the **/var/adm/SPlogs/pman** directory.
- Events for which registrations are as yet unacknowledged.
- Events for which actions are currently being taken.
- Events currently ready to be acted on by this daemon.

Creating Problem Management Subscriptions

The **pmandef** command is the mechanism provided for creating Problem Management subscriptions for the **pmand** daemon to Event Management services. The **pmandef** command provides for defining:

Event Manager events to register for
Actions to take when those events occur:
- Run a command
- Issue an SNMP trap
- Write to the AIX Error Log or BSD syslog facilities

The **pmandef** command also provides for:

Activating a Problem Management subscription
Deactivating a Problem Management subscription
Querying a Problem Management subscription
Removing a Problem Management subscription

For more information on **pmandef**, refer to *IBM Parallel System Support Programs for AIX: Command and Technical Reference*.

Security Considerations

The **pmandef** command is based on Sysctl, which uses Kerberos for user authentication. All users of the **pmandef** command must have valid Kerberos credentials. In addition, the user's Kerberos principal must be listed in the **/etc/sysctl.pman.acl** file on the local node, in order to store the subscription in the SDR, as well as on all the nodes that are affected by the new subscription, in order for the affected Problem Management daemons to be notified of the new subscription. If the user's Kerberos principal is listed only in the **/etc/sysctl.pman.acl** file on the local node, the subscription will be stored in the SDR, but the Problem Management daemons will not act on the new subscription until the next time they are restarted.

The user's Kerberos principal is more than just a mechanism to validate access to the Problem Management subsystem. The Kerberos principal that was in effect when the subscription was created, via the **pmandef -s** command, is stored as part of the subscription definition, and it is used to establish the ownership of that subscription. Modifications to the subscription, via the **pmandef** command with the **-u**, **-d** and **-a** flags, are allowed only by a user with the same Kerberos principal that is stored in the subscription definition. The Kerberos principal that is stored in the subscription definition is also used by the Problem Management daemon to decide whether the action that is to result from the occurrence of an event should be allowed.

After **pmand** receives notification that an event has occurred, and before it performs the action for that event, it checks to see whether the Kerberos principal that is stored in the subscription definition is authorized to perform the requested action on the node where **pmand** is running. If the requested action is to write an entry in the AIX Error Log and BSD syslog or to generate an SNMP trap, then the Kerberos principal that owns the subscription must have been listed in the root user's **$HOME/.klogin** file. If the requested action is to execute a command, then the Kerberos principal must be listed in the **$HOME/.klogin** file of the user which will be used to run the command. The user which will be used to run the command is by default the same user who issued the **pmandef -s** command to create the subscription. A different user may be specified to **pmandef** by using the **-U** flag.

Running a Command

Use **pmandef** to specify a command to run when a specified event or rearm event occurs. For example:

```
pmandef -s Program_Monitor \
-e 'IBM.PSSP.Prog.pcount:NodeNum=12;ProgName=mycmd;UserName=bob:X@ == ' \
-r "X@ ==1" -c "echo program has stopped >/tmp/myevent.out" \
-C "echo program has restarted >/tmp/myrearm.out"
```

Running this example on node 5 causes the command **echo program has stopped >/tmp/myevent.out** to run on node 5 whenever the number of processes named **mycmd** and owned by user **bob** on node 12 becomes 0 (the event). When this number increases back to 1 (the rearm event), the command **echo program has restarted >/tmp/myrearm.out** runs on node 5.

You can specify if you want the command to run on a node other than the one from which the **pmandef** command was issued. For example:

```
pmandef -s Program_Monitor \
-e 'IBM.PSSP.Prog.pcount:NodeNum=1-5,13;ProgName=mycmd;UserName=bob:X@ == ' \
-r "X@ ==1" -c /usr/local/bin/start_recovery \
-C /usr/local/bin/stop_recovery -n 1-3,7
```

This example causes the commands to run on nodes 1, 2, 3 and 7, whenever **bob**'s program dies or gets restarted on any of nodes 1, 2, 3, 4, 5 or 13. If **bob**'s program dies on node 4, then the command **/usr/local/bin/start_recovery** runs on nodes 1, 2, 3 and 7.

Any number of commands can run simultaneously.

You can specify a timeout, in seconds, for each command. The minimum timeout that can be specified is 10 seconds. If the command has not exited before the specified timeout, the command is killed.

For information on command termination status, use the **lssrc -ls** command.

The Command Environment: The Problem Management subsystem makes all of the contents of an Event Management notification available in the command's environment when the command is run:

PMAN_HANDLE
The name that identifies this subscription to the Problem Management subsystem. This name was given as the argument of the **-s** flag to **pmandef**.

PMAN_PRINCIPAL
The name of the Kerberos principal that owns this subscription.

PMAN_RVNAME
The Event Management resource variable.

PMAN_IVECTOR
The Event Management instance vector.

PMAN_PRED Either the Event Management predicate or rearm predicate, depending on whether this is an event or rearm event.

PMAN_TIME The time that the event was reported to the Problem Management subsystem.

PMAN_LOCATION
The node number of the node on which the event was generated, usually (but not always) the node on which the event occurred.

PMAN_RVTYPE
One of **long**, **float** or **sbs**, depending on whether the type of the resource variable value is a long integer, a floating point value or a Structured Byte String.

Note: If the PMAN_RVTYPE is either **long** or **float**, then the resource variable value is stored in PMAN_RVVALUE, and PMAN_RVVALUE is to be interpreted as type PMAN_RVTYPE.

Note: If PMAN_RVTYPE is **sbs**, then the resource variable value is composed of one or more structure fields. There is no PMAN_RVVALUE environment variable. Instead there is a separate environment variable for each field, and the PMAN_RVCOUNT environment variable defines the number of fields. For example, if there are 3 structure fields within the Structured Byte String, the PMAN_RVCOUNT will be 3, and there will be 3 separate environment variables for the 3 structure fields: PMAN_RVFIELD0, PMAN_RVFIELD1 and PMAN_RVFIELD2. Each of these 3 environment variables contains a *name=value* pair, where *name* is the structure field name, and *value* is the structure field value.

For example, the following command:

```
pmandef -s example \
-e 'IBM.PSSP.Prog.pcount:NodeNum=9-11;ProgName=mycmd;UserName=root:X@ == ' \
-c "/usr/local/bin/recovery_cmd" -n 12
```

requests the **/usr/local/bin/recovery_cmd** command to run on node 12, when the number of processes named **mycmd** and owned by root on nodes 9, 10, or 11 becomes zero. If the **mycmd** program terminates on node 10, the command **/usr/local/bin/recovery_cmd** runs on node 12, and the following environment variables are included in its environment:

PMAN_HANDLE (**example**)
PMAN_PRINCIPAL (**root.admin@PPD.POK.IBM.COM**)
PMAN_RVNAME (**IBM.PSSP.Prog.pcount**)
PMAN_IVECTOR (**ProgName=mycmd;UserName=root;NodeNum=10**)
PMAN_PRED (**X@0==0**)
PMAN_TIME (**Thu Aug 22 00:42:08 1996**)
PMAN_LOCATION (**10**)
PMAN_RVTYPE (**sbs**)
PMAN_RVCOUNT (**3**)
PMAN_RVFIELD0 (**CurPIDCount=0**)
PMAN_RVFIELD1 (**PrevPIDCount=1**)
PMAN_RVFIELD2 (**CurPIDList=**)

This information could be used by any command. Two utilities that report this information are provided as part of the Problem Management subsystem: **notify_event** and **log_event**. (These commands are provided to get you started; you may want to write more sophisticated commands.)

notify_event captures event information and mails it to the user running the command on the local node.

log_event captures event information and logs it to a wraparound file. The syntax for **log_event** is:

/usr/lpp/ssp/bin/log_event *log_filename*

log_event uses the AIX 4.1 **alog** command to write to a wraparound file. The size of the wraparound file is limited to 64K. The **alog** command must be used to read the file. Refer to the AIX 4.1 **alog** man page for more information on this command.

Issuing an SNMP Trap

Use the **pmandef** command to subscribe to an Event Management event and specify that an SNMP trap be issued for that event. For example:

```
pmandef -s Filesystem_Monitor \
-e 'IBM.PSSP.aixos.FS%totused:NodeNum=1 ;VG=myvg;LV=mylv:X>95' \
-t 1234 -n
```

In this example, whenever the file system associated with the **mylv** logical volume and **myvg** volume group on node 10 becomes more that 95% full, an SNMP trap will be generated on the control workstation.

For complete information on how **pmand** can be configured to issue an SNMP trap when an event for which it is registered occurs, refer to Chapter 25, "Managing SP System Events in a Network Environment" on page 381.

Logging an Event

You can specify that **pmand** write event notification information, along with some optional specified text, to the AIX Error Log and BSD syslog facilities. For example:

```
pmandef -s Filesystem_Monitor \
-e 'IBM.PSSP.aixos.FS%totused:NodeNum=11;VG=myvg;mylv:X>95' \
-l "filesystem is almost full" -h local
```

In this example, whenever the file system associated with the **mylv** logical volume and **myvg** volume group on node 11 becomes more than 95 % full, the text

filesystem is almost full gets written to the AIX Error Log and BSD syslog facilities on node 11 (via the **-h local** option).

Requesting Problem Management Subscription Information

Use the **pmanquery** command to query the SDR for a description of a Problem Management subscription. The **pmanquery** command ouputs the details of the subscription information in raw format, which can then be used by other applications. The following example queries all subscriptions:

```
pmanquery -n all -k all
```

For more information on **pmanquery**, refer to *IBM Parallel System Support Programs for AIX: Command and Technical Reference*.

Monitoring Default Events

The Problem Management subsystem provides for a set of default events to be monitored in the **/usr/lpp/ssp/install/bin/pmandefaults** script. This script contains a series of **pmandef** commands that request an event notification to be mailed to root on the control workstation, when the specified event occurs. These are events that would interest many system administrators. Events are defined for all nodes in the current system partition, and all events are monitored from the control workstation. The list of events includes:

- The /**var** file system is more than 95 percent full.
- The /**tmp** file system is more than 95 percent full.
- An error log record of type **PERM** has been written to the AIX Error Log.
- The **inetd** daemon has terminated.
- The **sdrd** daemon has terminated (control workstation only).
- The **sysctld** daemon has terminated.
- The **hrd** daemon has terminated (control workstation only).
- The **fsd** daemon has terminated (nodes only).

This script is a suggested starting point for configuring the Problem Management subsystem on your SP system. You can choose to run the script as is, or you can make your own copy of the script and modify it to suit your needs, or you can choose not to run the script at all.

The script takes no arguments, and it should be executed as the **admin** Kerberos user (for example, **root.admin@PPD.POK.IBM.COM**). It should be executed once for each system partition; specify the system partition by setting the SP_NAME environment variable.

Understanding the Problem Management Resource Monitor Daemon

The **pmanrmd** daemon is a resource monitor daemon that provides resource variables to Event Management services. When you install your SP system, a **pmanrmd** daemon is automatically configured on each node in a system partition. Additionally, there is a **pmanrmd** daemon running on the control workstation for each system partition in the SP system.

Controlling pmanrmd

The **pmanrmd** daemon is under the System Resource Controller (SRC). Only the following SRC commands are available for controlling the **pmanrmd** daemon:

startsrc -s pmanrm starts **pmanrmd**
stopsrc -s pmanrm stops **pmanrmd**

No command is provided to refresh the **pmanrmd** daemon while it is running. To refresh the daemon, you must stop and start the daemon (using **stopsrc** and **startsrc**).

Creating Resource Monitors

The Problem Management subsystem provides sixteen resource variables, named **IBM.PSSP.pm.User_state1** through **IBM.PSSP.pm.User_state16**. You can associate values with the provided resource variables and configure **pmanrmd** to supply the values that you associate to Event Management services.

Associating Values with Problem Management Resource Variables

Two mechanisms are provided for associating values with the Problem Management resource variables:

1. The **pmanrminput** command
2. The **pmanrmd.conf** sample configuration file

The **/usr/lpp/ssp/bin/pmanrminput** command requires root authority to use. The command can be run from within scripts. It takes as arguments:

The name of the SRC subsystem to which to communicate the request

A string that consists of:

- The name of one of the Problem Management supplied resource variables
- A delimiter (+)
- A string to provide as the value of the Problem Management supplied resource variable
- Another delimiter (+)

For example, if you have a server application running on node 10 and you sometimes want to tell users on nodes 1-9 to stop using the server application, you could use the Problem Management subsystem to accomplish this. Create a Problem Management subscription on nodes 1-9 to run the **wall** command to inform the users about the state of the server application:

```
pmandef -s ready_or_not \
-e 'IBM.PSSP.pm.User_state3:NodeNum=1 :X@ ==“READY”' \
-r 'X@ !=“READY”' \
-c “wall Node 1  is ready” -C “wall Stop using node 1 ” -n 1-9
```

When the server application is ready for users, it uses the **pmanrminput** command to communicate its state to the Problem Management subsystem:

```
pmanrminput -s pman -a “IBM.PSSP.pm.User_state3+READY+”
```

The Problem Management subsystem passes the string **READY** to Event Management as the value of the resource variable **IBM.PSSP.pm.User_state3**, and that value satisfies the event definition, so the command **wall Node 10 is ready** runs on nodes 1-9. When the server application has a problem, it can use **pmanrminput** to communicate its state change:

```
pmanrminput -s pman -a "IBM.PSSP.pm.User_state3+NOT READY+"
```

The resource value **NOT READY** satisfies the rearm event definition, so the command **wall Stop using node 10** runs on nodes 1-9.

Another way to associate a value with one of the Problem Management resource variables is to provide the name of the resource variable and a command in a configuration file, based on the **pmanrmd.conf** sample file (for more information on the **pmanrmd.conf** sample file refer to "Configuring pmanrmd to Provide Resource Variables"). The standard output of the command will be provided to Event Management services as the value for the specified Problem Management resource variable.

Configuring pmanrmd to Provide Resource Variables

In the example in "Associating Values with Problem Management Resource Variables" on page 377, the resource, an application server, was able to communicate its state change directly to the Problem Management subsystem. However, this is not always possible. For example, if you want to run a command whenever the contents of the **/etc** directory on nodes 1-5 change, the **/etc** directory is the resource, and a directory is not capable of stating when it has changed. In this case, the Problem Management subsystem needs to be configured to periodically send the name and time stamp of the last changed file in the **/etc** directory as the resource variable value to Event Management. This can be accomplished by defining a resource monitor command to the Problem Management subsystem.

A sample **pmanrmd.conf** file is provided. Copy, rename, and edit this file to specify:

- The node(s) on which **pmanrmd** is to be configured
- The name of the Problem Management resource variable to provide via **pmanrmd**
- A sampling interval (in seconds) for each resource variable
- The command to run to produce the standard output that will be provided as the value for the Problem Management resource variable

For example:

```
TargetType=NODE_RANGE
Target=1-5
Rvar=IBM.PSSP.pm.User_state2
SampInt=6
Command="/bin/ls -tl /etc | /bin/head -2 | /bin/grep -v total"
```

In this example, the **pmanrmd** daemon will run the command **/bin/ls -tl /etc | /bin/head -2 | /bin/grep -v total** every ten minutes, and it will send the output of that command as the value of the resource variable **IBM.PSSP.pm.User_state2** to Event Management. The output of this command is the name of the last file to be changed and its time stamp. You could then subscribe to changes in the **/etc** directory resource with the following command:

```
pmandef -s etc_file_changed \
-e 'IBM.PSSP.pm.User_state2:NodeNum=1-5:X@ !=X@P ' \
-c "echo somebody changed a file in /etc >/tmp/etc_change_log"
```

Whenever the most recent output of the **/bin/ls -tl /etc | /bin/head -2 | /bin/grep -v total** command is different from the previous output, the event definition will be

satisfied, and the **"echo somebody changed a file in /etc >/tmp/etc_change_log"** command will get executed.

Loading pmanrmd Configuration Information into the SDR

pmanrmdloadSDR reads the configuration file and loads the information in the stanzas into the SDR. **pmanrmdloadSDR** runs on the control workstation.

The syntax for **pmanrmdloadSDR** is :

pmanrmdloadSDR *configuration_file*

After you load the configuration information into the SDR, refresh the affected **pmanrmd** daemons.

Chapter 25. Managing SP System Events in a Network Environment

IBM Parallel System Support Programs for AIX includes a Problem Management subsystem (refer to Chapter 24, “Using the Problem Management Subsystem” on page 369) that provides access to Event Management function without the necessity of writing C programs that use the Event Management APIs. The Problem Management subsystem gives you the ability to subscribe to Event Management events and to specify actions for those events. Issuing an SNMP trap in response to an event (which can come either from Event Management or the AIX Error Log) is one of the actions you can specify.

The ability to issue an SNMP trap in response to an event allows you to report problem events occurring in your SP system to a network manager existing on a remote node (a network manager application is not supplied with the SP). The Problem Management subsystem provides an SP SNMP proxy agent, **sp_configd**, (sometimes referred to as a ‘subagent daemon’) that runs on the control workstation and every SP processor node. The SP proxy agent provides the following functions:

1. A Management Information Base (MIB)
2. SNMP **GET** and **GET NEXT** command support that allows data in the MIB to be accessed by a network manager application
3. Creation and transmittal of SNMP traps to an installation-defined network manager application when the following events occur on the node on which the SP proxy agent is running:

   A ‘cold start’ trap is issued when the agent is activated

   An enterprise-specific trap is issued when an entry with an “alert=true” attribute is written in the AIX Error Log

   An enterprise-specific trap is issued when an user-specified event is detected within Event Management services

This chapter provides overview and task information for making the SP part of a network management system. In such a system, for example, the SP presents error event information, in the form of SNMP traps, to a network management application, such as NetView for AIX, which displays and logs the trap, and may notify a problem management application, such as Trouble Ticket for AIX, of problem events occurring on the SP.

In order for the SP to be part of such a network management system, network managers must be notified when selected AIX Error Log entries are written and Event Management events occur. SP-specific configuration data is provided so that management applications can determine which nodes compose the SP system. You decide what Error Log and Event Management events will trigger manager notification.

We'll first discuss general concepts related to Network Management and then go on to present information specific to enabling the reporting of SP system events in a network environment. For further discussion of Simple Network Management Protocol and the Management Information Base, refer to *IBM AIX Version 4.1 Communications Programming Concepts*, SC23-2610.

© Copyright IBM Corp. 1996

Understanding Network Management

The Simple Network Management Protocol (SNMP) describes the means by which management data is packaged and transported through the network in order for SNMP agents and SNMP network managers to communicate. The Management Information Base (MIB) defines the structure of the data being transported.

Understanding SNMP

SNMP uses a client/server approach to management, defining two roles: Manager (client) and Agent (server).

The Manager (NetView for AIX is an example) oversees the overall network activity. Critical events in the form of SNMP traps are sent to this Manager to alert network operators of problems in the network. The Manager also acts as the central point for storing and displaying statistics maintained by the Agent (server). These statistics are maintained by the Agent in the form of MIB variables.

The Agent is responsible for reporting on and maintaining the data pertaining to a managed resource, such as a device or, in the case of the SP Agent, a set of configuration information identifying the nodes composing the SP system and tables of resource attributes within the SP which can be monitored. Agents can run on several different types of managed nodes.

Understanding the MIB

Every Agent supports a Management Information Base (MIB), a set of variables that represent the physical and logical resources of the managed systems or Agent. The MIB is not a database, in the sense of a monolithic collection of data, but rather it represents dynamic information. The values of the variables are maintained by different system functions such as the kernel, device drivers, or subsystems. The Agent obtains the variable values from these system functions.

The Manager can read variable values via these SNMP requests:

GET Requests the SNMP Agent to retrieve the value of the specified variable and return it to the requester (the Manager).

GET NEXT Requests the SNMP Agent to retrieve the value of the next variable, after the one specified in the request, and return it to the Manager. This is especially useful for retrieving tabular information, and multiple variables can be requested in it.

To summarize, the SNMP Manager determines the status of Agents by using the GET request to poll the Agents on a regular basis. However, if an Agent discovers an exceptional event, it will alert the Manager immediately, without waiting for a GET request, by sending a trap. The trap does not necessarily describe the problem in any detail; it simply informs the Manager that there is a change of state within the resources managed by the Agent, and the Manager may poll the Agent via an SNMP GET or GET NEXT request to determine more detail about the problem.

Enabling Network Management for the SP System

As we stated in the introduction to this chapter, in order for an SP to be part of a network management system, network managers must be notified of the AIX Error Log or Event Management events that you would like to centrally manage. A section of the SP proxy agent MIB (contained in the **/usr/lpp/ssp/config/snmp_proxy/ibmSPMIB.my** file) contains SP-specific SNMP configuration data, enabling Managers to view the SP as a system composed of processor nodes and a control workstation. In other words, there must be a way to create SNMP traps for SP error or problem events and there must be a MIB that contains configuration information specific to the SP system.

SNMP Traps for the SP from AIX Error Log Events

The **snmp_trap_gen** error notification method notifies the **sp_configd** daemon (or SP proxy agent) to create SNMP traps when selected types of errors are recorded in the AIX Error Log. (Note that the **trapgend** method from NetView for AIX also creates SNMP traps from AIX Error Log events. **trapgend** formats such events to be specifically compatible with SNA alerts. Only one of these methods should be used to avoid duplicate traps being generated for a single error.) The method and daemon run on each processor node and control workstation composing the SP system. In order for a trap to be created, using the **snmp_trap_gen** method, error log entries on each node and control workstation must contain an “alert=true” attribute. The following steps describe the flow of events for creating an SNMP trap for an AIX Error Log event:

1. An application or subsystem writes to the AIX Error Log facility.
2. If the Error Log entry contains an “alert=true” attribute, the **snmp_trap_gen** error notification method runs.
3. The **snmp_trap_gen** method uses the sequence number from the Error Log entry as input to the **errpt -a** command to obtain full information about the event.
4. **snmp_trap_gen** places the event information in a FIFO file, **/var/tmp/errlog_entry**
5. **sp_configd** reads the FIFO file, parses the information into objects within the **ibmSPErrlogVars** group of the **ibmSP** MIB and creates a trap from the objects whose instantiations contain non-null values.
6. **sp_configd** sends the trap to the AIX agent, **snmpd**, which in turn sends the trap to network managers specified in the **/etc/snmpd.conf** file existing on the node.

The following information is provided in SNMP traps from AIX Error Log events generated using the **snmp_trap_gen** method and the **sp_configd** daemon.

The **enterprise** field contains the OID (object identifier) of the **sp_configd** subagent (this is the OID assigned to the **ibmSP** MIB)

The **specific-trap** field contains the error ID from the Error Log entry (this is the convention expected by NetView for AIX when configuring events)

When the corresponding information exists in the Error Log entry, the **variable bindings** field contains the following object values paired with their OIDs:

- Error label
- Error ID

- Error log entry time stamp
- Unique sequence number
- Machine ID parameter
- Node ID parameter
- Error class
- Error type
- Resource name
- Resource class
- Resource type
- Identifies the location code of a device
- Vital product data
- Error description
- Probable causes
- User causes
- User actions
- Install causes
- Install actions
- Failure causes
- Failure actions
- Detail data

Other fields in the trap are as specified by SNMP protocol.

Designating AIX Error Log Entries as Alertable

In order for **sp_configd** to pass any entries in the system error log as traps to the **snmpd** daemon for transmission to Network Managers, the Error Record Templates for the entries must have the specific "alert" field set to the value **true**. (Make sure you are authenticated as the root user.) This may be accomplished as follows:

1. On the system in which the **sp_configd** daemon is running, use the **errpt -t** command to list the templates for entries in the AIX Error Log.
2. Select the IDs of the entries for which you wish to have traps issued when they occur.
3. Create a file containing the following two lines for each entry to be designated alertable:

   = *ID*:
   alert = true
4. From the command line, enter:

   errupdate *file*
5. To verify that the alert status has been changed to "true" (or 1), enter:

   errpt -tF alert=1 | grep *ID*

   listing the error template. You should see the specified ID listed.

SNMP Traps for the SP from Event Management Events

The following steps describe the flow of events for creating an SNMP trap for an Event Management event:

Designating that an SNMP Trap is Issued for an Event Management Event

1. Use the **pmandef** command to subscribe to an Event Management event and specify that an SNMP trap be issued for that event. For example:

   ```
   pmandef -s Filesystem_Monitor /
   -e'IBM.PSSP.aixos.FS%totused:NodeNum=1 ;VG=myvg;LV=mylv:X>95' /
   -t 1234 -n
   ```

   (In this example, whenever the file system associated with the **mylv** logical volume and **myvg** volume group on node 10 becomes more that 95% full, an SNMP trap will be generated on the control workstation.)

2. **pmand** (the Problem Management Event Management client daemon) subscribes to the event as specified in the **pmandef** command. The event must occur on the local node on which **pmand** is running.
3. **pmand** writes the contents of the Event Manager subsystem-supplied event response and the user-specified event configuration information into a FIFO file
4. **sp_configd** reads the data and creates an SNMP trap from it
5. **sp_configd** sends the trap to the SNMP managers specified in the **/etc/snmpd.conf** file on the node.
6. You specify a particular trap ID in the configuration information for an Event Manager event.

The following information is provided in SNMP traps from Event Management events generated by the **sp_configd** daemon.

The Contents of SNMP Traps Containing Event Management Events

The **enterprise** field contains the OID of the **ibmSP** MIB

The **specific-trap** field contains the trap ID field from the **pmand** configuration for this event. This allows configuration of NetView for AIX actions based on the type of event (which is done based on the specific trap ID)

The **variable bindings** field contains the following object values paired with their OIDs:

- The event ID
- Event flags
- A time stamp indicating the time the Event Manager generated the event
- The number of the node and the system partition address where the event occurred
- The resource variable name
- MIB table and instance information allowing further information about the variable involved with the event to be obtained from the **ibmSP** MIB.
- The value of the resource variable that triggered the event
- The predicate for which the event was triggered

Other fields in the trap are as specified by SNMP protocol.

The MIB for the SP

The **ibmSP** MIB is provided to define SP-specific information. The **ibmSP** MIB consists of an **ibmSPConfig** group, the **ibmSPErrlogVars** group, and the **ibmSPEMVariables** group. The **ibmSP** MIB is instantiated on each SP processor node and the control workstation.

The **ibmSP** MIB is defined in the source file **ibmSPMIB.my** in the **/usr/lpp/ssp/config/snmp_proxy** directory. For a sample of the **ibmSP** MIB file, see "ibmSPMIB.my" on page 481.

The **ibmSPConfig** group defines objects containing SP system configuration information. The group consists of objects containing the following information:

The node number of the node on which the queried subagent is running

The IP address of the system partition in which the queried node resides

The host name of the primary control workstation.

The operational state of the primary control workstation (up or down)

The host name of the backup control workstation (will be **NULL** if none exists)

The operational state of the backup control workstation (up or down)

The version number of the IBM Parallel System Support Programs for AIX running on the control workstation

A table of SPNodeEntrys. Each row in the table is indexed by the IP address of a system partition combined with a node number. Each row contains the following information about a node:

- The IP address of the system partition in which it resides
- The node number
- The number of the frame containing the node
- The lowest slot number in the frame containing the node
- The number of slots in the frame occupied by the node
- The reliable host name assigned to the node (this is the host name associated with the SP Ethernet)
- The initial host name assigned to the node (this is the host name assigned to the node during the customization based on data obtained from the SP Perspectives GUI)
- The name of the system partition in which the node resides
- The version number of the IBM Parallel System Support Programs for AIX running on the node

The **ibmSPConfig** group, when instantiated on the control workstation, will identify the default system partition IP address as the IP address of the system partition containing the reporting node and **0** as the node number of the reporting node.

The **ibmSPErrlogVars** group is instantiated on each SP processor node and the control workstation. The group consists of a sequence of objects containing information about the latest Error Log write that caused an SNMP trap to be issued on the reporting node.

The **ibmSPEMVariables** group is instantiated on each SP processor node and the control workstation. The group consists of a sequence of objects containing information about the last Event Management event that caused an SNMP trap to

be issued on the reporting node and tables of objects representing Event Management variables. The group contains up to three tables:

A table of objects (**ibmSPEMNodeDepVarsTable**) allowing access to the definitions of node-dependent resource variables (those defined with an Event Manager 'Locator' attribute). This table contains object instantiations for only those variables which have a Locator value equalling the node number of the node on which the SP proxy agent is running.

A table of objects (**ibmSPEMNodeIndepVarsTable**) allowing access to the definitions of node-independent resource variables (those defined without an Event Manager 'Locator' attribute)

A table of objects (**ibmSPEMVarValuesTable**) allowing access to the current values assigned to Event Management resource variables

The following information is provided through object instantiations for each of the Event Management resource variables in the **ibmSPEMNodeDepVarsTable** and **ibmSPEMNodeIndepVarsTable** tables:

Strings containing the following Event Management variable information:

- Name of the resource variable
- Resource variable description
- Variable type (counter, quantity, or state)
- Data type (long, float, structured byte string)
- Structure byte string format (only for a structured byte string data type)
- Initial variable value
- An index into the **ibmSPEMVarValuesTable**, which, when used in combination with an instantiation vector, provides access to an entry containing the current value of a variable instance
- Variable's resource class
- Instantiation vector definition
- Description of each element in the instantiation vector
- Name used to read and write the variable in the PTX shared (may be **NULL**) memory
- Default predicate for event notification (may be **NULL**)
- A description of the event
- Locator value specified for the variable
- Resource order group

The **ibmSPEMVarValuesTable** provides object instantiations for the current values of variables from both the **ibmSPEMNodeDepVarsTable** and the **ibmSPEMNodeIndepVarsTable** tables.

The portion of the **ibmSPEMVariables** group instantiated by the **sp_configd** subagent depends on the type of SP node the subagent is running on. The **ibmSPEMNodeDepVarsTable** and **ibmSPEMVarValuesTable** tables are instantiated on all the processor nodes and the control workstation. The **ibmSPEMNodeIndepVarsTable** table is instantiated on only the control workstation.

Viewing the ibmSP MIB

The AIX **snmpinfo** command may be used in place of a manager client, such as NetView for AIX, to view the contents of the **ibmSP** MIB. Use of this command requires **root** authentication. The following are examples of using **snmpinfo**:

1. To dump the contents of the **ibmSP** MIB located on the host *host_name* in verbose mode, enter the command:

   snmpinfo -m dump -v -h *host_name* ibmSP

2. To dump the contents of the **ibmSPConfig** group, enter the command:

   snmpinfo -m dump -v -h *host_name* ibmSPconfig

3. To get values for the MIB variable **SPhostnodenumber.0** located on the local host, enter:

   snmpinfo -m get ibmSPhostnodenumber.

4. To get the value for the MIB variable following the **ibmSPhostnodenumber** variable, enter:

   snmpinfo -m next ibmSPhostnodenumber.

Configuring the SP Proxy Agent

The SP proxy agents (**sp_configd**) residing on each managed SP node and control workstation are configured to be started automatically whenever AIX is started. The configuration changes necessary to accomplish this are performed during installation on the nodes and the control workstation.

If you have a management station that "listens" for traps, place the information for the trap destination in the **/etc/snmpd.conf** file on each SP node and control workstation. Instructions for doing this are in the **/etc/snmpd.conf** file. The following steps are performed automatically by the **syspar_ctrl** script on each node and the control workstation when the PSSP software is installed:

1. These steps presume that the standard AIX Agent is already configured to be managed by the desired Managers. If this is not the case, see the information contained in the **/etc/snmpd.conf** file for required updates to the file. This update must be made on every SP node and the control workstation.
2. The **/etc/snmpd.conf** file is modified to add the SP subagent (**sp_configd**) to the set of SMUX peers with a "smux" record
3. The SP MIB file **/usr/lpp/ssp/config/snmp_proxy/ibmSPMIB.defs** is appended to the **/etc/mibs.def** file. The **/usr/lpp/ssp/config/snmp_proxy/ibmSPMIB.defs** file is compiled from **ibmSPMIB.my**
4. Information for the new **sp_configd** subagent is added to the **/etc/snmpd.peers** file. The information is the OID of the Agent and the community name (password).
5. The **snmpd** daemon is refreshed.
6. The daemon is started under SRC control.
   a. An src_entry for **sp_configd** is added to the AIX Object Data Manager (ODM) so that the daemon can be controlled using the **startsrc** command. The **src_entry** invokes the **startsrc** command with the following parameters:

```
-s sp_configd -a "-t 6  " -p /usr/lpp/ssp/bin/spconfigd
-u   -i /dev/null -o /dev/null -e /dev/null -S -n 15 -F 15
```

b. An entry is added to the AIX **inittab** file to start the daemon by issuing the command:

```
/usr/bin/startsrc -s sp_configd
```

c. The **sp_configd** daemon supports a **-t** switch that determines the amount of time (in seconds) for which data is to be cached by the daemon and re-fetched when a new request for the data is received. The default value is 60 seconds. The default can be changed in the src_entry or the **startsrc** command contained in the **inittab** file, or by entering **startsrc** at the command line.

d. The daemon can be stopped by issuing:

```
/usr/bin/stopsrc -s sp_configd
```

Configuring NetView for AIX for the SP System

If you are using NetView for AIX as the network manager for your SP system, you must configure it to recognize SP trap and configuration information.

Configuring the SP MIB to be Used by NetView for AIX

1. Copy the file **/usr/lpp/ssp/config/snmp_proxy/ibmSPMIB.my** from an SP node into **/usr/OV/snmp_mibs** as **ibmSP.mib** on the node running NetView for AIX
2. Load the MIB using the NetView for AIX menu option **Load/Unload MIBs**.

Configuring SP Events to be Used by NetView for AIX

Traps generated containing events from SP nodes and the control workstation are displayed automatically at the NetView for AIX **Control Desk** if the node on which the event originated is a managed node. If the node is not managed, the event is recorded in the event history file and is displayed only when you query the event history file. The traps contain object values indicating:

- The trap ID
- The variable name
- The event predicate definition that caused the event
- The value of the variable at the time the event was generated
- Information necessary to use the NetView for AIX browse facility to query the node for the current value of the variable or for more information about the characteristics of the variable and its value

You may tailor the information in the traps by using the **Trap Customization** selection under the **Event/Configuration** AIX menu option.

Chapter 26. Managing Error Logs

The IBM RS/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 Parallel System Support Programs for AIX: 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 RS/6000 SP System Management

The RS/6000 SP System Management menu appears.

SELECT RS/6000 SP Log Management

The RS/6000 SP Log Management menu appears.

© Copyright IBM Corp. 1996

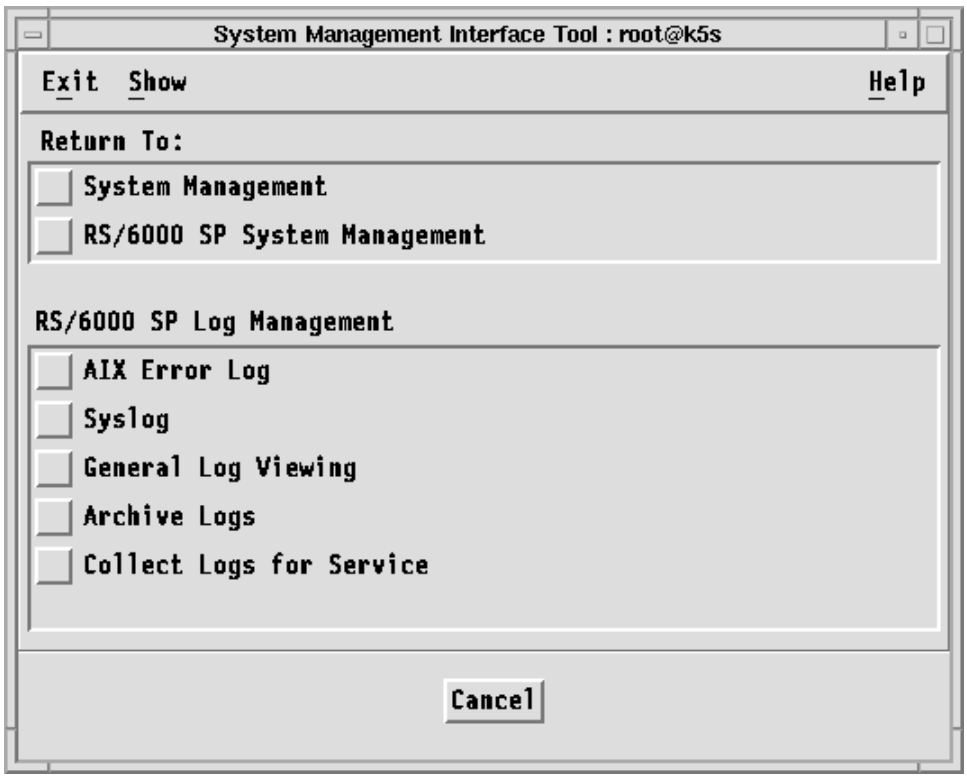


Figure 79. Log Management SMIT Panel

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 175 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 15, “Sysctl” on page 185 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 RS/6000 Problem Solving Guide

IBM General Concepts and Procedures for RS/6000

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
```

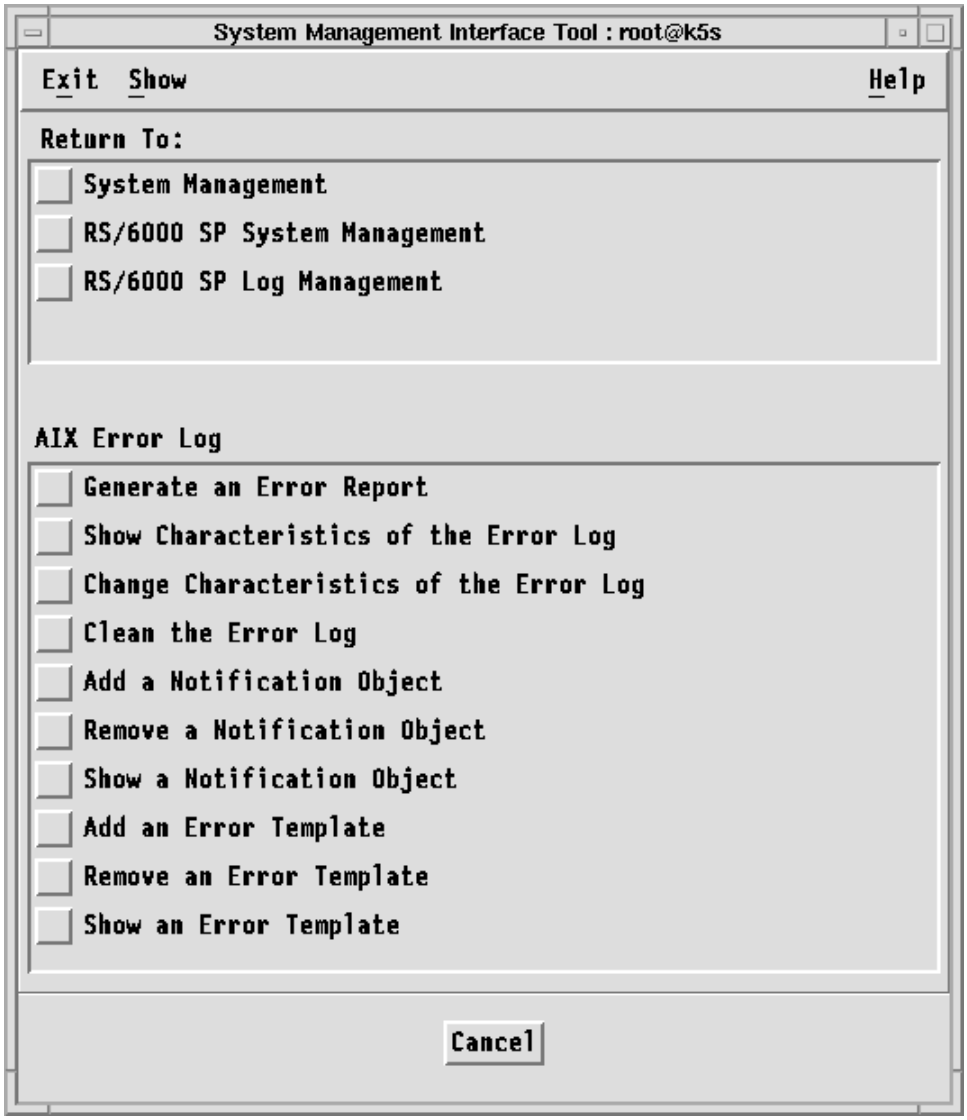


Figure 80. AIX Error Log SMIT Panel

Generating AIX Error Log Reports

You can generate reports on entries in the AIX Error Log on a number of nodes. The report can be displayed or written to a file on the local node.

From SMIT: The fastpath invocation for the Generate an Error Report menu is:

smit perrpt

Trimming AIX Error Logs

You can trim records from error logs on a set of nodes.

From SMIT: The fastpath invocation for the Clean the Error Log menu is:

smit perrclear

Configuring the AIX Error Log

You can display the configuration parameters of the AIX Error Log to the local node.

From SMIT: The fastpath invocation for the Show Characteristics of the Error Log menu is:

```
smit perrdemon_shw
```

You can alter one or more of the configuration parameters for the AIX Error Log on a set of nodes. Because of the additional entries generated by SP system software, we recommend AIX Error Log file size be a minimum of 4M.

From SMIT: The fastpath invocation for the Change Characteristics of the Error Log menu is:

```
smit perrdemon_chg
```

Managing Error Notification Objects

Error Notification Objects are ODM objects held in the class errnotify that are used by the AIX Error Notification Facility to invoke methods upon occurrence of an error event. Fields in the errnotify class match to fields in an Error Template for selection. If an error is logged matching the selection criteria defined in a notification object, the method associated with that object is invoked. For more information on using the AIX Error Notification Facility, refer to *IBM General Concepts and Procedures for RS/6000*.

You can add, remove, and show notification objects in parallel on the SP system.

From SMIT:

The fastpath invocation for the Add a Notification Object menu is:

```
smit padd_en
```

The fastpath invocation for the Remove a Notification Object menu is:

```
smit prem_en
```

The fastpath invocation for the Show a Notification Object menu is:

```
smit pshw_en
```

From the Command Line: To add, remove, or show error notification objects in parallel on the SP system, enter:

```
penotify -f show
```

For complete details on the command, refer to the *IBM Parallel System Support Programs for AIX: Command and Technical Reference*.

Following is a description of the actions taken by the notification method EN_pend located under the **/spdata/sys1/err_methods** directory. This method can be installed and used to invoke pre- and post- action scripts and mail a report of the logged error. This script provides a suggested structure for notification methods and can be reused with different pre- or post- action scripts as described in the sections.

EN_pend Method Flow

1. EN_pend looks under the directory it resides in for a file with the same name and a **.envs** suffix. If found, it sources the file to pick up environment variables EN_RUNDEFAULT and EN_MAILLOC. An **EN_pend.envs** script is installed under the same directory.
2. **EN_pend.envs** sets the EN_RUNDEFAULT environment variable. It also sets the EN_MAILLOC to root at the control workstation, if possible, or to root at the local node.
3. **EN_pend** checks for a pre-action script under the same directory and name with a **.pre** suffix and executes it if found.
4. **EN_pend** mails an expanded report of the error using the sequence of the error passed by the notification facility to the EN_MAILLOC, if EN_RUNDEFAULT and EN_MAILLOC variables are set.
5. **EN_pend** checks for a post-action script under the same directory and name with a **.post** suffix and executes it if found.

Installing a Notifcation Object

To add the **EN_pend** method to all nodes in the current partition to send a report whenever an error of type PEND (loss of availability of a device is imminent) occurs, enter:

```
penotify -a -n “PEND_err” -P -t “PEND” -m ‘/spdata/sys1/err_methods/EN_pend $1’

penotify -a -n “pend_err” -P -t “pend” -m ‘/spdata/sys1/err_methods/EN_pend $1’

penotify -a -n “Pend_err” -P -t “Pend” -m ‘/spdata/sys1/err_methods/EN_pend $1’
```

The **-P** flag will cause the object to persist after the system is restarted. Three objects are added with variations on PEND because upper case is not always adhered to by all AIX LPPs and vendor functions. The **$1** argument causes the Error Notification Facility to pass the error sequence number to the notify method.

The **EN_pend** and **EN_pend.envs** scripts can be used to invoke different pre- and post- action scripts for different error events by creating links to them. **EN_pend** looks for **.envs**, **.pre** and **.post** scripts under the directory it is called from, and by the same basename. For example, to use **EN_pend** for reporting hdisk0 errors on nodes h0, h1, h2 and h3 and perform pre- and post- action:

1. Create .pre and .post action scripts on one of the nodes, for example, EN_hdisk0.pre, EN_hdisk0.post under the /**spdata**/**sys1**/**err_methods** directory on node h0.
2. Copy the pre- and post- scripts to nodes h1, h2 and h3 using **pcp**:

   ```
   pcp -w h1,h2,h3 EN_hdisk .pre /spdata/sys1/err_methods/
   pcp -w h1,h2,h3 EN_hdisk .post /spdata/sys1/err_methods/
   ```

3. Create links to the **EN_pend** and **EN_pend.envs** scripts:

   ```
   dsh -w h ,h1,h2,h3 ln -s /spdata/sys1/err_methods/EN_pend /spdata/sys1/err_methods/EN_hdisk

   dsh -w h ,h1,h2,h3 ln -s /spdata/sys1/err_methods/EN_pend.envs /spdata/sys1/err_methods/
   EN_hdisk .envs
   ```

4. Add the notification object:

   ```
   penotify -w h ,h1,h2,h3 -f add -P -n “hdisk _err” -m ‘/spdata/sys1/err_methods/EN_hdisk  $1’ -N
   “hdisk ”
   ```

To display the notification object created, enter:

```
penotify -w h ,h1,h2,h3 -f show -n “hdisk _err”
```

To remove this notification object, enter:

```
penotify -w h ,h1,h2,h3 -f remove -n “hdisk _err”
```

Note: Notification methods need to be accessible to each node that the notify object is added to. We recommend that the notification scripts be kept local on each node in case of network failure. File collections should be used to maintain updates to notification methods.

Managing Error Templates

You can create a new error template in the Error Template repository for logging errors.

From SMIT: The fastpath invocation for the Add an Error Template menu is:

```
smit padd_et
```

You can remove a template from the Error Template Repository.

From SMIT: The fastpath invocation for the Remove an Error Template menu is:

```
smit prem_et
```

You can display entries from the Error Template Repository to the local host.

From SMIT: The fastpath invocation for the Show an Error Template menu is:

```
smit pshw_et
```

Managing the BSD syslog Facility

SP error logging utilizes the BSD syslog facility for recording error events.

From SMIT: To access the Syslog SMIT menu, enter:

```
smit spsyslog
```

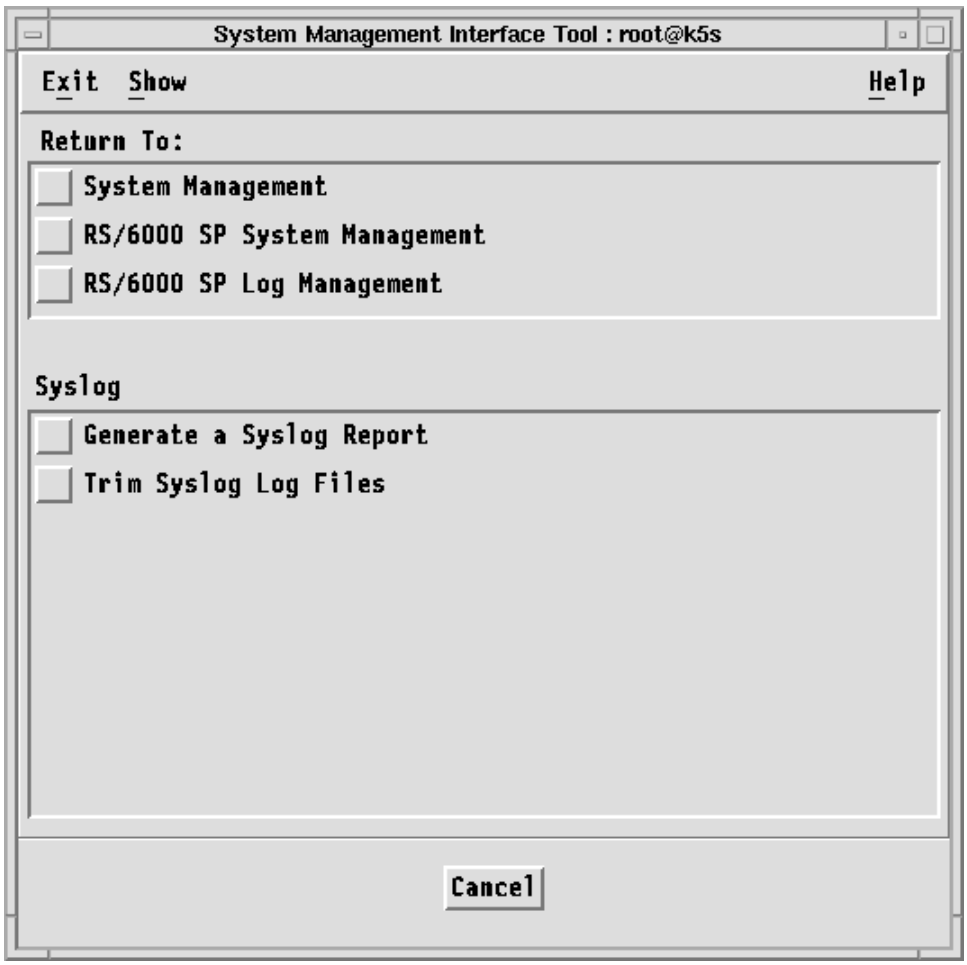


Figure 81. Syslog SMIT Panel

Generating Reports on BSD syslog Log Files

The **syslogd** daemon, which logs the errors, is configured with a file designating filters for the incoming messages to determine their destination. The default configuration file is /**etc**/**syslog.conf**. BSD syslog errors are classified by the facility that is issuing the error, and by the error's priority value. Refer to **syslogd** in the *IBM AIX Version 4.1 Commands Reference*, SBOF-1851, for a list of facility and priority values. Entries in the configuration file determine the destination for each error message based on these values. Destinations can be a file, user ID, or the **syslogd** daemon on another machine. We recommend that error messages be logged locally rather than forwarded to a remote **syslogd** because of the increased network traffic. File collections can be used to maintain consistent configuration of the syslog facility.

The format of a syslog log file is:

MMM DD HH:MM:SS node_name resource[pid]: msg

Where:

MMM DD HH:MM:SS Timestamp (month day hour:minute:second)

node_name The name of the node that the error occurred on

resource The name of the failing system

pid The optionally logged process ID of the failing resource.

msg A free form error message.

Errors logged by SP components will contain in the message section the following additional information pertaining to the logging resource:

LPP LPP name

Fn filename

SID SID_level_of_the_file

L# Line number or function

From SMIT: The fastpath invocation for the Generate a Syslog Report menu is:

```
smit spsyslrpt
```

From the Command Line: To report to the local node all records logged by **ftp** for all syslog log files on nodes in the current system partition, enter:

```
psyslrpt -a -r ftp
```

To report all records that logged to files selected for the daemon and user facilities starting on March 3 and to report the records to the local node from nodes host1 and host2, enter:

```
psyslrpt -w host1,host2 -f user,daemon -s  3 3
```

For complete details on the **psyslrpt** command, refer to the *IBM Parallel System Support Programs for AIX: Command and Technical Reference*.

Trimming BSD syslog Log Files

The size of syslog log files is not configurable and will continue to grow until manually trimmed.

From SMIT: The fastpath invocation for the Trim Syslog Log Files menu is:

```
smit spsyslclr
```

From the Command Line: To trim all records older than 30 days from the log file **/var/adm/msgs** on the local node, enter:

```
psyslclr -y 3  -l /var/adm/msgs
```

To trim all records from all log files found in the alternate syslog configuration file **/var/adm/syslog.conf**, enter:

```
psyslclr -g /var/adm/syslog.conf -y
```

Note: The **syslogd** daemon is stopped during the trimming process, then restarted with either the default configuration file or an alternate file, if used (**-g** option). For complete details on the **psyslclr** command, refer to the *IBM Parallel System Support Programs for AIX: Command and Technical Reference*.

psyslclr can be added as a **crontab** entry for performing scheduled syslog trimming. On the control workstation, **psyslclr** is used to trim daemon facility messages older than six days. This is done in **/usr/lpp/ssp/bin/cleanup.logs.ws**,

which is run from the control workstation's **crontabs** file. (Refer to Chapter 13, “Maintaining the crontabs File” on page 147 for more information.)

Viewing Error Logs

You can view error log data.

From SMIT: To access the General Log Viewing SMIT menu, enter:

```
smit splogview
```

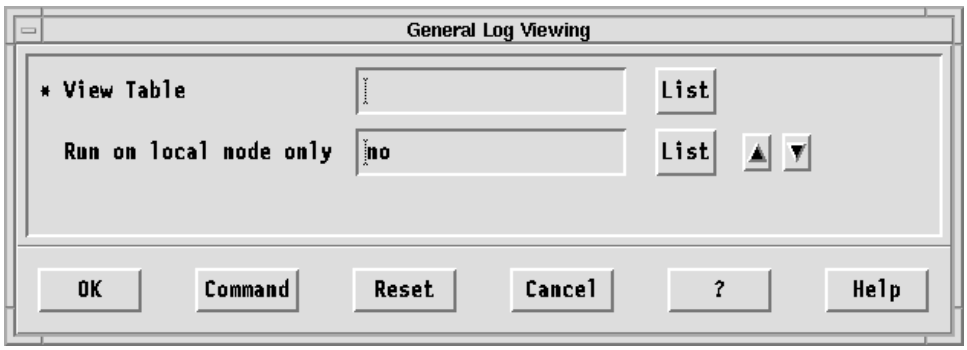


Figure 82. General Log Viewing SMIT Panel

From the Command Line: To cause the contents of each file entry and output of each command entry on the target nodes in the log table **amd.tab** to be displayed to stdout of the node the **splm** command is executed on, enter:

```
splm -a view -t /spdata/sys1/logtables/amd.tab
```

For complete details on the **splm** command, refer to the *IBM Parallel System Support Programs for AIX: Command and Technical Reference*.

Archiving Error Logs

Error log archives can be created, removed, or gathered to a central location where they can be optionally written to a tape device or mailed to another location.

From SMIT: To access the Archive Logs SMIT menu, enter:

```
smit sparchive
```

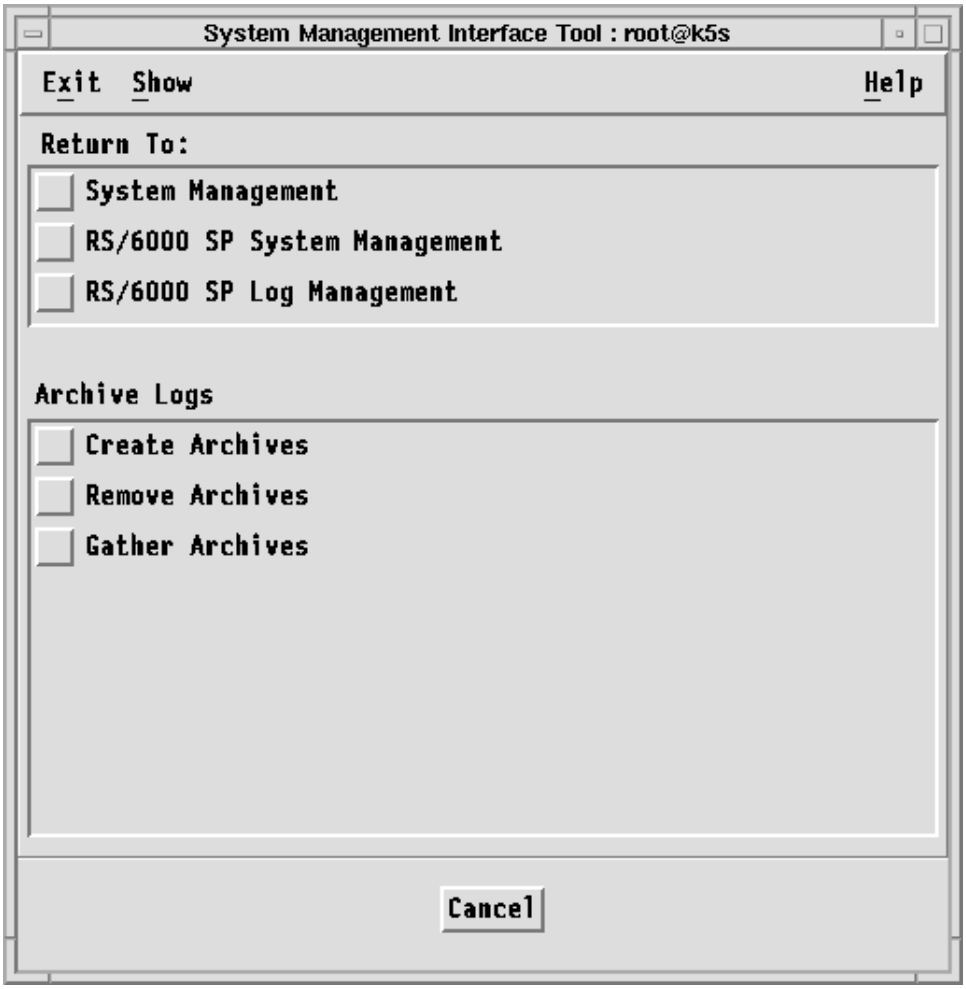


Figure 83. Archive Logs SMIT Panel

Creating Archives

From SMIT: The fastpath invocation for the Create Archives menu is:

```
smit spcreate_archive
```

From the Command Line: To create an archive on each target node under directory **/var/archives/arch_weekly.tab**, enter:

```
splm -a archive -t /spdata/sys1/logtables/weekly.tab -c -d /var/archives
```

Removing Archives

From SMIT: The fastpath invocation for the Remove Archives menu is:

```
smit spremove_archive
```

From the Command Line: To remove all files and directories under and including **/var/archives/arch_weekly.tab**, enter:

```
splm -a archive -t /spdata/sys1/logtables/weekly.tab -r -d /var/archives
```

Gathering Archives

From SMIT: The fastpath invocation for the Gather Archives menu is:

```
smit spgather_archive
```

From the Command Line: To gather the compressed tar files created on each target node under directory **/var/archives/arch_weekly.tab** to the directory **/var/logrepos** on the local node.

```
splm -a gather -k archive -t /spdata/sys1/logtables/weekly.tab -d /var/archives -l /var/logrepos
```

For complete details on the **splm** command, refer to the *IBM Parallel System Support Programs for AIX: Command and Technical Reference*.

Collecting System Data for IBM Service

Service collections can be created, removed and gathered in the same manner as the archive functions with the addition of interacting with the AIX command **snap** to gather additional system information. The default top level directory for service collections is **/tmp** and this path will end with **srvc**_*log_table_name*.

From SMIT: To access the Collect Logs for Service SMIT menu, enter:

```
smit spcollect
```

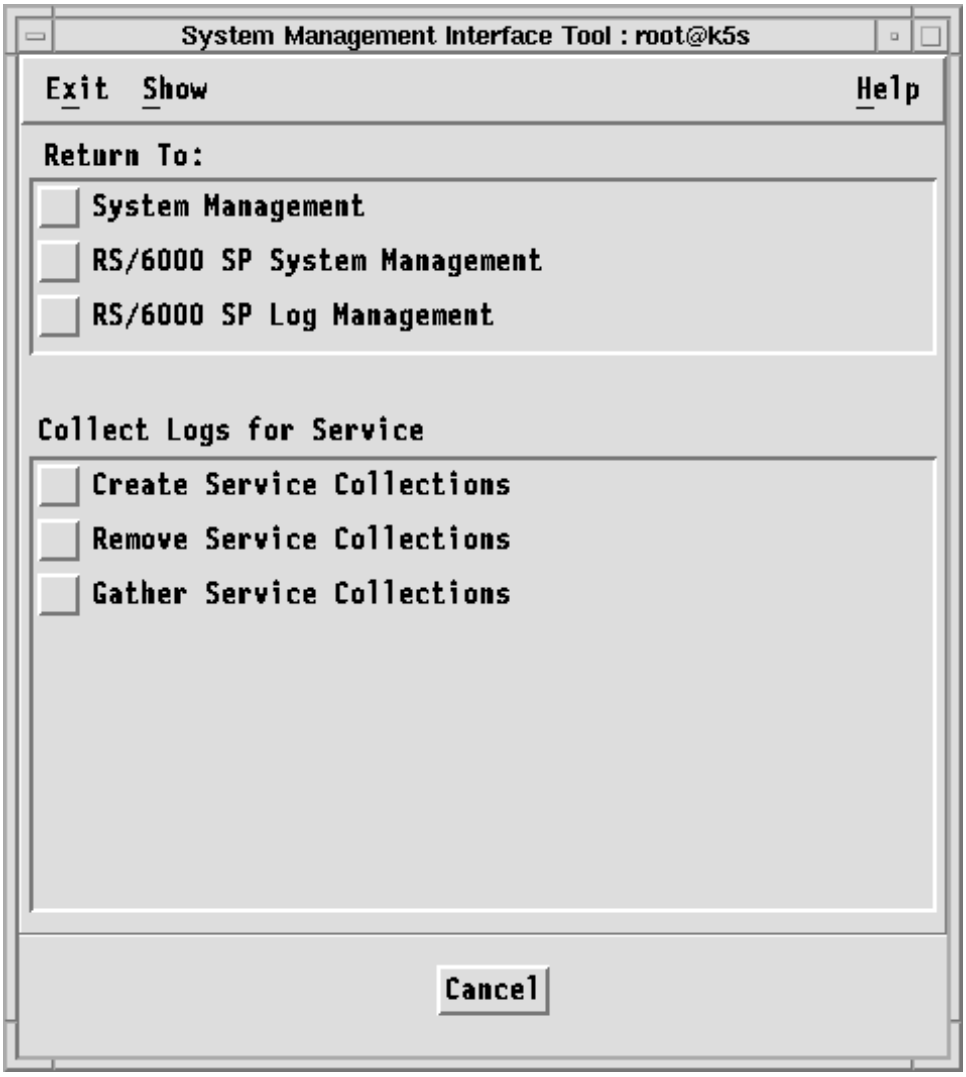


Figure 84. Collect Logs for Service SMIT Panel

Creating Service Collections

From SMIT: The fastpath invocation for the Create Service Collections menu is:

```
smit spcreate_collect
```

From the Command Line: In the following example service collections are created using table **ssp.tab**. **snap -gfk** will be called to gather general system information, file system information and kernel information. Additional log and system data designated in the **ssp.tab** table will be collected under the other directory created by **snap**. The *nodename*.**tar.Z** compressed tar file will be created.

```
splm -a service -t /spdata/sys1/logtables/ssp.tab -p 'gfk' -c
```

Removing Service Collections

From SMIT: The fastpath invocation for the Remove Service Collections menu is:

```
smit spremove_collect
```

From the Command Line: In the following example, the service collections that were created using table **ssp.tab** are removed:

```
splm -a service -t /spdata/sys1/logtables/ssp.tab -r
```

Gathering Service Collections

From SMIT: The fastpath invocation for the Gather Service Collections menu is:

```
smit spgather_collect
```

From the Command Line: In the following example the compressed tar files from the nodes in table **ssp.tab** are gathered to the local node and written to tape device /**dev**/**rmt0**. The **-s** flag specifies to stagger the process by gathering the file from one node, writing it to tape, then removing it before moving onto the next node. This provides a method for gathering all the compressed tar files to a tape or disk device, or mailing them, without requiring disk space on the local node for all the service tar files.

```
splm -a gather -k service -t /spdata/sys1/logtables/ssp.tab -d /tmp -s -o /dev/rmt
```

For complete details on the **splm** command, refer to the *IBM Parallel System Support Programs for AIX: Command and Technical Reference*.

For information on the **snap** command, refer to the *IBM AIX Version 4.1 Commands Reference*, SBOF-1851

System Performance

© Copyright IBM Corp. 1996

Chapter 27. Tuning the SP System

This chapter presents tuning information for the SP system. The chapter consists of six primary sections:

- The first section deals with how to approach tuning an SP system. In many cases tuning an SP system is different from, sometimes even opposite to, tuning a RS/6000 workstation.
- The second section deals with understanding the network topology of the SP system. Understanding the network topology aids in tuning for optimal performance.
- The third section deals specifically with tuning the networks. Network tuning is the key to achieving peak performance on the SP system. Network tuning must be done carefully due to the different approach to tuning required by the SP system.
- The fourth section addresses tuning the SP system switch adapters.
- The last two sections address specific problems and subsystems and how to adjust the tuning of the SP system for them.

We recommend that you read the first four sections before attempting to tune specific subsystems or to alleviate bottlenecks. Understanding all aspects of tuning an SP system before actually making any changes will help you to tune for optimal performance.

Once the SP system subsystems and networks are properly tuned, use standard AIX tuning within a node. Additional AIX tuning information can be found in *AIX Versions 3.2 and 4.1 Performance Tuning Guide*, SC23-2365.

Performance Tuning and Monitoring

When tuning or monitoring the performance of the SP system you must make sure that you know what you change, when you change it, when you collect any data, and what the data you collect is for. Not using change control, or having time stamps on performance data, makes tuning the SP system a very difficult task. We highly recommend that a change control system is used to track and monitor any changes to the tunables on any part of the SP system, including the control workstation and the nodes.

You must take into account the total system picture when you're tuning large configurations. For example, if you're tuning a partitioned SP system, subsystems that span several system partitions (such as the SP system Ethernet) can cause changes on more than one system partition.

Some administrative tasks, such as collecting and clearing log files, cause network traffic that could affect system performance. You should consider scheduling such tasks for off shift hours.

The approach to tuning the SP system is, in some situations, the opposite of how you would tune an AIX workstation. In tuning an AIX workstation or server, the most common approach is to tune the machine to handle the amount of traffic or services requested of it. In the case of a file server, the server is tuned to the

© Copyright IBM Corp. 1996

maximum amount of traffic it can receive. In general, the bottleneck in a high-end server is the capacity of the network through which services are requested.

In the SP system, a switch is faster than any other network available. With the non-blocking nature of a switch, the number of requests and volume of data that may be requested of a node can far exceed the node's capacity. To properly handle this situation, the application or tuning on the SP system need to manage the amount of services requested of a server or destination node. Instead of trying to tune a node for the maximum amount of services requested, each of the nodes requesting services needs to manage the volume of requests made of another node. In other words, you should reduce the number or volume of requests at the client, rather than increase the capacity of the server. It is very easy on large SP system configurations to require more services than the most powerful node can currently deliver.

Newly installed nodes will start off with the AIX default tuning settings. These are not appropriate for peak network performance, especially over a switch. You need to set the tunables for the SP system when you install the system. Information on setting tunables during installation can be found in Table 15 on page 417 and Table 16 on page 417.

Changing Network Options on the Nodes

There are two methods for changing the network option tunables on the nodes after you have installed your SP system.

Changing Network Options on the Node

To change the network option values set during boot of the nodes from the control workstation, take the following steps:

1. On the control workstation, copy the file **/usr/lpp/ssp/samples/tuning.cust** file to **/tftpboot/tuning.cust**.
2. Select the tunable values that are best for your system. For tuning specific network adapters, refer to the *IBM Parallel System Support Programs for AIX: Installation and Migration Guide*. For tuning for specific user environments, refer to Table 15 on page 417. This can be done only for AIX 4.1 nodes (not AIX 3.2.5).
3. Edit **tuning.cust**. Uncomment the **/usr/sbin/no** commands and change to the appropriate values.
4. Run **spbootins** to set the affected nodes to customize mode.
5. To make the changes to the nodes, you can either reboot the node or run **/spdata/sys1/install/pssp/pssp_script** on each affected node. Use **dsh** to run **pssp_script** on several nodes at once. If you are using boot/install servers, customize the boot/install server first before customizing its client nodes. The revised **tuning.cust** file will be transferred from the control workstation to the boot/install server and then from the boot/install server to the client nodes.

After the node customization has completed, all subsequent boots will use the new network tunables on the nodes.

Changing Network Options on Selected Nodes

You can change the network tunables for a node locally on the node. Take the following steps:

1. Login to the node as root and edit the file **/tftpboot/tuning.cust**. If there is no **/tftpboot/tuning.cust** file, make a new one by copying **/usr/lpp/ssp/samples/tuning.cust** to **/tftpboot/tuning.cust** on the node.
2. Select the tunable values that are best for your system. For tuning specific network adapters, refer to the *IBM Parallel System Support Programs for AIX: Installation and Migration Guide*. For specific user environments, refer to Table 15 on page 417 This can be done only for AIX 4.1 nodes (not AIX 3.2.5).
3. Edit **tuning.cust**. Uncomment the **/usr/sbin/no** commands and change the default tunable values.
4. Once you have made the changes, run the **tuning.cust** script locally to make the changes on the node. There is no need to reboot for the new values to become effective. You may need to change the permission on the file to be able to execute it.

   Note: This is a temporary change. The next time the node is customized, the **tuning.cust** file will be transferred from the node's boot/install server and executed on the node, thus replacing any changes made locally. To make the network options permanent, change the **tuning.cust** file on the node's boot/install server, which in some cases may be the control workstation.

System Topology Considerations

When you set up a network for a large system you need to consider:

The SP Ethernet
Outside network connections
Routers
Gateways
Switch traffic

The most important consideration in configuring the SP Ethernet is the number of subnets you configure. The routing through the SP Ethernet can be complicated because of the limitation on the number of simultaneous network installs per Ethernet subnet. More information regarding the topology of the SP Ethernet can be found in *IBM Parallel System Support Programs for AIX: System Planning*.

When configuring software that uses the SP Ethernet, consider such software's location in the SP Ethernet topology. For example, software such as LoadLeveler and POE use the SP Ethernet to communicate to other nodes. Installing such software on a far point in the network in a large SP configuration can cause bottlenecks on the network subnets and adapters connected to the control workstation. Installing such software on the control workstation causes the lowest possible traffic; however, the control workstation has to be powerful enough to act as the control workstation and, in addition, support the LoadLeveler and POE software.

If you connect the SP Ethernet to your external network, make sure that the user traffic does not overload the SP network. You can overload the SP Ethernet if you route high speed network (FDDI or ATM, for example) traffic through it. Route

gateway traffic over a switch for gateways to FDDI, ATM, and other high speed networks. Configure routers or gateways to distribute the network traffic so that one network or subnet is not a bottleneck.

Several gateways should be configured if a lot of traffic is expected. All traffic on these networks can be monitored using the standard network monitoring tools. Details about the monitoring tools and their usage can be found in *AIX Versions 3.2 and 4.1 Performance Tuning Guide*, SC23-2365.

In a system that consists of just the SP Ethernet and a switch, you cannot optimize so that you achieve peak performance on both networks. You have to decide which network you want to be affected.

A list of recommended network tunables for specific adapters can be found in the *IBM Parallel System Support Programs for AIX: Installation and Migration Guide*.

Tuning the SP Ethernet

When you tune the SP network, you must consider the number of frames, the number of nodes, and the number of networks you are configuring.

The lower the amount of traffic in each network, the better the performance or response time for the system. If you run a small system of fewer than two frames, you probably connected the SP network to your own network through a gateway. For larger configurations, we recommend that you dedicate one node per frame as a gateway to the rest of the SP system. Routing from the gateway node can be either through the SP Ethernet or across a switch.

When routing across a switch, you should give careful consideration to other traffic moving across the switch. An environment which has many parallel jobs will suffer in performance if a lot of the user traffic is routed through a switch. Such traffic can interrupt the parallel jobs sharing a switch.

In larger SP system configurations, the network topology and the name server can cause problems. If there are many name server queries at the same time, and the topology from a node to the name server is complex, performance can suffer. Under these circumstances, there are three possible solutions:

1. Create a name server within the SP system complex so you are not at risk for traffic problems to an external name server.
2. Don't use the name server, but create an /**etc**/**hosts** file of all addresses that the SP system and the applications need, and change the nodes to resolve their addresses locally.
3. For an AIX 4.1.4 system or later, specify the name lookup hierarchy to first look locally in the /**etc**/**hosts** file, then the name server. (This is not possible in AIX systems earlier than 4.1.4.)

SP System-Specific Tunable Recommendations

The tunable values are set to either AIX defaults or the tunables you set during the installation process. See “Tuning for User Environments” on page 416 for the recommended defaults. Use the **netstat** and **no** commands to display tunable values. These commands and their parameters are described in the *IBM Parallel*

System Support Programs for AIX: Command and Technical Reference. The values you can tune are:

thewall The upper bound on the amount of real memory that can be used by the communications subsystem. The units are in 1K increments.

lowmbuf The minimum number of 256-byte mbuf buffers kept available at all times. (AIX 3.2.5 nodes only)

lowclust The minimum number of 4096-byte mbuf clusters kept available at all times from the mbuf pool. (AIX 3.2.5 nodes only)

mb_cl_hiwat The maximum number of 4096-byte mbuf clusters that can be used from the mbuf pool. (AIX 3.2.5 nodes only)

sb_max Upper limit on the size of the TCP and UDP buffers in mbufs of allocated space to a connection.

ipforwarding Specifies whether the kernel should forward packets. A value of **1** forwards packets when they are not for the local network; a value of **0** prevents forwarding.

tcp_sendspace The default size of the TCP send space value.

tcp_recvspace The default size of the TCP receive space value.

udp_sendspace The default size of the UDP send buffer. The safe maximum is 65536 (64K).

udp_recvspace The size of the UDP receive buffer.

rfc1323 Turns on several enhancements to the tcp_sendspace and tcp_recvspace tunables for high speed networks:

The maximum window size is increased from 64K to 4 Gigabytes.

ACKs are time-stamped for better round trip time estimates.

Sequence numbers are protected against wrapping.

tcp_mssdflt The default maximum segment size used in communicating with remote networks.

MTU Maximum Transmission Unit on network adapters.

Refer to the *IBM Parallel System Support Programs for AIX: Installation and Migration Guide* for a list of initial tunable settings.

AIX 3.2.5 Differences

The way that TCP/IP works is that you allocate two different size allocations out of the mbuf pool. They are mbufs using 256 byte allocations, and mbuf clusters using 4096 byte allocations. Whenever the size of the buffer you are sending is greater than 935 bytes, you need to use an mbuf cluster for staging the data. In addition, if you need an mbuf cluster, you also need an mbuf to point to the mbuf cluster from the mbuf pool.

In AIX 3.2.5, the total mbuf pool can be available to either mbufs or mbuf clusters. The maximum amount of space for the mbuf pool is specified using the **thewall** value. This value is in 1K sizes. If you allocate 4096 as **thewall** value, you have 4

megabytes of space. The **thewall** space can be managed by adjusting allocation to mbufs or mbuf clusters. There are three parameters that allocate mbuf space to either mbufs or mbuf clusters:

lowmbuf
lowclust
mb_cl_hiwat

The **lowmbuf** tunable determines the minimum number of mbufs that are kept available for mbuf allocation. They are not available for mbuf clusters. The **lowclust** is the minimum number of mbuf clusters that are kept available for mbuf clusters. Note that the **lowmbuf** is in 256-byte increments, and the **lowclust** is in 4096-byte increments. The **mb_cl_hiwat** tunable is the maximum number of mbuf clusters that can be used as mbuf clusters.

If you are running AIX 3.2.5, you need to make sure that **mb_cl_hiwat** is big enough. If you have the same number of **lowmbuf** and **lowclust** and you try to use up all the mbuf clusters for staging 4K data blocks, you run out of mbufs. The mbufs are used for other things besides pointing to mbuf clusters, so the **lowmbuf** becomes exhausted first.

When tuning for large file transfers, keep a large number of mbuf clusters available so that when data is transferred in buffers larger than 932 bytes, an mbuf cluster is available. For pure data streaming, a good starting point is to set aside 80% of the mbuf pool to mbuf clusters. Use the **mb_cl_hiwat** tunable to set the percentage.

In addition to opening up the TCP/IP window, adjust the size and allocation of the mbuf pool to give more space to 4KB data blocks on nodes running AIX 3.2.5.

Preventing Network Adapters from Dropping Packets

If traffic is routed through other external network routers, you may not be able to use the optimal maximum transmission unit (MTU) or **tcp_mssdflt** size for that network. If this happens, adjust the **tcp_sendspace** and **tcp_recvspace** values accordingly.

To find the optimal **tcp_sendspace** and **tcp_recvspace** sizes, get the largest MTU size that the router will handle to other networks of the same type. To get the optimal **tcp_sendspace** and **tcp_recvspace** sizes for single stream transfers, use the formula:

$t = m \quad q$

where:

t Optimal **tcp_sendspace** and **tcp_recvspace** sizes

m Largest MTU size that the router will handle to other networks

q Smaller of the transmit queue and receive queue size for the adapter

This does not apply to a switch because the switch adapter does not have transmit and receive queues. See “Tuning the Switch Device Driver Buffer Pools” on page 417 for more information on tuning **rpoolsize** and **spoolsize**.

The number produced by the formula is the largest that you can use before the adapter drops packets and TCP/IP has to do a retry on the dropped packets. You

want to avoid dropping packets on the adapter to get optimal throughput performance.

When configuring the nodes, you must plan the amount of mbuf space for all interfaces to a node. If you have SP Ethernet, FDDI and a switch all on one node, the mbuf pool size must be sized for the aggregate traffic expected. An indication that the mbuf allocation (**thewall** in the **no** tunables) is too small is getting a request for memory denied count in the **netstat -m** output. Switch traffic can require a large number of mbufs. Monitoring and tuning the mbufs is detailed in the *AIX Versions 3.2 and 4.1 Performance Tuning Guide*, SC23-2365.

SP Specific Tuning

The following are rules of thumb for tuning an SP :

thewall	At least 25 % of real memory up to a limit of 64 meg
sb_max	At least twice the size of the largest of the **tcp_sendspace**, **tcp_recvspace**, **udp_sendspace** and **udp_recvspace** values described below.
subnetsarelocal	Always set to **1**
ipforwarding	Set to **1** if a gateway node
tcp_sendspace	Minimum of **65536** unless there are problems with external connections not handling greater than 32768 byte windows. Never higher than major network adapter transmit queue limit. To calculate this limit, use *Adapter queue size MTU*
tcp_recvspace	Minimum of **65536** unless there are problems with external connections not handling greater than 32768 byte windows. Never higher than major network adapter transmit queue limit. To calculate this limit, use *Adapter queue size MTU*
udp_sendspace	Always less than **udp_recvspace**, but never greater than **65536**
udp_recvspace	Always greater than **udp_sendspace** and sized to handle as many simultaneous UDP packets as can be expected. For single parent/multiple child configurations, set **udp_recvspace** to **udp_sendspace** times the maximum number of child nodes if UDP is used.
rfc1323	Always set to **1**
tcp_mssdflt	Set for major network adapter for routed IP traffic. *Adapter MTU - IP headersize - TCP headersize - / optional rfc1323 headersize* or *Adapter MTU* - 2 - 2 - optional 12 (52)

The following are additional details for setting the network tunables for the SP system.

thewall

For AIX 4.1 systems, set to at least 25% of real memory. For servers using IP, it might be necessary to increase this to 50% if you have a small memory node. The maximum value you can set this to is **65536** or 64 megabytes of memory. The maximum amount of memory you can allocate to **thewall** is 64 meg, even if this value indicates a higher amount.

For AIX 3.2.5 nodes, see the recommended **thewall** size in Table 16 on page 417.

sb_max

This value is aggregate across **tcp_sendspace**, **tcp_recvspace**, **udp_sendspace**, and **udp_recvspace** PER SOCKET. Because it is in mbuf and mbuf clusters allocated, **sb_max** needs to be larger than **tcp_sendspace** and **tcp_recvspace** combined. The guideline is to set it to twice the largest of the four values.

mb_cl_hiwat (AIX 3.2.5 only)

This tunable sets the upper limit of 4K mbuf clusters that can be allocated from the mbuf pool. This value should be set to correspond to between 20% and 80% of the **thewall** value. This allows the maximum number of mbuf clusters in the pool, plus some space for other small packets. The 80% is used because each mbuf cluster needs to be pointed to by a regular mbuf. That means that each 4K cluster is actually 4352 bytes of mbuf pool space and if set to 80% of the mbuf pool, will actually use up 85% of the pool. The last 15% of the mbuf pool is needed for mbufs for the socket connections, packet headers and protocol control blocks.

To set **mb_cl_hiwat** to allocate the maximum recommended 80% of the mbufs for mbuf clusters, use the formula:

$m = (t/4) \quad .8$

where:

m	**mb_cl_hiwat** setting
t	**thewall** value from **no -a**

For small data transfer environments, we recommend that you set aside at least 20% for mbuf clusters. Use the formula:

$m = (t/4) \quad .2$

where:

m	**mb_cl_hiwat** setting
t	**thewall** value from **no -a**

lowclust (AIX 3.2.5 only)

This tunable sets a “free list” size for mbuf clusters kept available for incoming packets. A good start is 20% of the **mb_cl_hiwat**. For large I/O systems, this is a good guide initially. This means that when the system checks the mbuf allocations to see if there is space for incoming data, it will make sure that 20% of the allocation mbuf clusters are free and available. If this number is too small, the traffic will slow down because if an mbuf cluster is not available, the system has to wait to free one up, usually resulting in dropped packets.

lowmbuf (AIX 3.2.5 only)

This tunable is the same as the one above, except for the regular mbufs, not the clusters. A good start is to have 100 more mbufs than clusters on the "free lists". If the two settings are too close in value, mbufs will run out before the cluster "free list" is exhausted, causing dropped packets even though there are free mbuf clusters.

tcp_recvspace and tcp_sendspace

These tunables establish the size of the TCP window on a per datagram socket connection. The size depends on the network you are sending over as well as the number of sockets open at any one time. The recommendations are set to avoid overrunning networks other than a switch. For more information related to these tunables, see "The Nagle Algorithm" on page 418 and "Adapter Queue Lengths" on page 420.

To properly set these tunables, you need a good understanding of the type of traffic your application will be sending. For optimal switch performance, these need to be set to **256K** or greater.

udp_sendspace

Set to **65536** because anything beyond 65536 is essentially ineffective. Since UDP transmits a packet as soon as it gets any data, and since IP has an upper limit of 65536 per packet, anything beyond 65536 runs the small risk of getting thrown away by IP.

udp_recvspace

We recommend starting at 10 times the **udp_sendspace** because UDP may not be able to pass a packet to an application before another one arrives. Also, several nodes can send to one node at the same time. To provide some staging space, this size is set to allow 10 packets to arrive before any are thrown away. For large parallel applications using UDP, the value may have to be increased.

rfc1323

If you are setting **tcp_recvspace** and **tcp_sendspace** to greater than 65536, you need to set **rfc1323** on both clients and servers. Without having **rfc1323** set, the effective values for **tcp_recvspace** and **tcp_sendspace** will be **65536** even if they are set higher.

tcp_mssdflt

This tunable is used to set the maximum packet size for communication with remote networks. This tunable is the same as the MTU for communication across a local network except for one small difference. The **tcp_mssdflt** size if for the size of the data only, in a packet. You need to reduce the **tcp_mssdflt** for the size of any headers so that you send full packets instead of full a packet and a fragment. The way to calculate this is as follows:

MTU of interface - TCP headersize - IP headersize - /
rfc1323 headersize

which is:

MTU - 2 - 2 - 12 (*MTU* - 52)

Limiting data to the MTU - 52 bytes ensures that, where possible, only full packets will be sent.

CSS MTU size

The MTU of a switch is **65520** and we recommend that you keep this setting. However, under some circumstances (for example, when you want to avoid the Nagle algorithm causing very slow traffic), it may be necessary to reduce this value. You can reduce the MTU of a switch to as low as 32678 with only a 2 - 10% loss in throughput. However, the CPU utilization will be slightly higher due to the per packet overhead.

To reduce the MTU of the switch, use:

ifconfig css mtu *new_size*

This takes effect immediately, and needs to be run as root user.

You should always use the same MTU across all nodes.

Tuning the SP Network for Specific Workloads

There are different network tunables that optimize the system for different types of network traffic. The following sections describe the way that the system utilizes these optimal tunables to benefit the types of network traffic generated.

A difficulty in tuning networks is that different mixes of traffic and the way applications utilize the network can work against each other. A set of tunables that benefits a single stream point-to-point large data transfer can cause severe problems for a Parallel Operating Environment (POE). In many cases you cannot optimize for two types of traffic. In addition, if your configuration has several types of network media (token-ring, Ethernet, FDDI and ATM), it is not always possible to optimize for all network types.

Tuning for User Environments

The following tables provide network tunable settings designed as initial values for various user environments. Table 15 on page 417 provides values for AIX 4.1 nodes and Table 16 on page 417 provides values for AIX 3.2.5 nodes. You need to change these values on each of the installed nodes. To temporarily change these values, use the **no** command. If you want these values to be preserved across booting the nodes, we recommend that you change the tuning script on each node. Details on how to change the script are found in the *IBM Parallel System Support Programs for AIX: Installation and Migration Guide*. See “tuning.cust File” on page 510 for a sample of the tuning script.

The settings in Table 15 on page 417 and Table 16 on page 417 are only initial recommendations. Start with them and realize you may need to change them.

Table 15. Initial Values for AIX 4.1

Tunable	Software Development Mixed Interactive	Commercial	Scientific/Engineering High Performance Large Data Streaming
thewall	16384	16384	16384
sb_max	163840	442368	1310720
subnetsarelocal	1	1	1
tcp_sendspace	65536	221184	327680
tcp_recvspace	65536	221184	327680
udp_sendspace	32768	59392	65536
udp_recvspace	65536	221184	655360
rfc1323	1	1	1
tcp_mssdflt	1448	1448	Varies

Table 16. Initial Values for AIX 3.2.5

Tunable	Software Development Mixed Interactive	Commercial	Scientific/Engineering High Performance Large Data Streaming
thewall	4096	6144	6144
lowclust	60	100	100
lowmbuf	50	200	200
mb_cl_hiwat	120	1220	1220
sb_max	163840	237568	1310720
subnetsarelocal	1	1	1
tcp_sendspace	65536	118784	655360
tcp_recvspace	65536	118784	655360
udp_sendspace	32768	59392	65536
udp_recvspace	65536	118784	655360
rfc1323	1	1	1
tcp_mssdflt	1448	1448	Varies

Tuning the Switch Device Driver Buffer Pools

PSSP V2R2 provides the ability to tune the buffer pool allocation in the switch device driver. Not setting these buffer sizes correctly can result in poor or unpredictable performance when using the switch interface on a node. The switch uses two buffer pools to stage data either incoming or outgoing on the interface. These buffer sizes can be set using the **chgcss** command. See *IBM Parallel System Support Programs for AIX: Command and Technical Reference* for additional information on the **chgcss** command. The command can be found in the **/usr/lpp/ssp/css directory**.

You can view the **rpoolsize** and **spoolsize** buffers used by the device driver by listing the attributes of the interface. For either switch type, enter the following command:

```
lsattr -E -l css
```

produces the following output containing default values:

```
bus_mem_addr      x 4          Bus memory address          False
int_level         xb           Bus interrupt level         False
int_priority      3            Interrupt  priority         False
dma_lv            8            DMA arbitration level       False
spoolsize         524288       Size of IP send buffer      True
rpoolsize         524288       Size of IP receive buffer   True
```

By default, the **rpoolsize** and **spoolsize** is set to 512K. However, if the aggregate amount of data staged incoming or outgoing is larger that 512K, buffer overruns will occur.

You must also ensure that the **rpoolsize** and **spoolsize** setting is equal to or larger than the larger of the **tcp_sendspace** and **tcp_recvspace** size on your nodes. If the **tcp_sendspace** is larger than the **spoolsize** and you transmit large amounts of data, you will overrun the **spoolsize** causing a failure count in the device driver statistics. Also, if the **rpoolsize** is smaller than the **tcp_recvspace** and another node transmits large amounts of data, you will overrun the **rpoolsize** buffer, again causing a failure count in the device driver statistics. This will also happen when multiple transfers are in progress and the aggregate amount of data is larger than the current **spoolsize** and **rpoolsize** settings.

If several connections are transmitting or receiving at the same time, then the **spoolsize** and **rpoolsize** may need to be for the aggregate traffic across all connections.

For 604 High Nodes, we recommend that the initial **spoolsize** be 512K times the number of processors in the nodes.

You can use the following command to change the **rpoolsize** and **spoolsize** values on each node:

```
chgcss -l css  -a spoolsize=newsize -a rpoolsize=newsize
```

New settings are not effective until the device driver is reloaded or the node is rebooted.

Tuning Problems and Issues

The section covers topics on specific problems that can be encountered on an SP system and ways to alleviate then.

The Nagle Algorithm

A problem that often occurs on the SP system is that an application runs very slowly when using a switch, while performance is significantly better on an Ehternet or FDDI network interface. This problem is sometimes caused by the Nagle Algorithm (used by TCP/IP) interacting with the delayed ACK (acknowledgment) timer.

The Nagle Algorithm states that under some circumstances, there will be a waiting period of 200 msec before data is sent. Such circumstances are:

- There are no outstanding packets on a socket to return an ACK
- The data is smaller than the segment size (MTU)
- If multiple buffers are sent equal to the TCP window and the window size is less than the MTU (when using a datagram socket, TCP will not allow sending more than the window size before receiving acknowledgment from the destination)

Under such circumstances the Nagle Algorithm causes a waiting period of up to 200 msec for other data that can also be included in a larger packet in order to reduce the number of packets on the network.

The Nagle Algorithm uses the following parameters for traffic over a switch:

- Segment size = MTU
- Window size = smaller of **tcp_sendspace** and **tcp_recvspace**
- Data size = application data buffer

The following are the specific rules used by the Nagle Algorithm in deciding when to send data:

- If a packet is equal to or larger than the segment size (or MTU), and the TCP window is not full, send an MTU-size buffer immediately.
- If the interface is idle, or the **TCP_NODELAY** flag is set, and the TCP window is not full, send the buffer immediately.
- If there is less than 1/2 of the TCP window in outstanding data, send the buffer immediately.
- If sending less than a segment size buffer, and if more than 1/2 the window is outstanding, and **TCP_NODELAY** is not set, wait up to 200 msec for more data before sending the buffer.

In addition to the Nagle Algorithm, a delayed ACK timer can cause slow data transfer over a switch. This timer is set when a single TCP/IP packet is received at the receive side of a connection. If a second packet is received, then the timer expires, and an acknowledgment is sent back to the sending side. You rarely see this on small segment size (MTU) networks because a large buffer of data results in more than one packet being transmitted. However, on large segment size networks like a switch, writing a 32K buffer results in only one packet. That same buffer on smaller segment size networks results in multiple packets, and there is no use of the delay ACK timer.

To summarize, a 200 msec delay in sending data over a switch can be caused by either the Nagle Algorithm being invoked or by the application writing buffers so that only one packet is sent over the interface and the delayed ACK timer causes a 200 msec delay on the acknowledgment.

The following recommendations will help to avoid the Nagle algorithm (and not affect other parts of any system):

1. Control the segment, window, and data sizes at the application socket.

2. From within an application, you can increase the TCP window size on the socket by setting a different size using the **SO_SNDBUF** and **SO_RCVBUF** settings on the socket. For good performance across a switch, we recommend that both **SO_SNDBUF** and **SO_RCVBUF** be set to at least **327680** on both the client and server nodes. You need to set both sides since TCP uses the lowest common size to determine the actual TCP window.

   If you set the **SO_SNDBUF** and **SO_RCVBUF** sizes above 65536, you need to set **TCP_RFC1323** also on the socket unless the **no** options already have it set. Setting **TCP_RFC1323** to **1** will take advantage of window sizes greater than 65536.

3. Set **TCP_NODELAY** on the socket of the sending side to turn off the Nagle algorithm.

   All data sent will go immediately, no matter what the data size. However, if the application sends very small buffers, you will significantly increase the total number of packets on the network.

4. If you are running an application and don't have access to the source code, use the **no** command to increase the TCP window.

   Increasing the TCP window may not always be effective because increasing the **tcp_sendspace** and **tcp_recvspace** size on the sending and receiving nodes may cause other negative affects on the SP system or to other applications running on the system.

5. Change the MTU size of the switch.

   Changing the MTU of the switch will move the window size and buffer size combination where the 200 msec delay is invoked. When writing 32K buffers to a TCP connection, if the TCP/IP window is 65536, only 5 packets per second are transmitted. If you change the MTU of the switch interface to 32768 there is no delay on transmitting a 32768 buffer because it is the same size at the segment size of the switch. However, reducing the MTU of the switch to 32768 degrades the peak TCP/IP throughput between 2 and 10 percent. Reducing the MTU even further degrades the peak throughput even more.

Adapter Queue Lengths

The high throughput of a switch can cause problems with other network adapters connected to gateway nodes on the SP system. There is a fixed amount of memory on each network adapter for staging network traffic. The transmit adapter queue length specifies the maximum number of packets for each adapter (except a switch).

The transmit adapter queue length is tunable through SMIT, and the typical default size is too small to handle SP system traffic. The typical value for this queue is 30 to 64, of which each entry can handle a maximum of the MTU of that adapter. For an Ethernet adapter, a queue length of 30 Ethernet packet works out to be a maximum total data size of 45000 bytes. This is less than a single packet from a switch.

If transmit adapter length queue size is exceeded, subsequent packets are thrown away by the adapter device driver and result in dropped packets. We recommend that the transmit adapter length queue size for all network adapters on all nodes of the SP system, as well as the control workstation, be set to their maximum.

A problem with adapter queue sizes is indicated by TCP/IP retries on the node in the **netstat -s** output or Ierr and Oerr in the **netstat -i** output for the adapter. This indicates that the adapter queue size is too small, or the amount of TCP/IP traffic too much for the adapter. Because **netstat -s** is for TCP/IP, and not for a specific adapter, you will have to look at the amount of errors in **netstat -i** for each adapter to determine which adapter is the one causing the problem. On AIX 4.1.4 and lower systems, view adapter errors using the **entstat**, **tokstat**, and **fddistat** commands.

Another way to tune IP for the adapter queue is to reduce the TCP/IP window and **udp_sendspace** so that it is less than the transmit queue size times the segment size on the network. This usually results in lower throughput and higher system overhead for network traffic.

For Ethernet, Token Ring and FDDI, the adapter statistics, including a count of dropped packets, can be seen by using the **entstat**, **tokstat**, and **fddistat** commands, respectively. These commands are AIX 4.1 commands and are not found on AIX 3.2.5 systems.

Tuning For Single Parent–Multiple Child Applications

Some applications, including the Parallel Operating Environment (POE), consist of a parent or master node with multiple child or remote nodes. Such applications have a potential for the child nodes all sending large numbers of messages to one parent node. Figure 85 on page 422 is a simplified view of a multiple-child application.

The following information on this configuration uses the Parallel Operating Environment (POE) as an example. The problem and solution are applicable to any parallel application that uses datagram sockets.

In a POE job that sends a large number of messages to standard out and standard error, the amount of traffic generated can use up a lot of mbufs on the system. The following tunables are recommended as initial values for systems that run large POE sessions or other applications that use similar resources:

lowclust = 60 (AIX 3.2.5 nodes only)

lowmbuf = 200 (AIX 3.2.5 nodes only)

thewall = 8192 (AIX 3.2.5) or 16834 (AIX 4.1)

mb_cl_hiwat = 524 (AIX 3.2.5 nodes only)

tcp_sendspace = 65536

tcp_recvspace = 65536

rfc1323 = 1

These are only initial tunable values. For large jobs, depending on the number of messages from the remote nodes, the value for **thewall** may need to be increased. If the master node gets a count of requests for memory denied in the **netstat -m** output, you will need to increase the **thewall** value. On AIX 3.2.5 nodes, the value for **mb_cl_hiwat** will have to be increased in proportion to the increase of **thewall**.

These initial values will have a negative affect on the performance of large data transfers across a switch.

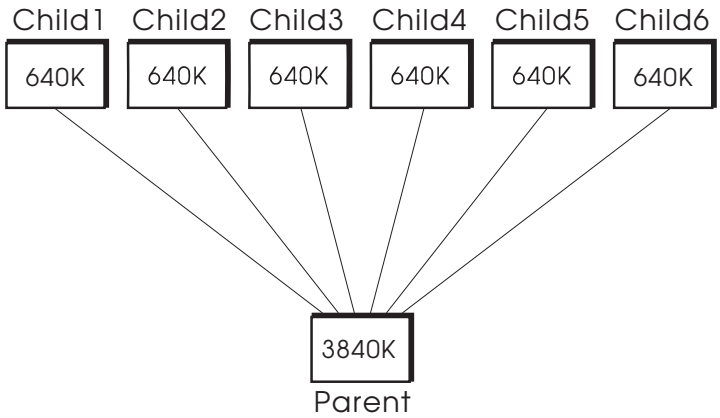


Figure 85. Simplified View of a Single Parent–Multiple Child Application

On any task assigned to the nodes assigned in a POE session, anything sent to standard out or standard error is routed back to the master node and then on to the user's session. Each of these messages requires its own mbuf. If you have a large parallel job, one using 32 or more nodes, the master node mbuf pool can be flooded by the traffic from the remote nodes. Because messages from standard out and standard error are synchronous, or blocking transmits, large numbers of messages will end up slowing down the execution of the program.

If messages are placed in the code that record a high number of events, the SP Ethernet (used to pass messages from the remote nodes to the master node, as well as for other uses) can become saturated. And, if you are connected to the master node either explicitly or implicitly using the SP Ethernet, you double the amount of traffic you are sending through the SP Ethernet. This causes backup of the network traffic and a rapid increase of mbufs needed on the master node.

One way to reduce the load on the Ethernet, as well as the mbuf requirements on the master node, is to reduce the size of the **tcp_recvspace** tunable on the master node. As explained earlier, the **tcp_recvspace** size determines the amount of data that a node can have outstanding on a socket. Shortening the TCP window reduces the amount of traffic on the SP Ethernet. The traffic that would otherwise flood the SP Ethernet is held on the remote node until it can be transmitted. This prevents flooding the SP Ethernet and reduces the amount of mbufs needed on the master node. However, it causes traffic to back up on the remote nodes, and can cause the application to slow down.

Another side effect of using a smaller **tcp_recvspace** is that other users of TCP/IP, particularly those used for large data transfers, will see a decrease in data transfer throughput.

At this point, you need to decide whether to affect the performance of large data transfers from potential POE remote nodes or try to increase the mbufs on the master node. Increasing the amount of mbufs on the master node runs the risk that you will flood the SP Ethernet and cause other problems on the system.

A better solution is to set the socket **SO_RECV** size from within an application, so other traffic on the node is not affected.

Consolidated Systems

Consolidated systems present a larger tuning challenge than homogeneous systems. Consolidated systems are those in which nodes assume several different roles. In such a case, make a full examination of what is running on each node, the traffic through the node, and any interaction between applications. Defining an appropriate set of tunables for a consolidated system requires accounting for everything running on the SP system.

External Servers

External servers can restrict performance on the SP system. For example, some external servers may not support **rfc1323**, and, therefore, can use a maximum TCP/IP window of only 65536 to transfer data to or from an SP system.

In some cases external servers have to communicate with additional external servers. When such external servers establish a connection, the negotiated tunables may be set to small values. For example, the TCP/IP window size value is set to the least common denominator. In the case of an external server having a **tcp_recvspace** of 16384, that is the TCP/IP window size used. Such a small TCP/IP window size provides slow throughput if the sending node is on the far side of a switch from a gateway node.

Subsystem Tuning

This section provides information for specific subsystems in an SP environment.

Tuning for Large Data Transfers

Several typical SP applications and utilities transfer large blocks of data across a switch using TCP/IP. These applications and utilities include:

ftp
rcp

To achieve peak data transfer across a switch using TCP/IP, you need to increase the tunables that affect buffer sizes, queue sizes, and the TCP/IP window. The tunables to adjust are:

lowclust (AIX 3.2.5 nodes only)
lowmbuf (AIX 3.2.5 nodes only)
thewall
mb_cl_hiwat (AIX 3.2.5 nodes only)
tcp_sendspace
tcp_recvspace
rfc1323

To get the best TCP/IP transfer rate, you need to size the TCP/IP window large enough to keep data streaming across a switch without stopping the IP stream. The switch has a Maximum Transmission Unit (MTU) of 65520 bytes. This is the largest buffer of data that it can send. When using TCP/IP, TCP will send as many buffers as it can until the total data sent without acknowledgment from the receiver reaches the **tcp_sendspace** value. Experimentation has found that having 4 to 10 buffers as the size of the window allows TCP/IP to achieve peak transfer rate. (The smaller number applies to AIX 4.1 and the larger to AIX 3.2.5.) Also, the faster nodes require a greater number of buffers. If you set **tcp_sendspace** and

tcp_recvspace to 655360, it can hurt the performance of the other network adapters connected to the node.

Gateway or Router Nodes

Gateway or router nodes direct traffic between external networks and the SP system. Because different types of networks can be connected to router nodes, plan carefully. For slow speed networks such as Ethernet and Token Ring, several adapters can be configured. For higher performance networks, however, we recommend a single adapter.

If a lot of traffic is routed through a router node, any parallel job using that node will be affected. We recommend that router or gateway nodes not be assigned to parallel jobs unless there are no other nodes available, or the amount of expected network traffic is small.

Network File System (NFS)

Three aspects of NFS that are particularly challenging to tune for large SP system configurations are:

- The number of concurrent NFS requests to an NFS server
- The number of NFS daemons (**nfsd** and **biod**) running
- The NFS socket size

The first problem related to NFS that we generally see in large SP system configurations occurs when a single node is acting as the NFS server for a large number of nodes. In this scenario, the aggregate number of NFS requests can overwhelm the NFS socket or **nfsd** daemons on the server, or enough daemons cannot be configured.

When NFS is configured on a node, by default AIX configures 8 **nfsd**s and 6 **biod**s. The **nfsd**s handle NFS requests between a client node and the server. The **biod**s handle block I/O, primarily in NFS writes from the client node.

For the server configuration, **nfsd**s are the primary concern. If you have a 64 node configuration and configured one NFS server for the other 63 nodes, and if all 63 client nodes made an NFS request at the same time, you will need at least 63 **nfsd**s on the server. If you had only 8 **nfsd**s, as in the default configuration, then 55 NFS requests would have to wait and they could time out. Obviously, average NFS response time will be very slow.

However, there is a limit on how many **nfsd**s you can configure before the amount of processing on behalf of the NFS traffic overwhelms the server. Generally, when you configure more than 64 **nfsd**s you start to see NFS performance degradation, depending on the characteristics of the NFS traffic. The size of the requests, the mix of NFS operations, and the number of processes on each node that generates NFS requests influence the amount of NFS performance degradation. Examine **nfsstat** command output to check indications that you are generating too much NFS traffic to a server. If you see RPC timeouts or RPC retransmits when checking on the client nodes, then you probably are overwhelming the current number of **nfsd**s or **nfs_socket** size on the server.

Determining the number of **biod**s to configure on the client node is a bit complicated. The **biod**s are used when writing to an NFS server from a client

node. Typically you'd configure enough **biod**s on a client node for the NFS write traffic that you expect. However, NFS has a limit of a maximum of 6 **biod**s on a client node per NFS mounted file system. If you have only one process on a client node writing to a single file system on the server, there is no reason to increase the number of **biod**s. And, if you have only one remote mounted NFS file system on each client, you can use only 6 **biod**s no matter how many processes are writing to the file system.

The limitation to 6 **biod**s on a client node per NFS mounted file system is imposed by NFS Version 2.0 (which is packaged with AIX 4.1). In NFS Version 2.0, all NFS writes use a fully synchronous 4K write model. When you write to an NFS mounted file system, you can send only 4K at a time from the client to the server. If you have only one I/O outstanding at a time, this is very slow. In a fully synchronous model the data must be written on the disk at the server before the client can send the next 4K block. To speed this up a bit, NFS allows staging 6 requests at the client per file system, but no more. Given the time it takes to write a block on disk in AIX, one block at a time is very slow.

If there is more than one NFS mounted file system on your client nodes, you can configure up to 6 **biod**s per file system. If the number of **biod**s gets too high, performance will suffer in the same way as when too many **nfsd**s are configured on the server. Again, a general rule is more than 64 **biod**s will start to degrade performance.

In general the same rules apply for **nfsd**s on the client node as on the server node. However, we don't recommend that you use the same number of **nfsd**s on the client nodes as the servers when a large number of nodes all use the same server (16 clients to 1 server for example). If all nodes have 16 **nfsd**s, each **nfsd** on the server can potentially be sent 16 NFS requests. In this situation, monitor **nfsstat** on both the client and server nodes to look for RPC timeouts and retransmits.

If you already have a lot of **nfsd**s configured on the server, the best solution is to split the NFS server duties up across several nodes, and keep the potential number of concurrent NFS requests below 64. As a general guideline, configuring 64 **nfsd**s on a server usually works well on the SP system.

For more detailed information regarding tuning NFS, refer to the *AIX 3.2.5 and 4.1 Performance Monitoring and Tuning Guide*, SC23-2365, and the *AIX System Management Guide: Communications and Networks*, GC23-2487.

ADSTAR Distributed Storage Manager (ADSM)

When running ADSM on the SP system, the performance of a backup or restore is affected by several tunable settings. However, depending on the backup destination, or restore source, the tunables and their settings are different. The following section lists the settings to initially try for running the ADSM server on MVS over an ESCON channel and running the ADSM server on an SP system node. Because ADSM is not always the only application running on a client node or server, some of these values may not be reasonable in your environment.

There are up to four places where various tunables can be set to optimize the performance of ADSM. The following sections list these places, and the initial values to use for ADSM over a switch and ADSM over ESCON to MVS.

In the following tables, use the switch settings for ADSM clients on another node. Use the ESCON to MVS settings for going to MVS on the client nodes.

SP Client Node

The following are the **no** tunable values that will achieve the best ADSM performance on the client nodes.

Parameter	Switch	ESCON to MVS
sb_max	1310720	131072
thewall	6000	6000
lowmbuf (3.2.5 only)	500	500
lowclust (3.2.5 only)	200	200
mb_cl_hiwat (3.2.5 only)	1200	1200
rfc1323	1	N/A
tcp_mssdflt	N/A	4056

If you already have a larger value for **thewall**, then you need to keep the larger **thewall** value, but keep the ratio between **mb_cl_hiwat** and **thewall** the same as in the above settings.

The following are the entries to use in the **/usr/lpp/adsm/bin/dsm.sys** file on the ADSM client nodes.

Parameter	Switch	ESCON to MVS
TCPWindowsize	256	64
TCPBuffsize	32	32
txnbyte	25600	25600
tcpnodelay	Y	Y

SP ADSM Server Node

The following are the **no** tunable values that will achieve the best ADSM performance on the SP ADSM server node.

Parameter	Switch	ESCON to MVS
sb_max	1310720	N/A
thewall	6000	N/A
lowmbuf (3.2.5 only)	500	N/A
lowclust (3.2.5 only)	200	N/A
mb_cl_hiwat (3.2.5 only)	1200	N/A
rfc1323	1	N/A
tcp_mssdflt	N/A	N/A

If you already have a larger value for **thewall**, you need to keep the larger **thewall** value, but keep the ratio between **mb_cl_hiwat** and **thewall** the same as in the above settings.

The following are the entries to use in the **/usr/lpp/adsm/bin/dsmserv.opt** file on the SP system ADSM server node.

Parameter	Switch	ESCON to MVS
TCPWindowsize	256	N/A
TCPBuffsize	32	N/A
txnbyte	25600	N/A
tcpnodelay	Y	N/A

ESCON Gateway Node

The following are the **no** tunable values that will achieve the best ADSM performance on the node acting as the gateway across ESCON.

Parameter	Switch	ESCON to MVS
sb_max	N/A	131072
thewall	N/A	6000
lowmbuf (3.2.5 only)	N/A	500
lowclust (3.2.5 only)	N/A	200
mb_cl_hiwat (3.2.5 only)	N/A	1200
rfc1323	N/A	N/A

If you already have a larger value for **thewall**, you need to keep the larger **thewall** value, but keep the ratio between **mb_cl_hiwat** and **thewall** the same as in the above settings.

You will also need to set the MTU size of the ESCON interface to 4096. To do that you can use:

```
ifconfig es  -mtu 4 96
```

MVS ADSM Server

The following tunable settings on MVS will get you peak throughput either across a switch, or across a single ESCON channel to MVS. Your actual throughput will be determined by the speed of the media being backed up to, or restored from, the SP system server or MVS server.

Table 17. TCP/IP(MVS) parameters/settings

DATABUFFERPOOLSIZE	240	32768
LARGEENVELOPEPOOLSIZE	240	8192

ESCON

If you are using CLIO to pass data from MVS to AIX, then TCP/IP over the switch to the thin node, you have to be careful on buffer sizes used. CLIO uses a 60K buffer to pass data in either direction, either to or from MVS. If the application running on the wide node is a catch and forward application, taking incoming traffic from one side, and passing it to the other side, then from the settings below, the TCP/IP packet size used across the switch is 64K. This can result in fragmentation, or re-assembly problems in the application on the wide node.

To achieve peak TCP/IP throughput over a switch and through ESCON to MVS, set the **tcp_mssdflt** size on the remote nodes to **4056** and the **MTU** of the ESCON interface to **4096**.

PCA Card and CLIO/S Tuning on the SP System

There are two sets of tunables to be concerned with when tuning the SP system for running with a PCA attached to MVS: the cat0 interface and the network mbufs.

Note: The PCA card is currently supported only on AIX 3.2.5. This section is included for your use if you are running AIX 3.2.5 in a system partition.

In setting up the cat0 interface, you need to define the number of buffers that are used for transmitting and receiving data from the channel. To display these parameters, enter the following command on the SP node:

```
lsattr -E -l cat
```

This output looks like this:

```
dma_bus_mem     x  F4    Bus memory address                    False
bus_intr_lvl 4           Bus interrupt level                   False
dma_lvl          x6        DMA arbitration level               False
xmitsz          4 96       Transmit  buffer  SIZE
xmitno           26          NUMBER of transmit buffers
recvsz          4 96       Receive  buffer  SIZE
recvno           26          NUMBER of receive buffers
rdto                       Receive data transfer OFFSET
speed            4           Channel  SPEED
first_sc         x9        STARTING subchannel address
num_sc            8          NUMBER of subchannel addresses
clawset           x9       CLAW mode subchannel set
switch          online     Online/Offline  SWITCH
indicator       online     Online/Offline  INDICATOR
ucode_type      1            N/A
host_name       KN1MVS       N/A
adapter_name P24N11A1      N/A
```

The items of interest are the **xmitno** and **recvno**. These parameters set up how much space is allocated in the PCA card buffer space for transmits and receives. In the example above we have set up the default of the same amount for both transmits and receives. If your system shows a large difference between transmits and receives rates, increase the number for the direction that is slow. You must not use more than 52 total buffers, assuming that **xmitsz** and **recvsz** are set to 4096. There is a total of 208K on the card; exceeding this total will render the interface unusable.

You also must not use fewer than 15 buffers for either transmits or receives if you are using CLIO/S across the interface. Because CLIO/S uses a 60K buffer size, you need a 60K minimum to store the data on the card. If you use fewer than 15 buffers, you will see transmission drop outs, stalls, or even communication shutdown on the node.

To change the number of buffers you can use the **chdev** command on the node. The command is:

```
chdev -P -l cat  -a xmitno=37 -a recvno=15
```

You will need to reboot the node after issuing this command.

You can use whatever number of buffers you want as long as you do not exceed 52 4K buffers, or 208K total.

The **thewall** and **lowmbuf** tunables are for total mbufs in the system. Because CLIO/S over the PCA uses at least 4K buffers, it uses the mbuf clusters not individual mbufs. These tunables are the **lowclust** and **mb_cl_hiwat** tunables.

For total mbufs, you should make sure that ANY communication interface has enough mbufs available. The total mbufs are for all Ethernet, token-ring, PCA and switch interfaces. If all these interfaces need more mbufs than are set using **thewall**, it will need to be increased.

To change the **thewall** value, enter:

```
no -o thewall=12228
```

If you suspect that you are running out of mbufs, look at the output of the **netstat -m** statistics. This output looks like:

```
527 mbufs in use:
        252 mbufs allocated to data
        11 mbufs allocated to packet headers
        1  mbufs allocated to socket structures
        137 mbufs allocated to protocol control blocks
        15 mbufs allocated to routing table entries
        8 mbufs allocated to socket names and addresses
        4 mbufs allocated to interface addresses
252/4 1 mapped pages in use
1735 Kbytes allocated to network (65% in use)
126 requests for memory denied
 requests for memory delayed
 calls to protocol drain routines
```

This listing shows entries in mbufs allocated to data and requests for memory denied. These two counters are the ones to watch. If you are getting an increasing count under requests for memory denied, increase the value for **thewall**. The mbufs allocated for data shows the current number of mbufs containing data. If this count gets high, examine the mbufs denied value. The mbufs denied counts the number of times an mbuf request was denied because one was not available since the system last booted. The mbufs allocated to data is the count at the time the data was collected.

You will normally see approximately 45 mbufs allocated to data when running only CLIO/S over the PCA. This number will be affected by traffic to other interfaces, as well as stalls on any interface. A stall can show up as an increase in mbufs allocated for data, and may stay at a high value.

If you run out of mbufs and generate a lot of mbuf denied counts, the interfaces can go down.

You might want to try the following:

1. Add buffers using the **thewall** setting default of 2048. If you are running other interfaces, such as a switch, you probably have to increase the value more than if you are not running a switch.
2. Increase the **lowclust** value because CLIO/S uses mainly clusters of mbufs.

3. Increase the **mb_cl_hiwat** to maintain a large enough number of clusters for CLIO/S traffic.

These are only initial recommendations. Depending on the type of PCA interface traffic, and other network impacts, these will need to be changed.

Additional information on general mbuf tuning can be found in *AIX Versions 3.2 and 4.1 Performance Tuning Guide*, SC23-2365.

Distributed Tape Management System Enablement

© Copyright IBM Corp. 1996

Chapter 28. Distributed Tape Management System Enablement

Note

This information on the distributed tape management system enablement facility is provided for backward compatibility purposes only.

We recommend the use of IBM Network Tape Access and Control System for AIX (NetTAPE). NetTAPE is a program that provides consolidated tape operations and transparent access to a wide variety of remote tape devices on an SP system.

Note: The description of the tape management enhancement logic in the physical tape device driver is included here for completeness relative to interaction with the pseudo tape device driver and support requirements by a tape management system application. This level of function is supplied with the device driver support obtained for 3480 and 3490 tape devices and is not provided with this product. Refer to the documentation supplied with those tape products for the latest interface/functional definition and availability.

This chapter provides information about the distributed tape management system (DTMS) enablement facility and describes the tasks involved in installing, defining, and configuring the enablement facilities.

Overview

The distributed tape management system (DTMS) enablement provides facilities that allow for the support of a fully distributed tape management system. A management application that utilizes the device driver enhancements and the pseudo tape device driver functionality could provide the support that would allow users operating in any node on SP systems to have transparent access to mainframe tape media or resources attached through (optionally) other SP nodes. Once access to tape media or resources is established, through use of either a command or API interface by a management application, the reading and writing (and control) of tape can be supported through the use of the standard AIX system call interface (**open**, **close**, **read**, **write**, and **ioctl**).

Through support of the enablement logic, a management application could manage both device and media resources and provide secure access to those resources. A management application, interacting with the enablement logic could provide transparent support for ANSI and IBM Standard Label and No Label tape formats.

The DTMS enablement facilities are comprised of the following components:

Magnetic Tape Extension (MTE) pseudo device driver (referred to as the pseudo tape device driver in this chapter)

- Tape management pseudo control point (TMPCP)
- Site-defined exit facility

Physical tape device driver with the Tape Management Enhancement (TMSDD)

© Copyright IBM Corp. 1996

- Management Protocol Interface (MPI)
- User Protocol Interface (UPI)
- Tape management control point (TMCP)

Magnetic Tape Extension (MTE) Pseudo Device Driver

The Magnetic Tape Extension (MTE) pseudo device driver provides enabling facilities that can be used to support a distributed, multinode tape management system. Support is provided to create and control local and local/remote pair logical device images. The support structure is also provided for the invocation of site-defined exit processing for AIX system calls to logical (tape) devices. Refer to the following section for a description of the tape management pseudo control point (TMPCP). Refer to the **mte_exits** file in *IBM Parallel System Support Programs for AIX: Command and Technical Reference* for an overview and interface definition of the site exit facility provided by the pseudo tape device driver.

Tape Management Pseudo Control Point (TMPCP)

A tape management pseudo control point (TMPCP) is a logical extraction of the pseudo tape device driver that is used to create and control User Protocol Interface (UPI) logical device images. Each (configured) TMPCP is capable of supporting 255 logical device images. One device-special file is created for each configured TMPCP for the pseudo tape device driver.

A TMPCP only accepts an **open()** that contains the MPI bit set in the device minor number (MPI **open()** request). Any **open()** issued to the TMPCP without the MPI bit set in the device minor number is rejected with an **errno** value of EINVAL. Each TMPCP only allows one **open()** process at a time and rejects additional **open()** requests with an **errno** value of EBUSY.

A TMPCP does not accept **read()** or **write()** system calls and rejects these requests with an **errno** value of EINVAL. Each TMPCP supports the following **ioctl()** system call requests:

MTTMSCSKE Creates a system security key.

MTTMSCUPIE Creates a User Protocol Interface (logical device image).

MTTMSIOCINFO Gets general device information.

MTTMSMUPIE Modifies a User Protocol Interface (logical device image).

MTTMSRUPIE Removes a User Protocol Interface (logical device image).

MTTMSSRSIE Sets and reads state information.

MTTMSEUPIE Inserts or extracts user data or **ioctl()** parameters (logical device image).

All other **ioctl()** requests are rejected with **errno** set to EINVAL.

The MTTMSCUPIE **ioctl()** system call request is used to create a UPI logical device image. The major/minor device number for the tape device that this logical device image is created for, is passed to the TMPCP in the *devmmn* field of the MTTMSCUPIE **ioctl()** structure. The MTTMSCUPIE **ioctl()** system call returns a set of UPI major/minor numbers to be used by a tape manager to create the device special files for the user of this UPI logical device image. The UPI major/minor numbers are the mechanism that allows the pseudo tape device driver to be called

for all references by the user to a UPI tape device. Refer to “Tape Management Control Point (TMCP)” on page 446 or to the **mt_tmcp** man page in *IBM Parallel System Support Programs for AIX: Command and Technical Reference* for information on creating UPI logical device images in the physical tape device driver. Logical device image types of MAGTAPE, CP_LOCAL, CP_REMOTE, and STUB are supported by the MTE pseudo tape device driver. Refer to the **mte_cupi** ioctl in *IBM Parallel System Support Programs for AIX: Command and Technical Reference* for information on the **ioctl()** used to create UPI logical device images in the pseudo tape device driver.

The MTTMSMUPIE **ioctl()** system call request is used to change or modify an existing UPI logical device image. Refer to the **mte_mupi** ioctl in *IBM Parallel System Support Programs for AIX: Command and Technical Reference* for information on modifying or changing UPI logical device images in the pseudo tape device driver.

The MTTMSRUPIE **ioctl()** system call request is used to remove a previously created logical device image from the TMPCP. Refer to the **mte_rupi** ioctl in *IBM Parallel System Support Programs for AIX: Command and Technical Reference* for information on removing UPI logical device images in the pseudo tape device driver.

The MTTMSCSKE **ioctl()** system call request is used to establish a security key in the pseudo tape device driver for tape management **ioctl()** access. Refer to the **mte_cskey** command for information on the **ioctl()** used to establish a security key in the pseudo tape device driver.

The MTTMSEUPIE **ioctl()** system call request is used for recovery operations to insert or extract user data or **ioctl()** parameters to or from a logical device image. Refer to the **mte_eupie** command for information on the **ioctl()** used for recovery operations.

A state of “MPI forced close” results when a **close()** system call is executed for the TMPCP and one or more logical device images created by a MTTMSCUPIE **ioctl()** still exist (not removed by a MTTMSRUPIE **ioctl()**). The control block structure for the TMPCP, as well as the control block structures established for each existing logical device image, remain intact until removed by a subsequent UNCONFIGURE operation that finds the TMPCP in the MPI forced-close condition, in which case all associated control block structures are removed. A subsequent **open()** system call for the TMPCP while in an MPI forced-close condition (not unconfigured) is accepted and the MPI forced-close condition is reset. All control block structures remain intact and subsequent logical device image requests continue unchanged.

Each UPI logical device image accepts **open()**, **close()**, **read()**, **write()**, and **ioctl()** system calls. For each system call received, the appropriate site-defined exit routine (optional) is called, and the request is passed to the physical tape device driver on the local or remote node. Refer to “Tape Management Control Point (TMCP)” on page 446 or to the **mt_tmcp** man page in *IBM Parallel System Support Programs for AIX: Command and Technical Reference* for information about the tape management extensions in the physical tape device driver.

Site-Defined Exit Facility

The pseudo tape device driver supports exits that can be redefined by a site for user access to tapes under the control of the DTMS. This facility is implemented through function calls to "stub" exit routines. These routines (whose interfaces are described by the **mte_exits** file in *IBM Parallel System Support Programs for AIX: Command and Technical Reference*) can optionally be replaced with site-defined routines. This exit capability allows individual, site-specific processing to be executed prior to and immediately after pseudo tape device driver execution of a system call to a UPI logical device image.

At the time of an **open()**, **close()**, **read()**, **write()**, or **ioctl()** request, a system-call dependent pre-call exit routine is invoked for the specified UPI logical device image. The pseudo tape device driver, upon return from the "pre-call" routine, invokes the physical tape device driver at the respective system call entry point. Upon return from the physical tape device driver, the system-call dependent "post-call" exit routine is executed before control is returned to the user.

Exit support is invoked for logical device image types of MAGTAPE and CP_REMOTE. Refer to the **mte_tmpcp** man page and the **mte_cupi** ioctl in *IBM Parallel System Support Programs for AIX: Command and Technical Reference* for information on creating UPI logical device images in the pseudo tape device driver.

Installing the MTE Pseudo Device Driver

Notes:

1. Before installing the MTE pseudo device driver, refer to *IBM Parallel System Support Programs for AIX: Installation and Migration Guide*.
2. Do not remove a previous version of the MTE **installp** image without performing the procedures in "Deconfiguring a Tape Management Pseudo Control Point (TMPCP)" on page 442.

The Magnetic Tape Extension (MTE) **installp** image can be installed in the following environments:

IBM RS/6000 single system environment
IBM RS/6000 wired cluster environment
IBM RS/6000 Scalable POWERparallel Systems environment

The MTE pseudo device driver needs to be installed on every node that requires remote tape access. It also needs to be installed on every node that has a physical tape device attached to it that will be used by DTMS.

The installation procedure can vary depending on whether the environment is single system or multinode. When multinode, the installation procedure further depends on whether a globally-mounted file system (/**usr**) exists, or the system supports the distributed shell (**dsh**).[1] Perform one of the following procedures to install the MTE **installp** image (in the SP environment, commands should be entered from the control node unless otherwise noted).

If the system has a globally-mounted /**usr** file system, the installation procedure is identical to the single system environment:

[1] The **dsh** command is part of the SP system.

– To install from a 3 1/2" disk, enter:

  ```
  # installp -a -d /dev/rfd  -X ssp.mte
  ```

– To install from tape, enter:

  ```
  # installp -a -d /dev/rmt  -X ssp.mte
  ```

– To install from a file system image *ssp.mte*, enter:

  ```
  # installp -a -d /full_pathname/ssp.mte -X ssp.mte
  ```

If the system does not have a globally-mounted /**usr** file system but supports the **dsh** command, perform the following steps:

1. Copy the MTE **installp** image file to a Network File System (NFS) mounted file system.

   In the following examples, assume that a file named *mte.image* is being copied from media to the /**work** directory which can be reached from all the nodes that need to install the MTE **installp** image. Use one of the following commands to copy the MTE **installp** image file to the /**work** directory:

   To copy from a 3 1/2" disk, enter:

   ```
   # dd if=/dev/rfd  of=/work/ssp.mte
   ```

   To copy from tape, enter:

   ```
   # dd if=/dev/rmt  of=/work/ssp.mte
   ```

2. Enter the following commands:

   For the **sh** or **ksh** shell, enter:

   ```
   # WCOLL=./wchosts                    / sh and ksh shell  /
     export  WCOLL
   ```

   For the **csh** shell, enter:

   ```
   # setenv WCOLL ./wchosts          / csh shell             /
   ```

   The **./wchosts** file contains a list of host names that will install the MTE **installp** image. The format of the **./wchosts** file is one host per line. An example of a **./wchosts** file follows:

   ```
   r15n 1
   r15n 2
   r15n 3
   r15n 4
   ```

3. Enter the following **dsh** command:

   ```
   # dsh /etc/installp -a -d /work/ssp.mte -X ssp.mte | dshbak
   ```

   Using the previous **./wchosts** file, this command applies the MTE **installp** image on nodes r15n01, r15n02, r15n03, and r15n04. Refer to the **dsh** command in *IBM Parallel System Support Programs for AIX: Command and Technical Reference* for additional information.

If the multinode system does not have a globally-mounted /**usr** file system and does not support the **dsh** command, perform the following steps:

1. Copy the MTE **installp** image file *mte.image* as appropriate to either a globally-mounted file system or any directory within each node.
2. Enter the following command from each node that needs to install the MTE **installp** image:

```
# installp -a -d /full_pathname/ssp.mte -X ssp.mte
```

Defining and Configuring a Tape Management Pseudo Control Point

After installing the MTE **installp** image, use one of the following procedures to define and configure a tape management pseudo control point (TMPCP). This procedure will load the MTE pseudo device driver. Repeat this procedure to define and configure additional TMPCPs.

If the system supports the **dsh** command, enter the following commands from the control node.

1. For **sh** or **ksh**, enter:

   ```
   # WCOLL=./wchosts                    / sh and ksh shell /
     export WCOLL
   ```

   For **csh** shell, enter:

   ```
   # setenv WCOLL ./wchosts          / csh shell          /
   ```

   The **./wchosts** file contains a list of host names that will define and configure a TMPCP. The format of the **./wchosts** file is one host per line. An example of a **./wchosts** file follows:

   ```
   r15n 1
   r15n 2
   r15n 3
   r15n 4
   ```

2. Enter the following **dsh** command:

   ```
   # dsh /etc/mkdev -ctape -snode -ttmpcp | dshbak
   ```

   Using the previous /**.wchosts** file, this command defines and configures a TMPCP on nodes r15n01, r15n02, r15n03, and r15n04. Refer to the **dsh** command in *IBM Parallel System Support Programs for AIX: Command and Technical Reference* for additional information.

If the system does not support the **dsh** command, enter the following command from each node that needs to define and configure a TMPCP:

```
# /etc/mkdev -ctape -snode -ttmpcp
```

or run SMIT from each node as described in the following section.

Defining and Configuring a Tape Management Pseudo Control Point Using SMIT

Perform the following procedure to define and configure a tape management pseudo control point (TMPCP):

1. Enter the following command:

   ```
   smit
   ```

   The SMIT main menu is displayed (Figure 86).

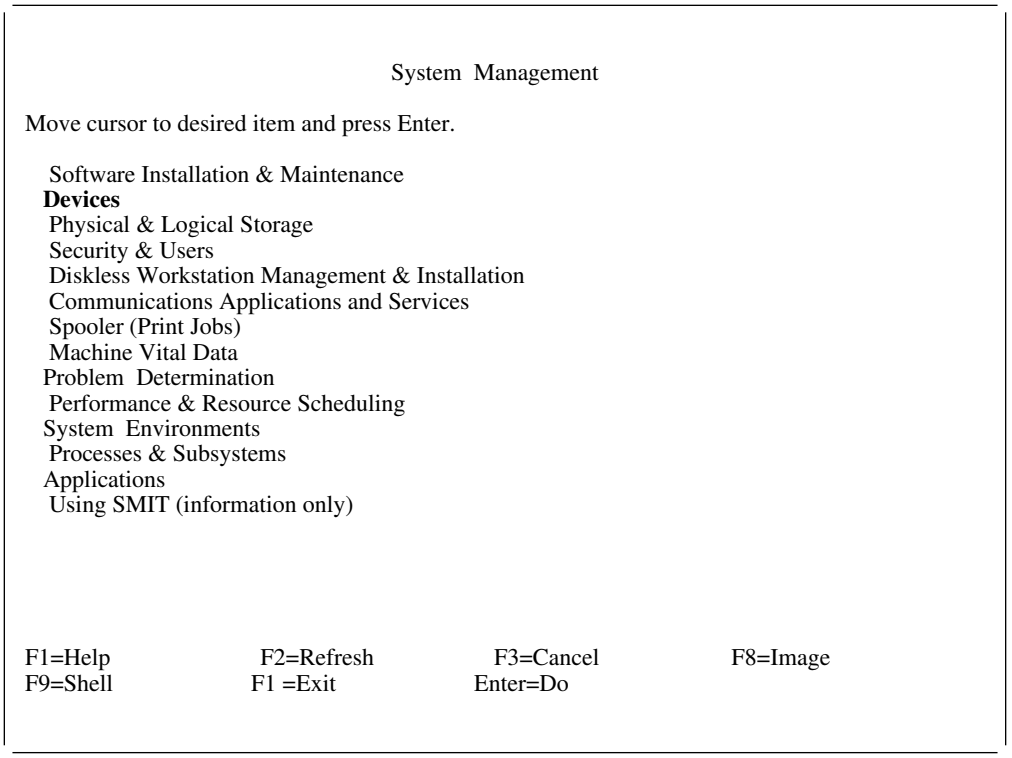


Figure 86. SMIT Main Menu

2. Select **Devices** from the SMIT main menu.

   Use the up (↑) and down (↓) arrow keys to move the cursor to the desired item on the menu.

   Press **Enter** or click on **Do** if you are processing **smit** in a windowed environment.

3. Select **Tape Drive** from the Devices menu (Figure 87).

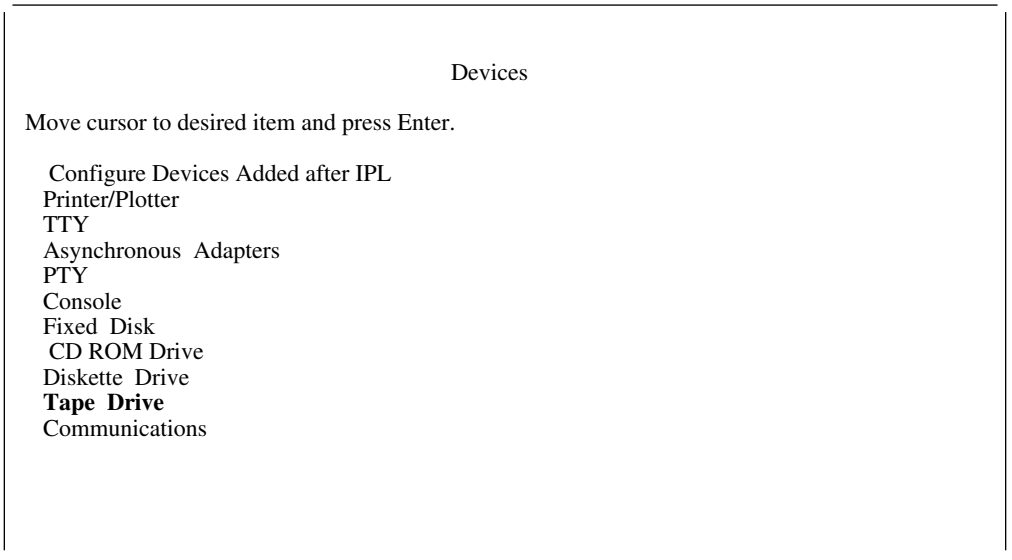


Figure 87. SMIT Devices Menu

The Tape Drive menu is displayed (Figure 88).

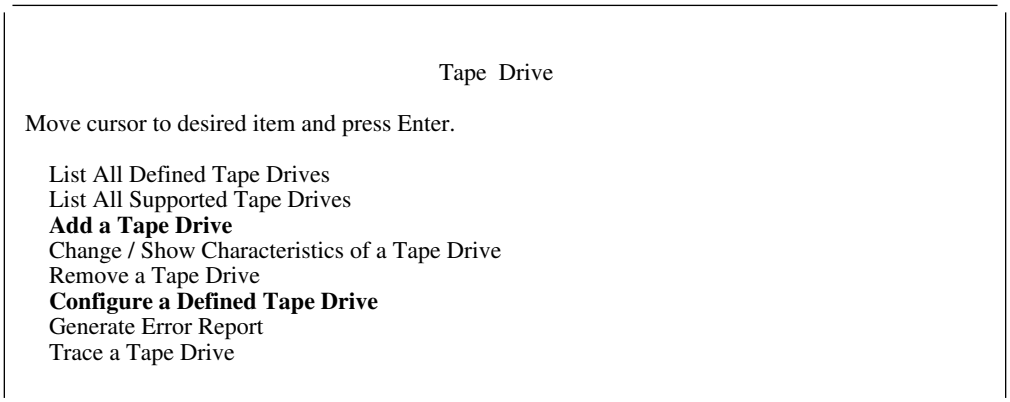


Figure 88. SMIT Tape Drive Menu

4. Select **Add a Tape Drive** from the Tape Drive menu (Figure 88). The Tape Drive Type menu is displayed (Figure 89).
5. Select the tape management pseudo control point from the Tape Drive Type menu (Figure 89).

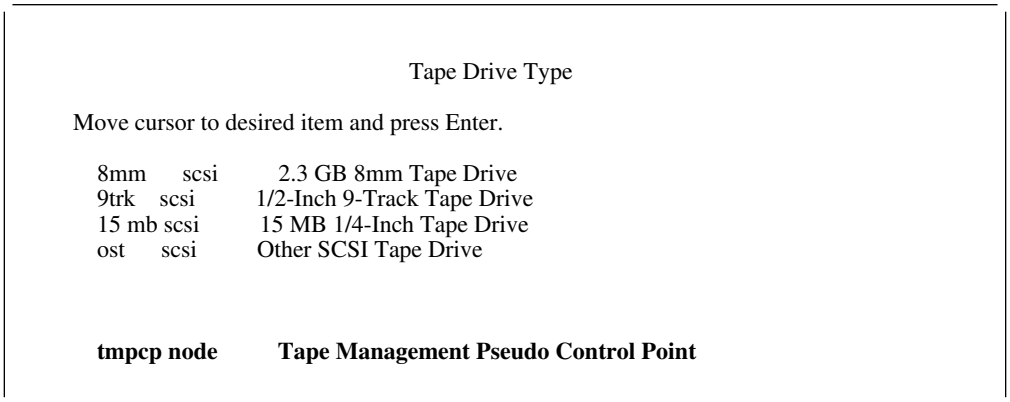


Figure 89. Tape Drive Type Menu

The Add a Tape Management Pseudo Control Point menu is displayed (Figure 90).

The Logical Name field for a tape management pseudo control point is optional. If the field is left blank, a unique name is assigned automatically. The assigned name is TMPCP***n***, where ***n*** is a number that indicates the TMPCP entry and starts with 0 for the first TMPCP defined.

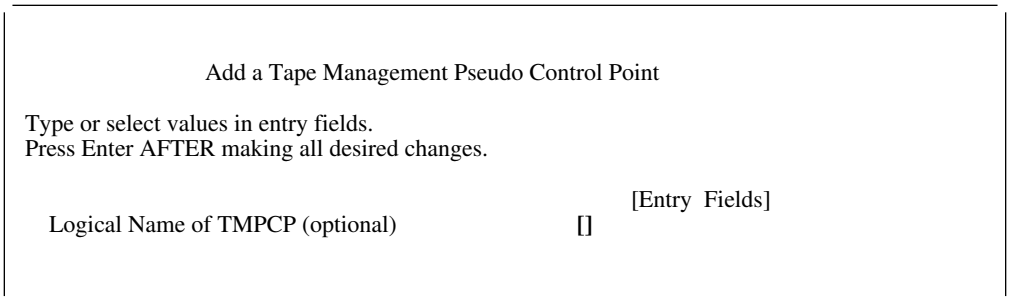


Figure 90. Add a Tape Management Pseudo Control Point Menu

6. Press **Enter** or click on **Do**.

   The COMMAND STATUS window is displayed with the **Command: OK** status. The logical name of the tape management pseudo control point is displayed as **Defined** (Figure 91).

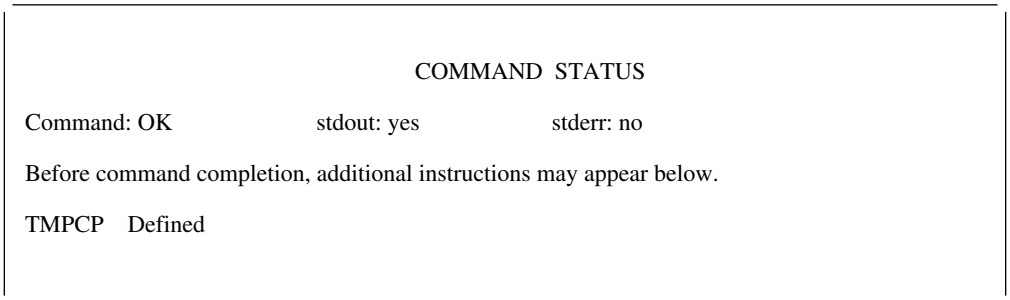


Figure 91. TMPCP COMMAND STATUS Window

7. Repeat steps 4 through 6 to define any additional tape management pseudo control points.
8. Return to the Tape Drive menu (Figure 88).
9. Select **Configure a Defined Tape Drive** from the Tape Drive menu. A list of TMPCP devices is displayed (Figure 92).

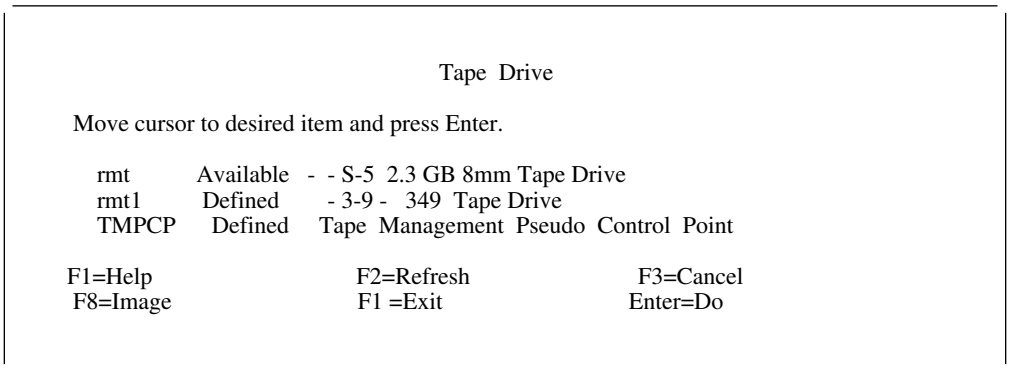


Figure 92. Configure a TMPCP Selection List

10. Select a defined tape management pseudo control point to configure from the TMPCP selection list.

    The COMMAND STATUS window is displayed with the **Command: OK** status, and the tape management pseudo control point is shown as **Available** (Figure 93).

```
                    COMMAND  STATUS

Command: OK          stdout: yes          stderr: no

Before command completion, additional instructions may appear below.

TMPCP    Available
```

Figure 93. Configure a Defined TMPCP COMMAND STATUS Window

11. Cancel the COMMAND STATUS window by pressing F3. The Tape Drive menu is displayed again (Figure 88).
12. Repeat steps 9 through 11 to configure additional tape management pseudo control points.
13. Exit from the SMIT menu by pressing F10.

    You have completed the configuration of tape management pseudo control points (TMPCPs) to the system.

Deconfiguring a Tape Management Pseudo Control Point (TMPCP)

Note: In the following examples, replace the letter **n** with the appropriate number for the selected TMPCP.

Use one of the following methods to deconfigure a TMPCP:

This method leaves the TMPCP defined in the configuration database. It is similar to bringing a device offline. The command unconfigures the TMPCP, but leaves it defined in the database. Enter the following command:

```
rmdev -l TMPCPn
```

This method removes the specified TMPCP definition from the device database. Enter the following command:

```
rmdev -l TMPCPn -d
```

The pseudo device driver modules are not unloaded from the kernel until the last TMPCP is deconfigured.

Physical Tape Device Driver with the Tape Management Enhancement (TMSDD)

Note: The description of the tape management enhancement logic in the tape device driver is included here for completeness relative to interaction with the pseudo tape device driver and support requirements by a tape management system application. This level of function is supplied with the

device driver support obtained with 3490 and 3590 tape devices and is not provided with this product. Refer to the documentation supplied with those tape products for the latest interface/functional definition and availability.

The tape management enhancement logic in the tape device driver (TDD) provides a set of functions that can be used by a management application to facilitate in the control of devices under the domain of a tape management system. This functionality provides a separate system call interface for tape manager access and a separate system call interface for end user access for the same physical tape device. The system call interface provided for tape manager access is referred to as the Management Protocol Interface (MPI). The system call interface provided for end user access is referred to as the User Protocol Interface (UPI).

The Management Protocol Interface (MPI) can be used to send tape position information, error and exception conditions, and operating mode information. It is also used by a tape management application to perform tape data access and control operations.

The User Protocol Interface (UPI) is the interface used for tape media access by the normal tape user for all read, write, and control operations.

Management Protocol Interface (MPI)

A Management Protocol Interface device is a device that was opened using a device special file that contains a major/minor number that has the (defined) MPI bit set. Refer to the section on “Tape Management Control Point (TMCP)” on page 446 and to the **mt_gdci** ioctl in *IBM Parallel System Support Programs for AIX: Command and Technical Reference* for information on obtaining MPI major/minor numbers for tape devices configured to the tape device driver.

When a tape device is opened in MPI mode, the tape device driver recognizes that device as being under the domain of a tape management system and no longer accepts an **open()** system call (for that device) using the normal (non-MPI/UPI) minor number definition to reference the device.

A tape device only allows one MPI **open()** process at a time and rejects any additional **open()** system calls with **errno** set to EBUSY. A second MPI **open()** request causes a *ddreason* of OPEN2 to be returned by a set/read state information (SRSI) **ioctl()** issued to or sleeping on the device opened in Management Protocol Interface mode. Refer to the **mt_srsi** ioctl in *IBM Parallel System Support Programs for AIX: Command and Technical Reference* for more information on this **ioctl()**.

A device opened in Management Protocol Interface mode supports the standard AIX system calls of **read()**, **write()**, and **close()**. Support is also provided for those ioctls supported by the standard (non-MPI) tape device, as well as support for additional ioctls designated for use by devices opened in MPI mode.

A tape management system can use the MPI ioctls to pass operating mode information and control information to the TDD. It can also use the MPI device to perform tape positioning functions and to perform label processing on behalf of, and transparent to, the user of the device or volume.

An MPI **ioctl()** is used to inform a tape management system when a condition requiring external processing is detected by the tape device driver. This interface is

used to communicate operational information and error and exception information from the TDD to the tape manager. When a condition that requires tape manager notification occurs, the TDD suspends all activity on the affected device for the end user request and notifies the tape manager. The TDD uses the SRSI **ioctl()** system call to communicate with the tape manager. When an SRSI **ioctl()** system call is outstanding for a device that requires tape management action, the TDD indicates the reason for notification in the *ddreason* field of the request and returns the request to the tape manager. When an SRSI **ioctl()** system call is not outstanding for a device that requires tape management action, the TDD queues the notification indication until an SRSI **ioctl()** system call is issued for the device. The UPI end user process, when suspended awaiting tape manager action, remains suspended until direction is received from the tape manager through use of one of the *ddaction* subcommands of the SRSI **ioctl()** system call.

A tape management system can be notified (through this interface mechanism) to handle Header, Trailer, and User Label processing as dictated by the operating mode the user is running in and the tape position information as maintained by the TDD. A tape management system can also be notified (through this interface mechanism) to handle device state changes, post error processing, and close processing prior to the TDD returning to the user. Refer to the **mt_srsi** ioctl in *IBM Parallel System Support Programs for AIX: Command and Technical Reference* for information on the **ioctl()** used to set or change UPI logical device images state information in the physical tape device driver.

The following ioctls are provided to allow a tape management system to create and control UPI logical device image devices for use by end user access. These ioctls are only supported by a device that was opened in MPI mode:

CUPI Creates a User Protocol Interface (logical device image)
RUPI Removes a User Protocol Interface (logical device image)
SRMI Sets or reads mode information (logical device image)
SRSI Sets or reads state information (logical device image)
SRSTI Sets or reads statistical information (logical device image)

CUPI ioctl() System Call: The CUPI **ioctl()** system call is used to create a UPI logical device image for the addressed device. The **ioctl()** returns a set of UPI major/minor numbers for use by a tape manager to create the device special files to be used by an end user designated to use this UPI logical device image. Refer to the **mt_cupi** ioctl in *IBM Parallel System Support Programs for AIX: Command and Technical Reference* for information on the **ioctl()** used to create UPI logical device images in the physical tape device driver.

SRMI ioctl() System Call: The SRMI **ioctl()** system call is used to set or read the operating mode of an existing UPI logical device image. Refer to the **mt_srmi** ioctl in *IBM Parallel System Support Programs for AIX: Command and Technical Reference* for information on the **ioctl()** used to set or read UPI logical device images mode information in the physical tape device driver.

RUPI ioctl() System Call: The RUPI **ioctl()** system call is used to remove a previously created UPI logical device image. Refer to the **mt_rupi** ioctl in *IBM Parallel System Support Programs for AIX: Command and Technical Reference* for information on the **ioctl()** used to remove UPI logical device images in the physical tape device driver.

SRSI ioctl() System Call: The SRSI **ioctl()** system call is used to set or read state information for an existing UPI logical device image. Refer to the **mt_srsi** ioctl in *IBM Parallel System Support Programs for AIX: Command and Technical Reference* for information on the **ioctl()** used to set or read UPI logical device images state information in the physical tape device driver.

SRSTI ioctl() System Call: The SRSTI **ioctl()** system call is used to set or read statistical information for a tape device. Refer to the **mt_srsti** ioctl in *IBM Parallel System Support Programs for AIX: Command and Technical Reference* for information on the **ioctl()** used to set or read statistical information in the physical tape device driver.

When a **close()** system call is executed for an MPI device while the associated UPI logical device image still exists (was not removed by a RUPI **ioctl()**), a state of MPI forced close is indicated for this MPI device. The control block structure for the MPI device, as well as the control block structures established for the UPI logical device image remain intact until removed by a subsequent UNCONFIGURE operation that finds the device in the MPI forced-close condition. In this case, all the associated control block structures for both the MPI and the UPI device are removed. A subsequent MPI **open()** system call for this device while in an MPI forced-close condition (not unconfigured) is accepted and the MPI forced-close condition is reset. All control block structures remain intact and subsequent UPI logical device image requests continue unchanged. Any UPI logical device image request that requires tape management action or is waiting for tape management action while the device is in an MPI forced close-condition is returned to the user with an **errno** of ENXIO.

User Protocol Interface (UPI)

A User Protocol Interface device is a logical device image created by a tape management system to provide a secure path for end user access through an environment controlled by that tape manager. The User Protocol Interface device is created by a tape management system through use of the CUPI **ioctl()** system call issued to a device opened in Management Protocol Interface mode. Refer to the section on Management Protocol Interface mode described previously and the **mt_cupi** ioctl in *IBM Parallel System Support Programs for AIX: Command and Technical Reference* for information on the creation of a UPI logical device image and the UPI major/minor numbers used to reference the device. A User Protocol Interface device can only be opened using a device special file that contains a major/minor number that has the (defined) UPI bit set.

The interface protocol support consists of the standard AIX system calls which include **open()**, **close()**, **read()**, and **write()**. The **ioctl()** system call is also supported, but the functions provided through the **ioctl()** interface protocol vary depending on the operating mode allocated for a given user by the Tape Manager, as communicated through the Management Protocol Interface (MPI), and the support provided by the tape management system.

Tape Management Control Point (TMCP)

A tape management control point (TMCP) is a logical extraction of the tape device driver. It is provided by the TMS enhancement logic and can be used to extract configuration information from the tape device driver, to provide notification relative to configuration change events within the tape device driver, and to provide device driver program failure notification.

A TMCP device special file (/**dev**/**TMCP.major#**) is created automatically by the TMS enhancement logic when the first tape device is configured, and is removed when the last tape device is unconfigured. A tape manager can use this TMCP entry to extract device configuration information from the tape device driver and to provide indirect access to MPI management **ioctl()** system calls that manipulate, initialize, and control logical device entries. A tape manager can also use the TMCP to receive configuration change information within the tape device driver, as well as exception condition reporting.

A TMCP only accepts an **open()** system call that contains the (defined) MPI bit set in the device minor number (MPI open request). Any **open()** system call issued to the TMCP without the MPI bit set in the minor number is rejected with **errno** set to EINVAL. A TMCP only allows one MPI **open()** process at a time and rejects any additional **open()** system calls with **errno** set to EBUSY. The second MPI **open()** also causes a *ddreason* of OPEN2 to be returned by a set/read state information (SRSI) **ioctl()** issued to or sleeping on the TMCP.

A TMCP uses the SRSI **ioctl()** system call to communicate with the tape manager. If an SRSI **ioctl()** system call is outstanding when an event occurs that requires tape management notification, the TMCP will indicate the reason for notification in the *ddreason* field of the request and returns the request to the tape manager. If an SRSI **ioctl()** system call is not outstanding when an event occurs that requires tape management notification, the TMCP will queue the notification indication until an SRSI **ioctl()** system call is issued for the TMCP.

A TMCP does not accept **read()** or **write()** system calls and rejects these requests with an **errno** value of EINVAL. A TMCP supports the following **ioctl()** requests:

CSK Creates the security key.

GDCI Gets device configuration information.

IOCINFO Gets general device information.

SRSI Set or reads state information. The only subcommands supported are RDMODE, SRSIWAIT, and WAITTIME, if the request is ultimately directed toward the TMCP.

With the exception of the other MPI **ioctl()** system call listed under the section on Management Protocol Interface (using the indirect addressing function of those ioctls), all other **ioctl()** system calls are rejected with **errno** set to EINVAL.

The GDCI **ioctl()** system call is used to obtain tape device driver configuration information from a TMCP. The subcommands provided with this **ioctl()** allow the requestor to obtain all configuration entries from the device driver, or only those new entries or configuration change requests that were received. Each device configuration entry returned includes the major/minor number with the (defined) MPI bit set that is required to open that device as a Management Protocol Interface (MPI) device. Refer to the **mt_gdci** ioctl in *IBM Parallel System Support Programs*

for AIX: Command and Technical Reference for information on the **ioctl()** used to get device configuration information from the physical tape device driver.

The SRSI **ioctl()** system call is used to obtain event information from a TMCP. The PACTION subcommands are not supported when issued to a TMCP. Refer to the **mt_srsi** ioctl in *IBM Parallel System Support Programs for AIX: Command and Technical Reference* for information on the **ioctl()** used to set or read UPI logical device images state information in the physical tape device driver.

The CSK **ioctl()** system call is used to create a security key for the TDD. This system call must be issued immediately following the **open()** system call issued to a TMCP. The security key established for the tape device driver must be included in the MPI **ioctl()** system calls on systems that do not support the “tmsmpi” C2 security privilege. Refer to the **mt_cskey** ioctl in *IBM Parallel System Support Programs for AIX: Command and Technical Reference* for information on the **ioctl()** used to create a security key for the physical tape device driver.

Installing the Enhanced Physical Tape Device Driver (TMSDD)

The enhanced physical tape device driver (TMSDD) is shipped as a separate **installp** image (tmsdd.driver). Users should install the base physical tape device drivers (for example, the AIX Parallel and ESCON channel attachment or SCSI tape devices) before the enhanced TMSDD device driver. Refer to your product’s associated physical tape device driver documentation for additional information.

The following is only included as an example for installing the TMSDD device driver with the AIX Parallel and ESCON channel attached tape devices or the SCSI tape devices. Users should follow *IBM SCSI Tape Drive and Library Device Drivers Installation and User's Guide* or *AIX Parallel and ESCON Channel Tape Attachment/6000 Installation and User’s Guide* to perform the installation. The enhanced TMSDD needs to be installed on every node that has a physical tape device attached to it.

After the base physical tape device drivers have been successfully installed, the following procedure can be used to install the enhanced TMSDD:

To install from a 3 1/2" disk, enter:

```
# installp -ac -d /dev/rfd  -X ssp.tms
```

To install from tape, enter:

```
# installp -ac -d /dev/rmt  -X ssp.tms
```

To install from an **installp** image file, enter:

```
# installp -ac -d /full_pathname/ssp.tms -X ssp.tms
```

In the following examples, assume that the user has completed the installation and configuration of the channel emulator device drivers (S370 or S390) and the tape control unit (3480cu0 or 3490cu0) or the SCSI tape devices (3490E or 3590). For detailed SCSI device driver information, please refer to *IBM SCSI Tape Drive and Library Device Drivers Installation and User's Guide*. For detailed channel emulator device driver information, please refer to *AIX Parallel and ESCON Channel Tape Attachment/6000 Installation and User’s Guide*.

Defining and Configuring a 3480/3490 Tape Device

Note: You must configure the S370 Parallel Channel Emulator or the S390 ESCON Channel Emulator before you can define any tape devices.

The following is included only as an example for defining and configuring tape device drivers. Refer to your product's associated device driver documentation for additional information.

Perform the following procedure to define and configure a tape device:

1. Enter the following command:

   smit

   The SMIT main menu is displayed (Figure 94).

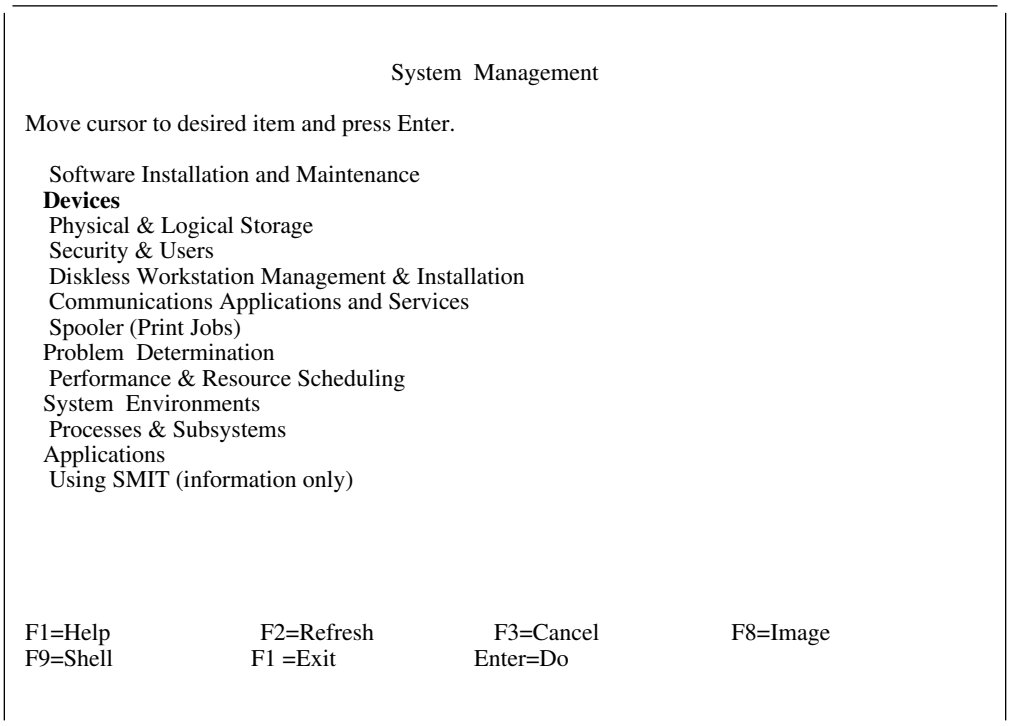


Figure 94. SMIT Main Menu

2. Select **Devices** (item 2) from the SMIT main menu.

   Use the up (↑) and down (↓) arrow keys to move the cursor to the desired item on the menu.

   Press **Enter** or click on **Do** if you are processing **smit** in a windowed environment.

3. Select **Tape Drive** from the Devices menu (Figure 95).

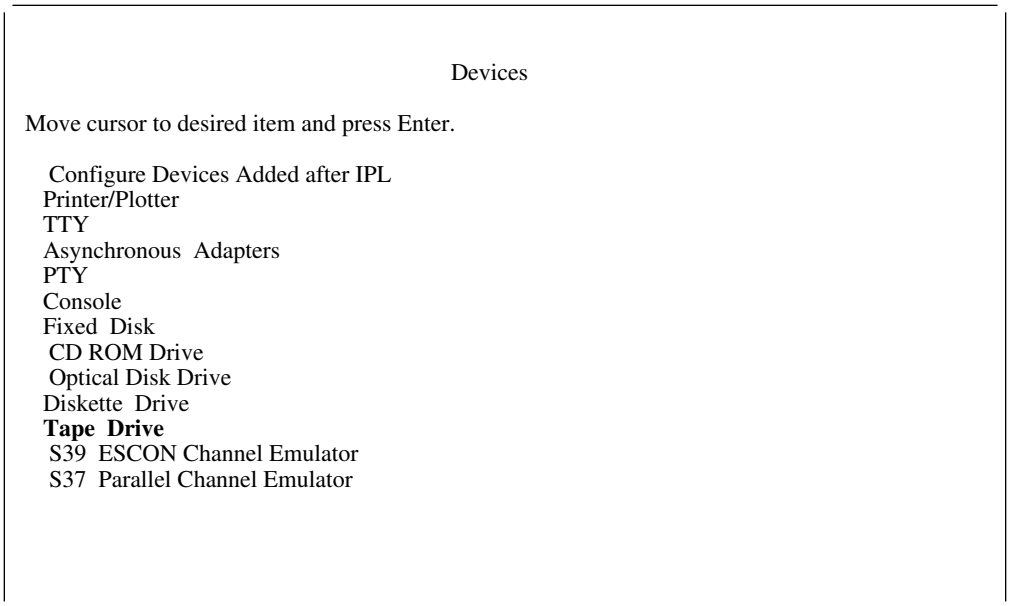


Figure 95. SMIT Devices Menu

The Tape Drive menu is displayed (Figure 96).

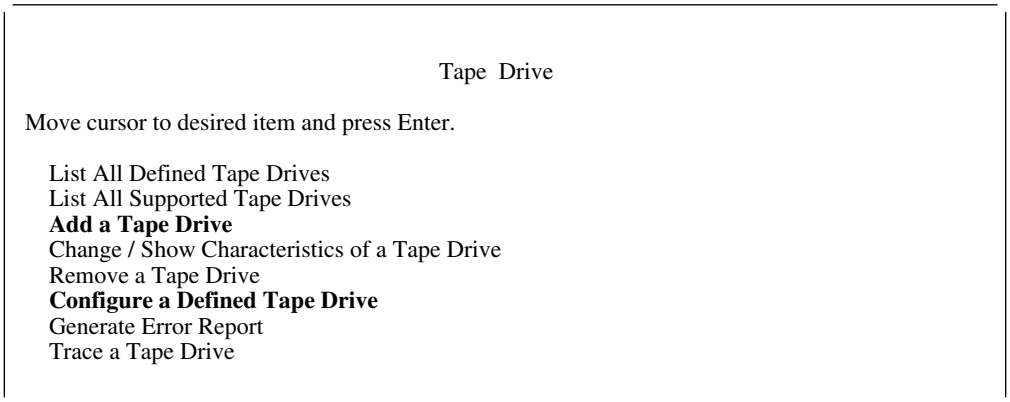


Figure 96. SMIT Tape Drive Menu

4. Select **Add a Tape Drive** from the Tape Drive menu. The Tape Drive Type menu is displayed (Figure 97).

   Note: The 3420, 3480, and 3490 devices support the concept of a control unit. A control unit must be added to the configuration before any 34xx tape drive can be defined. Each control unit in the configuration must have a unique name. The name must not conflict with any other names for tape devices defined for the system. The control unit name is referenced when adding tape drives. The control units are defined in the S370 Parallel Channel Emulator or S390 ESCON Channel Emulator menus.

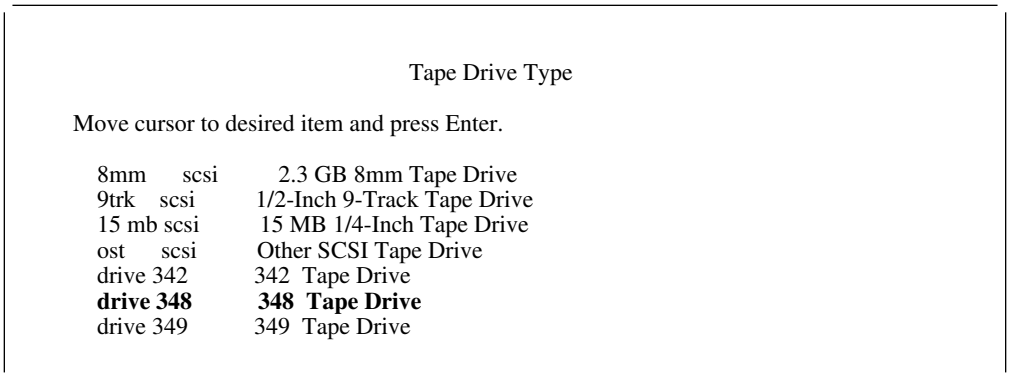


Figure 97. Tape Drive Type Menu

5. Select the type of tape drive (3420, 3480, or 3490).
6. Select the parent control unit of the tape drive. The parent control unit is a previously defined control unit to which the tape drive is attached (Figure 98).

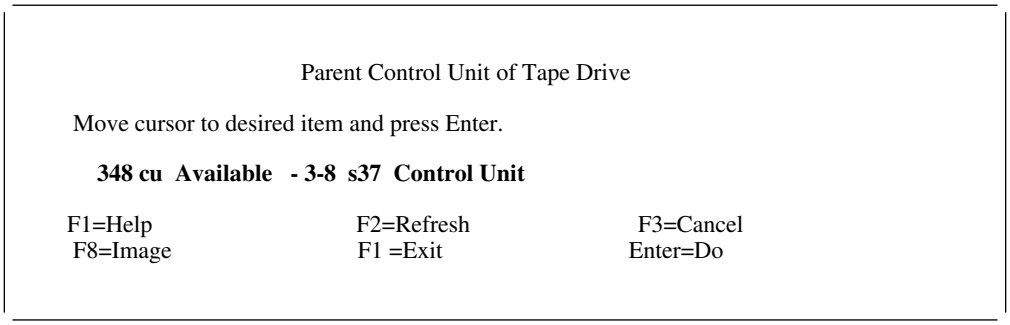


Figure 98. Parent Control Unit Menu

The Add a 3480 Tape Drive menu is displayed (Figure 99).

The Logical Name field for the tape drive is optional. If the field is left blank, the assigned name is rmt***n***, where ***n*** is a number that indicates the ***n***th tape device defined in the system. Typically, because the identifier for an 8-mm or a 1/4-inch tape device is already defined, the identifier for 34xx tape devices begins with **rmt1**. This identifier is also used for the special file name that is associated with the device. If a user-created logical name is added, it must not conflict with the logical names of other tape devices or control units.

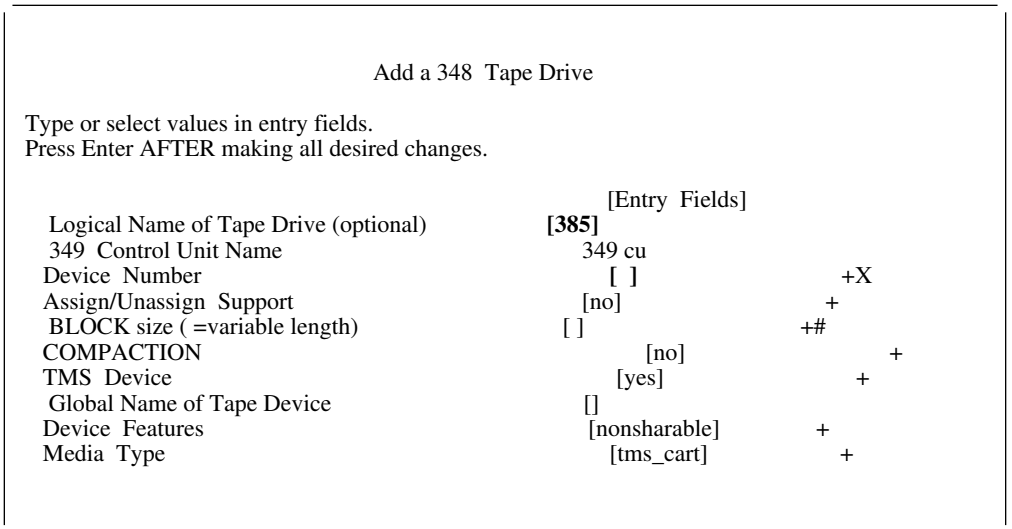


Figure 99. Add a 3480 Tape Drive Menu

7. Enter the device number of the tape drive.
8. Specify the assign/unassign option.

   If this field is set to **yes**, the tape drive is automatically assigned (reserved) to the RS/6000 host at device configuration. Similarly, the tape device is automatically unassigned when the device is changed from the configured state to the defined state. The default is **no** for no automatic assignment.

   Notes:

   a. If the device being configured is already assigned by another host and the assignment field is set to **yes**, the configuration will fail.

   b. The Global Name of Tape Device default is set to the same as the Logical Name of Tape Drive.

   c. The TMS Device default is set to no.

   d. The Media Type default for 3480 and 3490 tape drives is set to tms_cart. The Media Type default for a 3420 tape drive is set to tms_reel.

   e. The COMPACTION default is set to yes.
9. Press **Enter** or click on **Do**.

   The COMMAND STATUS window is displayed with the **Command: OK** status. The logical name of the tape device must be shown as **Defined** (Figure 100).

```
                    COMMAND  STATUS

Command: OK          stdout: yes          stderr: no

Before command completion, additional instructions may appear below.

385 Defined
```

Figure 100. Add a 3490 Tape Drive COMMAND STATUS Window

10. Repeat steps 5 through 9 to define the remaining tape drives for the control unit.
11. Repeat steps 4 through 9 to define any additional tape drives.
12. Return to the Tape Drive menu (Figure 96).
13. Select **Configure a Defined Tape Drive** from the Tape Drive menu. A list of tape devices is displayed (Figure 101).

```
                    Tape  Drive

  Move cursor to desired item and press Enter.

    384      Available   - - S-5  2.3  GB  8mm  Tape  Drive
    385      Defined      - 3-9 -   349  Tape  Drive

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

Figure 101. Configure a Defined Tape Drive Selection List

14. Select a defined tape drive to configure from the Defined Tape Drive Selection list.

    The COMMAND STATUS window is displayed with the **Command: OK** status, and the tape drive is shown as **Available** (Figure 102).

```
                    COMMAND  STATUS

Command: OK          stdout: yes          stderr: no

Before command completion, additional instructions may appear below.

385   Available
```

Figure 102. Configure a Defined Tape Drive COMMAND STATUS Window

15. Cancel the COMMAND STATUS window by pressing F3. The Tape Drive window is displayed again.
16. Continue selecting tape drives for configuration until all of the devices are configured.

    Note: You can define more than one drive at the same address on an adapter or control unit; however, you can only configure one of these drives at a time.
17. Exit from the SMIT menu.

Changing 3480/3490 Tape Device Attributes

Perform the following procedure to change tape device attributes:

1. Enter the following command:

   smit

   The SMIT main menu is displayed (Figure 103).

```
                                System  Management

Move cursor to desired item and press Enter.

  Software Installation and Maintenance
 Devices
  Physical & Logical Storage
  Security & Users
  Diskless Workstation Management & Installation
  Communications Applications and Services
  Spooler (Print Jobs)
 Problem  Determination
  Performance & Resource Scheduling
 System  Environments
  Processes & Subsystems
 Applications
  Using SMIT (information only)



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

Figure 103. SMIT Main Menu

2. Select **Devices** (item 2) from the SMIT main menu.

   Use the up (↑) and down (↓) arrow keys to move the cursor to the desired item on the menu.

   Press **Enter** or click on **Do** if you are processing **smit** in a windowed environment.
3. Select **Tape Drive** from the Devices menu (Figure 104).

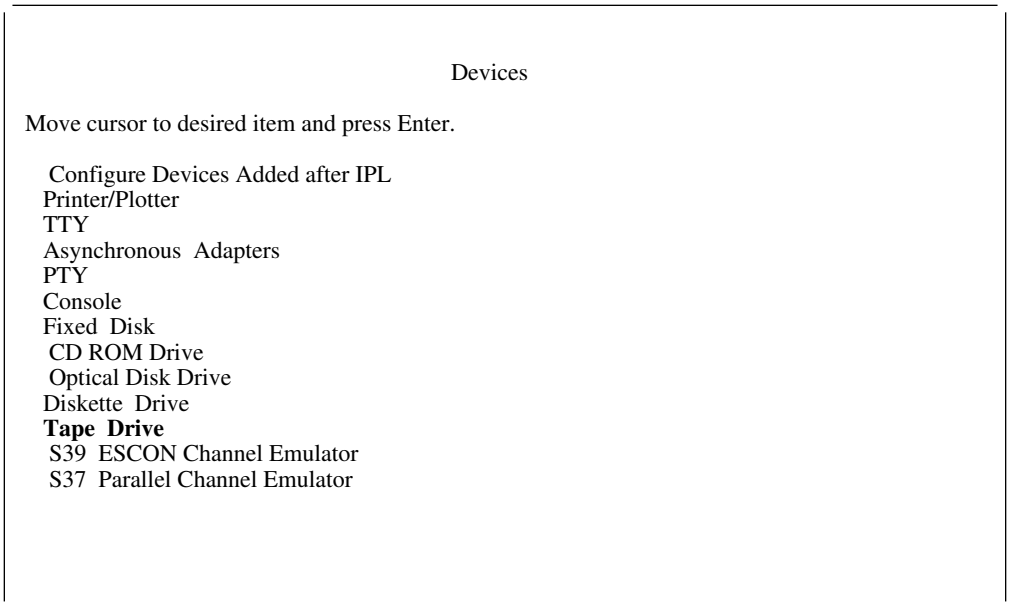


Figure 104. SMIT Devices Menu

4. Select **Change / Show Characteristics of a Tape Drive** from the Tape Drive Menu. The Tape Drive Type menu is displayed (Figure 105).

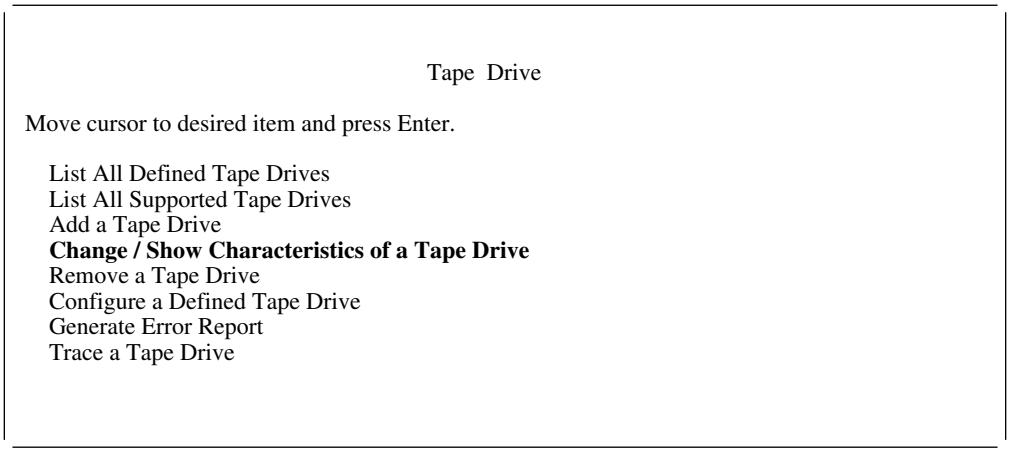


Figure 105. Tape Drive Type Menu

5. Select the tape device that you want to change (for example, 384).

```
                                Tape  Drive

 Move cursor to desired item and press Enter.

 rmt  Available   - 8-  -2  2.3 GB 8mm Tape Drive
 384  Defined      - 3-A - 4  348   Tape  Drive
 385  Available   - 3-A - 5  348   Tape  Drive

 F1=Help                    F2=Refresh                 F3=Cancel
 Esc+8=Image                Esc+ =Exit                 Enter=Do
 /=Find                     n=Find  Next
```

Figure 106. Tape Drive Submenu

6. Select **TMS Device** and change the value from **no** (default) to **yes**. The device must be configured under the TMS domain in order to use the tape management enhancement functions.
7. Select **Device Features** to change the device features.

```
                Change / Show Characteristics of a 348  Tape Drive

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

                                                    [Entry  Fields]
  Logical Name of Tape Drive                        384
  348  Control Unit Name                            348 cu
  Device  Number                                      4
  Assign/Unassign  Support                          [no]                +
  BLOCK size ( =variable length)                    [ ]                 +#
  COMPACTION                                          [no]                      +
  TMS  Device                                         [yes]               +
  Global Name of Tape Drive                         [384]
  Device  Features                                    [nonsharable]       +
  Media  Type                                         [tms_cart]          +

 F1=Help              F2=Refresh           F3=Cancel            F4=List
 Esc+5=Reset          Esc+6=Command        Esc+7=Edit           Esc+8=Image
 Esc+9=Shell          Esc+ =Exit           Enter=Do
```

Figure 107. Change / Show Characteristics of a 3480 Tape Drive Menu

8. To change the device features, select **Device Features**, then press F4 to list all the available device features.

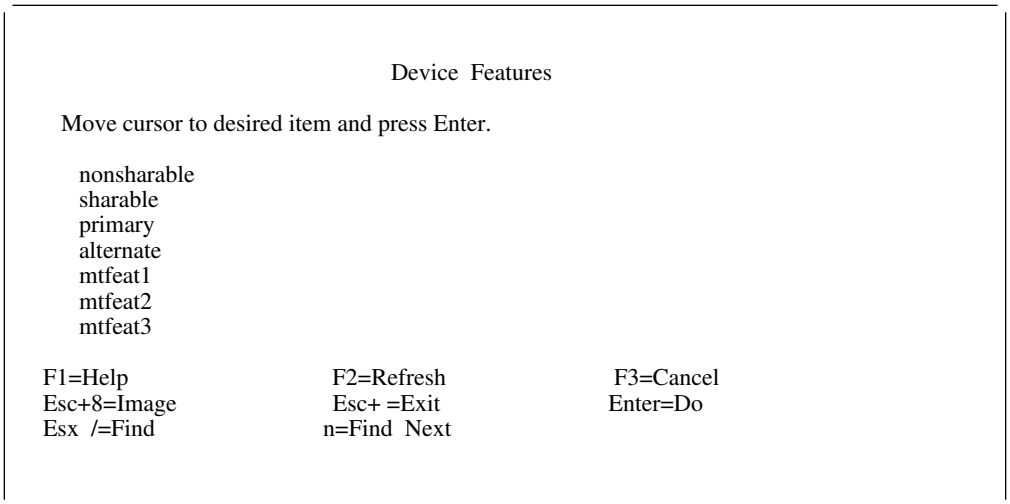


Figure 108. Device Features List Submenu

In this section, a "path" refers to a defined or configured tape drive (see Figure 105) and "device" refers to the associated physical device.

The following features imply that the device is sharable with other hosts using the assign/unassign protocol.

nonsharable Indicates that TMS access to the device is only through this path. The device cannot be shared with another tape management system.

sharable Indicates that TMS access to the device is only through this path. The device can also be known to other (atomic) tape management systems.

primary Indicates that the device is accessible to the TMS through multiple paths across one or more nodes and this path should be used as the primary access path. At most, one primary path should be defined for a device. If this option is selected, the **Global Name of Tape Drive** must be specified for each path.

alternate Indicates that this device is accessible to the TMS through multiple paths across one or more nodes and this path should be used as an alternate access path. All of the paths to a device can be defined as alternate paths. If this option is selected, the **Global Name of Tape Drive** must be specified for each path.

mtfeat1, mtfeat2, mtfeat3
Reserved for future use.

9. To change the media type, select **Media Type**, then press F4 to list all available media types.

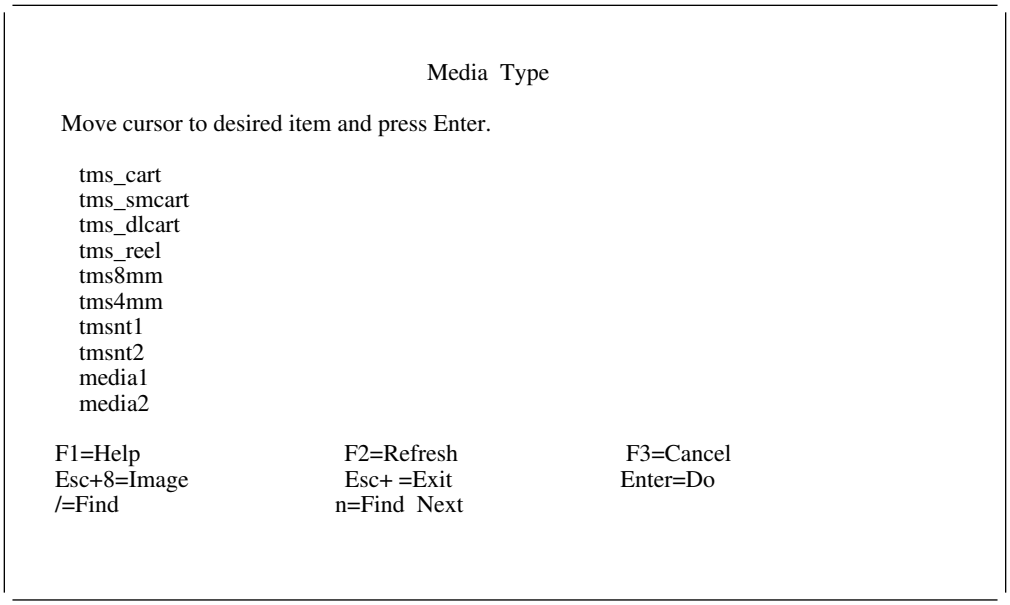


Figure 109. Media Type List Submenu

tms_cart Indicates a cartridge type device (3480, 3490).

tms_dlcart Indicates a double length cartridge type device (3490E).

tms_reel Indicates a reel type device (3420).

tms_smcart Indicates a small cartridge type device.

tmsnt1 Indicates a NT1 type device.

tmsnt2 Indicates a NT2 type device.

tms4mm Indicates a 4mm type device.

tms8mm Indicates an 8mm type device.

media1, media2
Reserved for future use.

10. Press enter when you have completed your selections or changes.

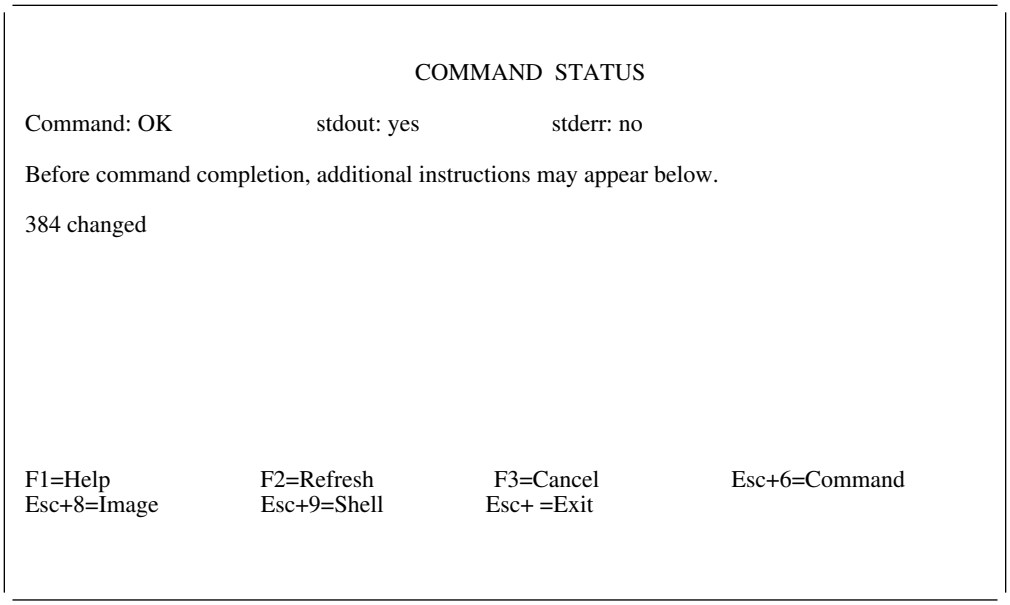


Figure 110. Command Status Menu

11. Exit from the SMIT menu.

    You have completed the installation and definition of 34xx tape devices to the system.

Changing SCSI (3490E or 3590) Tape Device Attributes

The following is included only as an example for defining and configuring tape device drivers. Refer to your product's associated device driver documentation for additional information.

Perform the following procedure to change tape device attributes:

1. Enter the following command:

   smit

   The SMIT main menu is displayed (Figure 111).

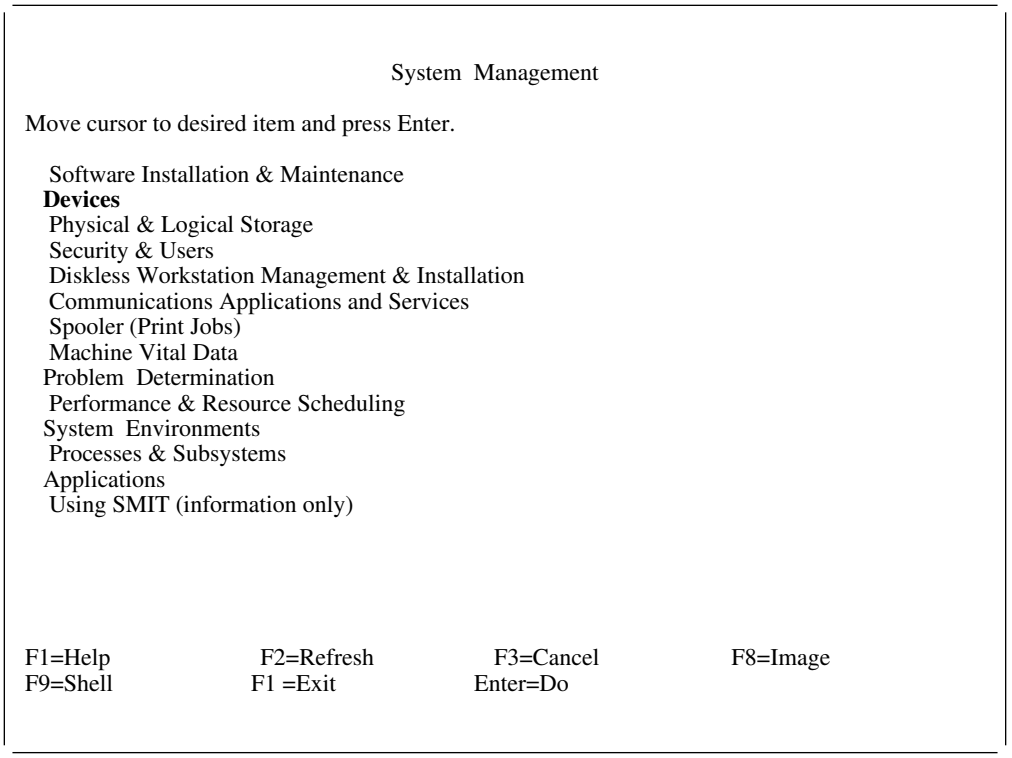


Figure 111. SMIT Main Menu

2. Select **Devices** from the SMIT main menu.

   Use the up (↑) and down (↓) arrow keys to move the cursor to the desired item on the menu.

   Press **Enter** or click on **Do** if you are processing **smit** in a windowed environment.

3. Select **Tape Drive** from the Devices menu (Figure 112).

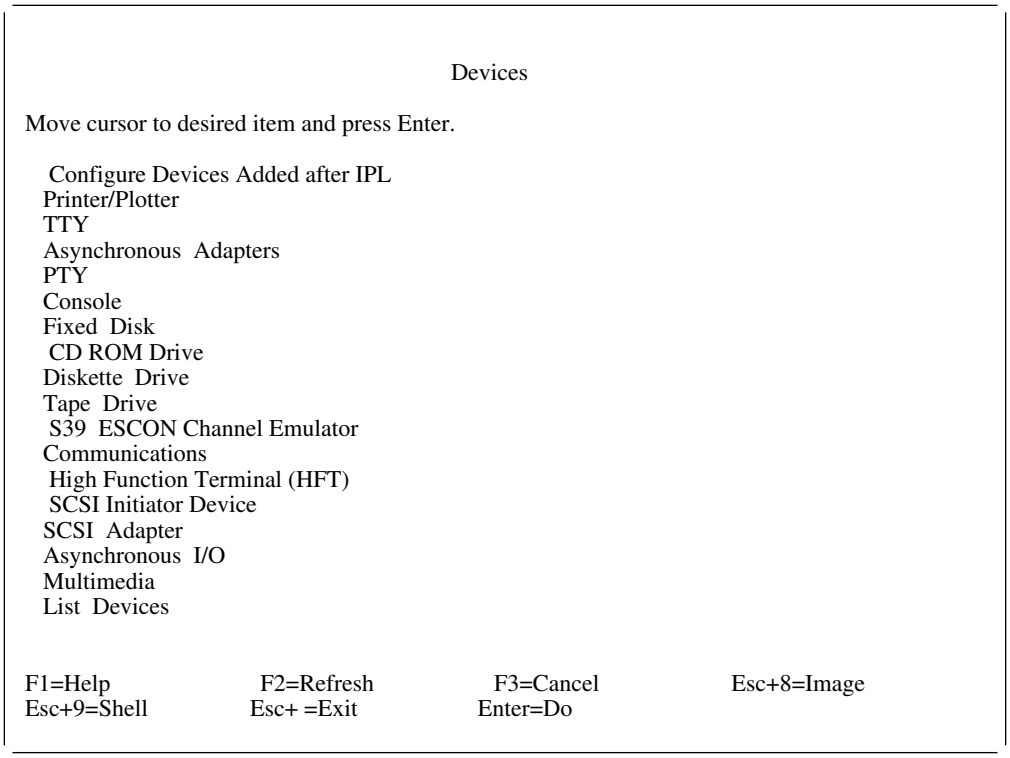


Figure 112. SMIT Devices Menu

4. Select **Change / Show Characteristics of a Tape Drive** from the Tape Drive Menu. The Tape Drive Type menu is displayed (Figure 113).

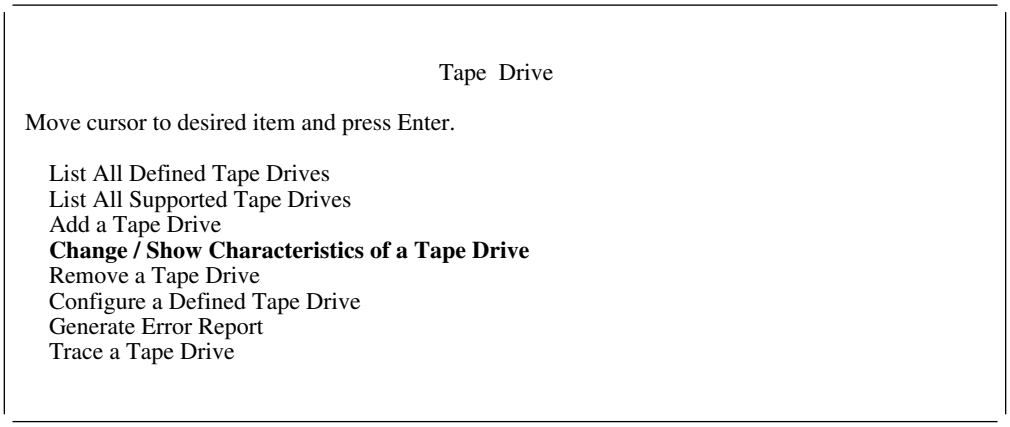


Figure 113. Tape Drive Type Menu

5. Select the tape device that you want to change (for example, rmt0).

```
                                    Tape  Drive

  Move cursor to desired item and press Enter.

    rmt        Available   - 2- 1-   IBM  SSD  349 E  tape  device
     TMPCP      Available                 Tape Management Pseudo Control Point

  F1=Help                    F2=Refresh                  F3=Cancel
  Esc+8=Image                Esc+ =Exit                  Enter=Do
  F1x  /=Find                n=Find  Next
```

Figure 114. Tape Drive Submenu

6. Select TMS device and change the value from **no** (default) to **yes**. To use the tape management enhancement functions, the device must be configured under the TMS domain.
7. Select **Device Features** to change the device features.

```
                    Change / Show Characteristics of a Tape Drive

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

 [TOP]                                                    [Entry  Fields]
   Tape  Drive                                             rmt
    Tape Drive type                                       A349 E
    Tape Drive interface                                 scsi
    Description                                           IBM SSD 349 E tape device
   Status                                                  Available
   Location                                                 - 2- 1-
   Parent  adapter                                        vscsi1
   Connection  address
    Block size ( =variable length)                  [ ]
    Use DEVICE BUFFERS during writes                      yes
    Use data compression                                 [yes]
    Allow device to dominate SCSI bus                   [no]
    Turn on volume information logging                 [yes]
    Maximum size of log file (in # of entries)   [5  ]
    Backward Space/Forward Space Record Mode          SCSI
   TMS  Device                                             [yes]
    Global Name of Tape Drive                            []
   Device  Features                                      [nonsharable]
   Media  Type                                             [tms_cart]
 [BOTTOM]



 F1=Help              F2=Refresh           F3=Cancel            F4=List
 Esc+5=Reset          Esc+6=Command        Esc+7=Edit           Esc+8=Image
 Esc+9=Shell          Esc+ =Exit           Enter=Do
```

Figure 115. Change / Show Characteristics of a 3490E Tape Drive Menu

8. To change the device features, select **Device Features**, then press F4 to list all the available device features.

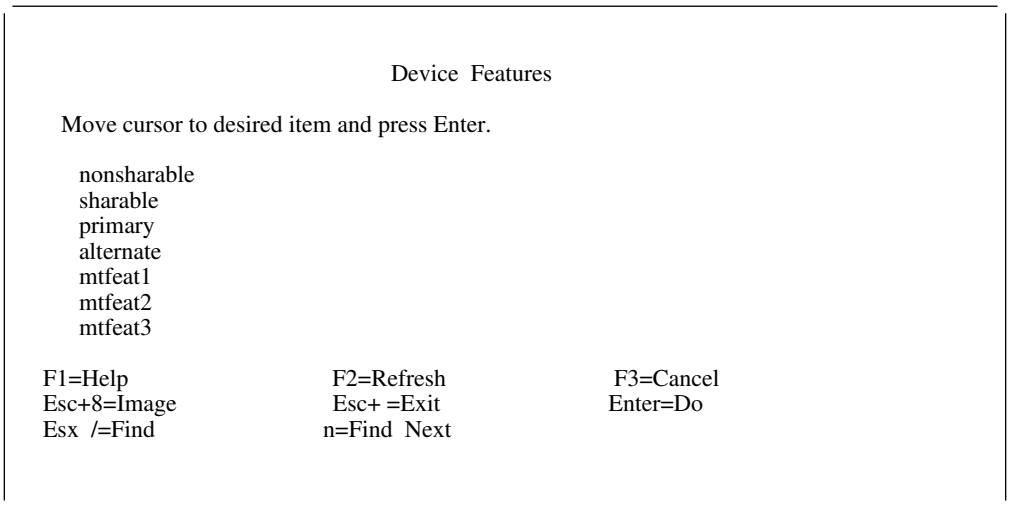


Figure 116. Device Features List Submenu

In this section, a "path" refers to a defined or configured tape drive and "device" refers to the associated physical device.

The following features imply that the device is sharable with other hosts using the assign/unassign protocol.

nonsharable Indicates that TMS access to the device is only through this path. The device cannot be shared with another tape management system.

sharable Indicates that TMS access to the device is only through this path. The device can also be known to other (atomic) tape management systems.

primary Indicates that the device is accessible to the TMS through multiple paths across one or more nodes and this path should be used as the primary access path. At most, one primary path should be defined for a device. If this option is selected, the **Global Name of Tape Drive** must be specified for each path.

alternate Indicates that this device is accessible to the TMS through multiple paths across one or more nodes and this path should be used as an alternate access path. All of the paths to a device can be defined as alternate paths. If this option is selected, the **Global Name of Tape Drive** must be specified for each path.

mtfeat1, mtfeat2, mtfeat3
Reserved for future use.

9. To change the media type, select **Media Type**, then press F4 to list all available media types.

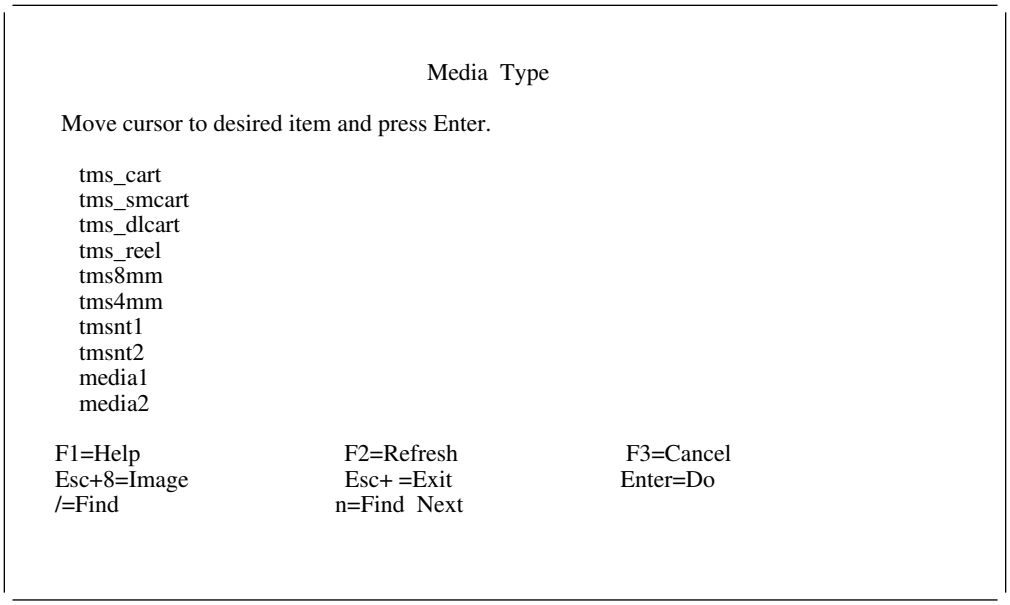


Figure 117. Media Type List Submenu

tms_cart Indicates a cartridge type device (3480, 3490).

tms_dlcart Indicates a double length cartridge type device (3490E).

tms_reel Indicates a reel type device (3420).

tms_smcart Indicates a small cartridge type device.

tmsnt1 Indicates a NT1 type device.

tmsnt2 Indicates a NT2 type device.

tms4mm Indicates a 4mm type device.

tms8mm Indicates an 8mm type device.

media1, media2
Reserved for future use.

10. Press enter when you have completed your selections or changes.

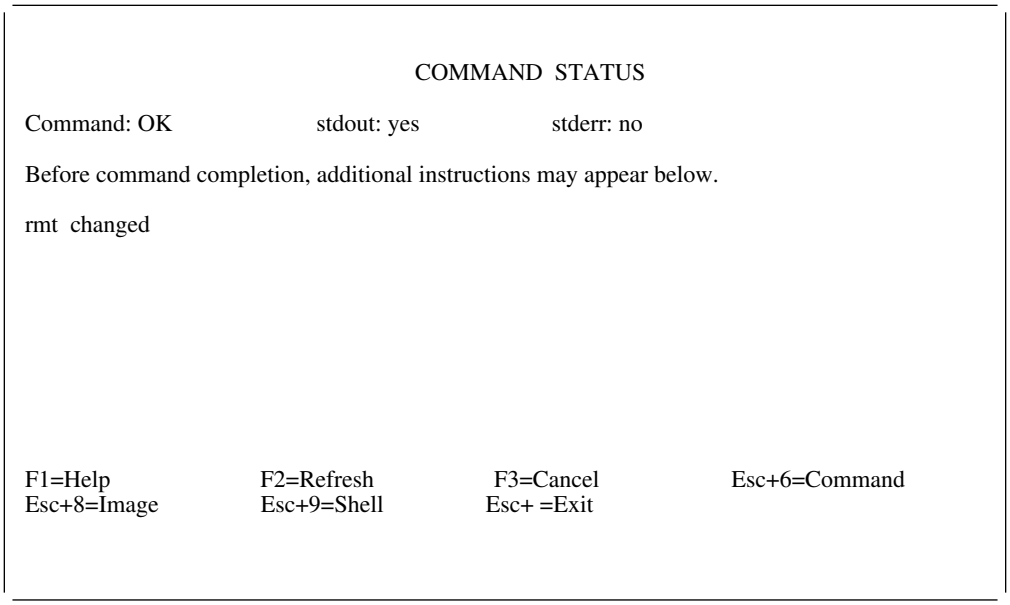


Figure 118. Command Status Menu

11. Exit from the SMIT menu.

    You have completed the installation and definition of 34xx tape devices to the system.

Refer to your product's associated base physical tape device driver documentation for additional information about defining, configuring, changing attributes, and unconfiguring a tape device.

Appendixes

© Copyright IBM Corp. 1996

Appendix A. Using an Alternate Boot System Image

The Alternate Boot System Image function provides the ability to configure two different **rootvg** volume groups for an SP node or the control workstation. The **rootvg** volume group contains logical volumes for:

Operating system file systems
JFS logs (journaled file system logs)
Dump space
Paging space

You may select one of the two levels of software to execute by changing the boot list in NVRAM (nonvolatile random access memory) and booting the node or control workstation with the disk that contains the desired software level. The node or control workstation will, in effect, have two **rootvg** volume groups. Only one is active at any time and no changes from one **rootvg** volume group are propagated to the other **rootvg** volume group. For example, /**tmp** file changes on one **rootvg** volume group are not changed on the dormant **rootvg** volume group. External or non-**rootvg** volume groups on additional disks are available via export and import of the volume group on both **rootvg** volume groups.

This function is useful when a quick backout of a new software level is required. Such a backout may be desired for testing or to recover from problems found soon after a new software level is put into production.

Enabling Alternate Boot System Image Function

To use the Alternate Boot System Image function, a node or the control workstation must have the following hardware:

1. Two hard disks (one for each **rootvg** volume group)
2. A virtual battery on the SP nodes

SP Virtual Battery

The SP virtual battery is a power source that keeps NVRAM powered and the time-of-day clock running while the SP frame is plugged into a power outlet with power applied or until a node is removed from the SP frame. Use of the virtual battery provides for maintaining the boot list while the node is powered off. Power is supplied by the node supervisor card in each node. This card provides line power even when the node is powered off.

On wide SP nodes, a virtual battery is installed from the factory. On thin SP nodes, a no charge Engineering Change Action (ECA) can be ordered by your IBM representative. Order ECA number ECA007. On the control workstation, a battery is installed and no virtual battery ECA is required.

© Copyright IBM Corp. 1996

Using the Alternate Boot System Image Function

To use the Alternate Boot System Image function on an SP node, take the following steps:

1. Perform the following steps to install each system image on a single disk:
   a. Enter **spbootins -r install -h hdisk0 -l** *node_list*
   b. Run **setup_server** on the boot/install server
   c. Select **Select** and **Do Net Boot** from the Global Controls display on the SP System Monitor.
   d. Redo steps a-c and change the **spbootins -h** option to select the other install disk.
2. Set the boot list to the disk containing the image you want to boot.
   a. On the SP node, enter:

      ```
      /bin/bootlist -m prevboot -i
      ```

      This invalidates the previous boot list.
   b. On the SP node, enter:

      ```
      /bin/bootlist -m normal ent  hdisk
      ```

      This sets the boot list to boot from the SP Ethernet first (so an install can be done if required.) In this example, if an installation is not being performed, the boot list will boot from **hdisk0**.
   c. On the SP node, enter:

      ```
      spbootins -h hdisk  -l 5
      ```

      This updates the SDR with the correct boot disk information. In this example, for **node_number 5** in the Node class of the SDR, the **bootdisk** attribute is set to **hdisk0**.

   All of the boot list manipulations can be done on the control workstation, using **dsh** for a group of nodes. The SDR command needs to be done for each node in the group.
3. Export any non-**rootvg** volume groups (**exportvg** *volume_group_name*).
4. Reboot the node to bring up the system image that was installed on **hdisk0**.
5. Import any non-**rootvg** volume groups (**importvg -y** *volume_group_name* **hdisk***x*).

To switch back to the other system image, redo step 2 and change the **-h hdisk0** value to the physical disk name of the alternate system image.

Setting the Boot List Manually

When a frame drops power or a node is removed, you must manually set the boot list on every node in that frame. In the case of nodes being removed from the frame, just the boot list on that node needs to be manually changed. To manually change the boot list for a specific node, perform the following steps from the SP System Monitor GUI on the control workstation:

1. Select **SP** → **Display Layouts**

2. Select the number of the frame you want to see
3. Select the number of the node within the frame
4. Select **Select Node Front Panel Layout**
5. Press **Display Node Layout**
6. Set the key switch position to **Secure**
7. Open the console for that node
8. **Power on** the node
9. Wait for the LED display to get to 200
10. Set the key switch position to **Service**
11. Press the **Reset** button
12. The Diagnostic Menu appears in the console window
13. Hit **Enter**
14. Select **Service Aids**
15. Select **Display/Alter Bootlist**
16. Select **Normal Mode Bootlist**
17. Select **Alter Current Bootlist**
18. Assign **1** to **en0** and **2** to **hdisk***x*
19. Press F7 to commit the change
20. Reboot the node

Using Alternate System Image on the Control Workstation

The alternate system image can be used on a control workstation with at least two hard disks available for the **rootvg**. Each hard disk must be large enough to hold the entire operating system and the /**spdata** directory.

Because the control workstation has a battery, the SP virtual battery is not needed.

To use the alternate system image function, you must first install both levels of software (AIX and PSSP) on each disk (one image per disk). Configure each level of software as a control workstation. When using alternate system images, complete and separate copies of the /**spdata** information are also made. The control workstation cannot have multiple system images while the SP nodes have multiple system images. You must first migrate the control workstation to a new software level and commit that software level (that is, you are not able to change back to the old level) before you create additional system images for the SP nodes. All alternate system images for the SP nodes must be created while the control workstation is at the new software level.

Because changes are not propagated to dormant **rootvg**s, you should not make any node configuration changes during the period when alternate system images are in use on the control workstation.

Steps to Switch Between Alternate System Images

1. Install the control workstation software (AIX and PSSP) on a single hard disk for the first system image.

   Use the AIX and PSSP documentation to install to a single disk. As an example, use **hdisk0**.

2. Install the control workstation software (AIX and PSSP) on a single hard disk for the second system image.

   Use the AIX and PSSP documentation to install to a single disk. As an example, use **hdisk1**.

3. Set the boot list to **hdisk0**.

   a. Enter the **diag** command at the shell prompt.

   b. The Diagnostic Menu appears in the console window

   c. Hit **Enter**

   d. Select **Service Aids**

   e. Select **Display/Alter Bootlist**

   f. Select **Normal Mode Bootlist**

   g. Select **Alter Current Bootlist**

   h. Assign **1** to **hdisk0***x*

   i. Press F7 to commit the change

   j. Reboot the node

To switch to the new level assign **1** to **hdisk1** in the **Alter Current Bootlist** menu, commit the change and reboot the node.

Appendix B. SP Daemons

The following daemons are specific to the SP servers and processor nodes:

amd	Automount daemon; automatically mounts a directory on request based on map entries
fault_service_Worm_RTG	Fault-handling daemon for the High Performance Switch
fault_service_Worm_SP	Fault-handling daemon for the SP Switch
fs_monitor	Sticky interrupt handler for the High Performance Switch
fs_monitor_SP	Sticky interrupt handler for the SP Switch
haemd	Event Manager daemon; matches information about the state of system resources with information about resource conditions that are of interest to client programs to create events; runs on all nodes of a system partition; one instance per system partition runs on the control workstation.
hagsd	Group Services daemon; provides services for coordinating and monitoring changes to the state of a client program that is running on a set of nodes in an SP system; runs on all nodes of a system partition; one instance per system partition runs on the control workstation.
hagsglsmd	A Group Services subsystem daemon that provides global synchronization services for the High Performance Switch adapter membership group; runs on all nodes of a system partition; one instance per system partition runs on the control workstation.
hardmon	Interfaces to the serial port on each SP frame; does tasks such as polling hardware for changes in hardware status, sending state changes to interested parties, sending commands to change power state and key lock position, and providing an interface to the S1 serial port of each processing node
hatsd	Topology Services daemon; provides information to other PSSP subsystems about the state of the nodes and adapters on the SP system; runs on all nodes of a system partition; one instance per system partition runs on the control workstation.
hbd	Heartbeat daemon; runs on all nodes, including the control workstation

© Copyright IBM Corp. 1996

hrd	Host responds daemon; runs only on the control workstation (See Appendix F, “The System Data Repository” on page 541 for more information.)
jmd	SP Resource Manager
kerberos	Kerberos authentication server (doesn't run on nodes)
kpropd	Kerberos backup communication daemon (doesn't run on nodes)
kshd	Authenticated server for SP remote commands **rsh** and **rcp**
pmand	The Event Management client daemon that dispatches actions based on subscribed-to events
pmanrmd	Provides resource variables for problem management
sdrd	System Data Repository server
sp_configd	Creates and sends SNMP traps when selected types of errors are recorded
splogd	Logs SP hardware state changes, reports SP hardware errors, and provides a state change alert
supfilesrv	Provides services for file collections
switchtbld	Switch Table API daemon
sysctld	Sysctl daemon; runs on the control workstation, all nodes, and other RS/6000s
xntpd	Network time protocol daemon, keeps clocks synchronized across the network

The following AIX daemons may also run on the SP servers and processor nodes:

cron	Runs shell commands at specified intervals
errdemon	Standard AIX error daemon for recording error log information
inetd	Provides Internet server management for a network
named	Provides server functions for the Domain Name Protocol
nfsd,biod, statd, lockd	Standard NFS daemons
portmap	Provides **RPC** services
qdaemon	Schedules print jobs from the **enq** command
rlogind	Provides **rlogin** services
rpe.mountd	Responds to client NFS mount requests
rpestatd, rpe.lockd	Provide System V style of file and record locking with NFS
sendmail	Routes mail for local or network delivery

snmpd	Provides the agent functions of the simple network management protocol
srcmstr	System resource controller master daemon
syslogd	Logs system messages
telnetd	Provides **telnet** services
uprintfd	Constructs and writes kernel messages
writesrv	Allows the **write** command to send and receive messages
xmservd	Performance Toolbox
ypserv	Runs on NIS servers and slaves to provide NIS services
ypbind	Runs on all NIS clients to communicate with respective server

Appendix C. Sample Files

afsclient.cust

This is a sample of the **afsclient.cust** file.

```
| ##############################################################################
| #
| # Module: <afsclient.cust>
| #
| #--------------------------------------------------------------
| #
| # Description: This script is a sample which is designed to be used
| #            as a guide for users who may want to install AFS on SP nodes.
| #            This script is based on a premise that the Control Workstation (CW)
| #            has already been installed and activated as an AFS client.
| #
| #            This sample script is to be executed by "root" user on the SP node.
| #            This may be used with the /tftpboot/script.cust with modification
| #            based on the customer requirements and configuration.
| #
| ##############################################################################
| PATH=/bin:/usr/bin:/etc:/usr/sbin:/usr/bin/X11:/var/sysman:/var/sysman/etc:
| /etc/amd:/usr/lpp/ssp/bin:/usr/lpp/ssp/install/bin:/usr/afsws/etc:/usr/vice/etc:

| SERVER=‘cat /etc/ssp/server_hostname | cut -d" " -f3’

| ##############################################################################
| # Create the required AFS directory on /usr    filesystem. If the /usr is to
| # be used as client/server make sure the directories are installed on the
| # SP node server first, and then made available to the SP node clients.
| # Make sure you have adequate space (about 5   K) is required to load the
| # AFS execucatbles and files on the /usr filesystem.
| # We will create the directories on /usr, and then copy the required files
| # from the CW to the /usr/vice/etc directory on the SP node.
| # For our sample configuration we will reference the CW as "$SERVER"
| /usr/sbin/chfs    -a size='+1    ' /usr

| /bin/mkdir -p /usr/vice/etc/C
| /bin/mkdir /usr/vice/etc/dkload
| # Activate the ticket granting ticket so the remote copy will work successfully
| #to bring AFS files from the CW to the SP node.
| /usr/lpp/ssp/rcmd/bin/rcmdtgt

| # Copy the files from CW /usr/vice/etc directory to the SP node /usr/vice/etc
| # directory on the SP node.

| /usr/lpp/ssp/rcmd/bin/rcp $SERVER:/usr/vice/etc/CellServDB /usr/vice/etc/CellServDB
| /usr/lpp/ssp/rcmd/bin/rcp $SERVER:/usr/vice/etc/ThisCell      /usr/vice/etc/ThisCell
| /usr/lpp/ssp/rcmd/bin/rcp $SERVER:/usr/vice/etc/afsd        /usr/vice/etc/afsd
| /usr/lpp/ssp/rcmd/bin/rcp $SERVER:/usr/vice/etc/cacheinfo /usr/vice/etc/cacheinfo
| /usr/lpp/ssp/rcmd/bin/rcp $SERVER:/usr/vice/etc/C/  /usr/vice/etc/C/
| /usr/lpp/ssp/rcmd/bin/rcp $SERVER:/usr/vice/etc/dkload/  /usr/vice/etc/dkload/
| /usr/lpp/ssp/rcmd/bin/rcp $SERVER:/usr/vice/etc/rc.afs    /usr/vice/etc/rc.afs
```

© Copyright IBM Corp. 1996

```
| ###############################################################################
| #    This next step will setup the cache that will be used on the SP nodes
| #    These values can be modified, based on the amount of users that will be
| #    using afs services. We will use the same values as placed on the CW for our
| #    configuration being used in the sample script. The values to be modified are
| #    the filesystem size for /usr/vice/cache, and /usr/vice/etc/cacheinfo    size.
| #    This configuration will setup for a medium afs configuration of 4-8 users.
| #    We will use a cacheinfo of 8    K, and filesystem of 9    K.
| #    We will create the /usr/vice/cache directory and then create a JFS that will
| #   be  mounted
| #
| /bin/mkdir /usr/vice/cache

| /usr/sbin/crfs -v jfs -g rootvg -a size=18     -m '/usr/vice/cache' -A 'yes' -p 'rw'
| /bin/sleep 5
| /usr/sbin/mount /usr/vice/cache

| ###############################################################################
| #    This next step will setup the the afs as one of the supported filesystems
| #    in the /etc/vfs    file. You can either "edit" and place the information
| #   afs         4              none                                  none
| #    in the file, or append the information using the "echo" command.
| #

| /bin/echo "afs         4              none                                  none" >> /etc/vfs

| ###############################################################################
| #    This next step will setup the the afs directory, and then activiate the afs
| #    daemon (afsd) that connect to the AFS server. This will be executed by
| #    starting the /etc/rc.afs script.
| #
| #    The /etc/rc.afs will setup and execute the AFS libraries to be used as
| #    kernel extension in the SP nodes. You may want to use different execution
| #    variables based on you configuration and its use with nfs services
| #    We expect the files to be available in the /usr/vice/etc/dkload directory
| #
| #    /usr/vice/etc/dkload/cfgexport -a export.ext
| #   /usr/vice/etc/dkload/cfgafs   -a  afs.ext
| #
| ###############################################################################

| /bin/mkdir   /afs
| /usr/vice/etc/rc.afs

| /bin/sleep 6
| ###############################################################################
| #    This next step will setup a soft link of the /afs/<cellname>/@sys/usr/afsws
| #    directory to /usr/afsws on the SP node. The exact pathnames will need
| #    to reflect    the afs configuration cell name, and the system directory.
| #    This will be based on where the afsws executables are located.
| #
| #    /bin/ln -s /afs/<cellname>/@sys/usr/afsws    /usr/afsws
| #    My cell execution will be the following:
| /bin/ln -s /afs/ppd.pok.ibm.com/rs_aix41/usr/afsws      /usr/afsws

| ###############################################################################
| #    This next step will setup the itab entry for afs in the /etc/inittab file.
| #
```

```
/usr/sbin/mkitab "rcafs:2:wait:/usr/vice/etc/rc.afs>/dev/console 2>&1 /
# Start AFS daemon "

################################################################################
#    This next step will add the /usr/afsws/etc /usr/afsws/bin and the
#    /usr/vice/etc into your current path environment. You may want to
#    update the path in .profile, or ksh.rc or other login setup.
PATH=$PATH:/usr/afsws/etc:/usr/afsws/bin:/usr/vice/etc:
```

block_usr_sample

This script creates a file of user names in a column format called called /**tmp**/**usr.input**. The **usr.input** file can be used with the **spacs_cntrl** command. Once you are comfortable using Login Control, you can uncomment the **spacs_cntrl** command in this script.

```
#!/usr/lpp/ssp/perl/bin/perl
#########################################################
#  Description:
#
#     This sample script will build a file of users from the /etc/passwd
#     file to input to spacs_cntrl.
#
#     The following items should be checked and possibly changed to conform
#     to the policy at your site.
#
#     1)    uid threshhold to start adding users to the file.    Should be large
#           enough to prevent system users such as root, adm, bin, lpd, etc.
#           from being added to the file for spacs_cntrl.    The arbitary value
#           used is 125.
#
#     2)    flags for spacs_cntrl. The logging flag is issued to
#           allow you to follow the actions of spacs_cntrl.    You may wish to
#           remove this flag once you are familiar with running spacs_cntrl.
#           The -s flag suppresses error messages which are logged then -l is
#          issued.
#
#
#
#
#########################################################


# required files.
$usrfile="/tmp/usr.input";


$uidstart = 125;

$allokay =  ;
$syserror = 2;



# open the usr.input file

unless ( open(USRFILE,">$usrfile") ) {
          print "block_usr:    Cannot open usr.input file.\n";
         exit  ($syserror);
         }

# write any names with a uid of 125 or above into file

while ( ($uname,$passwd,$uid) = getpwent ) {
       if ( $uid >= $uidstart ) {
          print USRFILE $uname,"\n";
```

```
      }
}


close(USRFILE);

# make sure file has entries

if ( -z $usrfile ) {
     print "block_usr:    No entries in $usrfile.    Not executing spacs_cntrl.\n";
    exit($syserror);
}


# Uncomment the following for this script to automatically run the spacs_cntrl command
# and block users.
# issue spacs_cntrl to block users in the file.

#system "/usr/lpp/ssp/bin/spacs_cntrl -s -l -f $usrfile block";
#$rc=$?;
#if ( $rc !=   ) {
#      $rc = ($rc >> 8);
#      print "block_usr:    Possible error from spacs_cntrl.    Return code = $rc\n";
#     exit($rc);
#}

exit ($allokay)
```

bootptab.info File

If it exists, the /**etc**/**bootptab.info** file is read by the **sphrdwrad** command for hardware Ethernet addresses to be placed in the SDR. If it does not exist, then the **sphrdwrad** command still runs but takes longer to collect the information from the hardware.

Each line should list a SP node, specified by node number or frame,slot, followed by a blank and the hardware Ethernet address. The file should look similar to this:

```
17 02608C2E48D9
19 02608C2D6712
21 02608C2E49A4
23 02608C2E48E2
```

If you are an SP1 user, these hardware Ethernet addresses would have been in the /**etc**/**bootptab** file.

Using the frame,slot format, the file would look like this:

```
2,1 02608C2E48D9
2,3 02608C2D6712
2,5 02608C2E49A4
2,7 02608C2E48E2
```

ibmSPMIB.my

A copy of this MIB can be found in file **/usr/lpp/ssp/config/snmp_proxy/ibmSPMIB.my**. The file defines SP-specific information for the creation of SNMP traps for the SP from AIX Error Log and Event Management Events.

```
# Licensed Materials - Property of IBM
#
# 5765-529
#
# (C) Copyright IBM Corp. 1996 All Rights Reserved.
#
# US Government Users Restricted Rights - Use, duplication or disclosure
# restricted by GSA ADP Schedule Contract with IBM Corp.
#
# Script Name:    ibmSPMIB.my
#
# Description:
#          definition of the ibmSP mib.
#

IBMSP-MIB    DEFINITIONS ::= BEGIN

IMPORTS
            Counter, Gauge, TimeTicks, IpAddress, DisplayString, enterprises
  FROM  RFC1155-SMI
 TRAP-TYPE
  FROM  RFC1215;

ibm         OBJECT IDENTIFIER ::= { enterprises 2 }

ibmProd OBJECT IDENTIFIER ::= { ibm 6 }

ibmSP      OBJECT IDENTIFIER ::= { ibmProd 117 }

#-------------------------------------------------------------
#

#                       IBM SP MIB

#

ibmSPConfig OBJECT IDENTIFIER ::= { ibmSP 1 }

ibmSPhostnodenumber OBJECT-TYPE
 SYNTAX INTEGER
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
   "A number from   to the total number of slots.    It identifies
   the relative node number assigned to this processor."
 ::= { ibmSPConfig 1 }

ibmSPhostpartaddr OBJECT-TYPE
 SYNTAX IpAddress
 ACCESS    read-only
 STATUS mandatory
```

```
|    DESCRIPTION
|      "ip address assigned to the system partition in which this host
|      resides. If this host is acting as a control workstation (i.e.
|      the value of ibmSPhostnodenumber.  is  ), this will be the ip
|      address of the default partition."
|    ::= { ibmSPConfig 2 }

|   ibmSPCWScodeversion OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "The ssp release that is installed on the control workstation.
|       Values are of the form 2. , 2.1, etc. A NULL value means the
|       release is not known."
|    ::= { ibmSPConfig 3 }

|   ibmSPprimaryCWSname OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "hostname of the primary control workstation."
|    ::= { ibmSPConfig 4 }

|   ibmSPprimaryCWSoperstatus OBJECT-TYPE
|    SYNTAX INTEGER {
|     up(1),
|     down(2)
|     }
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "operational state of the primary control workstation."
|    ::= { ibmSPConfig 5 }

|   ibmSPbackupCWSname OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "hostname of the backup control workstation."
|    ::= { ibmSPConfig 6 }

|   ibmSPbackupCWSoperstatus OBJECT-TYPE
|    SYNTAX INTEGER {
|     up(1),
|     down(2)
|     }
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "operational state of the backup control workstation."
|    ::= { ibmSPConfig 7 }

|   ibmSPSystemTable OBJECT-TYPE
|    SYNTAX SEQUENCE OF IbmSPNodeEntry
|    ACCESS    not-accessible
```

```
    STATUS mandatory
    DESCRIPTION
      "A list of SPNodeEntrys."
    ::= { ibmSPConfig 8 }

  ibmSPNodeEntry OBJECT-TYPE
    SYNTAX IbmSPNodeEntry
    ACCESS    not-accessible
    STATUS mandatory
    DESCRIPTION
      "Identifies a processor node residing in an SP frame."
    INDEX { ibmSPpartitionaddr, ibmSPnodenumber }
    ::= { ibmSPSystemTable 1 }

  IbmSPNodeEntry ::=
    SEQUENCE {
      ibmSPpartitionaddr
        IpAddress,
      ibmSPnodenumber
        INTEGER,
      ibmSPframenumber
        INTEGER,
      ibmSPslotnumber
        INTEGER,
      ibmSPslotsused
        INTEGER,
      ibmSPinitialhostname
        DisplayString,
      ibmSPreliablehostname
        DisplayString,
      ibmSPsysparname
        DisplayString,
      ibmSPcodeversion
        DisplayString
      }

  ibmSPpartitionaddr OBJECT-TYPE
    SYNTAX IpAddress
    ACCESS    read-only
    STATUS mandatory
    DESCRIPTION
      "ip address assigned to the partition in which this node resides."
    ::= { ibmSPNodeEntry 1 }

  ibmSPnodenumber OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
      "A number from 1 to the total number of slots.    It identifies
      the relative node number assigned to the processor."
    ::= { ibmSPNodeEntry 2 }

  ibmSPframenumber OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
```

```
|     "A number from 1 to the number of frames.    It identifies
|     the number of the frame in which the processor node resides."
|   ::= { ibmSPNodeEntry 3 }

|   ibmSPslotnumber OBJECT-TYPE
|    SYNTAX INTEGER
|    ACCESS read-only
|    STATUS mandatory
|    DESCRIPTION
|      "A number from 1 to the number of slots in a frame.    It
|      identifies the number of the first slot occupied by the
|      processor node within the frame."
|   ::= { ibmSPNodeEntry 4 }

|   ibmSPslotsused OBJECT-TYPE
|    SYNTAX INTEGER
|    ACCESS read-only
|    STATUS mandatory
|    DESCRIPTION
|      "A number from 1 to two.    It identifies the number of slots
|      occupied by the processor."
|   ::= { ibmSPNodeEntry 5 }

|   ibmSPinitialhostname OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS read-only
|    STATUS mandatory
|    DESCRIPTION
|      "The initial hostname assigned to the node during the SP customization
|     phase."
|   ::= { ibmSPNodeEntry 6 }

|   ibmSPreliablehostname OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS read-only
|    STATUS mandatory
|    DESCRIPTION
|      "The hostname associated with the SP ethernet."
|   ::= { ibmSPNodeEntry 7 }

|   ibmSPsysparname OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS read-only
|    STATUS mandatory
|    DESCRIPTION
|      "The name of the system partition containing this processor
|     node."
|   ::= { ibmSPNodeEntry 8}

|   ibmSPcodeversion OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "The ssp release that is installed on the processor node.
|      Values are of the form 2. , 2.1, etc. A NULL value means the
|      release is not known."
|   ::= { ibmSPNodeEntry 9 }
```

```
|       #   The format of traps containing errlog entries whose tmplates
|       #   are defined with 'Alert=yes'.

|  #  ibmSPErrlogTrap  TRAP-TYPE
|  #    ENTERPRISE  ibmSP
|  #     VARIABLES    { ibmSPellabel, ibmSPelidentifier, ibmSPeldatetime,
|  #       ibmSPelsequencenum, ibmSPelmachineid, ibmSPelnodeid,
|  #       ibmSPelclass, ibmSPeltype, ibmSPelresource,
|  #       ibmSPelrscclass, ibmSPelrsctype, ibmSPellocation,
|  #       ibmSPelvpd, ibmSPetdescription, ibmSPetprobcauses,
|  #       ibmSPetusercauses, ibmSPetuseraction, ibmSPetinstcauses,
|  #       ibmSPetinstaction, ibmSPetfailcauses, ibmSPetfailaction,
|  #      ibmSPetdetaildata}
|  #         DESCRIPTION
|  #              "These traps contain the contents of errlog entries formatted
|  #              into objects defining the contents of the errlog entry.
|  #              Since any single errlog entry does not contain all of the
|  #              fields defined in the collection of errlog templates, when a
|  #              object contains a null value, it will not be included in the
|  #             trap."
|  #      ::=  ibmSPelidentifier.

|  ibmSPErrlogVars OBJECT-TYPE
|   SYNTAX SEQUENCE OF IbmSPErrlogEntry
|   ACCESS    not-accessible
|   STATUS mandatory
|   DESCRIPTION
|     "A single SPNodeErrlogEntry."
|   ::= { ibmSP 2 }

|  IbmSPErrlogEntry ::=
|   SEQUENCE {
|     ibmSPellabel
|       DisplayString,
|     ibmSPelidentifier
|       DisplayString,
|     ibmSPeldatetime
|       DisplayString,
|     ibmSPelsequencenum
|       DisplayString,
|     ibmSPelmachineid
|       DisplayString,
|     ibmSPelnodeid
|       DisplayString,
|     ibmSPelclass
|       DisplayString,
|     ibmSPeltype
|       DisplayString,
|     ibmSPelresource
|       DisplayString,
|     ibmSPelrscclass
|       DisplayString,
|     ibmSPelrsctype
|       DisplayString,
|     ibmSPellocation
|       DisplayString,
|     ibmSPelvpd
```

```
|       DisplayString,
|     ibmSPetdescription
|       DisplayString,
|     ibmSPetprobcauses
|       DisplayString,
|     ibmSPetusercauses
|       DisplayString,
|     ibmSPetuseraction
|       DisplayString,
|     ibmSPetinstcauses
|       DisplayString,
|     ibmSPetinstaction
|       DisplayString,
|     ibmSPetfailcauses
|       DisplayString,
|     ibmSPetfailaction
|       DisplayString,
|     ibmSPetdetaildata
|       DisplayString
|     }

|   ibmSPellabel OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "The label associated with the error identifier defined by
|      the ibmSPelidentifier object."
|    ::= { ibmSPErrlogVars 1 }

|   ibmSPelidentifier OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "A unique identifier defining the type of error entry written
|      to the system error log existing on the host from which the
|     trap  originated."
|    ::= { ibmSPErrlogVars 2 }

|   ibmSPeldatetime OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "A timestamp identifying the time that this error log entry
|      was written to the system error log.    It is of the form:
|      day     month     day_of_month     hour:min:sec"
|    ::= { ibmSPErrlogVars 3 }

|   ibmSPelsequencenum OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "A decimal number which is the sequence number assigned to this
|     entry.  This  is  the  value  specified  on  the  -l  switch  of  the
|      errpt command used to obtain the trap data on the host from
```

```
|       which the trap originated."
|     ::= { ibmSPErrlogVars 4 }

|   ibmSPelmachineid OBJECT-TYPE
|     SYNTAX DisplayString
|     ACCESS    read-only
|     STATUS mandatory
|     DESCRIPTION
|       "A decimal number which is the machine ID of the host on which
|       the trap originated. This is the value returned by the AIX
|       'uname -m' command when issued on the host from which the trap
|      originated."
|     ::= { ibmSPErrlogVars 5 }

|   ibmSPelnodeid OBJECT-TYPE
|     SYNTAX DisplayString
|     ACCESS    read-only
|     STATUS mandatory
|     DESCRIPTION
|       "An alpha-numeric string which is the node ID of the host on
|       which the trap originated.    This is the value returned by the AIX
|       'uname -m' command when issued on the host from which the trap
|      originated."
|     ::= { ibmSPErrlogVars 6 }

|   ibmSPelclass OBJECT-TYPE
|     SYNTAX DisplayString
|     ACCESS    read-only
|     STATUS mandatory
|     DESCRIPTION
|       "An alphabetic string which is the error class of this entry:
|       H (hardware), S (software), O (errlogger command messages)."
|     ::= { ibmSPErrlogVars 7 }

|   ibmSPeltype OBJECT-TYPE
|     SYNTAX DisplayString
|     ACCESS    read-only
|     STATUS mandatory
|     DESCRIPTION
|       "An alphabetic string which is the severity of the error entry:
|       'PEND' (impending loss of availability), 'PERF' (unacceptable
|       performance degradation), 'PERM' (permanent), 'TEMP' (temporary),
|      'UNKN'  (unknown)."
|     ::= { ibmSPErrlogVars 8 }

|   ibmSPelresource OBJECT-TYPE
|     SYNTAX DisplayString
|     ACCESS    read-only
|     STATUS mandatory
|     DESCRIPTION
|       "The resource name associated with the error. For hardware
|       errors this is a device name, for software errors this is
|       the name of the failing executable, for operator command
|       messages this is 'OPERATOR'."
|     ::= { ibmSPErrlogVars 9 }

|   ibmSPelrscclass OBJECT-TYPE
|     SYNTAX DisplayString
```

```
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "The resource class associated with the error. For hardware
|      errors this is a device class (or 'NONE')."
|    ::= { ibmSPErrlogVars 1  }

|   ibmSPelrsctype OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "The resource type associated with the error.    For hardware
|      errors this is a device type (or 'NONE'), for software errors
|      (when specified) this is an LPP."
|    ::= { ibmSPErrlogVars 11 }

|   ibmSPellocation OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "For hardware errors, information about the location of the
|      failing device (or 'NONE')."
|    ::= { ibmSPErrlogVars 12 }

|   ibmSPelvpd OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "IBM or user supplied vital product data."
|    ::= { ibmSPErrlogVars 13}

|   ibmSPetdescription OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|     "Error  description."
|    ::= { ibmSPErrlogVars 14}

|   ibmSPetprobcauses OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "Probable causes of the error."
|    ::= { ibmSPErrlogVars 15 }

|   ibmSPetusercauses OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "User actions which may have caused the error."
|    ::= { ibmSPErrlogVars 16 }
```

```
| ibmSPetuseraction OBJECT-TYPE
|  SYNTAX DisplayString
|  ACCESS    read-only
|  STATUS mandatory
|  DESCRIPTION
|    "Recommended actions the user may take to correct the error."
|  ::= { ibmSPErrlogVars 17 }

| ibmSPetinstcauses OBJECT-TYPE
|  SYNTAX DisplayString
|  ACCESS    read-only
|  STATUS mandatory
|  DESCRIPTION
|    "Installation causes of the error."
|  ::= { ibmSPErrlogVars 18 }

| ibmSPetinstaction OBJECT-TYPE
|  SYNTAX DisplayString
|  ACCESS    read-only
|  STATUS mandatory
|  DESCRIPTION
|    "User actions which may have been performed during installation
|    to cause the error."
|  ::= { ibmSPErrlogVars 19 }

| ibmSPetfailcauses OBJECT-TYPE
|  SYNTAX DisplayString
|  ACCESS    read-only
|  STATUS mandatory
|  DESCRIPTION
|    "A list of candidates which may be the source of the error."
|  ::= { ibmSPErrlogVars 2  }

| ibmSPetfailaction OBJECT-TYPE
|  SYNTAX DisplayString
|  ACCESS    read-only
|  STATUS mandatory
|  DESCRIPTION
|    "A list of reccommended actions which may be taken to correct
|    the possible failures."
|  ::= { ibmSPErrlogVars 21 }

| ibmSPeldetaildata OBJECT-TYPE
|  SYNTAX DisplayString
|  ACCESS    read-only
|  STATUS mandatory
|  DESCRIPTION
|    "Detailed data about this particular error."
|  ::= { ibmSPErrlogVars 22 }

| ibmSPEMVariables OBJECT IDENTIFIER ::= { ibmSP 3 }

| # The format of traps containing errlog entries whose tmplates
| # are defined with 'Alert=yes'.

| #  ibmSPEMEventTrap  TRAP-TYPE
| #  ENTERPRISE  ibmSP
|  ibmSPEMEventVarValue
```

```
|     DisplayString,
|   ibmSPEMEventPredicate
|     DisplayString
|   #   VARIABLES   {  ibmSPEMEventID,  ibmSPEMEventFlags,  ibmSPEMEventTime,
|   #     ibmSPEMEventLocation,  ibmSPEMEventPartitionAddress,
|   #     ibmSPEMEventVarsTableName,  ibmSPEMEventVarsTableInstanceID,
|   #     ibmSPEMEventVarName,  ibmSPEMEventVarValueInstanceVector,
|   #     ibmSPEMEventVarValuesTableInstanceID,
|   #     ibmSPEMEventVariableValue,  ibmSPEMEventPredicate,
|   #   DESCRIPTION
|   #       "These traps contain the contents of events generated from the
|   #       PSSP Event Manager.    The events have been formatted
|   #       into objects defining the contents of the event and where
|   #       the variable pertaining to the event and its value are located
|   #       in the ibmSP mib.
|   #   ::=  ibmSPEMEventID.

|   ibmSPEMEvent OBJECT-TYPE
|     SYNTAX SEQUENCE OF IbmSPEMEventEntry
|     ACCESS    not-accessible
|     STATUS mandatory
|     DESCRIPTION
|       "A single IbmSPEMEventEntry."
|     ::= { ibmSPEMVariables 1 }

|   ibmSPEMEventEntry ::=
|     SEQUENCE {
|       ibmSPEMEventID
|         INTEGER,
|       ibmSPEMEventFlags
|         INTEGER,
|       ibmSPEMEventTime
|         TimeTicks,
|       ibmSPEMEventLocation
|         INTEGER,
|       ibmSPEMEventPartitionAddress
|         IpAddress,
|       ibmSPEMEventVarsTableName
|         DisplayString,
|       ibmSPEMEventVarsTableInstanceID
|         DisplayString,
|       ibmSPEMEventVarName
|         DisplayString,
|       ibmSPEMEventVarValueInstanceVector
|         DisplayString,
|       ibmSPEMEventVarValuesTableInstanceID
|         DisplayString,
|       ibmSPEMEventVarValue
|         DisplayString,
|       ibmSPEMEventPredicate
|         DisplayString
|       }

|   ibmSPEMEventID OBJECT-TYPE
|     SYNTAX INTEGER
|     ACCESS    read-only
|     STATUS mandatory
|     DESCRIPTION
```

```
|      "The trap identifier assigned to the occurence of the Event."
|    ::= { ibmSPEMEvent 1}

|   ibmSPEMEventFlags OBJECT-TYPE
|    SYNTAX INTEGER {
|     re-arm(1),
|     false-predicate(2),
|     unregister-event(4)
|     }
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|     "EM  Flags."
|    ::= { ibmSPEMEvent 2}

|   ibmSPEMEventTime OBJECT-TYPE
|    SYNTAX TimeTicks
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "Elapsed time between the activation of the SP proxy sub-agent
|      and the occurence of the event."
|    ::= { ibmSPEMEvent 3}

|   ibmSPEMEventLocation OBJECT-TYPE
|    SYNTAX INTEGER
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "The number of the SP node from which the event was generated."
|    ::= { ibmSPEMEvent 4}

|   ibmSPEMEventPartitionAddress OBJECT-TYPE
|    SYNTAX IpAddress
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "The ip address of the SP partition from which the event was
|     generated."
|    ::= { ibmSPEMEvent 5}

|   ibmSPEMEventVarsTableName OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "The name of the table in the SP MIB in which contains the variable
|     definition  objects."
|    ::= { ibmSPEMEvent 6}

|   ibmSPEMEventVarsTableInstanceID OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "The instance identifier used to locate the row in the table named
|      by the ibmSPEMEventVarsTableName object value which contains
|      further information about the EM Event variable definition."
```

```
|    ::= { ibmSPEMEvent 7}

|   ibmSPEMEventVarName OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "The name of the EM variable    about which the event is recorded."
|    ::= { ibmSPEMEvent 8}

|   ibmSPEMEventVarValueInstanceVector OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "The instantiation vector of the EM variable instance that resulted
|      in the event."
|    ::= { ibmSPEMEvent 9}

|   ibmSPEMEventVarValuesTableInstanceID OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "The instance identifier used to locate the row in the
|      ibmSPEMVarValuesTable identifying for which instantiation of the named
|      EM variable the value is reported. This is used to obtain the current
|      variable value in the table to see if it has changed since the time
|      the trap was issued."
|    ::= { ibmSPEMEvent 1 }

|   ibmSPEMEventVarValue OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "The value of the variable instance at the time of the event."
|    ::= { ibmSPEMEvent 11}

|   ibmSPEMEventPredicate OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "The predicate string which caused the event."
|    ::= { ibmSPEMEvent 12}

|   ibmSPEMNodeDepVarsTable OBJECT-TYPE
|    SYNTAX SEQUENCE OF IbmSPEMNodeDepVarEntry
|    ACCESS    not-accessible
|    STATUS mandatory
|    DESCRIPTION
|      "A table of node-dependent EM Variable attributes. Variables
|      in this table are only instantiated on the node containing the
|      resource monitor. See the ibmSPEMNodeDepVarLocator object
|     description."
|    ::= { ibmSPEMVariables 2 }
```

```
| ibmSPEMNodeDepVarEntry OBJECT-TYPE
|  SYNTAX IbmSPEMNodeDepVarEntry
|  ACCESS    not-accessible
|  STATUS mandatory
|  DESCRIPTION
|    "The attributes of an event manager variable.    Each octet of
|    the variable-length index string is encoded in a separate
|    sub-identifier. The number of the sub-identifiers in the index
|   is  IMPLIED."
|  INDEX { ibmSPEMNodeDepVarName }
|  ::= { ibmSPEMNodeDepVarsTable 1 }

| IbmSPEMNodeDepVarEntry ::=
|  SEQUENCE {
|    ibmSPEMNodeDepVarName
|      DisplayString,
|    ibmSPEMNodeDepVarDescr
|      DisplayString,
|    ibmSPEMNodeDepVarType
|      DisplayString,
|    ibmSPEMNodeDepVarDataType
|      DisplayString,
|    ibmSPEMNodeDepVarSBSFormat
|      DisplayString,
|    ibmSPEMNodeDepVarInitValue
|      DisplayString,
|    ibmSPEMNodeDepVarCurValueIndex
|      INTEGER,
|    ibmSPEMNodeDepVarClass
|      DisplayString,
|    ibmSPEMNodeDepVarVecElDefn
|      DisplayString,
|    ibmSPEMNodeDepVarVecElDescr
|      DisplayString,
|    ibmSPEMNodeDepVarPTXName
|      DisplayString,
|    ibmSPEMNodeDepVarDefPred
|      DisplayString,
|    ibmSPEMNodeDepVarEventDescr
|      DisplayString,
|    ibmSPEMNodeDepVarLocator
|      DisplayString,
|    ibmSPEMNodeDepVarOrderGroup
|      DisplayString
|    }

| ibmSPEMNodeDepVarName OBJECT-TYPE
|  SYNTAX DisplayString
|  ACCESS    read-only
|  STATUS mandatory
|  DESCRIPTION
|    "A resource variable name as defined to the Event Manager."
|  ::= { ibmSPEMNodeDepVarEntry 1 }

| ibmSPEMNodeDepVarDescr OBJECT-TYPE
|  SYNTAX DisplayString
|  ACCESS    read-only
|  STATUS mandatory
```

```
|    DESCRIPTION
|      "A description of the variable, including its semantics."
|    ::= { ibmSPEMNodeDepVarEntry 2 }

|   ibmSPEMNodeDepVarType OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "One of the strings Counter, Quantity or State."
|    ::= { ibmSPEMNodeDepVarEntry 3 }

|   ibmSPEMNodeDepVarDataType OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "One of the strings long, float or SBS. SBS is only permitted
|      for a variable of the type State. If the value of this
|      object is SBS, then the definitions of the structured fields
|      that comprise the structured byte string contained in the
|     ibmSPEMNodeDepVarSBSFormat  object."
|    ::= { ibmSPEMNodeDepVarEntry 4 }

|   ibmSPEMNodeDepVarSBSFormat OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "If the ibmSPEMNodeDepVarDataType object with the corresponding
|      instance id has a value of SBS, then this object dscribes the
|      structured fields within the variable. Included for each structured
|      field is the structured field name, followed by an equal sign,
|      followed by the data type for the field. The structured fields
|      are defined sequentially beginning with sequence number  ."
|    ::= { ibmSPEMNodeDepVarEntry 5 }

|   ibmSPEMNodeDepVarInitValue OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "The initial value of a resource variable before it is observed
|      for the first time."
|    ::= { ibmSPEMNodeDepVarEntry 6 }

|   ibmSPEMNodeDepVarCurValueIndex OBJECT-TYPE
|    SYNTAX INTEGER
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "An index into the ibmSPEMVarValuesTable to locate value
|      instances for this variable."
|    ::= { ibmSPEMNodeDepVarEntry 7 }

|   ibmSPEMNodeDepVarClass OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
```

```
|    STATUS mandatory
|    DESCRIPTION
|      "The name of the class to which this resource variable belongs."
|    ::= { ibmSPEMNodeDepVarEntry 8 }

|   ibmSPEMNodeDepVarVecElDefn OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "The name(s) of instantiation vector elements associated with the
|     resource  variable."
|    ::= { ibmSPEMNodeDepVarEntry 9 }

|   ibmSPEMNodeDepVarVecElDescr OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "A description of the instantiation vector elements associated with
|      the resource variable."
|    ::= { ibmSPEMNodeDepVarEntry 1  }

|   ibmSPEMNodeDepVarPTXName OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "The name used to read and write the resource variable in the PTX
|     shared  memory."
|    ::= { ibmSPEMNodeDepVarEntry 11 }

|   ibmSPEMNodeDepVarDefPred OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "The default predicate to be applied to the resource variable."
|    ::= { ibmSPEMNodeDepVarEntry 12 }

|   ibmSPEMNodeDepVarEventDescr OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "A description of the event generated by the application of the
|     default  predicate."
|    ::= { ibmSPEMNodeDepVarEntry 13 }

|   ibmSPEMNodeDepVarLocator OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "The name of the vector element whose value is the number of the
|      node containing the variable instance."
|    ::= { ibmSPEMNodeDepVarEntry 14 }
```

```
| ibmSPEMNodeDepVarOrderGroup OBJECT-TYPE
|  SYNTAX DisplayString
|  ACCESS    read-only
|  STATUS mandatory
|  DESCRIPTION
|    "The name of an Availability group. All events generated by the
|    default predicates of resource variables within this order group
|    name are guaranteed ordered delivery with respect to one
|   another."
|  ::= { ibmSPEMNodeDepVarEntry 15 }

| ibmSPEMNodeIndepVarsTable OBJECT-TYPE
|  SYNTAX SEQUENCE OF IbmSPEMNodeIndepVarEntry
|  ACCESS    not-accessible
|  STATUS mandatory
|  DESCRIPTION
|    "A table of node-independent EM Variable attributes.    No EM Locator
|    element is defined for these variables so the location of the
|    resource monitor is unknown within the partition. Variables
|    in this table are only instantiated on the Control Work Station.
|    See the ibmSPEMNodeDepVarLocator object description."
|  ::= { ibmSPEMVariables 3 }

| ibmSPEMNodeIndepVarEntry OBJECT-TYPE
|  SYNTAX IbmSPEMNodeIndepVarEntry
|  ACCESS    not-accessible
|  STATUS mandatory
|  DESCRIPTION
|    "The attributes of an event manager variable.    Each octet of
|    the variable-length index string is encoded in a separate
|    sub-identifier. The number of the sub-identifiers in the index
|   is  IMPLIED."
|  INDEX { ibmSPEMNodeIndepPartaddr, ibmSPEMNodeIndepVarName }
|  ::= { ibmSPEMNodeIndepVarsTable 1 }

| IbmSPEMNodeIndepVarEntry ::=
|  SEQUENCE {
|   ibmSPEMNodeIndepPartaddr
|     IpAddress,
|   ibmSPEMNodeIndepVarName
|     DisplayString,
|   ibmSPEMNodeIndepVarDescr
|     DisplayString,
|   ibmSPEMNodeIndepVarType
|     DisplayString,
|   ibmSPEMNodeIndepVarDataType
|     DisplayString,
|   ibmSPEMNodeIndepVarSBSFormat
|     DisplayString,
|   ibmSPEMNodeIndepVarInitValue
|     DisplayString,
|   ibmSPEMNodeIndepVarCurValueIndex
|     INTEGER,
|   ibmSPEMNodeIndepVarClass
|     DisplayString,
|   ibmSPEMNodeIndepVarVecElDefn
|     DisplayString,
|   ibmSPEMNodeIndepVarVecElDescr
```

```
|       DisplayString,
|     ibmSPEMNodeIndepVarPTXName
|       DisplayString,
|     ibmSPEMNodeIndepVarDefPred
|       DisplayString,
|     ibmSPEMNodeIndepVarEventDescr
|       DisplayString,
|     ibmSPEMNodeIndepVarOrderGroup
|       DisplayString
|     }

|   ibmSPEMNodeIndepPartaddr OBJECT-TYPE
|    SYNTAX IpAddress
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "ip address assigned to the system partition in which the
|      this resource variable resides."
|    ::= { ibmSPEMNodeIndepVarEntry 1 }

|   ibmSPEMNodeIndepVarName OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "A resource variable name as defined to the Event Manager."
|    ::= { ibmSPEMNodeIndepVarEntry 2 }

|   ibmSPEMNodeIndepVarDescr OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "A description of the variable, including its semantics."
|    ::= { ibmSPEMNodeIndepVarEntry 3 }

|   ibmSPEMNodeIndepVarType OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "One of the strings Counter, Quantity or State."
|    ::= { ibmSPEMNodeIndepVarEntry 4 }

|   ibmSPEMNodeIndepVarDataType OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "One of the strings long, float or SBS. SBS is only permitted
|      for a variable of the type State. If the value of this
|      object is SBS, then the definitions of the structured fields
|      that comprise the structured byte string contained in the
|     ibmSPEMNodeIndepVarSBSFormat  object."
|    ::= { ibmSPEMNodeIndepVarEntry 5 }

|   ibmSPEMNodeIndepVarSBSFormat OBJECT-TYPE
|    SYNTAX DisplayString
```

```
    ACCESS    read-only
    STATUS mandatory
    DESCRIPTION
      "If the ibmSPEMNodeIndepVarDataType object with the corresponding
      instance id has a value of SBS, then this object describes the
      structured fields within the variable. Included for each structured
      field is the structured field name, followed by an equal sign,
      followed by the data type for the field. The structured fields
      are defined sequentially beginning with sequence number  ."
    ::= { ibmSPEMNodeIndepVarEntry 6 }

  ibmSPEMNodeIndepVarInitValue OBJECT-TYPE
    SYNTAX DisplayString
    ACCESS    read-only
    STATUS mandatory
    DESCRIPTION
      "The initial value of a resource variable before it is observed
      for the first time."
    ::= { ibmSPEMNodeIndepVarEntry 7 }

  ibmSPEMNodeIndepVarCurValueIndex OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS    read-only
    STATUS mandatory
    DESCRIPTION
      "An index into the ibmSPEMVarValuesTable table to locate value
      instances for this variable."
    ::= { ibmSPEMNodeIndepVarEntry 8 }

  ibmSPEMNodeIndepVarClass OBJECT-TYPE
    SYNTAX DisplayString
    ACCESS    read-only
    STATUS mandatory
    DESCRIPTION
      "The name of the class to which this resource variable belongs."
    ::= { ibmSPEMNodeIndepVarEntry 9 }

  ibmSPEMNodeIndepVarVecElDefn OBJECT-TYPE
    SYNTAX DisplayString
    ACCESS    read-only
    STATUS mandatory
    DESCRIPTION
      "The name(s) of instantiation vector elements associated with the
     resource  variable."
    ::= { ibmSPEMNodeIndepVarEntry 1  }

  ibmSPEMNodeIndepVarVecElDescr OBJECT-TYPE
    SYNTAX DisplayString
    ACCESS    read-only
    STATUS mandatory
    DESCRIPTION
      "A description of the instantiation vector elements associated with
      the resource variable."
    ::= { ibmSPEMNodeIndepVarEntry 11 }

  ibmSPEMNodeIndepVarPTXName OBJECT-TYPE
    SYNTAX DisplayString
    ACCESS    read-only
```

```
|    STATUS mandatory
|    DESCRIPTION
|      "The name used to read and write the resource variable in the PTX
|     shared  memory."
|    ::= { ibmSPEMNodeIndepVarEntry 12 }

|   ibmSPEMNodeIndepVarDefPred OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "The default predicate to be applied to the resource variable."
|    ::= { ibmSPEMNodeIndepVarEntry 13 }

|   ibmSPEMNodeIndepVarEventDescr OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "A description of the event generated by the application of the
|     default  predicate."
|    ::= { ibmSPEMNodeIndepVarEntry 14 }

|   ibmSPEMNodeIndepVarOrderGroup OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|      "The name of an Availability group. All events generated by the
|      default predicates of resource variables within this order group
|      name are guaranteed ordered delivery with respect to one
|     another."
|    ::= { ibmSPEMNodeIndepVarEntry 15 }

|   ibmSPEMVarValuesTable OBJECT-TYPE
|    SYNTAX SEQUENCE OF IbmSPEMVarValuesEntry
|    ACCESS    not-accessible
|    STATUS mandatory
|    DESCRIPTION
|      "A table of current EM Variable values. Variables."
|    ::= { ibmSPEMVariables 4 }

|   ibmSPEMVarValuesEntry OBJECT-TYPE
|    SYNTAX IbmSPEMVarValuesEntry
|    ACCESS    not-accessible
|    STATUS mandatory
|    DESCRIPTION
|      "The current value of an event manager variable instantiation.
|      The value of the ibmSPEMVarValueIndex is assigned when the
|      SP sub-agent (sp_configd) is initialized; it is contained in the
|      ibmSPEMNodeDepVarCurValueIndex object within the ibmSPNodeDepVarsTable
|      (if the variable is node-dependent) or in the
|      ibmSPEMNodeIndepVarCurValueIndex object within the
|      ibmSPEMNodeIndepVarTable (if the variable is node-independent).
|      Each octet of the variable-length ibmSPEMVarValueInstanceVector value
|      string is encoded in a separate sub-identifier, preceded by its
|      length which may be   if its value is null. A   length indicates
|      the function represented by the EM variable is not being monitored."
```

```
|    INDEX { ibmSPEMVarValueIndex, ibmSPEMVarValueInstanceVector }
|    ::= { ibmSPEMVarValuesTable 1 }

|   IbmSPEMVarValuesEntry ::=
|    SEQUENCE {
|      ibmSPEMVarValueIndex
|        INTEGER,
|      ibmSPEMVarValueInstanceVector
|        DisplayString,
|      ibmSPEMVarValuePartaddr
|        IpAddress,
|      ibmSPEMVarValueName
|        DisplayString,
|      ibmSPEMVarValue
|        DisplayString
|      }

|   ibmSPEMVarValueIndex OBJECT-TYPE
|    SYNTAX INTEGER
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|       "An index used as part of the instance id to identify the
|       object instance containing the current value of an EM
|       resource variable instance "
|    ::= { ibmSPEMVarValuesEntry 1}

|   ibmSPEMVarValueInstanceVector OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|       "The instantiation vector elements associated with the
|      resource  variable."
|    ::= { ibmSPEMVarValuesEntry 2}

|   ibmSPEMVarValuePartaddr OBJECT-TYPE
|    SYNTAX IpAddress
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|       "ip address assigned to the system partition in which the
|       this resource variable resides."
|    ::= { ibmSPEMVarValuesEntry 3}

|   ibmSPEMVarValueName OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
|       "A resource variable name as defined to the Event Manager."
|    ::= { ibmSPEMVarValuesEntry 4}

|   ibmSPEMVarValue OBJECT-TYPE
|    SYNTAX DisplayString
|    ACCESS    read-only
|    STATUS mandatory
|    DESCRIPTION
```

```
|       "The current value of a resource variable."
|    ::= { ibmSPEMVarValuesEntry 5}

|  END
```

jmd_config

This is a sample of the **jmd_config** file. You can find it in **/etc/**.

```
# General rules:
# 1) Comment lines start with a # in the first column.
# 2) All keywords must start in the first column of the file.
# 3) To continue a line put \ at the end of the line to be continued.
# 4) If a keyword has several values they must be separated by ";".
# 5) Any hostnames specified here must have a
#       corresponding IP address and node number in the SDR.
# 6) All blank lines must have "\n" (carriage return) in the first column.
#       Any lines with "\n" preceded by one or more blanks will cause a syntax
#      error.
#
# This file contains an example of an SP2 with sixteen wide nodes.
#
# JM_LIST contains the list of SP2 hostnames that are candidates to run
# the JM server daemon.    This list will be searched in the order specified.
#
# The JM_LIST keyword can be repeated on subsequent lines and the
# lists will be concatenated.
JM_LIST=r 7n 1;r 7n 3;r 7n 5

# ACCESS_CONTROL specifies if the system administrator wants the JM to use the
# access control system management tool (/usr/lpp/ssp/bin/spacs_cntrl)
# to allow user access to a parallel node while it is allocated.
# If not specified resource manager will default to no.
# The ACCESS_CONTROL keyword can be repeated on subsequent lines but the
# last value will be the only one used.
# Specify    <yes or no>
ACCESS_CONTROL = yes

| # EN_ADAPTER selects the ethernet adapter network that will
| # be used to allocate resources for job application requests.
| # Valid values for EN_ADAPTER are either   or 1 for en  or en1,
| # respectively.    The default value is   for en .    The EN_ADAPTER
| # value will determine which adapter is used regardless of what
| # the node names are for the Pool members.    Specify <  or 1>.
| EN_ADAPTER =

# General rules for pools:
# 1) To describe a pool, the POOL_ID and ATTRIBUTES keywords have to be
#       specified. If you specify none of the MEMBERS_INTERACTIVE,
#       MEMBERS_BATCH, MEMBERS_GENERAL keywords (i.e. the pool has no
#       member nodes) then the pool will be ignored.
# 2) POOL_ID must be an unique integer greater than or equal to -1.
# 3) ATTRIBUTES must be a string with no blanks or "="
#       within the string and less than 1 24 characters.
# 4) Any of the MEMBERS_INTERACTIVE, MEMBERS_BATCH, or MEMBERS_GENERAL,
#       keywords can be repeated on subsequent lines and the
#       corresponding lists will be concatenated.
# 5) Each node on the SP2 can be in only one pool and only one subpool
#       within that pool.    Since different IP hostnames can be used
#       to refer to the same node, you will receive an error if you
#       use different names referring to the same node within
#      multiple  pools.
```

```
#
# Serial pool:
#
#    There can only be one serial pool. This information is defined
#    here so that the system administrator can record her decision
#    on where the batch serial jobs run (serial batch nodes), and
#    where the users are allowed to log into (serial interactive
#    nodes). Serial nodes that can be batch and interactive are put
#    into the general subpools. For the sake of distinguishing
#    between serial and parallel pools, the serial pool is required
#    to have a pool_id of -1.    The resource manager does not allocate
#    nodes in the serial pool.

POOL_ID= -1
ATTRIBUTES = This_is_the_serial_test_pool
# In this example:
#      r 7n 1 will be where the users are allowed to log into,
#      r 7n 3 will be where the batch serial jobs run,
#      r 7n 5 can be used for either of the above reasons.
MEMBERS_INTERACTIVE = r 7n 1
MEMBERS_BATCH            =  r 7n 3
MEMBERS_GENERAL         =  r 7n 5

# Parallel pools:
#
#    There can be several parallel pools, each pool having its own
#   characteristics.  The  Resource  Manager  allocates  nodes  from  the
#    interactive subpools for use by interactive parallel jobs and
#    from the batch subpools for use by jobs submitted to batch
#   schedulers  like
#   LoadLeveler.  Nodes  will  be  allocated  from  the  general  subpools
#    if there are no nodes available from the more specific
#   subpools.

POOL_ID =
ATTRIBUTES = IP_pool

# In this example:
#      r 7n 7    will be allocated to interactive parallel jobs,
#      r 7n 9 and r 7n11 will be allocated to jobs submitted to a batch
#                  scheduler (like LoadLeveler).
#      r 7n 13, r 7n15, r 7n17 can be used for either of the above reasons.

MEMBERS_INTERACTIVE = r 7n 7;
MEMBERS_BATCH      = r 7n 9; \
                           r 7n11;
MEMBERS_GENERAL = r 7n13;
MEMBERS_GENERAL = r 7n15;
MEMBERS_GENERAL = r 7n17;

POOL_ID = 1

# In this example:
#      r 7n19 will be allocated to interactive parallel jobs,
#      r 7n21 will be allocated to jobs submitted to a batch
#                  scheduler (like LoadLeveler).
#      r 7n23-r 7n31 can be used for either of the above reasons.
```

```
ATTRIBUTES = user_space_pool
MEMBERS_INTERACTIVE = r 7n19
MEMBERS_BATCH = r 7n 21;
MEMBERS_GENERAL = r 7n23;r 7n25;r 7n27;r 7n29;r 7n31;
```

script.cust File

The **script.cust** file is located in **/usr/lpp/ssp/samples**.

```
#!/bin/ksh
#####################################################################
#
# Module: <script.cust>
#
#--------------------------------------------------------------------
#
# Description: This script is a sample which is designed to be used by the
#       user as a guide for some functions that might be performed
#            after a node has been netinstalled, but before it boots for the
#           first  time.
#
#       If you want    to use some of the examples in this file, then you need
#       to either copy the entire script to /tftpboot/script.cust and
#       edit the lines you want to execute, or just copy over the portions
#      of this sample that you want executed to /tftpboot/script.cust
#
#
#####################################################################

PATH=/bin:/usr/bin:/etc:/usr/sbin:/usr/bin/X11:/var/sysman:
/var/sysman/etc:/etc/amd:/usr/lpp/ssp/bin:/usr/lpp/ssp/install/bin:
SSPDIR=/usr/lpp/ssp
SERVER= cat /etc/ssp/server_hostname | cut -d" " -f3
#
# Set the time zone
# use smit chtz to see the choices for time zones
# (EST5)                  Eastern U.S.; Colombia    - daylight time unobserved
# (EST5EDT)               Eastern U.S.; Colombia    - daylight time observed
#chtz EST5EDT
#
#####################################################################
# Importing existing volume groups
#####################################################################
#
#importvg -y uservg hdisk1
#
# To activate a volume group use varyonvg.
#varyonvg uservg
#
#####################################################################
# defining name resolution
# name resolution is required on the nodes. If you have not created an
# image that contains a method for resolving hostnames and addresses, then
# one of these methods must be selected.
#####################################################################
#
#####################################################################
# To perform kerberos authenticated rcp use the ssp rcmdtgt to obtain
# a kerberos ticket first.
#####################################################################
#
# Define the file that will store the ticket
```

```
#export KRBTKFILE=/tmp/tkt_rcmd$$
#
# Get the kerberos ticket
#/usr/lpp/ssp/rcmd/bin/rcmdtgt
#
# If you are using /etc/resolv.conf to resolve hostnames and addresses
# then use this to copy the /etc/resolv.conf file from the boot/install server
#
# Perform kerberos authenticated rcp
#/usr/lpp/ssp/rcmd/bin/rcp $SERVER:/etc/resolv.conf /etc/resolv.conf
#
# If you are using the /etc/hosts file to resolve hostnames and addresses
# then use the following commands to copy and append the /etc/hosts
# file from the boot/install server
#
# Perform kerberos authenticated rcp
#/usr/lpp/ssp/rcmd/bin/rcp $SERVER:/etc/hosts /tmp/hosts
#/bin/cat /tmp/hosts >>/etc/hosts
#/bin/rm /tmp/hosts
#
# Remove the ticket file
#/bin/rm $KRBTKFILE
#
# If you are using NIS to resolve hostnames and addresses then use the
# next section to define you NIS environment
#
###############################################################################
# defining nis domains, servers, and clients
###############################################################################
#
# To set the NIS domainname use the domainname command.
# To list the NIS domainname, issue domainname with no parameters and it
# will list the NIS domainname.
#
#/usr/etc/yp/chypdom '-B' <NIS domainname>
#
# Make this node an NIS server
# note: replace <NIS mastername> with the hostname of your NIS master
# note: issue: chmaster -s <node hostname>
# on the NIS master to add this node as an NIS server prior to issuing the
# mkslve command on the node.
#
#mkslave <NIS mastername>
#
# Make this node an NIS client
#
#mkclient
#
###############################################################################
# installing other lpp's
# If you installed other lpp's on your system, made a mksysb image of that
#    system, and are planning to install /usr clients from that mksysb image,
#    you do not need to issue any installp commands here related to those
#  lpp's.  The /usr parts will be available on your /usr server (assuming
#    you installed the server with the same mksysb image), and the root parts
#    will get updated automatically by the finishrclient script, which is
#    run automatically after every /usr client is installed.
###############################################################################
```

```
#
# To install Fortran on a node create a backup image of the installp in
# the /usr/sys/inst.images/ssp directory of the control workstation and
# name it xlf.installp (make sure the installp image has read and execute
# permissions by all - chmod +rx xlf.installp).
#
# Check to see if fortran is already installed on this system
#if [[ -z  lslpp -h | grep xlf  ]]
#then
#    # Fortran not installed - Mount the installp image and perform the installp
|   #    /usr/sbin/mount <control workstation address>:/usr/sys/inst.images/ssp /mnt
#    [[ -f /mnt/ssp.installp ]] && installp -ac -X -d/mnt/xlf.installp all
#
#    # Unmount the installp image
|   #   /usr/sbin/unmount  /mnt
#fi
#
################################################################################
# Modifying page space
################################################################################
#
#
# To make a new logical volume mypage for page space with 5 logical
# partitions (4 MB per partition - 2  MB total) on the volume group vg 4,
# issue the mklv command.
# -t specifies the logical volume type (jfs, jfslog, or paging)
# -y specifies the name of the logical volume to be used instead of the system
# generated logical volume name(pagelvXX - where XX starts at    and increase
# by one).
# Use lsps -a to list all available paging space and its status.
#
#mklv -t paging -y mypage vg 4 5
#
# To cause the /dev/mypage device to become available, use the swapon command.
#
#swapon /dev/mypage
#
# To change page space to be active on the next and subsequent reboots
# us the chps command.
# -a specifies to use page spae at next reboot
# y specifies to use page space at subsequent reboots
#
#chps -a y mypage
#
# Change the paging space - to add 1   logical partitions (4   MB )
# to mypage page space
# note: The default logical partition size is 4 MB.
# note: To use smit to change page space issue: smit chps
#
# Sample check to see if we are greater than what we are asking for
# NOTE: read comments for increasing logical volume space below
#
#num_pps_tox=1                  # number of PPs to extend page space by
#
## determine cur size of mypage in PPs
|   #pps= /usr/sbin/lsps -c mypage | tail +2 | cut -d":" -f4
#if (( pps < num_pps_tox ))
#then
```

```
#       chps -s$num_pps_tox mypage
#fi
#
# Create a 4  logical partition (default 4 MB partions - 16  MB) page space
# on the root volume group
# -a option actviates page space on subsequent reboots
# -n option actviates page space immediately
# -s option is the size of the page space in partitions
#
#mkps -s'4 ' -n -a rootvg
#
################################################################################
# increasing logical volume space
#
# Code below is a sample test for increasing LV one time and for a large
# amount... If you need to extend for small amounts, uncomment just
# the extendlv command line or do the extend manually, but remember to
# comment it out for subsequent customizes; otherwise the LV will be
# extended by that amount on ALL customizes.
#
# Assumption: LV is a minimal size (4MB = 1PPs) and want it to be 16 MB
# at initial install time ...
################################################################################
#
## increasing logical volume space 4  logical partition
## (default 4 MB partions - 16  MB)
## pps -> physical partitions equate to logical partitions when LV is COPY=1
#
#num_pps_tox=1                          # number of PPs to extend LV by
#
## determine current number PPs in hd3
#cur_pps_str= lslv hd3 | grep "¬LPs" | cut -d":" -f2
#cur_pps=${cur_pps_str%%PP }
#
#if (( num_pps_tox >= cur_pps ))      # already that big
#then
#       extendlv hd3 $num_pps_tox               # extend by 1
#fi
#
#
################################################################################
# increasing a filesystem size
################################################################################
# NOTE: read comments above on increasing logical volume space ...
#
## increasing the /var filesystem by 4 MB
#
##check to see if we did the increase once, by checking size > extend amount
#
#increase=8192          # specified in 512 byte blocks; represents 4M increase
#fssize= /usr/sbin/lsfs -c /var | tail +2 | cut -d":" -f6
#if (( fssize < increase ))
#then
#       chfs -a size=+$increase /var
#fi
#
#
################################################################################
```

```
#
################################################################################
# changing the maximum number of of users that can be concurrently logged
# in according to the one-time AIX license fee for the system. The change is
# put into effect after the next restart of the system.
################################################################################
#
#chlicense -u3
#
################################################################################
#
# To set the allcast attribute of a Token Ring Network Interface to
# not confine broadcast to the local LAN:
#
# chdev -l tr  -a allcast=off -P
################################################################################
################################################################################
#
# Copy user management files from server to node.
# These are only some of the files that are copy over.
# The server address is gotten from the /etc/ssp/server_hostname file.
# If the server can not get to the correct files (passwd, group, etc) then
# place the correct ip address of where those files are in place of $SERVER.
#      (e.g.) /usr/bin/rcp -p 129.4 .62.11:/etc/passwd /etc/passwd
# Check the /var/adm/SPlogs/sysman/<hostname>.config.log log file for errors
#
################################################################################

# /usr/bin/rcp -p $SERVER:/etc/passwd /etc/passwd
# /usr/bin/rcp -p $SERVER:/etc/group /etc/group
# /usr/bin/rcp -p $SERVER:/etc/security/passwd /etc/security/passwd
# /usr/bin/rcp -p $SERVER:/etc/security/group /etc/security/group
# /usr/bin/rcp -p $SERVER:/etc/security/user /etc/security/user
```

tuning.cust File

The **tuning.cust** file is located in **/usr/lpp/ssp/samples**.

```
#!/bin/ksh
#"@(#)96    1.2    src/ssp/samples/tuning.cust, sysman, ssp_r2.4 6/12/95 19:55:43"
#########################################################################
#
# Module: <tuning.cust>
#
#-------------------------------------------------------------------
#
# Description: This script is a sample which is designed to be used by the
#       user as a guide to change the network options for TCP/IP and UDP/IP
#     communications on the SP system
#
#       If you want to use some of the examples in this file, then you need
#       to either copy the entire script to /tftpboot/tuning.cust and
#       edit the lines you want to execute, or just copy over the portions
#      of this sample that you want executed to /tftpboot/tuning.cust
#
#     These settings are only a sample and should be changes to the preferred
#     settings for your system.
#
#     For further recommendations on tuning, refer to the Administration Guide
#
#-------------------------------------------------------------------

#
#     Set default mbuf pool size for all TCP/IP and thread buffer
#
#    /usr/sbin/no -o thewall=16384

#
#     Set default sb_max size for all IP connections
#
#    /usr/sbin/no -o sb_max=16384

#
#     Set default TCP send window size for all TCP/IP connections
#
#    /usr/sbin/no -o tcp_sendspace=65536

#
#     Set default TCP receive window size for all TCP/IP connections
#
#    /usr/sbin/no -o tcp_recvspace=65536

#
#     Set default UDP send window size for all UDP/IP connections
#     Effective maximum is limited to 65536
#
#    /usr/sbin/no -o udp_sendspace=32768

#
#     Set default UDP receive window size for all UDP/IP connections
```

```
#
# /usr/sbin/no -o udp_recvspace=65536

#
#   Set rfc1323 feature on for faster throughput for High Performance Networks.
#
#  /usr/sbin/no -o rfc1323=1

#
#   Set default Maximum Transmission Unit size for transfers across networks.
#
# /usr/sbin/no -o tcp_mssdflt=1448
#   Enable the kernel to forward packets
# /usr/sbin/no -o ipforwarding=1
```

/usr/lib/X11/app-defaults/Spmon

```
! "@(#)37     1.2     src/ssp/spmon/gui/Spmon, spmon, ssp_r2.1 9/16/94 1 :29:33" :
! X resource file for the spmon -g or spmon_gui commands:
! tcl wants colon : in comment lines too
! There arent many fonts common across all servers:
Tk font: 8x13bold
Dialog font: 8x13bold
```

Appendix D. System Monitor Variables

These variables can be queried from the **spmon** command interface, the **hmmon** command interface, or from the System Monitor GUI **spmon -g**.

Notes:

1. Responses provided (in parentheses) with descriptions refer to **spmon** command values. Responses provided [in brackets] refer to **hmmon** command values. Colors refer to the System Monitor GUI.
2. On the System Monitor GUI in the All Node Summary, not all variables are valid for all node types. When a variable is not supported by the target node, that variable is blanked out.

Use the **hmmon** command to determine the type of frame, node, and switch.

For example, to determine the frame type for Frame 1:

```
hmmon -G -Q -s -v type 1:
   1        type              18     x883a    supervisor  type
```

To determine the node type in Frame 1, slot 4:

```
hmmon -Q -s -v type 1:4
   1    4    type              97     x943a    supervisor  type
```

or

```
hmmon -G -Q -s -v type 1:4
   1    4    type              97     x943a    supervisor  type
```

To determine the switch type in Frame 1, slot 17:

```
hmmon -G -Q -s -v type 1:17
   1    17   type              49     x8c3a    supervisor  type
```

or to determine the switch type in Frame 11, slot 2:

```
hmmon -G -Q -s -v type 11:2
   11   2    type              49     x8c3a    supervisor  type
```

Refer to the *IBM Parallel System Support Programs for AIX: Command and Technical Reference* for more information on the **hmmon** command.

Frame Hardware Variables Common to All Frame Supervisors

Variable Name	Display Type	Description
codeVersion	Constant	Supervisor microcode version number
controllerResponds	Boolean	Indicator that supervisor is responding to request and commands from host. (1 [True] (green) = okay, 0 [False] (red) = no response)
daemonPollRate	Integer	Daemon poll rate in seconds
diagByte	Integer	Diagnosis byte. Correlates with error indicators. This variable is not currently being used.

© Copyright IBM Corp. 1996

Variable Name	Display Type	Description
nodefail 1 ⋮ nodefail 16	Boolean	Loss of communication from frame supervisor to specified node supervisor (1 [True] (red) = communication loss, 0 [False] (green) = communication okay)
nodefail17	Boolean	Loss of communication from frame supervisor to switch supervisor (1 [True] (red) = communication loss, 0 [False] (green) = communication okay)
nodeLinkOpen 1-16	Multistate	Node serial link1 open (1 [True]=open, 0 [False]=closed)
rs232CTS	Boolean	Indicator that the control node has received a Clear To Send signal over the RS232 link (1[True]=Clear To Send signal received, 0[False]=Clear To Send signal not received)
rs232DCD	Boolean	Indicator that the control node has received a Carrier Detect signal over the RS232 link (1[True]=Carrier Detect signal received, 0[False]=Carrier Detect signal not received)
serialNum	Constant	Supervisor serial number. This variable is not currently being used.
temp	Celsius	Temperature
tempRange	Boolean	Temperature out of range (1 [True] = temperature out of range, 0 [False] = temperature within range)
timeTicks	Integer	Frame Supervisor “heartbeat” counter (in numbers of 100 ms tics)
type	Constant	Frame controller card type (in decimal)
voltP48	Volts	Voltage of +48 volt line
voltP48Range	Boolean	+48 voltage out of range (1 [True] = voltage out of range, 0 [False] = voltage within range)

Frame Hardware Variables Unique to Card Type 17 and 20 Frame Supervisor

Variable Name	Display Type	Description
frACLED	Multistate	Represents the bottom green frame supervisor LED, showing status of AC power (actually monitors +12VDC present at frame supervisor) (2/3 [2] (gray) = LED blinking: AC Power failure, 1 [1] (green) = LED lit: AC power okay, 0 [0] (black) = LED not lit: no AC power exists)
frDCLED	Multistate	Represents of the top green frame supervisor LED, showing status of frame AC-to-DC power supplies. (2/3 [2] (gray) = LED blinking: DC Power failure, 1 [1] (green) = LED lit: DC power okay, 0 [0] (black) = LED not lit: no DC power exists)
frPowerDC_A	Boolean	Status of DC power supply A (1 [True] (green) = okay, 0 [False] (red) = not okay)
frPowerDC_B	Boolean	Status of DC power supply B (1 [True] (green) = okay, 0 [False] (red) = not okay)
frPowerDC_C	Boolean	Status of DC power supply C (1 [True] (green) = okay, 0 [False] (red) = not okay)
frPowerDC_D	Boolean	Status of DC power supply D (1 [True] (green) = okay, 0 [False] (red) = not okay)

Variable Name	Display Type	Description
frPowerOff_A	Boolean	Power switch for power supply A. Use the **-power** parameter on the **spmon** command to turn the power on or off. The frame supervisor returns values indicating whether the power is on or off. (1[True]=off, 0[False]=on)
frPowerOff_B	Boolean	Power switch for power supply B. Use the **-power** parameter on the **spmon** command to turn the power on or off. The frame supervisor returns values indicating whether the power is on or off. (1[True]=off, 0[False]=on)
frPowerOff_C	Boolean	Power switch for power supply C. Use the **-power** parameter on the **spmon** command to turn the power on or off. The frame supervisor returns values indicating whether power is on or off. (1[True]=off, 0[False]=on)
frPowerOff_D	Boolean	Power switch for power supply D. Use the **-power** parameter on the **spmon** command to turn the power on or off. The frame supervisor returns values indicating whether power is on or off. (1[True]=off, 0[False]=on)

Frame Hardware Variables Unique to SEPBU (Type 18) and HACWS SEPBU (Type 22) Frame Supervisors

Variable Name	Display Type	Description
amp_A	Amps	Current output of DC Module A
amp_ARange	Boolean	A current out of range (1 [True] = current out of range, 0 [False] = current within range)
amp_B	Amps	Current output of DC Module B
amp_BRange	Boolean	B current out of range (1 [True] = current out of range, 0 [False] = current within range)
amp_C	Amps	Current output of DC Module C
amp_CRange	Boolean	C current out of range (1 [True] = current out of range, 0 [False] = current within range)
amp_D	Amps	Current output of DC Module D
amp_DRange	Boolean	D current out of range (1 [True] = current out of range, 0 [False] = current within range)
frNoPowerModA	Boolean	Indicates if the DC power module A is installed (0 [False] = installed, 1 [True] = not installed)
frNoPowerModB	Boolean	Indicates if the DC power module B is installed (0 [False] = installed, 1 [True] = not installed)
frNoPowerModC	Boolean	Indicates if the DC power module C is installed (0 [False] = installed, 1 [True] = not installed)
frNoPowerModD	Boolean	Indicates if the DC power module D is installed (0 [False] = installed, 1 [True] = not installed)
frPowerModAbad	Boolean	Status of DC power module A (0 [False] (green) = okay, 1 [True] (red) = not okay)
frPowerModBbad	Boolean	Status of DC power module B (0 [False] (green) = okay, 1 [True] (red) = not okay)

Variable Name	Display Type	Description
frPowerModCbad	Boolean	Status of DC power module C (0 [False] (green) = okay, 1 [True] (red) = not okay)
frPowerModDbad	Boolean	Status of DC power module D (0 [False] (green) = okay, 1 [True] (red) = not okay)
frPowerLED	Boolean	Status of power supply (1 [1] = LED lit, power OK, 0 [0] = LED not lit, no power)
frPowerMods	Multistate	Represents the bottom green frame supervisor LED, showing whether power modules are present (2/3 [2] (gray)=LED blinking: no modules are present, 1 [1] (green)=LED lit: at least 1 module is present, 0 [0] (black)=LED not lit)
frPowerOff	Boolean	Frame power (1 [True] = off, 0 [False] = on)

Frame Hardware Variables Unique to HACWS (Type 20) and SEPBU HACWS (Type 22) Frame Supervisors

Variable Name	Display Type	Description
controllerIDMismatch	Boolean	(0 [True] = the frame ID stored in the HACWS supervisor card does not match the configured frame ID, 1 [False] = the frame ID stored in the HACWS supervisor card matches the configured frame ID)
controllerTail	Integer	(1 [1] = the monitor is connected via S1 to the frame supervisor, 2 [2] = the monitor is connected via S2 to the frame supervisor)
controllerTailActive	Boolean	(1 [True] (green) = the RS232 connection between the control workstation and the frame is active; the frame supervisor accepts hardware control commands from the monitor, 0 [False] (red) = the RS232 connection between the control workstation and the frame is not active; the frame supervisor does not accept hardware control commands from the monitor) In an HACWS environment, only one of the two connections should be active.
controllerTail1LED	Integer	Represents green LED 4 frame supervisor LED, showing the status of the S1 tail (1 [1] (green) LED on: S1 tail active, 0 [0] (red) LED off: S1 tail inactive)
controllerTail2LED	Integer	Represents green LED 3 frame supervisor LED, showing the status of the S2 tail (1 [1] (green) LED on: S2 tail active, 0 [0] (red) LED off: S2 tail inactive)
frRS232Active1	Integer	Represents the yellow LED 8 frame supervisor LED, showing the status of the RS-232Active1 communication with the control workstation (1 [1] (yellow) = LED on: RS232 active, 2/3 [2] (gray) = LED blinking: RS232 error, 0 [0] (black) = LED off: RS232 inactive)
frRS232Active2	Integer	Represents the yellow LED 7 frame supervisor LED, showing the status of the RS-232Active2 communication with the control workstation (1 [1] (yellow) = LED on: RS232 active, 2/3 [2] (gray) = LED blinking: RS232 error, 0 [0] (black) = LED off: RS232 inactive)

Variable Name	Display Type	Description
frTTNodeComm	Integer	Represents the yellow LED 6 frame supervisor LED, showing status of communication with nodes over the supervisor bus (I2C bus) (2/3 (gray)=LED blinking: bus dead-no nodes responding, 1 (yellow) = LED lit: 1 or more nodes not responding, 0 (black) = LED not lit: all nodes responding)

Frame Hardware Variables Unique to Card Type 17 and SEPBU (Type 18) Frame Superviors

Variable Name	Display Type	Description
frNodeComm	Integer	Represents the top yellow frame supervisor LED, showing status of communication with nodes over the supervisor bus (I2C link) (2/3 [2] (gray)=LED blinking: bus dead-no nodes responding, 1 [1] (yellow) = LED lit: 1 or more nodes not responding, 0 [0] (black) = LED not lit: all nodes responding)
frRS232Active	Integer	Represents the bottom yellow frame supervisor LED, showing status of serial (RS/232) communication with the control console (2/3 [2] (gray)=LED blinking: RS232 error, 1 [1] (yellow) = LED lit: RS232 activity, 0 [0] (black) = LED not lit: RS232 no activity)

Node Hardware Variables Common to All Node Supervisors

Variable Name	Display Type	Description
codeVersion	Constant	Supervisor microcode version number
commFail	Boolean	Indicates loss of communication to node supervisor from frame supervisor (1 [True] (red) = communication failure, 0 [False] (green) = communication okay) Display using the **spmon** command, use **-q -t frame***/**nodefail***/**value**
console	openConsole	Opens a window to the node's serial 1 port (no values are available for this object). The serialLinkOpen object indicates if a serial 1 port is open. To open a window to the serial 1 port using the **spmon** command, use the -open parameter or **s1term** command. Can't be queried with **hmmon** or **spmon**; can be selected from the All Node Summary of the System Monitor GUI.
envLED	Multistate	Reflects environment light on front of node. (2/3 [2] (gray) = LED blinking: power is shut off due to detection of a critical error (voltages, fans, temp., etc), 1 [1] (yellow) = LED lit: warning indicator-not yet critical, 0 [0] (black) = LED not lit: okay)
hostResponds	Boolean	Status of communication between the processor node and the control workstation (1 (green) = communication okay, 0 (red) = communication failure) Can't be queried from **hmmon**; can be selected from All Node Summary of System Monitor GUI.

Variable Name	Display Type	Description
keyModeSwitch	keySwitch	Three position key switch. To control the setting of the keyModeSwitch using the **spmon** command, use the **-key** parameter. To display the setting of the **keyModeSwitch** using the **spmon** command, use the -Key parmeter. The node supervisors return values indicating the position of the keyModeSwitch. (2 [2] (red) = service, 1 [1] (yellow) = secure, 0 [0] (green) = normal)
LED7SegA	7seg	** 1st digit of 3 digit display
LED7SegB	7seg	** 2nd digit of 3 digit display
LED7SegC	7seg	** 3rd digit of 3 digit display
nodePower	powerSwitch	To control the power using the **spmon** command, use the -power parameter and the -reset parameter. The node supervisors return values indicating whether nodePower is on or off. (1 [True] (green) = power switch on, 0 [False] (red) = power switch off)
powerLED	Multistate	Reflects power light on front of node. (2/3 [2] (gray) = LED blinking: power off but available, 1 [1] (green) = LED lit: power on, 0 [0] (black) = LED not lit: no power exists)
P48OK	Boolean	+48 volt input is okay (1 [True] (green) = okay, 0 [False] (red) = not okay)
serialLinkOpen	Multistate	Indicates whether the serial 1 port is open (1 [True] (yellow) = port opened, 0 [False] (black) = port closed)
switchResponds	Boolean	This variable is only defined when a switch is installed on the frame in which the node resides. On High Performance Switch systems, it indicates whether the switch is active on the node (1 (green) = active, 0 (red)= inactive, 2(yellow) = isolated) On SP Switch systems, it indicates whether the switch is active on the node (1 (green) = active, 0 (red) = not active on the switch for one of the following reasons It was “fenced” without the **-autojoin** option It faulted and was removed from the switch The node's adapter failed to configure properly The node's SP Switch adapter was recently reset by the **/usr/lpp/ssp/css/rc.switch** script. 2 (yellow) = “fenced” with the **-autojoin** option) Can't be queried from **hmmon**; can be selected from All Node Summary of System Monitor GUI.
temp	Celsius	Temperature (red = node shutdown due to temp failure, yellow = warning, temp is outside proper threshold, green = temp is OK)
tempRange	Boolean	Temperature out of range (1 [True] = temperature out of range, 0 [False] = temperature within range)
timeTicks	Integer	Node Supervisor “heartbeat” counter (in numbers of 100 ms tics)
type	Constant	Node Supervisor card type (in decimal)
3DigitDisplay	ledDisplay	Three-digit display. This variable cannot be used as a target of the **spmon** command. To display the contents of a three-digit display using the **spmon** command, use the -Led parameter. Information from **spmon -Led** can't be used with a 604 High Node. Can't be queried with **hmmon** or **spmon**; can be selected from the All Node Summary of the System Monitor GUI.

** Note: when querying the 3-digit LED values, the integers correspond to the following display characters:

Table 18. LED Translation Values

192 = 0	248 = 7
249 = 1	128 = 8
164 = 2	144 = 9
176 = 3	167 = c
153 = 4	179 = d
146 = 5	157 = u
130 = 6	255 = blank

Node-Specific Hardware Variables for Card Types 33 and 65 Node Supervisor

Variable Name	Display Type	Description
currentShutdown	Boolean	Current shutdown (1 [True] = shutdown, 0 [False] = current OK)
diagByte	Integer	Diagnosis byte (correlates with error indicators) This variable is not currently being used.
fanfail1-3	Boolean	Environmental failure: Shutdown due to fan out of spec condition (1 [True] (red) = shutdown, 0 [False] (green) = fan okay)
fanwarning1-3	Boolean	Environmental warning: Fan nearing out of spec condition (1 [True] (red) = warning, 0 [False] (green) = fan okay)
memoryProtect	Boolean	Represents the MemoryProtect signal received from the I/O planar of the processor node. The MemoryProtect signal indicates whether the memory cards in the processor node are the correct type (voltage) for this processor node. (0 [False] (green) = okay, 1 [True] (red) = incorrect voltage type memory card)
nodePowerOn10Sec	Boolean	Indicates node has been operational more than 10 seconds and remains so (1 [True] (green) = powered on, 0 [False] (black) = powered off)
P48Off10Sec	Boolean	+48 volt input has been terminated due to error and remains off (1 [True] (red) = powered off, 0 [False] (black) = okay)
resetActive	Boolean	Indicates that the node is resetting (1 [True] = node is resetting, 0 [False] = node is not resetting)
serialNum	Constant	Supervisor serial number. This variable is not currently being used
shutdownN12High	Boolean	Environmental failure: Node shutdown due to -12 volt line out of specifications (high) (1 [True] (red) = shutdown, 0 [False] (green) = voltage okay)
shutdownN12Low	Boolean	Environmental failure: Node shutdown due to -12 volt line out of specifications (low) (1 [True] (red) = shutdown, 0 [False] (green) = voltage okay)
shutdownP5High	Boolean	Environmental failure: Node shutdown due to +5 volt line out of specifications (high) (1 [True] (red) = shutdown, 0 [False] (green) = voltage okay)

Variable Name	Display Type	Description
shutdownP5Low	Boolean	Environmental failure: Node shutdown due to +5 volt line out of specifications (low) (1 [True] (red) = shutdown, 0 [False] (green) = voltage okay)
shutdownP12High	Boolean	Environmental failure: Node shutdown due to +12 volt line out of specifications (high) (1 [True] (red) = shutdown, 0 [False] (green) = voltage okay)
shutdownP12Low	Boolean	Environmental failure: Node shutdown due to +12 volt line out of specifications (low) (1 [True] (red) = shutdown, 0 [False] (green) = voltage okay)
shutdownP48Low	Boolean	Environmental failure: Node shutdown due to +48 volt line out of specifications (low) (1 [True] (red) = shutdown, 0 [False] (green) = voltage okay)
shutdownTemp	Boolean	Environmental failure: Shutdown due to temperature out of spec (1 [True] (red) = shutdown, 0 [False] (green) = temperature okay)
s1PortDTR	Boolean	Indicates serial 1 port DTR asserted. A console cannot be opened to the node's serial 1 port unless this flag is on. (1 [True] (green) = port asserted, 0 [False] (red) = port not asserted)
voltN12	Volts	Voltage on -12 volt line (red = node shutdown due to voltage failure, yellow = warning, voltage is outside proper threshold, green = voltage is OK)
voltN12Range	Boolean	-12 voltage out of range (1 [True] = voltage out of range, 0 [False] = voltage within range)
voltP5	Volts	Voltage on +5 volt line (red = node shutdown due to voltage failure, yellow = warning, voltage is outside proper threshold, green = voltage is OK)
voltP5Range	Boolean	+5 voltage out of range (1 [True] = voltage out of range, 0 [False] = voltage within range)
voltP12	Volts	Voltage on +12 volt line (red = node shutdown due to voltage failure, yellow = warning, voltage is outside proper threshold, green = voltage is OK)
voltP12Range	Boolean	+12 voltage out of range (1 [True] = voltage out of range, 0 [False] = voltage within range)
voltP48	Volts	Voltage on +48 volt line
voltP48Range	Boolean	+48 voltage out of range (1 [True] = voltage out of range, 0 [False] = voltage within range)
warningN12High	Boolean	Environmental warning: Node warning due to marginal spec condition on -12 volt line (high) (1 [True] (red) = warning, 0 [False] (green) = voltage okay)
warningN12Low	Boolean	Environmental warning: Node warning due to marginal spec condition on -12 volt line (low) (1 [True] (red) = warning, 0 [False] (green) = voltage okay)
warningP5High	Boolean	Environmental warning: Node warning due to marginal spec condition on +5 volt line (high) (1 [True] (red) = warning, 0 [False] (green) = voltage okay)
warningP5Low	Boolean	Environmental warning: Node warning due to marginal spec condition on +5 volt line (low) (1 [True] (red) = warning, 0 [False] (green) = voltage okay)

Variable Name	Display Type	Description
warningP12High	Boolean	Environmental warning: Node warning due to marginal spec condition on +12 volt line (high) (1 [True] (red) = warning, 0 [False] (green) = voltage okay)
warningP12Low	Boolean	Environmental warning: Node warning due to marginal spec condition on +12 volt line (low) (1 [True] (red) = warning, 0 [False] (green) = voltage okay)
7segChanged	Multistate	Indicates the digits within the 3-digit display are changing faster than once per second. (1 [True] (black) digits changing faster than once per second, 0 [False] (yellow) digits changing less than once per second)

Node-Specific Hardware Variables for Card Type 97 Node Supervisors

Variable Name	Display Type	Description
diagByte	Integer	Diagnosis byte (correlates with error indicators) This variable is not currently being used.
fanfail1-3	Boolean	Environmental failure: Shutdown due to fan out of spec condition (1 [True] (red) = shutdown, 0 [False] (green) = fan okay)
fanwarning1-3	Boolean	Environmental warning: Fan nearing out of spec condition (1 [True] (red) = warning, 0 [False] (green) = fan okay)
memoryProtect	Boolean	Represents the MemoryProtect signal received from the I/O planar of the processor node. The MemoryProtect signal indicates whether the memory cards in the processor node are the correct type (voltage) for this processor node. (0 [False] (green) = okay, 1 [True] (red) = incorrect voltage type memory card)
nodePowerOn10Sec	Boolean	Indicates node has been operational more than 10 seconds and remains so (1 [True] (green) = powered on, 0 [False] (black) = powered off)
P4dPresent	Multistate	Indicates daughter card 4v supply installed (1 [True] (green) = 4v card installed, 0 [False] (black) = 4v card not installed)
P48Off10Sec	Boolean	+48 volt input has been terminated due to error and remains off (1 [True] (red) = powered off, 0 [False] (black) = okay)
P5DCok	Boolean	Voltage on +5 volt line is good (1 [True] (green) = voltage ok, 0 [False] (red) = voltage not on)
P2_5dPresent	Multistate	Indicates daughter card 2.5v supply installed (1 [True] (green) = 2.5v card installed, 0 [False] (black) = 2.5v card not installed)
resetActive	Boolean	Indicates that the node is resetting (1 [True] = node is resetting, 0 [False] = node is not resetting)
serialNum	Constant	Supervisor serial number. This variable is not currently being used
shutdownN12High	Boolean	Environmental failure: Node shutdown due to -12 volt line out of specifications (high) (1 [True] (red) = shutdown, 0 [False] (green) = voltage okay)
shutdownN12Low	Boolean	Environmental failure: Node shutdown due to -12 volt line out of specifications (low) (1 [True] (red) = shutdown, 0 [False] (green) = voltage okay)

Variable Name	Display Type	Description
shutdownP4dHigh	Boolean	Environmental failure: Node shutdown due to +4 volt daughter card line out of specifications (high) (1 [True] (red) = shutdown, 0 [False] (green) = voltage okay) / not installed
shutdownP4dLow	Boolean	Environmental failure: Node shutdown due to +4 volt daughter line out of specifications (low) (1 [True] (red) = shutdown, 0 [False] (green) = voltage okay) / not installed
shutdownP5High	Boolean	Environmental failure: Node shutdown due to +5 volt line out of specifications (high) (1 [True] (red) = shutdown, 0 [False] (green) = voltage okay)
shutdownP5Low	Boolean	Environmental failure: Node shutdown due to +5 volt line out of specifications (low) (1 [True] (red) = shutdown, 0 [False] (green) = voltage okay)
shutdownP2_5dHigh	Boolean	Environmental failure: Node shutdown due to +2.5 volt daughter card line out of specifications (high) (1 [True] (red) = shutdown, 0 [False] (green) = voltage okay) / not installed
shutdownP2_5dLow	Boolean	Environmental failure: Node shutdown due to +2.5 volt daughter line out of specifications (low) (1 [True] (red) = shutdown, 0 [False] (green) = voltage okay) / not installed
shutdownP12High	Boolean	Environmental failure: Node shutdown due to +12 volt line out of specifications (high) (1 [True] (red) = shutdown, 0 [False] (green) = voltage okay)
shutdownP12Low	Boolean	Environmental failure: Node shutdown due to +12 volt line out of specifications (low) (1 [True] (red) = shutdown, 0 [False] (green) = voltage okay)
shutdownTemp	Boolean	Environmental failure: Shutdown due to temperature out of spec (1 [True] (red) = shutdown, 0 [False] (green) = temperature okay)
s1PortDTR	Boolean	Indicates serial 1 port DTR asserted. A console cannot be opened to the node's serial 1 port unless this flag is on. (1 [True] (green) = port asserted, 0 [False] (red) = port not asserted)
voltN12	Volts	Voltage on -12 volt line (red = node shutdown due to voltage failure, yellow = warning, voltage is outside proper threshold, green = voltage is OK)
voltN12Range	Boolean	-12 voltage out of range (1 [True] = voltage out of range, 0 [False] = voltage within range)
voltP4d	Volts	Voltage on +4 volt daughter card line; only valid when the P4dPresent value is 1 (red = node shutdown due to voltage failure, yellow = warning, voltage is outside proper threshold, green = voltage is OK)
voltP4dRange	Boolean	+4 volt daughter out of range (1 [True] = voltage out of range, 0 [False] = voltage within range)
voltP5	Volts	Voltage on +5 volt line (red = node shutdown due to voltage failure, yellow = warning, voltage is outside proper threshold, green = voltage is OK)
voltP5Range	Boolean	+5 voltage out of range (1 [True] = voltage out of range, 0 [False] = voltage within range)
voltP2_5d	Volts	Voltage on +2.5 volt daughter card line; only valid when the P2_5dPresent value is 1 (red = node shutdown due to voltage failure, yellow = warning, voltage is outside proper threshold, green = voltage is OK)

Variable Name	Display Type	Description
voltP2_5dRange	Boolean	+2.5 volt daughter out of range (1 [True] = voltage out of range, 0 [False] = voltage within range)
voltP12	Volts	Voltage on +12 volt line (red = node shutdown due to voltage failure, yellow = warning, voltage is outside proper threshold, green = voltage is OK)
voltP12Range	Boolean	+12 voltage out of range (1 [True] = voltage out of range, 0 [False] = voltage within range)
warningN12High	Boolean	Environmental warning: Node warning due to marginal spec condition on -12 volt line (high) (1 [True] (red) = warning, 0 [False] (green) = voltage okay)
warningN12Low	Boolean	Environmental warning: Node warning due to marginal spec condition on -12 volt line (low) (1 [True] (red) = warning, 0 [False] (green) = voltage okay)
warningP4dHigh	Boolean	Environmental warning: Node warning due to marginal spec condition on +4 volt daughter card line (high) (1 [True] (red) = warning, 0 [False] (green) = voltage okay) / not installed
warningP4dLow	Boolean	Environmental warning: Node warning due to marginal spec condition on +4 volt daughter card line (low) (1 [True] (red) = warning, 0 [False] (green) = voltage okay) / not installed
warningP5High	Boolean	Environmental warning: Node warning due to marginal spec condition on +5 volt line (high) (1 [True] (red) = warning, 0 [False] (green) = voltage okay)
warningP5Low	Boolean	Environmental warning: Node warning due to marginal spec condition on +5 volt line (low) (1 [True] (red) = warning, 0 [False] (green) = voltage okay)
warningP2_5dHigh	Boolean	Environmental warning: Node warning due to marginal spec condition on +2.5 volt daughter card line (high) (1 [True] (red) = warning, 0 [False] (green) = voltage okay) / not installed
warningP2_5dLow	Boolean	Environmental warning: Node warning due to marginal spec condition on +2.5 volt daughter card line (low) (1 [True] (red) = warning, 0 [False] (green) = voltage okay) / not installed
warningP12High	Boolean	Environmental warning: Node warning due to marginal spec condition on +12 volt line (high) (1 [True] (red) = warning, 0 [False] (green) = voltage okay)
warningP12Low	Boolean	Environmental warning: Node warning due to marginal spec condition on +12 volt line (low) (1 [True] (red) = warning, 0 [False] (green) = voltage okay)
warningTemp	Boolean	Environmental warning: Temperature nearing out of spec condition (1 [True] (red) = warning, 0 [False] (green) = temperature okay)
7segChanged	Multistate	Indicates the digits within the 3-digit display are changing faster than once per second. (1 [True] (black) digits changing faster than once per second, 0 [False] (yellow) digits changing less than once per second)

Node-Specific Hardware Variables for Card Types 81 and 83 Node Supervisors

Variable Name	Display Type	Description
diagByte	Integer	Diagnosis byte (correlates with error indicators) This variable is not currently being used.
fanfail1d-5d	Boolean	Environmental failure: Shutdown due to fan out of spec condition (1 [True] (red) = shutdown, 0 [False] (green) = fan okay)
fanwarning1-3	Boolean	Environmental warning: Fan nearing out of spec condition (1 [True] (red) = warning, 0 [False] (green) = fan okay)
fanwarning4-5	Boolean	Environmental warning: Fan nearing out of spec condition (1 [True] (red) = warning, 0 [False] (green) = fan okay)
nodePowerOn10Sec	Boolean	Indicates node has been operational more than 10 seconds and remains so (1 [True] (green) = powered on, 0 [False] (black) = powered off)
P48Off10Sec	Boolean	+48 volt input has been terminated due to error and remains off (1 [True] (red) = powered off, 0 [False] (black) = okay)
P5DCok	Boolean	Voltage on +5 volt line is good (1 [True] (green) = voltage ok, 0 [False] (red) = voltage not on)
resetActive	Boolean	Indicates that the node is resetting (1 [True] = node is resetting, 0 [False] = node is not resetting)
serialNum	Constant	Supervisor serial number. This variable is not currently being used
shutdownN12High	Boolean	Environmental failure: Node shutdown due to -12 volt line out of specifications (high) (1 [True] (red) = shutdown, 0 [False] (green) = voltage okay)
shutdownN12Low	Boolean	Environmental failure: Node shutdown due to -12 volt line out of specifications (low) (1 [True] (red) = shutdown, 0 [False] (green) = voltage okay)
shutdownP4High	Boolean	Environmental failure: Node shutdown due to +4 volt line out of specifications (high) (1 [True] (red) = shutdown, 0 [False] (green) = voltage okay)
shutdownP4Low	Boolean	Environmental failure: Node shutdown due to +4 volt line out of specifications (low) (1 [True] (red) = shutdown, 0 [False] (green) = voltage okay)
shutdownP5iHigh	Boolean	Environmental failure: Node shutdown due to +5 volt I/O supply line out of specifications (high) (1 [True] (red) = shutdown, 0 [False] (green) = voltage okay)
shutdownP5iLow	Boolean	Environmental failure: Node shutdown due to +5 volt I/O supply line out of specifications (high) (1 [True] (red) = shutdown, 0 [False] (green) = voltage okay)
shutdownP5mHigh	Boolean	Environmental failure: Node shutdown due to +5 volt memory supply line out of specifications (high) (1 [True] (red) = shutdown, 0 [False] (green) = voltage okay)
shutdownP5mLow	Boolean	Environmental failure: Node shutdown due to +5 volt memory supply line out of specifications (high) (1 [True] (red) = shutdown, 0 [False] (green) = voltage okay)

Variable Name	Display Type	Description
shutdownP12High	Boolean	Environmental failure: Node shutdown due to +12 volt line out of specifications (high) (1 [True] (red) = shutdown, 0 [False] (green) = voltage okay)
shutdownP12Low	Boolean	Environmental failure: Node shutdown due to +12 volt line out of specifications (low) (1 [True] (red) = shutdown, 0 [False] (green) = voltage okay)
shutdownTemp	Boolean	Environmental failure: Shutdown due to temperature out of spec (1 [True] (red) = shutdown, 0 [False] (green) = temperature okay)
s1PortDTR	Boolean	Indicates serial 1 port DTR asserted. A console cannot be opened to the node's serial 1 port unless this flag is on. (1 [True] (green) = port asserted, 0 [False] (red) = port not asserted)
voltN12	Volts	Voltage on -12 volt line (red = node shutdown due to voltage failure, yellow = warning, voltage is outside proper threshold, green = voltage is OK)
voltN12Range	Boolean	-12 voltage out of range (1 [True] = voltage out of range, 0 [False] = voltage within range)
voltP4	Volts	Voltage on +4 volt supply line (red = node shutdown due to voltage failure, yellow = warning, voltage is outside proper threshold, green = voltage is OK)
voltP5i	Volts	Voltage on +5 volt I/O supply line (red = node shutdown due to voltage failure, yellow = warning, voltage is outside proper threshold, green = voltage is OK)
voltP5iRange	Boolean	+5 I/O voltage out of range (1 [True] = voltage out of range, 0 [False] = voltage within range)
voltP5m	Volts	Voltage on +5 volt memory supply line (red = node shutdown due to voltage failure, yellow = warning, voltage is outside proper threshold, green = voltage is OK)
voltP5mRange	Boolean	+5 memory voltage out of range (1 [True] = voltage out of range, 0 [False] = voltage within range)
voltP12	Volts	Voltage on +12 volt line (red = node shutdown due to voltage failure, yellow = warning, voltage is outside proper threshold, green = voltage is OK)
voltP12Range	Boolean	+12 voltage out of range (1 [True] = voltage out of range, 0 [False] = voltage within range)
warningN12High	Boolean	Environmental warning: Node warning due to marginal spec condition on -12 volt line (high) (1 [True] (red) = warning, 0 [False] (green) = voltage okay)
warningN12Low	Boolean	Environmental warning: Node warning due to marginal spec condition on -12 volt line (low) (1 [True] (red) = warning, 0 [False] (green) = voltage okay)
warningP4High	Boolean	Environmental warning: Node warning due to marginal spec condition on +4 volt line (high) (1 [True] (red) = warning, 0 [False] (green) = voltage okay)
warningP4Low	Boolean	Environmental warning: Node warning due to marginal spec condition on +4 volt line (low) (1 [True] (red) = warning, 0 [False] (green) = voltage okay)
warningP5iHigh	Boolean	Environmental warning: Node warning due to marginal spec condition on +5 volt I/O supply line (high) (1 [True] (red) = warning, 0 [False] (green) = voltage okay)

Variable Name	Display Type	Description
warningP5iLow	Boolean	Environmental warning: Node warning due to marginal spec condition on +5 volt I/O supply line (low) (1 [True] (red) = warning, 0 [False] (green) = voltage okay)
warningP5mHigh	Boolean	Environmental warning: Node warning due to marginal spec condition on +5 volt memory supply line (high) (1 [True] (red) = warning, 0 [False] (green) = voltage okay)
warningP5mLow	Boolean	Environmental warning: Node warning due to marginal spec condition on +5 volt memory supply line (low) (1 [True] (red) = warning, 0 [False] (green) = voltage okay)
warningP12High	Boolean	Environmental warning: Node warning due to marginal spec condition on +12 volt line (high) (1 [True] (red) = warning, 0 [False] (green) = voltage okay)
warningP12Low	Boolean	Environmental warning: Node warning due to marginal spec condition on +12 volt line (low) (1 [True] (red) = warning, 0 [False] (green) = voltage okay)
warningTemp	Boolean	Environmental warning: Temperature nearing out of spec condition (1 [True] (red) = warning, 0 [False] (green) = temperature okay)
7segChanged	Multistate	Indicates the digits within the 3-digit display are changing faster than once per second. (1 [True] (black) digits changing faster than once per second, 0 [False] (yellow) digits changing less than once per second)

Node-Specific Hardware Variables For Card Type 161 Node Supervisor

Table 19 (Page 1 of 2). Node-Specific Hardware Variables For Card Type 161 Node Supervisor

Variable Name	Display Type	Description
boardLevel	Constant	Incremental number indicating level of hardware.
hardwareKeyID	Constant	A merging of “boardLevel” and “supLevel” constants to produce a Key ID that uniquely defines this level of hardware.
hardwareStatus	Hex	Hardware diagnostics byte as follows: bit 7: 1 = Busy condition on LCD bit 6: 1 = Bounces on power On/Off Switch bit 5: 1 = Bounces on reset button bit 4: 1 = Bounces on key switch bit 3: 1 = Wrong length command from BUMP bit 2: 1 = SIF timeout bit 1: 1 = BUMP timeout bit 0: 1 = Unknown command from BUMP
lcd1	String	16 bytes of ASCII data that is the first line of the LCD display.
lcd1flash	Boolean	Indicates line 1 of the LCD display is flashing. (1 [True] = flashing, [False] 0 = not flashing)
lcd2	String	16 bytes of ASCII data that is the second line of the LCD display.
lcd2flash	Boolean	Indicates line 2 of the LCD display is flashing. (1 [True] = flashing, [False] 0 = not flashing)

Table 19 (Page 2 of 2). Node-Specific Hardware Variables For Card Type 161 Node Supervisor		
Variable Name	**Display Type**	**Description**
powerSwitchStatus	Boolean	Indicates power switch position and whether or not the node is capable of being powered on. (1 (on) = capable of being powered on, 0 (off) can not be powered on)
P5DCok	Boolean	Voltage on +5 volt line is good. (1 [True] (green) = voltage okay, 0 [False] (red) = voltage not on)
smpDiagLEDoff	Boolean	Environmental error: LED 5 (0 [False] = error condition, 1 [True] = no errors detected)
smpPowerLEDoff	Boolean	Indicates power status. LED 1 (0 [False] = DC power okay, 1 [True] = no DC power supplied)
supLevel	Constant	Incremental number indicating level of application supervisor installed on board.
warningTemp	Boolean	Environmental warning: Temperature nearing out of spec condition. (1 [True] (red) = warning, 0 [False] (green) = temperature okay)

Common Switch Variables

Variable Name	Display Type	Description
codeVersion	Constant	Supervisor microcode version number
diagByte	Integer	Diagnosis byte (correlates with error indicators). This variable is not currently being used.
envLED	Multistate	Reflects environment light on front of switch (2/3 [2] (gray) = LED blinking: power is shut off due to detection of a critical error (voltages, fans, temp., etc), 1 [1] (yellow) = LED lit: warning indicator-not yet critical, 0 [0] (black) = LED not lit: okay)
fanfail 1-5	Boolean	Environmental failure: Shutdown due to fan out of spec condition (1 [True] (red) = shutdown, 0 [False] (green) = fan okay)
nodePower	powerSwitch	To control the power using the **spmon** command, use the **-power** parameter. The switch supervisor returns a value indicating whether switchPower is on or off. (1 [True] (green) = power switch on, 0 [False] (red) = power switch off)
nodePowerOn10Sec	Boolean	Indicates switch has been operational more than 10 seconds and remains so (1 [True] (green) = powered on, 0 [False] (black) = powered off)
powerLED	Multistate	Reflects power light on front of switch (2/3 [2] (gray) = LED blinking: power off but available, 1 [1] (green) = LED lit: power on, 0 [0] (black) = LED not lit: no power exists)
P48Off10Sec	Boolean	+48 volt input has been terminated due to error and remains off (1 [True] (red) = powered off, 0 [False] (black) = okay)
P48OK	Boolean	+48 volt input is okay (1 [True] (green) = okay, 0 [False] (red) = not okay)
serialNum	Constant	Supervisor serial number. This variable is not currently being used.
shutdownTemp	Boolean	Environmental failure: Shutdown due to temperature out of spec (1 [True] (red) = shutdown, 0 [False] (green) = temperature okay)

Variable Name	Display Type	Description
temp	Celsius	Temperature (red = node shutdown due to temp failure, yellow = warning, temp is outside proper threshold, green = temp is OK)
tempRange	Boolean	Temperature out of range. (1 [True] = temp out of range, 0 [False] = temp within range)
timeTicks	Integer	Supervisor “heartbeat” counter (in numbers of 100 ms tics)
type	Constant	Switch Supervisor card type (in decimal)
warningTemp	Boolean	Environmental warning: Temperature nearing out of spec condition (1 [True] (red) = warning, 0 [False] (green) = temperature okay)

High Performance Switch and High Performance Switch-LC8 Card Types 49 and 50

Variable Name	Display Type	Description
fanwarning 1-5	Boolean	Environmental warning: Fan nearing out of spec condition (1 [True] (red) = warning, 0 [False] (green) = fan okay)
mux	Integer	Multiplexor setting to control the clock selection at each frame. Use the **-mux** parameter on the **spmon** command to select the proper input for the clock (3 [3] = input 3, 2[2] = input 2, 1[1] = input 1, 0[0] = internal clock)
shutdownN5High	Boolean	Environmental failure: Node shutdown due to -5 volt line out of specifications (high) (1 [True] (red) = shutdown, 0 [False] (green) = voltage okay)
shutdownN5Low	Boolean	Environmental failure: Node shutdown due to -5 volt line out of specifications (low) (1 [True] (red) = shutdown, 0 [False] (green) = voltage okay)
shutdownP5High	Boolean	Environmental failure: Node shutdown due to +5 volt line out of specifications (high) (1 [True] (red) = shutdown, 0 [False] (green) = voltage okay)
shutdownP5Low	Boolean	Environmental failure: Node shutdown due to +5 volt line out of specifications (low) (1 [True] (red) = shutdown, 0 [False] (green) = voltage okay)
shutdownP12High	Boolean	Environmental failure: Node shutdown due to +12 volt line out of specifications (high) (1 [True] (red) = shutdown, 0 [False] (green) = voltage okay)
shutdownP12Low	Boolean	Environmental failure: Node shutdown due to +12 volt line out of specifications (low) (1 [True] (red) = shutdown, 0 [False] (green) = voltage okay)
shutdownP48Low	Boolean	Environmental failure: Node shutdown due to +48 volt line out of specifications (low) (1 [True] (red) = shutdown, 0 [False] (green) = voltage okay)
voltN5	Volts	Voltage on -5 volt line (red = node shutdown due to voltage failure, yellow = warning—voltage is outside proper threshold, green = voltage is okay)
voltN5Range	Boolean	-5 voltage out of range (1 [True] = voltage out of range, 0 [False] = voltage within range)

Variable Name	Display Type	Description
voltP5	Volts	Voltage on +5 volt line (red = node shutdown due to voltage failure, yellow = warning—voltage is outside proper threshold, green) = voltage is okay)
voltP5Range	Boolean	+5 voltage out of range (1 [True] = voltage out of range, 0 [False] = voltage within range)
voltP12	Volts	Voltage on +12 volt line (red = node shutdown due to voltage failure, yellow = warning—voltage is outside proper threshold, green) = voltage is okay)
voltP12Range	Boolean	+12 voltage out of range (1 [True] = voltage out of range, 0 [False] = voltage within range)
voltP48	Volts	Voltage on +48 volt line (red = node shutdown due to voltage failure, yellow = warning—voltage is outside proper threshold, green) = voltage is okay)
voltP48Range	Boolean	+48 voltage out of range (1 [True] = voltage out of range, 0 [False] = voltage within range)
warningN5High	Boolean	Environmental warning: Node warning due to marginal spec condition on -5 volt line (high) (1 [True] (red) = warning, 0 [False] (green) = voltage okay)
warningN5Low	Boolean	Environmental warning: Node warning due to marginal spec condition on -5 volt line (low) (1 [True] (red) = warning, 0 [False] (green) = voltage okay)
warningP5High	Boolean	Environmental warning: Node warning due to marginal spec condition on +5 volt line (high) (1 [True] (red) = warning, 0 [False] (green) = voltage okay)
warningP5Low	Boolean	Environmental warning: Node warning due to marginal spec condition on +5 volt line (low) (1 [True] (red) = warning, 0 [False] (green) = voltage okay)
warningP12High	Boolean	Environmental warning: Node warning due to marginal spec condition on +12 volt line (high) (1 [True] (red) = warning, 0 [False] (green) = voltage okay)
warningP12Low	Boolean	Environmental warning: Node warning due to marginal spec condition on +12 volt line (low) (1 [True] (red) = warning, 0 [False] (green) = voltage okay)

SP Switch Variables for Card Type 129

Variable Name	Display Type	Description
boardLevel	Integer	Incremental number indicating level of hardware.
chip	Integer	Indicates which of the 8 chips the active clock is on when the value of the “clockSource” variable is set to 2 (STI). Value can be 0, 1, 2, 3, 4, 5, 6, or 7.
chipClkMissing	Boolean	Indicates whether the active chip, as defined by “chip,” is receiving its system clock from the current master clock source, as defined by “clockSource” ([True] = not receiving system clock, [False] = receiving system clock)

Variable Name	Display Type	Description
chipReceivedInit	Boolean	Indicates whether or the active chip, as defined by “chip,” has received its initialization packet from the primary service node. The **Estart** command sets and distributes initialization data to chips. ([True] = received initialization data, [False] = did not receive initialization data)
chipSelftest	Boolean	Indicates whether the active chip, as defined by “chip,” is currently in selftest mode as a result of a power on reset, as defined by “powerOnReset” (]True[= in selftest mode, [False] = not in selftest mode)
clockSource	Integer	Indicates whether the active clock source is Local (internal) or STI (external). (0 = Local, 2 = STI)
epromErase	Boolean	Indicates that FEEPROM was erased. This erasure is typically the result of an supervisor application microcode download ([True] = currently erased, [FALSE] = a supervisor application is loaded)
epromProgram	Boolean	Indicates whether FEEPROM was ever programmed with a supervisor application ([True] = programmed, [False] = not programmed)
mux	Integer	Indicates which of the 32 jacks the active clock is on as a function of chip and port when the value of the “clockSource” variable is set to 2 (STI). Value can be any of 2 through 34 per the following matrix: Port 1 2 3 Chip 34 33 32 31 1 1 9 8 7 2 26 25 24 23 3 18 17 16 15 4 22 21 2 19 5 14 13 12 11 6 3 29 28 27 7 6 5 4 3 Or, if the value is zero, indicates that the active clock is Local. In this case, the “clockSource” variable will be set to Local.
osc	Integer	Indicates which of the 2 Local (internal) oscillators are being used when the value of the “clockSource” variable is set to 0 (Local). Value can be 2 or 4.
pll	Integer	Indicates which of the four Phase Lock Loops are being used. Value can be 2, 3, 4, or 5.
port	Integer	Indicates which of the 4 ports the active clock is on when the value of the “clockSource” variable is set to 2 (STI). Value can be 0, 1, 2, or 3.
portClkMissing	Boolean	Indicates whether the active port, as defined by “port,” is receiving a valid STI clock, as defined by “clockSource” ([True] = not receiving STI clock, [False] = receiving STI clock)
powerOnReset	Boolean	Indicates power-on reset state of board ([True] = in state, [False] = not in state)
powerS1	Integer	Power supply #1 key ID.
powerS2	Integer	Power supply #2 key ID.
psParallel	Volts	Parallel power voltage.

Variable Name	Display Type	Description
psParallelFail	Boolean	Power detector failure: This variable indicates whether the ability to detect power supply failures is operational (0 [false] (green) = okay, 1 [true] (red) = not okay) If the value is 1 and either Power Supply #1 or Power Supply #2 failed, “ps1Fail” or “ps2Fail” could not be set to indicate the failure.
psParallelRange	Boolean	Parallel power voltage out of range ([True] = out of range, [False] = not out of range)
ps1Fail	Boolean	Power supply failure: Power Supply #1 failed and should be replaced (0 [false] (green) = okay, 1 [true] (red) = not okay)
ps1FuseGood	Volts	Fuse supply #1 voltage.
ps1FuseGoodRange	Boolean	Fuse supply #1 voltage out of range ([True] = out of range, [False] = not out of range)
ps1PowerGood	Volts	Power supply #1 voltage.
ps1PowerGoodRange	Boolean	Power supply #1 voltage out of range ([True] = out of range, [False] = not out of range)
ps2Fail	Boolean	Power supply failure: Power Supply #2 failed and should be replaced (0 [false] (green) = okay, 1 [true] (red) = not okay)
ps2FuseGood	Volts	Fuse supply #2 voltage.
ps2FuseGoodRange	Boolean	Fuse supply #2 voltage out of range ([True] = out of range, [False] = not out of range)
ps2PowerGood	Volts	Power supply #2 voltage.
ps2PowerGoodRange	Boolean	Power supply #2 voltage out of range ([True] = out of range, [False] = not out of range)
recPortNotTuned	Boolean	Indicates whether a port's receive port is communicating with it ([True] = receive port not communicating with active port, as defined by “port,”, [False] = receive port is communicating with active port, as defined by “port”)
sendPortNotTuned	Boolean	Indicates whether the active port, as defined by “port,” is communicating with its corresponding receive port ([True] = active port not communicating with receive port, [False] = active port is communicating with its receive port)
shutdownOC	Boolean	Threshold shutdown: Either Power Supply #1 or Power Supply #2 has exceeded maximum over current threshold. Power supply should be replaced (0 [false] (green) = okay, 1 [true] (red) = not okay)
shutdownP3_3High	Boolean	Environmental failure: Node shutdown due to +3.3 volt line out of specifications (high) (1 [True] (red) = shutdown, 0 [False] (green) = voltage okay)
shutdownP3_3Low	Boolean	Environmental failure: Node shutdown due to +3.3 volt line out of specifications (low) (1 [True] (red) = shutdown, 0 [False] (green) = voltage okay)
subtype	Boolean	Indicates the subtype of the SP Switch board. ([True] = SP Switch (16 port), [False] = SP Switch-8 (8 port).
supLevel	Integer	Incremental number indicating level of application Supervisor installed on board.
synchReset	Boolean	Indicates synchronous reset state of board ([True] = in state, [False] = not in state)

Variable Name	Display Type	Description
warningOC	Boolean	Threshold warning: Either Power Supply #1 or Power Supply #2 is approaching maximum over current threshold (0 [false] (green) = okay, 1 [true] (red) = not okay)
warningP3_3High	Boolean	Environmental warning: Node warning due to marginal spec condition on +3.3 volt line (high) (1 [True] (red) = warning, 0 [False] (green) = voltage okay)
warningP3_3Low	Boolean	Environmental warning: Node warning due to marginal spec condition on +3.3 volt line (low) (1 [True] (red) = warning, 0 [False] (green) = voltage okay)
voltP3_3	Volts	Voltage on +3.3 volt line (red = node shutdown due to voltage failure, yellow = warning—voltage is outside proper threshold, green = voltage is okay)
voltP3_3Range	Boolean	+3.3 voltage out of range (1 [True] = voltage out of range, 0 [False] = voltage within range)

Basecode Mode Hardware Variables for the SP Switch Card Type 128, SEPBU Frame Card Type 16 and 604 High Node Type 16

Variable Name	Display Type	Description
applicationLevel	Integer	Is the level of application Supervisor loaded in FEEPROM.
checksumError	Boolean	Indicates whether the application Supervisor download process detected a data transmission checksum error ([True] = checksum error, [False] = no checksum error)
feepromEmpty	Boolean	Indicates whether the programmable FEEPROM is empty (not programmed) ([True] = not programmed, [False] = programmed)
feepromEraseCount	Integer	Incremental value that indicates the number of times the programmable FEEPROM was erased
hardwareKeyID	Hex	Is the value of the Hardware Key taken from the hardware (card) itself. The value is an incremental version number.
keyidMismatch	Boolean	Indicates whether a compatible supervisor application was downloaded to the card based on a match of the “softwareKeyID” and “hardwareKeyID” ([True] = mismatch, [False] = no mismatch)
softwareKeyID	Hex	Is the value of the Software Key taken from the supervisor application microcode file during download to the card. The value is an incremental version number.
supDataChecksum	Hex	Checksum for the application Supervisor loaded in FEEPROM.
supFailure	Boolean	Indicates that the application Supervisor loaded into FEEPROM cannot be executed due to an error generated during download or because FEEPROM was not programmed ([True] = not executable, [False] = executable) supFailure will be True if 1 or more of the following conditions occur: keyidMismatch = True checksumError = True feepromEmpty = True supDataChecksum != supRomChecksum

Variable Name	Display Type	Description
supRomChecksum	Hex	Saved supDataChecksum. If checksumError is False, this will match supDataChecksum.

Appendix E. Listing SP System Resource Variables

As described in Chapter 23, "The Event Management Subsystem" on page 349, PSSP supplies a number of resource monitors that collect resource data and send it to the Event Management subsystem. You may also install resource monitors that:

- Are supplied by other IBM products or third parties.
- Have been written by programmers at your installation.

Once all of these resource monitors are running, you may wish to determine what resource data is being collected in the system.

There are several ways to do this:

- You can look in the load list files that contain the Event Management configuration data for each of the resource monitors in your system.

  These files contain the data that is loaded into the SDR as Event Management objects by the **haemloadcfg** command.

- You can use commands.

  You can use the PSSP **SDRGetObjects** command to list Event Management objects that have been stored in the SDR. You can use the AIX **dspmsg** and **dspcat** commands to list individual messages and the contents of message catalogs that contain descriptions of resource variables.

- You can use the SP Perspectives graphical user interface.

  You can use the Event Management portion of SP Perspectives to list information about the resource variables and instance vectors that have been defined in the system.

Looking at the Event Management Load List Files

The load list file for the resource variables supplied by PSSP is the **/usr/lpp/ssp/install/config/haemloadlist** file. This file is a plain text file that consists of stanzas, each of which describes an object in one of the Event Management SDR classes. For detailed information about the format of a load list file, see the man page for the **haemloadlist** file in *PSSP Command and Technical Reference*.

If you have installed other resource monitors, look at the load list files that contain their configuration data.

Getting Information about Event Management Objects in the SDR

To get information about all of the event management objects in the SDR, you can issue the **SDRGetObjects** command for each class of objects. You can also direct the output to a file, which you can then view or print. To do this, enter the following commands:

```
SDRGetObjects -G EM_Resource_Variable > rv_data
SDRGetObjects -G EM_Resource_Class > class_data
SDRGetObjects -G EM_Structured_Byte_String > sbs_data
SDRGetObjects -G EM_Instance_Vector > iv_data
SDRGetObjects -G EM_Resource_Monitor > rm_data
```

© Copyright IBM Corp. 1996

Specifying the **-G** flag ensures that you receive information about all of the resource variables in all of the system partitions in the SP system. If you are only interested in the resource variables in the current system partition, omit the **-G** flag.

Figure 119 on page 537 contains an example of the output of each **SDRGetObjects** command for the **IBM.PSSP.SP_HW.Node.lcd1** resource variable and its associated resource class, structured byte string, instance vector, and resource monitor definitions. For ease of presentation, the output has been reformatted vertically. The actual output of the command is horizontally formatted.

For a complete description of the attributes of Event Management SDR classes, see *IBM Parallel System Support Programs for AIX: Event Management Programming Guide and Reference*.

You can also restrict the information you receive. For example, to get only the **rvName** and **rvClass** information for the **IBM.PSSP.SP_HW.Node.lcd1** resource variable, enter:

```
SDRGetObjects -G EM_Resource_Variable \
    rvName==IBM.PSSP.SP_HW.Node.lcd1 rvName rvClass
```

In response, the system displays:

```
rvName          rvClass
IBM.PSSP.SP_HW.Node.lcd1          IBM.PSSP.SP_HW
```

For more information about the **SDRGetObjects** command, see *PSSP Command and Technical Reference*.

```
Information from EM_Resource_Variable class for Resource Variable IBM.PSSP.SP_HW.Node.lcd1

        rvName                  IBM.PSSP.SP_HW.Node.lcd1
        rvDescription           244
        rvValue_type            State
        rvData_type             SBS
        rvInitial_value
        rvClass                 IBM.PSSP.SP_HW
        rvPTX_name                ""
        rvPTX_description       ""
        rvPTX_min                 ""
        rvPTX_max                 ""
        rvPredicate             ""
        rvEvent_description     ""
        rvLocator                 ""
        rvOrder_group             ""
        rvDynamic_instance
        rvIndex_vector          NodeNum

Information from EM_Resource_Class class for Resource Class IBM.PSSP.SP_HW

        rcClass                 IBM.PSSP.SP_HW
        rcResource_monitor      IBM.PSSP.hmrmd
        rcObservation_interval
        rcReporting_interval

Information from EM_Structured_Byte_String class for Resource Variable IBM.PSSP.SP_HW.Node.lcd1

        sbsVariable_name        IBM.PSSP.SP_HW.Node.lcd1
        sbsField_name           STRING
        sbsField_type           cstring
        sbsField_SN
        sbsField_init_val       ""

Information from EM_Instance_Vector class for Resource Name IBM.PSSP.SP_HW.Node

        ivResource_name         IBM.PSSP.SP_HW.Node
        ivElement_name          NodeNum
        ivElement_description   256

Information from EM_Resource_Monitor class for Resource Monitor IBM.PSSP.hmrmd

        rmName                    IBM.PSSP.hmrmd
        rmPath                  /usr/lpp/ssp/bin/haemRM/hmrmd
        rmArguments               IBM.PSSP.hmrmd
        rmMessage_file          hmrmd_des.cat
        rmMessage_set           1
        rmConnect_type          server
        rmPTX_prefix
        rmPTX_description
        rmPTX_asnno
```

Figure 119. Sample Output from PSSP SDRGetObjects commands

Displaying Event Management Descriptions

As shown in Figure 119 on page 537, the description of resource variable **IBM.PSSP.SP_HW.Node.lcd1** is message number 244 of message set 1 of the **hmrmd_des.cat** message catalog.

To display the description, enter:

```
dspmsg -s 1 hmrmd_des.cat 244
```

In response, the system displays:

```
"(string) lcd1: 'LCD line 1'."
```

To list all of the messages in the catalog and direct the output to a file, enter:

```
dspcat hmrmd_des.cat > hmrmdcat
```

For more information on the AIX **dspmsg** and **dspcat** commands, see *AIX Version 4.1 Commands Reference*.

Getting Resource Variable Information from SP Perspectives

You can use the Event Management portion of SP Perspectives to get information about the resource variables and instance vectors that have been defined in the system. To do this:

Start the Event Management portion of SP Perspectives.

Enter:

```
spevent
```

If you have more than one system partition defined, click on the icon that represents the system partition in which you are interested.

Open the Create Condition dialog box.

From the Menu bar of the Event Perspective, click on the Actions button and, from the cascading menus, select Event Definitions and then Create. In response, the system displays a notebook that is open to the Create Event Definition page.

On the Create Event Definition page, click on the Create Condition button. In response, the system displays the Create Condition dialog box.

Click the down arrow next to the Resource Variable Name field. In response, the system displays a list of all of the resource variables that are defined in the system partition you selected.

To get information about any particular resource variable, select it. In response, the system displays the resource variable's name in the Resource Variable Name field, its description in the Resource Variable Description field, and the format of the instance vector in the Resource Identifier Format field.

For an example of the Create Condition dialog box after the **IBM.PSSP.Prog.pcount** resource variable has been selected, see Figure 120 on page 539.

Note that SP Perspectives uses slightly different terminology than the Event Management APIs, as follows:

Where the Event Management APIs use the term "predicate," SP Perspectives uses "expression."

Where the Event Management APIs use the term "instance vector," SP Perspectives uses "resource identifier."

For complete information on using SP Perspectives, see the *PSSP Administration Guide* and the SP Perspectives help panels.

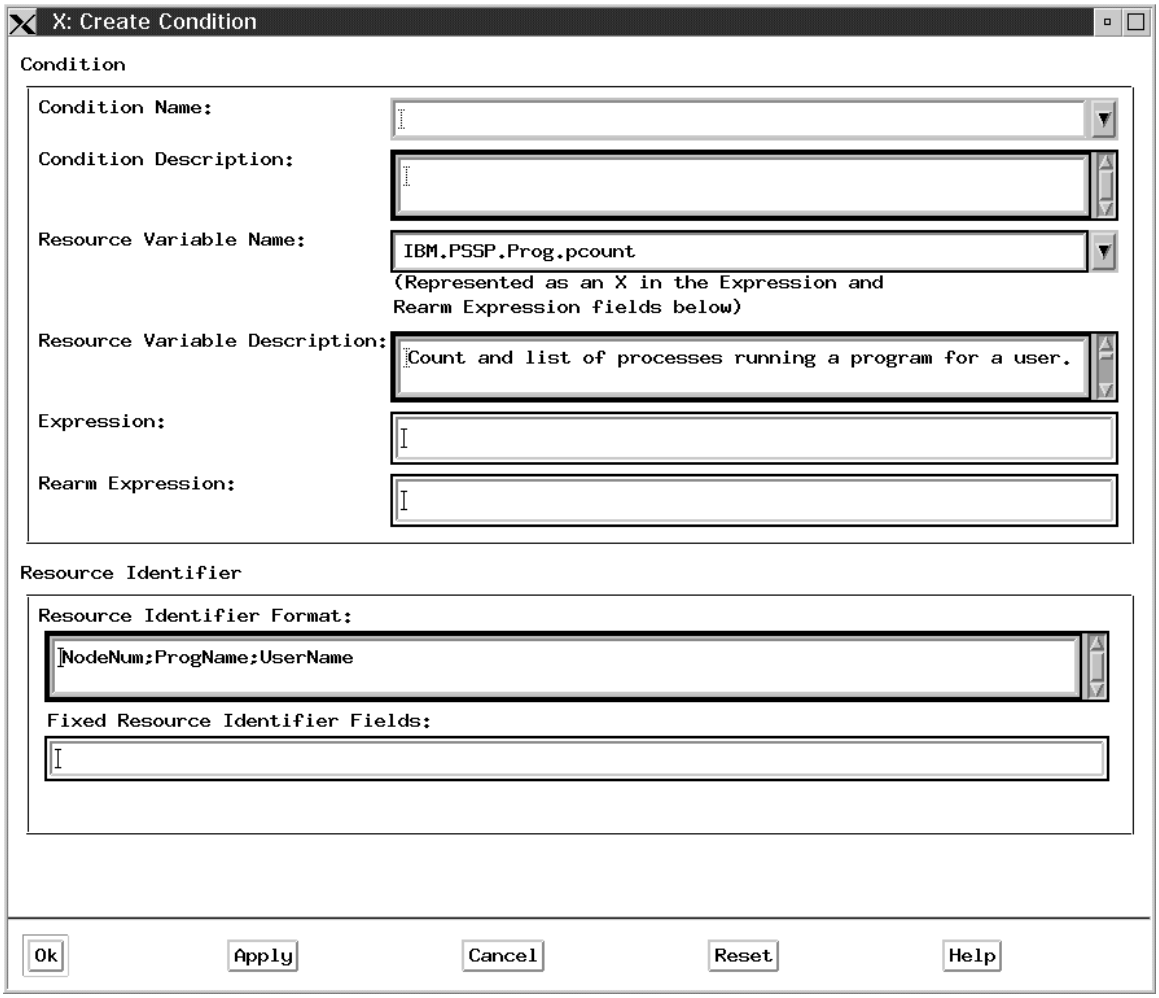


Figure 120. Using the Event Perspective to Get Resource Variable Information

Appendix F. The System Data Repository

The System Data Repository (SDR) is where your SP System configuration is stored. Information entered through the SMIT panels during installation goes into the SDR.

Understanding the System Data Repository

The SDR is composed of objects and their attributes. Objects differ according to class. An object belonging to, say, the node class, has a different set of attributes from an object in the adapter class.

Objects are *instances* of the object class. Some classes have many more objects than others. For example, there is only one object in the SP class for each SP System, but there are as many objects in the node class as there are nodes in the system.

An SDR object, however, is really only a specific set of attributes. Object attributes contain the meaningful data. To illustrate, Figure 121 compares two object classes in a sample configuration, the Frame class and the Adapter class.

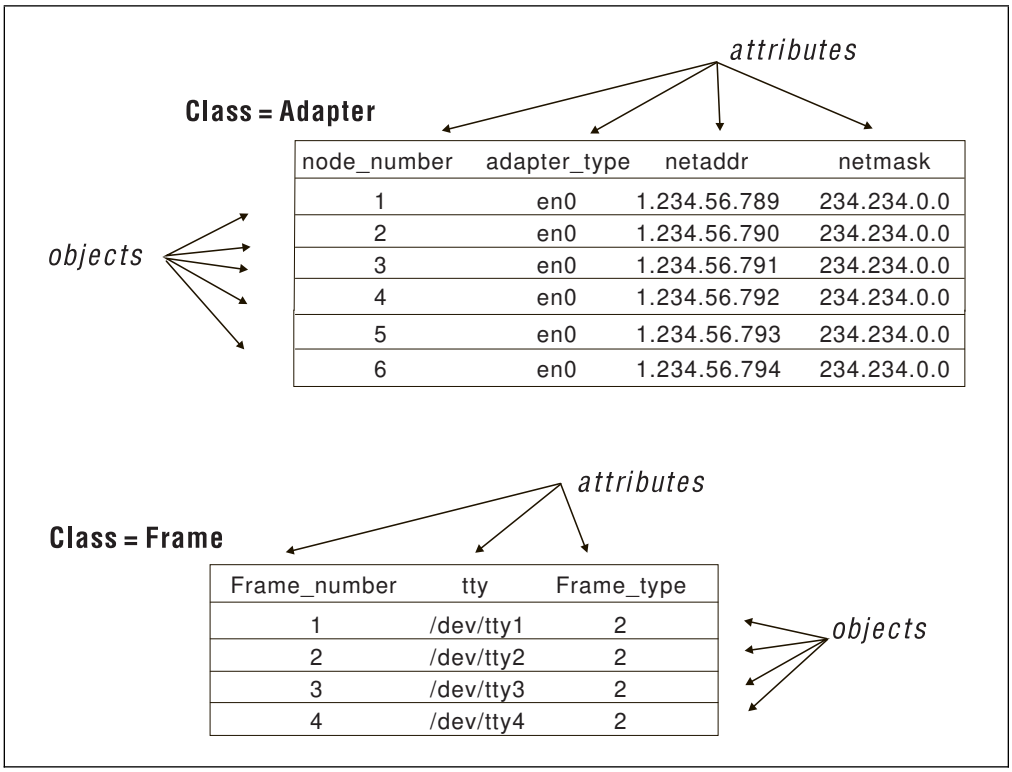


Figure 121. Comparing Two SDR Classes

The SP system provides *system partitioning* as a method for organizing the system into groups of nodes for various purposes such as testing new software and creating multiple production environments. Therefore there are two types of SDR classes: system and partitioned.

System classes (contain objects that are common to all system partitions)

© Copyright IBM Corp. 1996

- Class = Frame
- Class = SP
- Class = SP_ports
- Class = Switch
- Class = SysNodeGroup
- Class = Syspar_map

Partitioned classes (contain objects that are specific to a system partition. The *definition* of a Partitioned class is common to all system partitions, but each system partition contains different objects for the class.)

- Class = Node
- Class = Pool
- Class = Adapter
- Class = host_responds
- Class = switch_responds
- Class = Dont_care_pool_list
- Class = JM_server_nodes
- Class = JM_domain_info
- Class = JM_job_info
- Class = JM_node_usage
- Class = Switch_partition
- Class = VSD_Global_Volume_Group
- Class = HSD_Table
- Class = VSD_Table
- Class = VSD_Minor_Number
- Class = HSD_Minor_Number
- Class = GMT_Global_smt_nds
- Class = Syspar
- Class = pmandConfig
- Class = pmanrmdConfig
- Class = NodeGroup
- Class = TS_Config
- Class = Syspar_ports

For a complete list of attributes for each object, see "System Data Repository Classes and Attributes" on page 552.

In addition, there is a set of partitioned classes that are used by Event Management application programs:

Class = EM_Resource_Variable

Class = EM_Structured_Byte_String

Class = EM_Instantiation_Vector

Class = EM_Resource_Class

Class = EM_Resource_Monitor

For more information about these classes and their attributes, see *IBM Parallel System Support Programs for AIX: Event Management Programming Guide and Reference*.

Accessing the System Data Repository

There are two methods you can use to access the data in the SDR: SMIT panels and the SDR command line interface. The SDR command line interface is used by various SP commands. There should be no need to use the SDR command line directly and you should not do so unless you are instructed to by an IBM representative. In addition, many system management commands, such as **spbootins**, manipulate SDR data. You must be authorized in order to access the SDR. The SDR also performs authentication before allowing read-write access.

Authorization

The SDR has two levels of authorization, read-write and read-only. Table 20 shows the conditions under which each level of authorization is given.

Table 20. SDR Authorizations

Machine	Root User	Non-Root User
Control Workstation	read-write	read-only
SP Node	read-write	read-only
Other	read-only	read-only

Command entries made without necessary authorization will fail.

Authentication

Read-write authorization is given after authentication. This consists of checking the user ID. If a root user is running on the same machine as the SDR (the control workstation), then read-write authority is given. For a root user who is not on the control workstation, the SDR server checks that the adapter connecting the client machine has an adapter class object in the SDR. If so, the user is given read-write authority. If not, the user gets read-only authority.

All nonroot users get read-only authority.

Locating the SDR Server

There are three ways in which a process can locate the SDR server. They are selected in the following order of preference but when one method is selected, no other methods are attempted. This means that if method 1 is available, only method 1 is attempted. If it fails, methods 2 and 3 are *not* attempted.

1. The destination is passed as the third parameter in the **SDROpenSession** command. This only works for library routines and therefore may be used by some SP subsystems, such as the Resource Manager. The destination is the hostname or TCP/IP address of the control workstation where the SDR runs.
2. The **SP_NAME** environment variable is set to the hostname or TCP/IP address of the control workstation where the SDR runs.
3. The **/etc/SDR_dest_info** file is present and has the **primary** record set to the hostname or TCP/IP address of the control workstation where the SDR runs. This file is installed on the control workstation and all SP System nodes along with the IBM Parallel System Support Programs for AIX.

The SDR_dest_info File

The **/etc/SDR_dest_info** file is created on the control workstation at system installation, and propagated to all nodes in the SP system. The **/etc/SDR_dest_info** file has the following format:

```
 comments have an asterisk in column 1
default: <default system partition>
primary: <TCP/IP address of control workstation>
```

The **default** record identifies the default system partition. The default record is used at boot time so that the node can determine if it has changed system partitions.

Only the **primary** record is used by the SDR to locate the control workstation where the SDR server runs.

The SDR Daemon Log

The SDR daemon writes information to a log named **/var/adm/SPlogs/sdr/sdrdlog.***syspar_ip_addr.pid*, where *syspar_ip_addr* is the IP address of the system partition and *pid* is the process ID of the SDR daemon (**sdrd** process). This log will contain the date and time the process started, as well as problems encountered by the daemon in the course of operation.

Method 1 — Using SMIT

As displayed on the SP System Management SMIT Panel, the SMIT interface offers menu choices that provide different paths to the configuration data in the SDR.

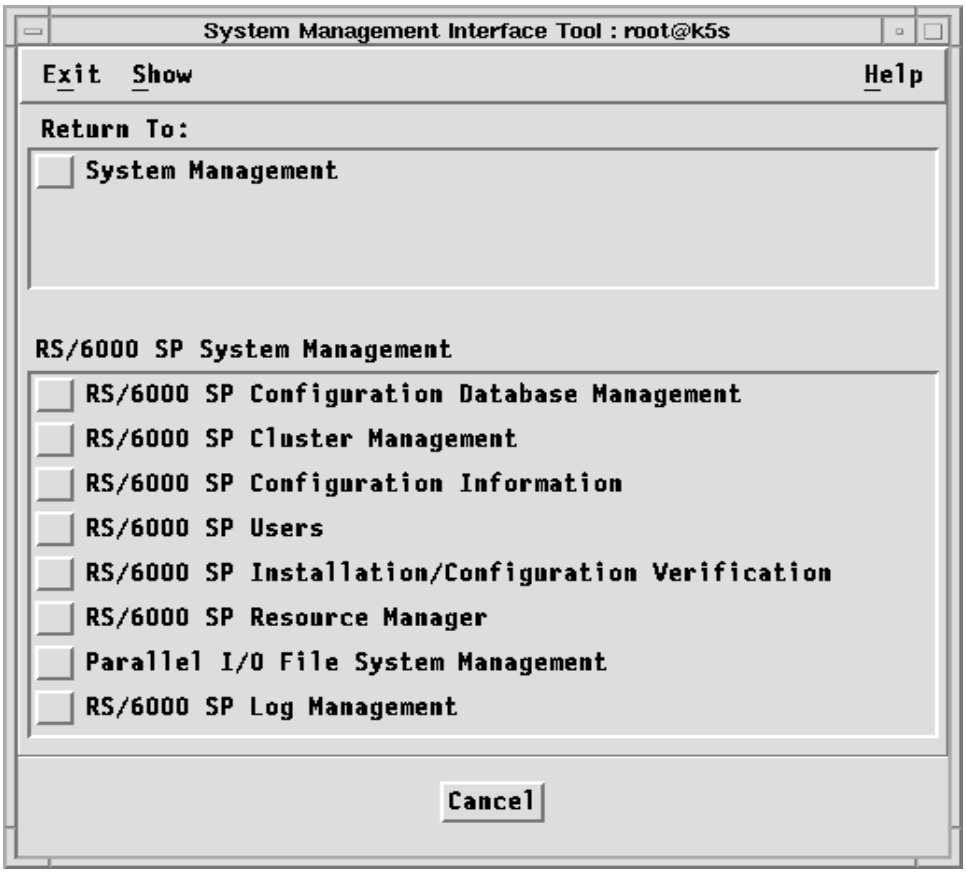


Figure 122. SP System Management SMIT Panel

These menu choices offer the following functions:

1. RS/6000 SP Configuration Database Management

   The dialogs available through this path allow you to enter, list, and change information during the installation process about your nodes, primary and secondary external LANs, and switch connections, as well as site environment information. These tasks are explained in the *IBM Parallel System Support Programs for AIX: Installation and Migration Guide*.

   The choices from this menu are:

   Enter Database Information

   - *Site Environment Information*
   - *Frame Information*
   - *Node Database Information*
   - *Node Group Information*
   - *System Partition Configuration*
   - *Run setup_server Command*
   - *VSD Database Information*

   List Database Information

   - *List Site Environment Database Information*
   - *List Frame Database Information*
   - *List Node Database Information*
   - *List System Partition Database Information*
   - *List LAN Database Information*

- *List VSD Database Information*

Delete Database Information

- *Delete Frame Information*
- *Delete Node Information*
- *Delete Adapter Information*
- *Delete Node Group Information*
- *Delete VSD Information*

2. RS/6000 SP Cluster Management

   The choices from this menu are:

   Run setup_server Command
   Perform Switch Operations
   VSD Management
   HSD Management

3. RS/6000 SP Configuration Information

   Using this path, you can display configuration information about your nodes, networks, file systems, and paging spaces.

   The choices from this menu are:

   List Node Hardware Information
   List Node Network Information
   List Node File System Information

4. RS/6000 SP Users

   Using this path, you can add and delete users, as well as change user management attributes such as passwords.

   The choices from this menu are:

   Add a user
   Change/Show Characteristics of a User
   Remove a User

5. RS/6000 SP Installation/Configuration Verification

   Use this path to check that your IBM Parallel System Support Programs for AIX options are installed correctly.

   The choices are:

   System Monitor Installation
   System Monitor Configuration
   System Data Repository
   System Management
   Communication Subsystem
   System Partition Configuration
   Switch Table API Installation
   Resource Manager Installation
   Resource Manager Configuration

6. RS/6000 SP Resource Manager

   The choices are:

   Change/Show Configuration Data
   Start the Resource Manager
   Reconfigure the Resource Manager

Stop the Resource Manager

7. RS/6000 SP Log Management

   The choices are:

   AIX Error Log
   Syslog
   General Log Viewing
   Archive Logs
   Collect Logs for Service

The menu selections run standard AIX commands and, in some cases, add information from the Hardware Monitor and reformat the output for useability. Let's examine the options on the list menu in more detail.Figure 123 contains options that invoke the AIX commands listed in Table 21 on page 548.

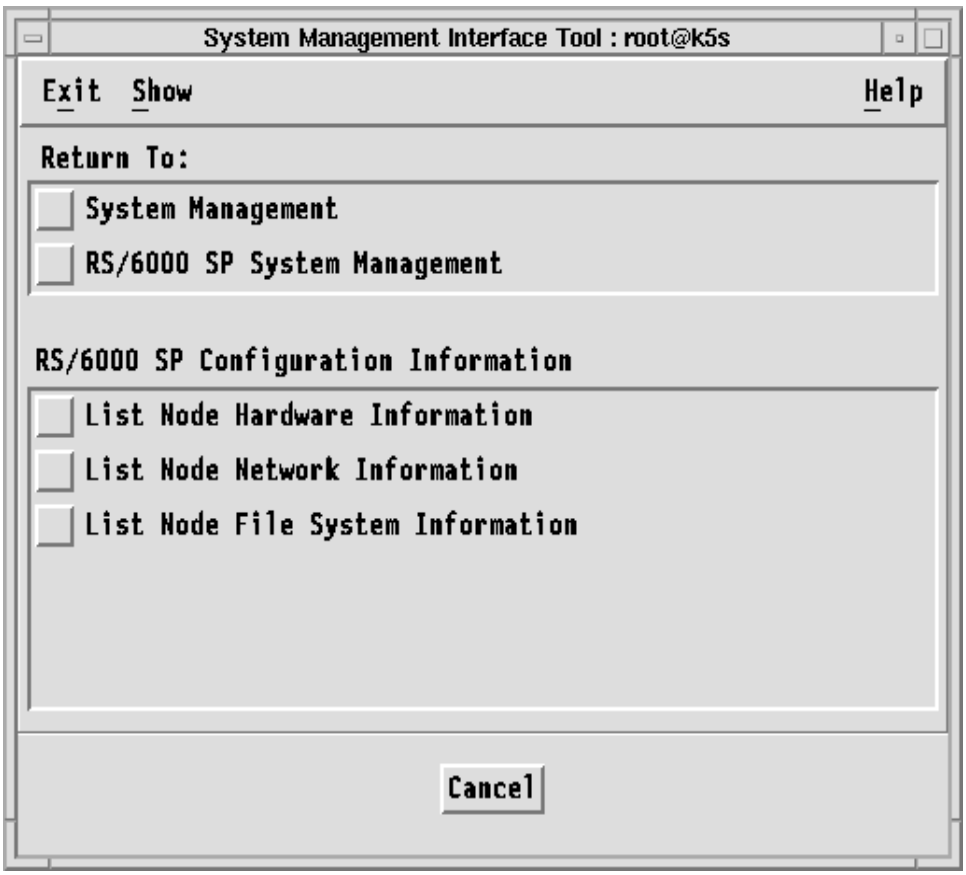


Figure 123. List RS/6000 SP Configuration Information

Table 21. Commands Invoked by SMIT Panels	
Select:	**To:**
List Node Hardware Information	Run the AIX **lscfg** command to display the name, location, and description of the devices related to the nodes
List Node Network Information	Run the AIX **netstat -in** command to show the state of the nodes' configured interfaces
List Node File System Information	Run the AIX **df** command to display the total spaces and available space on the node file systems

SELECT Any of the three choices

SMIT runs the corresponding AIX command for all the nodes and displays the information in a scrollable window.

Figure 124 shows the display for *List Node File System Information*.

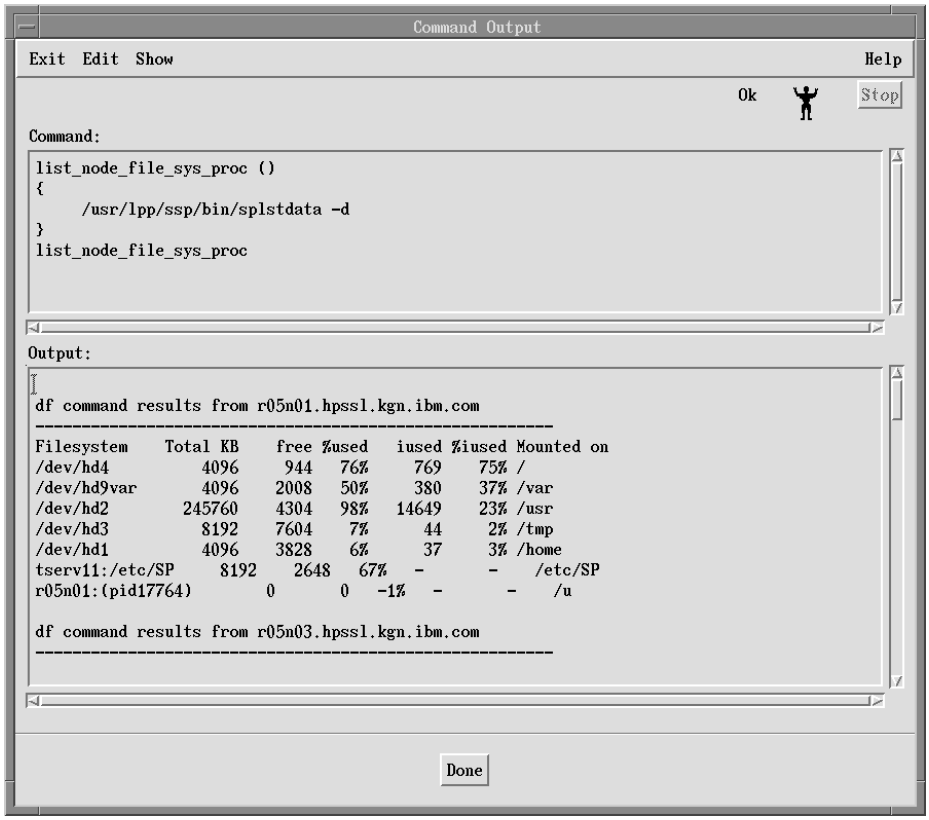


Figure 124. List Node File System Information Panel

You can also list the configuration data using the **splstdata** command. Refer to the *IBM Parallel System Support Programs for AIX: Command and Technical Reference* for complete syntax and examples.

Method 2 — - Using the SDR Command Line Interface

A command line interface allows you to display, change, or delete the contents of an SDR object without invoking SMIT panels.

These commands are used by the IBM Parallel System Support Programs for AIX to operate on SDR data. You should not need to use these commands directly. Should you choose to use them, do so with caution. SDR contents can be corrupted or made inaccessible.

The following list briefly describes these commands. See the *IBM Parallel System Support Programs for AIX: Command and Technical Reference* for exact syntax.

SDRAddSyspar
The IBM Parallel System Support Programs for AIX use this command to create a new daemon using the System Resource Controller (SRC).

SDRArchive
The IBM Parallel System Support Programs for AIX use this command to create an archive file containing all current SDR classes attributes.

SDRChangeAttrValues
The IBM Parallel System Support Programs for AIX use this command to change the attribute values of an existing object.

SDRClearLock
The IBM Parallel System Support Programs for AIX use this command to unlock a class that is locked, regardless of who has the lock. This is for system administration use only and should be used with caution.

SDRCreateAttrs
The IBM Parallel System Support Programs for AIX use this command to create new attributes for an SDR class.

SDRCreateClass
The IBM Parallel System Support Programs for AIX use this command to create a new class of objects and its attributes.

SDRCreateFile
The IBM Parallel System Support Programs for AIX use this command to create an SDR file from an AIX file.

SDRCreateObjects
The IBM Parallel System Support Programs for AIX use this command to create one or more new objects and define their attribute values.

SDRCreateSystemClass
The IBM Parallel System Support Programs for AIX use this command to create a system class.

SDRCreateSystemFile
The IBM Parallel System Support Programs for AIX use this command to create a file that can be retrieved from any system partition.

SDRDeleteFile
The IBM Parallel System Support Programs for AIX use this command to delete an SDR file.

SDRDeleteObjects
The IBM Parallel System Support Programs for AIX use this command to delete target objects.

SDRGetObjects
The IBM Parallel System Support Programs for AIX use this command to query the values of target objects and attributes and prints them to stdout.

SDRListClasses
The IBM Parallel System Support Programs for AIX use this command to list the class names in the SDR.

SDRListFiles
The IBM Parallel System Support Programs for AIX use this command to first list all the files in the system area, then list all the files in the system partition area.

SDRMoveObjects
The IBM Parallel System Support Programs for AIX use this command to move objects from one system partition to another.

SDRRemoveSyspar
The IBM Parallel System Support Programs for AIX use this command to remove the entire contents of the subdirectory under system partitions. It uses the SRC to remove the daemon that serves the system partition.

SDRReplaceFile
The IBM Parallel System Support Programs for AIX use this command to replace the specified SDR file with the specified AIX file.

SDRRestore
The IBM Parallel System Support Programs for AIX use this command to overwrite the current SDR with the contents of an archived SDR file.

SDRRetrieveFile
The IBM Parallel System Support Programs for AIX use this command to create an AIX file from an SDR file.

SDR_test
The IBM Parallel System Support Programs for AIX use this command to verify that the installation and configuration of the SDR completed successfully.

SDRWhoHasLock
The IBM Parallel System Support Programs for AIX use this command to query the lock transaction ID for a specified object class.

Updating the host_responds Class (the hrd Daemon)

The **host_responds** class in the SDR is updated automatically by the **hrd** daemon. The **hrd** daemon is called from the **hr** script, which is under SRC control. **hr** will respawn if it, or the **hrd** daemon, is killed. **hr** and **hrd** run on the control workstation. Both the **hr** script and the **hrd** daemon are in directory **/usr/lpp/ssp/bin**. (The **hrctrl** script provides the same function as the **hr** script, but it follows the syntax of the **syspar_ctrl** command.)

The **hrd** daemon monitors the SP nodes in one of two ways. The default method is to use the system heartbeat (**hbd** daemon). The **hrd** daemon communicates with the **hbd** daemon on the control workstation, getting updates about the status of the nodes. The **hbd** runs on the control workstation and all of the SP nodes.

The second method by which **hrd** can get node status is to use **fping**. **fping** is a program that runs **ping** to many nodes asyncronously. This second method also issues **snmpinfo** calls to each node, to detect a situation where **ping** succeeds but the node is not responding to user requests.

The **hr** script determines which method will be used. An environment variable named HR_FPING is set to **0** for heartbeat or **1** for **fping**. The default is **0**.

The line that chooses between the methods looks like this:

```
typeset -x HR_FPING=
```

After making a change to this file, you should reset the **hrd** daemon by issuing **hr reset**. This will cause it to respawn and the change will be put in effect. Other options for configuring **hrd** are documented in comments in the **hr** script (in **/usr/lpp/ssp/bin**).

The reason for the second method (**fping**) is that it has different characteristics than the system heartbeat. It does not require a daemon to be on each node, so it may reduce cpu usage on the nodes by a small amount.

Backing Up the SDR

Backing up the SDR regularly is a good way to insure against the loss of your data. Whenever you change your SP system configuration, use the **SDRArchive** command to take a snapshot of the SDR contents and save it in a **tar** file. The *IBM Parallel System Support Programs for AIX: Command and Technical Reference* contains syntax and usage information.

You can restore a saved image of the SDR using the **SDRRestore** command. Before restoring the SDR, be sure to stop the Resource Manager. (Refer to Chapter 7, “Managing SP Resources” on page 85.)

SDR Shadow Files

The SDR classes are stored in subdirectories under directory **/spdata/sys1/sdr**. The subdirectories of interest are **/system/classes** and **/partitions/***partition_ip_address***/classes**. The names of the files in those directories correspond to the names of the classes.

Before SDR commits changes to a class, the class contents are moved to a backup file in the same directory as the class. The backup file is named *class*.**shadow**, where *class* is the name of the class being written. If a power loss occurs or the SDR daemon is killed while the SDR class is being written, the class file that was being written at the time may not exist or may be corrupted. You should check for the existence of a *class*.**shadow** file. If one exists, the following steps should be taken to restore the class.

1. The corrupted *class* file (if one exists) should be removed.
2. The *class*.**shadow** file should be renamed to *class*.
3. The SDR daemon (**sdrd**) should be restarted. This can be accomplished by issuing **sdr -g sdr reset**.

The SDR daemon will restart automatically and recognize the renamed shadow file as the class.

System Data Repository Classes and Attributes

The tables below show all the SDR classes. You can specify these classes as targets in **SDRxxxxx** commands. The value in the **Type** column is the datatype of the attribute: S=string, I=integer, F=floating point.

Class = Node

Attribute Name	*Type*	*Description*	*Comments*
node_number	I	Relative node number ((frame#-1)*16)+slot	This number is computed from frame info and put here by system management.
hdw_enet_addr	I	Universal hardware ethernet identifier for this node.	Obtained via system monitor during system initialization.
frame_number	I	Frame number	Number from 1 to number of frames. Set during object creation.
slot_number	I	Node slot in frame	Number from 1 to number of nodes. Set during object creation.
slots_used	I	Nodes use 1, 2 or 4 slots. Distinguishes thin, wide or 604 High Node nodes.	Set by data obtained from system monitor.
switch_node_number	I	Switch relative node number.	Set during object creation from data obtained from system monitor.
switch_chip_port	I	Switch port this node is connected to.	Set during object creation from data obtained from system monitor.
switch_chip	I	Switch chip number this node is connected to.	Set during object creation from data obtained from system monitor.
switch_number	I	Switch number this node is connected to.	Set during object creation from data obtained from system monitor.
initial_hostname	S	Hostname assigned to this node during customization.	Customer supplied from CMI at installation.

Attribute Name	*Type*	*Description*	*Comments*
reliable_hostname	S	Hostname associated with SP ethernet.	Set during CMI processing from name entered for en0 interface name.
default_route	S	IP address of default route.	Customer supplied from CMI at installation.
boot_server	I	Relative node number of node that will answer bootp request when necessary. (0 represent control workstation.)	Set during creation to be node 1 of frame. For node 1 in frame set to 0 to represent control workstation.
bootdisk	S		
install_image	S	Path of net install image for this node. This field is initially set to **DEFAULT**, meaning that the default image is used.	Optionally setable by customer from CMI interface.
install_disk	S	The hard disks on which the install image is installed	
last_install_time	S	Time and date that this system was last successfully installed	Set during network installation.
last_install_image	S	server:/filepath of last image installed	Set during network installation.
switch_protocol	S	This is the string **LSP** or **CSSCI**	
switch_partition_number	I	Switch partition this node belongs to	Default is **1**
bootp_response	S	**install**=full net install and customize; **disk**=use local disk to boot (ignore bootp request); **maintenance**=on reboot, go into Installation/Maintenance menus; **customize**=just use information from SDR to customize node's ODM; **restore**=restore of backup image via network install	Initially set to **customize** during object creation. Changeable through CMI interface.
boot_device			
usr_maint	S	If set to **true** a node will perform usr client maintenance on boot	Initialized to **false**, set through CMI

Attribute Name	*Type*	*Description*	*Comments*
JM_domain_id	S	Identifier for a JM domain	
VSD_adapter	S	The adapter on this node used by VSD	This is an adapter type to be joined with the adapter_type attribute of the Adapter class, where the node and Adapter node_number fields match.
VSD_max_buffer_count	I	Buffer cache size for vsd	
VSD_request_blocks	I	The number of request blocks the VSD should allocate	
VSD_pbufs	I	The number of pbufs the VSD should allocate	
VSD_init_buffer_count	I	Buffer cache size for VSD	
VSD_min_buddy_buffer_size	I	The minimum VSD buddy buffer size allocated to a single request	
VSD_max_buddy_buffer_size	I	The maximum VSD buddy buffer size allocated to a single request	
VSD_max_buddy_buffers	I	The size of the VSD buddy buffer, given in the number of buffers of max_buddy size.	
cfs_adapter	S	The adapter on this node used by Calypso and Global Mount	
acct_class_id	S	An ID selected by administrator for grouping and merging acct data.	Default = **default**
acct_enable	S	Whether acct is enabled for a node	Default = **default**
acct_job_charge	S	Used to determine the number of charge fee units to charge a user for exclusive use of the node. Its value is in units of "seconds per charge fee unit"	Default = **1.0**

Attribute Name	*Type*	*Description*	*Comments*
acct_excluse_enable	S	Indicates whether stop and end job accounting records will be generated for jobs having exclusive use of a node.	Default = **false**
usr_server_ip	S	This is either an IP address or **local**	Default is **local**
usr_client_adapter	S	This will be **en0**, **en1**, **tr0** or **tr1**	Default is **en0**
has_usr_clients	S	This will be either true or false	Default is false
code_version	S	Contains the ssp release that is installed on this node.	Default is nothing (unknown)
usr_gateway_ip	S	The IP address of the gateway to get to /**usr** server	Default is **0**, meaning /**usr** server is not on a gateway.
VSD_maxIPmsgsz	I	Largest sized block of data that the IBM Virtual Shared Disk sends over the network for an I/O request	Default is **24576**
processor_type	S	Specifies if the node is a uniprocessor or multiprocessor.	**UP** or **MP**
processors_installed	I	The total number of processors in the node.	

Class = Frame

Attribute Name	*Type*	*Description*	*Comments*
frame_number	I	Frame number	
tty	S	The tty port to be used for this frame (like /dev/tty1)	From customer via CMI during configuration
frame_type	S	Switch or no switch	From information returned by the monitor.
MACN	S	Stands for Monitor and Control Node	Hostname of the machine that monitors this frame

Attribute Name	*Type*	*Description*	*Comments*
backup_MACN	S	The backup monitor and control node for hardware monitoring with High Availability Control Workstation Connectivity function.	
slots	I	Frame height	A full-size frame is 79" (2.00 meters). An SP Switch-8 frame is 49" (1.25 meters).

Class = Pool

Attribute Name	*Type*	*Description*	*Comments*
pool_id	I	The ID of the pool this node belongs to.	Identifies the pool a node is in.
node_name	S	A name that can be used to contact the node.	Comes from node_type_file.
pool_type	I	InteractiveOnly, BatchOnly, or General (1,2, or 3)	
attribute	S	User description of pool	Customer description of pool

Class = Switch

Attribute Name	*Type*	*Description*	*Comments*
switch_number	S	Switch Number	If less than 1000, switches connected to nodes. If greater than 1000, intermediate switches.
frame_number	I	Frame number	
slot_number	I	Slot number within frame	Node switches equal 17. Intermediate switches equal 2,4,6,8,10,12,14,16.

Attribute Name	*Type*	*Description*	*Comments*
switch_partition_number	I	Switch partition that the switch is on.	Default is **1**
switch_type	I	Switch supervisor card type (49, 50, or 129)	
clock_input	S	Clock input source	High Performance Switch possible values: 0, 1, 2, 3 SP Switch possible values: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 27, 28, 29, 30, 31, 32, 33, 34
switch_level	I	Microcode level from the switch supervisor	
switch_name	S	Short model name of the switch	
clock_source	I	Switch number of the switch providing the clocking signal for the current switch	
clock_change	S	Clock input has changed since last **Eclock**	

Class = Switch_partition

Attribute Name	*Type*	*Description*	*Comments*
switch_partition_number	I	The partition number for High Performance Switch use	
topology_filename	S	The filename of the topology file that is stored in the SDR	
primary_name	S	The reliable hostname of the switch primary node	
arp_enabled	S	Address resolution protocol	Default is **no**
switch_node_number_used	S	Switch adapter's IP addresses were assigned based on the switch node numbers	Default is **yes**
run_phase_duration	I	Length of time in minutes that the switch stays in run phase. Range is 1–117281 minutes.	Default is **2**

Attribute Name	*Type*	*Description*	*Comments*
primary_backup_name	S	Reliable host name of this system partition's primary backup node	
oncoming_primary_name	S	Reliable host name of the node that will become the primary node on the next **Estart**	
oncoming_primary_backup_name	S	Reliable host name of the node that will become the primary backup node on the next **Estart**	
num_nodes_success	I	Number of nodes in the system partition that successfully received the topology file	

Class = Adapter

Attribute Name	*Type*	*Description*	*Comments*
node_number	I	Relative node number	
adapter_type	S	Adapter type as recorded in ODM (lsp0, enet0, enet1, fi0, fi1 etc.)	Set via customer through CMI. Only en0 is required.
netaddr	S	ip address for this interface.	Set via customer through CMI.
netmask	S	netmask for this interface.	Set via customer through CMI.
tok_rate	I	Speed of token rate, (4 or 16 mb)	
enet_type	S	**bnc** or **dix** or **N/A**	

Class = host_responds

Attribute Name	*Type*	*Description*	*Comments*
node_number	I	Relative node number	This attribute gets mapped to frame/slot to retrieve DE value
host_responds	I	0 or 1 to indicate host is down or up.	This is really a query into the frame info written by the System Monitor.

Class = switch_responds

Attribute Name	*Type*	*Description*	*Comments*
node_number	I	Relative node number.	
switch_responds	I	0 or 1 if switch is inactive/active on the node.	This is written by switch code.
autojoin	I	Should this node join switch on startup.	
isolated	I	**1** or **0** if this node is isolated from the switch or not	
adapter_config_status	S	Status of the switch adapter in the node	

Class = SP

Attribute Name	*Type*	*Description*	*Comments*
control_workstation	S	Hostname of control workstation	
cw_ipaddrs	S	IP addresses of control workstation	
install_image	S	Full path of net install image.	Customer supplied from CMI at installation
remove_image	S	If set to **true** the install images are erased after all nodes have been installed from this server.	Customer supplied from CMI at installation
primary_node	S	Ethernet hostname of primary node.	Default is node 1 frame 1.

Attribute Name	*Type*	*Description*	*Comments*
ntp_config	S	The kind of NTP installation done: **none**=Do not configure NTP for this site. **consensus**=setup NTP to configure Control Workstation as the NTP server and boot servers as NTP peers. **timemaster**=site has an existing NTP server. Configure NTP to user these. NTP_SERVERS will contain the NTP server hostnames. **internet**=The Control Workstation has access to the internet. Configure the Control Workstation to be an NTP server using the internet time server(s) defined in NTP_SERVERS.	From customer input from CMI.
ntp_server	S	Hostname of NTP server.	From customer input from CMI
ntp_version	S	Default is 3.	Customer can override using CMI
amd_config	S	**True** or **False** for whether amd will be installed	Customer can input in CMI, default is false.
print_config	S	**false**=do not print **secure**=install print in secure mode **open**=install print in open mode	Default is false, changed from customer via CMI
print_id	S	User name to rsh to for secure mode printing	From customer via CMI
usermgmt_config	S	**false**=do not install SP user management **true**=install SP user management code and SMIT interface	Default is false, changed from customer via CMI
passwd_file	S	File containing password information (/etc/passwd, or other)	Default is /etc/passwd
passwd_file_loc	S	Location of password file	
homedir_server	S	Hostname of default user directory server for user management code	From customer via CMI, default is control workstation

Attribute Name	*Type*	*Description*	*Comments*
homedir_path	S	Default path to user home directories.	From customer via CMI, default is /home/$homedir_server
filecoll_config	S	**True** or **false** if file collection management code should be installed	From customer via CMI, default is false
supman_uid	S	UID for supman	
supfilesrv_port	I	File collections port number	
spacct_enable	S	Flag indicating whether, by default, accounting is enabled on all nodes that have an accounting enabled attribute of **default**	default is **false**
spacct_actnode_thresh	I	The SP accounting active node threshold indicates the percentage of nodes for which accounting data must be available, for merging and reporting of the data for a cycle to take place.	default is 80
spacct_excluse_enable	S	The cluster exclusive use accounting enabled attribute-indicates whether accounting start and end job records will be generated for jobs having exclusive use of the node.	True or false. Default is **false**.
acct_master	I	Specifies which node is to act as the accounting master.	default is 0
cw_has_usr_clients	S	Either **true** or **false**	Default is **false**
code_version	S	Contains the ssp release that is installed on the CWS	Default is nothing (unknown)
layout_dir	S	Pathname of layout directory for current system partition configuration	

Attribute Name	*Type*	*Description*	*Comments*
authent_server	S	The type of authentication server used. The possible values are **ssp** (SP authentication services) (**ssp.authent** server, either primary or slave, on the control workstation), **afs** (afs server on either the control workstation or elsewhere), **krb** (SP authentication services or other MIT Kerberos 4 server(s) on other workstations only).	The default is **krb**. The value is set by **install_cw**.
backup_cw	S	Kernel hostname of backup control workstation	If value is null, configuration is not HACWS
ipaddrs_bucw	S	IP addresses of backup control workstation that are not service adapters and are not involved with fail over	If value is null, the backup control workstation is not reachable as the backup during fail over periods
active_cw	S	Active control workstation, either control workstation or backup control workstation	If value is null, configuration is not HACWS

Class = Dont_care_pool_list

Attribute Name	*Type*	*Description*	*Comments*
pool_id	I	A pool in the don't care list.	
order_number	I	The position of the pool in the dont_care_pool_list	User must enforce no duplicates or holes.

Class = JM_server_nodes

Attribute Name	*Type*	*Description*	*Comments*
node_name	S	A name that can be used to contact the node.	Comes from node_type_file.
order_number	I	The position of this node in the JM_server_nodes list	User must enforce no duplicates or holes.

Class = JM_domain_info

Attribute Name	*Type*	*Description*	*Comments*
JM_domain_id	S	Identifier for a JM domain	
server_name	S	The node on which the JM server in this domain is running.	This may change as JM moves to a backup node.
pid	I	The process ID for the JM server in this domain.	
backup_name	S	The node that should server as backup to the JM server.	
backup_pid	I	The process ID of the JM server on the backup node.	Backup servers are always running. They just wake up when needed.
access_control	I	Whether authentication should be checked for user on node	0=**No**=Don't check, 1=**Yes**=Check
num_pools	I	Number of pools	
num_server_nodes	I	Number of JM server candidate nodes	
num_dont_care_pools	I	Number of don't care pools	

Class = JM_job_info

Attribute Name	*Type*	*Description*	*Comments*
job_id	I	Identifier for a JM job	Job ID for whole job. Subjobs identified by this ID + node.
group_id	I	The job group this job belongs to.	
client_node	S	The node name of the client that asked for the job.	This a hostname.
client_pid	I	The process ID of the client this job belongs to.	
client_uid	I	The user ID of the client this job belongs to.	
job_description	S	The job description associated with the job	Only set when **node_number=1**
adapter_type	I	The type of adapter the job is using: Ethernet, FDDI, and so on.	
pool_type	I	The type of request pool. Interactive = **0**, batch = **1**	Only set when **node_number=1**
req_auth	I	What action must be taken during deallocation	Only set when **node_number=1**
time_alloc	S	The time when the request was received by the server	Only set when **node_number=1**
node_number	I	Data about entire job for node number **1**. Data about specific node in the job for all other node numbers.	
cpu_usage	I	What type of usage was requested of node	Only set when node number is > 0.
adapter_usage	I	What type of usage was requested of node	Only set when node number is > 0.
num_virtual_tasks	I	The number of virtual tasks running on the node	Only set when node number is > 0.
virtual_task_ids	S	The virtual task IDs running on the node	Only set when node number is > 0.
rc	I	Indicates the return code associated with the node	Only set when node number is > 0.

Class = JM_node_usage

Attribute Name	*Type*	*Description*	*Comments*
node_number	I	The number of the node whose usage data is described in this object	
adapter_type	I	The adapter type whose usage data is described in this object	
current	I	How many jobs are currently using the specified CPU or adapter	
remaining	I	Indicates whether another job can use this CPU or adapter	**0** = no other job can use; **1** = other jobs can use if willing to share

Class = VSD_Global_Volume_Group

Attribute Name	*Type*	*Description*	*Comments*
global_group_name	S	Global Volume Group name	
local_group_name	S	Local Volume Group name	
primary_node	I	Primary node for this volume group	This is a relative node number.
secondary_node	I	Secondary node for this volume group	This is a relative node number and is OPTIONAL.

Class = HSD_Table

Attribute Name	*Type*	*Description*	*Comments*
minor_number	I	Unique minor numbers	
stripe_size	I	Stripe width	
num_vsds	I	Number of VSDs in HSD	

Attribute Name	*Type*	*Description*	*Comments*
option	S	Whether the HSD will skip the first stripe on each underlying IBM Virtual Shared Disk in an HSD	Value is **protect_lvcb** or **not_protect_lvcb**
HSD_name	S	Hashed Shared Device name	
VSD_name	S	Virtual Shared Device name	

Class = VSD_Table

Attribute Name	*Type*	*Description*	*Comments*
VSD_name	S	Volume group name	
global_group_name	S	Global Volume Group name	Same names as global_group_name in VSD_Global_Volume_group
logical_volume_name	S	The name of the logical volume comprising this vsd.	
minor_number	I	Minor number from minor table	
option	S	Options associated with this vsd	**cache** or **nocache** are the permissible values

Class = VSD_Minor_Number

Attribute Name	*Type*	*Description*	*Comments*
minor_number	I	Unique minor numbers	

Class = HSD_Minor_Number

Attribute Name	*Type*	*Description*	*Comments*
minor_number	I	Unique minor numbers	

Class = SP_ports

Attribute Name	*Type*	*Description*	*Comments*
daemon	S	Name of a daemon	
hostname	S	Host that the daemon runs on	
port	I	Port that the daemon uses on node specified in **hostname**	

Class = GMT_Global_smt_nds

Attribute Name	*Type*	*Description*	*Comments*
node_1	I	Global Mount uses this node as primary smart node.	
node_2	I	Global Mount uses this node as backup smart node.	
node_3	I	Global Mount uses this node as backup smart node.	
node_4	I	Global Mount uses this node as backup smart node.	
node_5	I	Global Mount uses this node as backup smart node.	

Class = Syspar_map

Attribute Name	*Type*	*Description*	*Comments*
syspar_name	S	Short hostname of a system partition	
syspar_addr	S	IP address of a system partition	
node_number	I	A node number in this system partition	
switch_node_number	I	The switch node number of this node	

Attribute Name	*Type*	*Description*	*Comments*
used	I	Is this node on IC bus of frame controller?	

Class = Syspar

Attribute Name	*Type*	*Description*	*Comments*
syspar_name	S	Hostname of system partition	
ip_address	S	IP address of system partition	
install_image	S	The image installed on this system partition	
syspar_dir	S	The directory from which this system partition was created	
code_version	S	IBM Parallel System Support Programs for AIX code version in this partition	
haem_cdb_version	S	Event Management Configuration Database (EMCDB) version string	

Class = pmandConfig

Attribute Name	*Type*	*Description*	*Comments*
pmNodenumber	I	The node number of the host on which **pmand** is to subscribe to the event.	If null, the **pmand** daemon on all nodes in the partition will subscribe to the event.
pmEventid	I	The ID for this event on this node as returned by the Event Manager	
pmTargetType	S	Identifies the type of the parameter in the **pmTarget** field.	**NODE_LIST**, **NODE_RANGE**, or **NODE_GROUP**

Attribute Name	*Type*	*Description*	*Comments*
pmTarget	S	Lists the targets on which daemons are to be configured. Can be a list of host names (**NODE_LIST**) separated by commas, a range of nodes (**NODE_RANGE**) as specified by **-n** argument to **hostlist**, or the name of a node group (**NODE_GROUP**).	A pre-defined node group, **ALLPMAN**, is provided. This node group consists of all the nodes in the current system partition.
pmRvar	S	Specifies the resource variable name for the event.	Required.
pmIvec	S	Specifies a fully-qualified instantiation vector for the event.	Required.
pmPred	S	Specifies a predicate for the event.	Required.
pmRearmPred	S	Specifies a predicate for the rearm event.	Optional.
pmCommand	S	The command to issue when one of the specified events occurs.	Optional.
pmCommandTimeout	S	The amount of time, in seconds, to wait before assuming that the **pmCommand** has hung, and killing it.	Optional.
pmHandle	S	An identifier (handle) for the event-action pair represented by this entry.	Required.
pmTrapid	I	Specifies an ID to be put into an SNMP trap to be sent when the event occurs.	Optional.
pmPPSlog	I	If non-zero, specifies logging via ppslog().	Optional.
pmRearmCommand	S	The command to issue when the rearm event occurs.	Optional.

Attribute Name	*Type*	*Description*	*Comments*
pmRearmCommandTimeout	S	The amount of time to wait, in seconds, before assuming the command has hung, and killing it.	Optional.
pmRearmTrapid	I	Specifies a specific id to be put into an SNMP trap to be sent when the rearm event occurs.	Optional.
pmRearmPPSlog	I	If non-zero, specifies logging via ppslog().	Optional.
pmUsername	S	The user name of the subscriber event.	Defaults to root.
pmPrincipal	S	The Kerberos principal of the requesting client (the client authorized to modify the subscription).	Required.
pmHost	S	The host from which the subscription was commanded.	Defaults to the local node.
pmActivated	S	Indicates whether a specific event-action pair is activated (subscribed to) or not.	Default is activated.
pmText	S	Optional text to be placed in the log record.	
pmRearmtext	S	Optional text to be placed into the log record.	
pmUserLabel	S	Stores the name of the **Named_Condition** from the Event Management subsystem, if one was used to subscribe to a problem management record.	Used by the SP Perspectives GUI.

Class = pmanrmdConfig

Attribute Name	*Type*	*Description*	*Comments*
pmrmNodenumber	I	The node number of the host on which **pmanrmd** is to instantiate the resource variable being defined.	
pmrmTargetType	S	Identifies the type of the parameter in the **pmTarget** field.	**NODE_LIST**, **NODE_RANGE**, or **NODE_GROUP**
pmrmTarget	S	Lists the targets on which daemons are to be configured. Can be a list of host names (**NODE_LIST**) separated by commas, a range of nodes (**NODE_RANGE**) as specified by the **-n** argument to **hostlist**, or the name of a node group (**NODE_GROUP**).	
pmrmRvar	S	Specifies the resource variable being defined.	Must be **IBM.PSSP.User_state***nn*, where *nn* is 1-16.
pmrmSampInt	I	Specifies a sampling interval, in seconds, for the resource variable.	
pmrmCommand	S	Specifies a command to run at the specified interval.	

Class = SysNodeGroup

Attribute Name	*Type*	*Description*	*Comments*
name	S	Name associated with a node group.	The first character of a node group name must be a letter. The remaining characters may be letters, numbers, . (period), or _ (underbar).

Attribute Name	*Type*	*Description*	*Comments*
defList	S	Nodes and node groups which are members of this node group.	For a node group named “A”, a list of nodes and node groups that belong to “A” directly (not through member node groups).

Class = NodeGroup

Attribute Name	*Type*	*Description*	*Comments*
name	S	Name associated with a node group.	The first character of a node group name must be a letter. The remaining characters may be letters, numbers, . (period), or _ (underbar).
defList	S	Nodes and node groups which are members of this node group.	For a node group named “A”, a list of nodes and node groups that belong to “A” directly (not through member node groups).

Class = TS_Config

For the default settings of the attributes in this class, see the man pages for the **hbctrl** and **hatsctrl** commands in *PSSP Command and Technical Reference*.

Attribute Name	*Type*	*Description*	*Comments*
Frequency	I	The frequency, in heartbeats per second, with which heartbeats are sent out	
Sensitivity	I	The number of heartbeats from the neighboring node that can be missed before the neighbor is declared inoperative	

Attribute Name	*Type*	*Description*	*Comments*
Run_FixPri	I	Whether the execution priority is fixed or floating (1=fixed, 0=floating)	
FixPri_Value	I	The value of the execution priority that is used on the **set_priority** system call	

Class = Syspar_ports

Attribute Name	*Type*	*Description*	*Comments*
subsystem	S	Name of a system partition-sensitive subsystem	
port	I	Port number used for intrasubsystem communication	

Appendix G. IP Address and Host Name Changes for SP Systems

The information in this appendix describes the activities that the administrator needs to perform when changing either the primary Ethernet IP address or host name for the SP nodes, or the primary IP address and host name for the control workstation. We strongly recommend that you avoid making any host name or IP address changes if possible. The tasks are tedious and in some cases require rerunning the installation steps. For more information on SP networks, reference the section “Planning your Network” located in *IBM Parallel System Support Programs for AIX: System Planning*.

Changing the IP address or host names of SP nodes affect the entire SP system. IP address and host names are located in the System Data Repository (SDR) using objects and attributes. IP address and host names are also kept in system-related files that are located on SP nodes and the control workstation. It is imperative that all new IP addresses and host names being used are resolvable on your SP system.

The following PSSP components must be considered when changing the IP address and host names:

- Network Installation Manager (NIM)
- System partitioning
- IBM Virtual Shared Disk
- High Availability Control Workstation (HACWS)
- Availability Services
- Problem Management subsystem
- Performance Monitor Services

SDR Objects with IP Address or Host Names

The following SDR objects reference the host name and IP address in the SP system for PSSP systems:

Adapter	Specifies the IP addresses used with the **ent**, **css0**, **tr0**, and **fi0** adapters
Frame	Specifies the Monitor and Control Node **MACN**, and HACWS **backup_MACN** attributes on the control workstation that work with host names
JM_domain_info	Works with the host names for Resource Manager domains
JM_Server_Nodes	Works with the host names for Resource Manager server nodes
Node	Works with the initial or reliable host names and uses the IP address for SP nodes and boot servers. The nodes are organized by system partitions.
Pool	Works with host names for Resource Manager pools

© Copyright IBM Corp. 1996

SP	Works with control workstation IP address and host names. Uses the host name when working with Network Time Protocol (NTP) printing, user management, and accounting services
SP_ports	Works with the host name used with **hardmon** and the control workstation
Switch_partition	Works with host name for primary and backup nodes used to support the **css** SP switch
Syspar	Works with the IP address and SP_NAME with system partitions
Syspar_map	Provides the host name and IP address on the control workstation for system partitions

The following SDR objects are added for PSSP 2.2 systems:

pmandConfig	Captures the SP node host name data working problem management
SPDM	Works with host name for Performance Monitor status data
SPDM_NODES	Works with host name for SP nodes and organized by system partition

SP System Files with IP Address and Host Names

The following files contain the IP address or host name that exist on SP nodes and the control workstation. We recommend that you look through these files when completing the procedures for changing host names and IP addresses for your SP system. The following files are available for PSSP systems:

/.rhosts	Contains host names used exclusively with **rcmd** services.
/.klogin	Contains host names used with authentication **rcmd** services.
/etc/hosts	Contains IP addresses and host names used with the SP system.
/etc/resolv.conf	Contains the IP address for Domain Name Service (DNS) (Optional).
/var/yp/*nis*	References the host name and IP address with Network Information Service (NIS).
/etc/krb.conf	Works with the host name for the authentication server.
/etc/krb.realms	Works with the host name of the SP nodes and authentication realm.
/etc/krb-srvtab	Provides the authentication service key using host name.
/etc/SDR_dest_info	Specifies the IP address of the control workstation and the SDR
/etc/ssp/cw_name	Specifies IP address of control workstation host name on SP nodes that work with Node installation and customization.

/etc/ssp/server_name
Specifies IP address and host name of the SP boot/install servers on SP nodes working with Node customization.

/etc/ssp/server_hostname
Specifies IP address and host name of the SP install servers on SP nodes working with Node installation.

/etc/ssp/reliable_hostname
Specifies IP address and host name of the SP node working with Node installation and customization.

/etc/ntp.conf Works with the IP address of the NTP server (Optional).

/etc/filesystems
Can contain the IP address or host name of NFS systems (mainly used on **/usr** client systems).

/tftpboot/*host***.config_info**
Contains the IP address and host name for each SP node. It is found on the control workstation and boot servers.

/tftpboot/*host***.intstall_info**
Contains the IP address and host name for each SP node. It is found on the control workstation and boot servers.

/tftpboot/*host***-new-srvtab**
Provides authentication service keys using host name. It is found on the control workstation and boot servers.

/etc/rc.net Contains the alias IP addresses used with system partitions

/etc/niminfo Works with the NIM configuration for NIM master information

/etc/sysctl.acl Uses hostname that works with Sysctl ACL support

/etc/logmgt.acl
Uses hostname that works with Error Log Mgt ACL support

/spdata/sys1/spmon/hmacls
Uses short hostname that works with hardmon authentication services

/etc/jmd_config.*SP_NAME*
Works with host names for Resource Management on the control workstation for all defined SP_NAME syspars.

/usr/lpp/csd/vsdfiles/VSD_ipaddr
Contains the SP node IBM Virtual Shared Disk adapter IP address

The following are additional files and directories for PSSP 2.2 systems:

/spdata/sys1/ha/cfg/em.*SP_NAME***cdb.***data*
Uses Syspar host name that works with configuration files for Event Management services.

/var/ha/run/*Availability Services*
Uses Syspar host name that contains the run files for the Availability Services, which include **haem** (Event Manager), **hats** (Topology Services) **hags** (Group Services-Ethernet), and **hagsglam** (Group Services- switch).

/var/ha/log/*Availability Services*
Uses Syspar host name that contains the log files for the Availability Services.

/var/adm/SPlogs/pman/*data*
Uses Syspar host name that contains the log files for the Problem Management subsystem.

/etc/services Specifies short host name based on *SP_NAME* partition that work with Availability Services port numbers

Selecting the Correct PSSP Procedure

You must understand that the procedure you select for changing the IP address and host name is based on the **PSSP release of the control workstation** for your SP system. The procedures described are based on PSSP levels for PSSP 1.2 (AIX 3.2.5), PSSP 2.1 (AIX 4.1), and PSSP 2.2 (AIX 4.1-4.2) systems. The PSSP 2.1 and PSSP 2.2 releases support system partitioning, where SP nodes can support PSSP 1.2, PSSP 2.1, and PSSP 2.2 code versions. There are special considerations relating to PSSP 1.2 level nodes when working with system partitions.

The PSSP 2.2 release allows new support of SP node coexistence, where SP nodes support either PSSP 1.2, and PSSP 2.2, or PSSP 2.1 and PSSP 2.2 code versions working in the same system partition. For more information on support for mixed levels of PSSP in a system partition, refer to *Implementing Coexistence with PSSP 2.2*, available from your IBM marketing representative or at **http://www.rs6000.ibm.com/tech**.

For PSSP 1.2 systems, see "Changing the IP Address and Host Names for PSSP 1.2 Systems."

For PSSP 2.1 systems, see "Changing the IP Address and Host Names for PSSP 2.1 Systems" on page 582.

For PSSP 2.2 systems, see "Changing the IP Address and Host Names for PSSP 2.2 Systems" on page 592.

Changing the IP Address and Host Names for PSSP 1.2 Systems

Changing the IP Address and host names on your SP system will cause changes to your control workstation and SP nodes. You may want to create a **mksysb** backup for the control workstation and back up critical file systems on the control workstation and SP nodes.

The administrator must perform the following steps for the PSSP 1.2 control workstation and the PSSP 1.2 nodes. These steps should be performed by the admin (root) user from a control workstation session. The control workstation steps include changes to the network configuration, SDR, and system files to reflect the new IP addresses and host names. The administrator then sets the boot response to **customize**, and performs a network boot for the PSSP 1.2 nodes. After the PSSP 1.2 nodes complete the customization, you need to verify that the PSSP 1.2 nodes reflect the new IP address and host name changes. The SP system is not

active until all SP nodes are reconfigured with the proper IP address and host name changes.

Execution on the PSSP 1.2 Control Workstation

Steps 1 through 5 are required only for IP address and host name changes being made for the control workstation. If changes are only being made to the SP nodes, you may go to step 6

1. Using SMIT **chinet** or the **chdev** command, update any new IP address or host name changes required for the affected adapters on the control workstation.
2. This step is performed for host name changes for the control workstation. It is not required for changes made to IP addresses or with SP node updates.

   Issue **setup_authent** from "Initialize 9076 SP2 Authentication Services" in *IBM 9076 Scalable POWERparallel Systems: SP2 Installation Guide* to create authentication services for the new host names being used. Manually check that the authentication files **/etc/krb.conf**, **/etc/krb.realms**, and **/etc/krb-srvtab** reference the new host name.
3. Manually issue **SDRChangeAttrValues** on the control workstation for SDR updates.

   To change the **hostname** attribute using the new host name for **SP_ports**, enter:

   SDRChangeAttrValues SP_ports hostname=*new CWS hostname*

   To change the **MACN** attribute using the new host name for **Frame**, enter:

   SDRChangeAttrValues Frame MACN=*new CWS hostname*

   To change the **control_workstation** attribute using the short host name for **SP**, enter:

   SDRChangeAttrValues SP control_workstation=*new CWS short hostname*
4. Manually update the **/etc/SDR_dest_info** and **/etc/hmacls** files to specify the new IP address and host name used with the SDR and **hardmon** services.
5. You may want to **reboot** the control workstation now. This establishes a clean system that reflects the IP address and host name changes. As an alternative, you can stop (kill) the **sdrd**, **hr_sdr**, **hb**, and **hardmon** daemon processes, and restart them on the control workstation.

   Usage Note

   The remaining steps are required for IP address and host name changes for both the control workstation and the SP nodes. Some of the steps are optional based on the changes being implemented.
6. This step includes the necessary updates required in the SDR objects for the SP nodes. See *IBM 9076 Scalable POWERparallel Systems: SP2 Installation Guide* for helpful descriptions. You can use the SP commands or the SMIT-based Configuration Management Interface (CMI).

   a. Using CMI or the **spsitenv** command, specify any host name changes that can be referenced for NTP, printing, and user management. See Step 12 "Enter Site Environment Information" in *IBM 9076 Scalable POWERparallel Systems: SP2 Installation Guide* (Optional).

b. Using CMI or the **spethernt** command, specify the new Ethernet IP address or host name change required for the SP nodes. See Step 14 "Enter Required Node Information" in *IBM 9076 Scalable POWERparallel Systems: SP2 Installation Guide*.

c. Using CMI or the **spadaptrs** command, reset **css0** and additional adapters that need to reference the new Ethernet IP address and host name. See Step 18 "Configure Additional Adapters for Nodes" in *IBM 9076 Scalable POWERparallel Systems: SP2 Installation Guide*.

d. Using CMI or the **sphostnam** command, reset the initial host name to be used for your SP system. See Step 19 "Configure Initial Hostnames for Nodes" in *IBM 9076 Scalable POWERparallel Systems: SP2 Installation Guide* (Optional).

e. Using CMI or the **spbootins** command, reset the boot/install and /**usr** server to reference the new Ethernet IP address and host name. See Step 21 "Set Up Nodes to be Installed" and Step 22 "Set Up Nodes to be /**usr** Clients" in *IBM 9076 Scalable POWERparallel Systems: SP2 Installation Guide*. Using **spbootins**, enter:

spbootins -u *usr_serv_id* -n *boot_server* -l *node_list*

Using CMI or the **spbootins** command, set the bootp response attribute to **customize** for the SP nodes. Using **spbootins**, enter:

spbootins -r customize -l *node_list*

You can issue the **splstdata -b** command to verify the correct boot information for the SP nodes.

You can manually verify that the following files were successfully updated after performing the previous CMI activities:

/etc/ntp.conf
/tftpboot/*host***.config_info**
/tftpboot/*host***.install_info**
/tftpboot/*host***-new-srvtab**

7. Update all other files on the control workstation that reflect the IP address or host name changes being made.

   a. Update any files that are involved with host name resolution. The files are **/etc/hosts**, **/etc/resolv.conf** (DNS), and **/etc/yp/*** (NIS).

   b. Update the **.rhosts** files that exist for PSSP 1.2 systems.

   c. Update the **/etc/filesystems** and **/etc/jmd_config** files for control workstation file systems and the Resource Manager configuration.

   d. You should also check to make sure your **/tftpboot/firstboot.cust** file is updated to reflect the IP address or host name changes. Instead of hardcoding host names, you may want to reference the $SERVER and $CWS variables.

8. Some files cannot be updated on the SP nodes during customization. You may want to **rcp** these files from the control workstation to the SP nodes by including them in the **/tftpboot/firstboot.cust** file. These include the **/etc/resolv.conf**, **/.rhosts**, and other SP customer-owned files.

   You can now perform a **NETWORK BOOT** on each of the SP nodes by issuing the **nodecond** command or by using the System Monitor GUI. You should follow the same installation sequence by first *customizing* the boot server

nodes. When the boot server nodes have completed successfully, you can then customize the remaining SP nodes. Since the SP nodes are in **customize** mode, most configuration files are updated to reflect the updated IP address and host name changes. You should now validate the PSSP 1.2 nodes. Refer to "Execution on PSSP 1.2 Nodes."

Execution on PSSP 1.2 Nodes

The following steps are performed on each PSSP 1.2 node to validate the IP address and host name updates for the control workstation and SP nodes. Perform the following verification checks on each of the SP nodes:

1. After the network boot is complete and the SP nodes are initialized, verify that they are using the correct IP address and host name.

   Updated Ethernet and additional adapters are using the correct address for each SP node

   Updated default routes are correct

   Host name resolution files were updated correctly (/**etc**/**hosts**, **dns**)

   The /**etc**/**SDR_dest_info** file points to the control workstation and the SDR

   The acct_master host name is updated for the NFS export list for the /**var**/**adm**/**acct** directory (SP Accounting)

   The /**etc**/**ssp**/**cw_name**, /**etc**/**ssp**/**server_hostname**, and /**etc**/**ssp**/**server_name** files point to the correct IP address and host names

   The /**etc**/**krb.conf**, /**etc**/**krb.realms**, and /**etc**/**krb-srvtab** files have the correct host names defined.

2. You may need to modify the following files on the SP nodes to reflect the updated IP addresses and host name definitions:

   /etc/filesystems
   /etc/ntp.conf
   /etc/resolv.conf
   /.rhosts
   /.klogin

   If any of the files are not correct, make the proper updates for the correct IP address and host name manually.

3. You may have to reset the routing tables after PSSP 1.2 node customization. You should verify that the current route tables are correct by issuing **netstat -nr**. If they are not correct, you can *flush* the route table and remove the route stanzas from the ODM. You can do this using the SMIT TCPIP panels or by issuing the **route -n -f** and **odmdelete** commands.

   Make sure that you re-create the proper default route tables using SMIT TCPIP or the **mktcpip** command.

4. If the node positions were changed, make sure that the correct node number is listed for the PSSP 1.2 boot server node. If the node number in the SDR is not listed as a boot server node, the **setup_server** command will remove required boot server files from the /**usr**/**sys**/**inst.images**/**ssp** directory.

   Make sure that the node number on the PSSP 1.2 boot server node matches the boot server node number listed in the SDR. To do this, on the PSSP 1.2 node, enter:

odmget -q attribute=node_number CuAt

Compare the value to the node number listed in the SDR using the **splstdata -b** command.

If the node numbers are different, you will need to update the SDR or the ODM to reflect the proper node number. You may need to use the **odme** editor or work with the **odmdelete** and **odmadd** commands.

5. We recommend issuing the **savebase** command to save your current ODM configuration to the boot device.

   If any manual updates were required to reflect the new IP address and host name changes, you should reboot the SP node. When the SP nodes are initialized, the SP system should now be using the new IP address and host names.

Changing the IP Address and Host Names for PSSP 2.1 Systems

Changing the IP address and host names on your SP system will cause changes to your control workstation and SP nodes. You may want to create a **mksysb** backup for the control workstation and backup critical file systems on the control workstation and SP nodes.

Changing a host name or IP address for a PSSP 2.1 node or control workstation requires the administrator to perform the following activities. There are network and SP configuration updates required on both the control workstation and SP nodes. We recommend that you document and understand all the required IP address and host name changes prior to performing this activity. It is also necessary to understand which SP nodes will be the boot server nodes in your SP configuration.

You will need to update the **en0** network interface and update system files on each SP node that have IP address or host name changes. You will need to know the new selected IP addresses and host names being used by the SP nodes, control workstation, and gateway servers. This is required so that the SP node can properly communicate with the control workstation during reboot activity.

You must work on the control workstation and make the necessary changes to the SDR. This includes updates to reset PSSP daemons and system partitions (syspar) files. You need to update files on the control workstation to reflect the new host names and IP addresses. During this activity, the SP system should not be available to users. The SP system is not ready until the SP nodes reference the new IP address and host name changes.

You must set the SDR boot response to **customize** and then execute a reboot for the SP nodes. After the SP nodes complete the reboot and are customized, you should verify that the files on the SP nodes reflect the new IP address and host name changes.

The installation support in AIX 4.1 for NIM requires the admin user to replace the NIM master objects. It may be necessary to update the NIM network and machine resources to match the new IP address or host name. The NIM updates are needed for the control workstation and boot server nodes.

If you use system partitioning on your SP system, you must rerun the system partitioning steps on your SP system. This allows the necessary Syspar config files

to reflect any changes. If you support a PSSP 1.2 (AIX 3.2.5) system partition, you must manually change the SP-based files on the PSSP 1.2 boot server node.

If your SP configuration supports HACWS, you must reconfigure the backup control workstation after completing the IP address and host name changes on the control workstation and SP system. First, you need to remove the current HACWS configuration prior to executing the IP address and host name procedures. You should rerun the HACWS installation steps using the new IP address and host names for the updated HACWS configuration.

If your SP system supports IBM Virtual Shared Disk configurations, you must execute some IBM Virtual Shared Disk commands on the SP nodes. You may need to stop and unconfigure the IBM Virtual Shared Disk, make IP address and host name changes, and then configure and start the IBM Virtual Shared Disk on the SP nodes.

Updating the PSSP 2.1 Node Interfaces

You need to make the following updates on all PSSP 2.1 nodes prior to making updates to the control workstation.

Note to the SP Administrator

If your SP node is running at PSSP 1.2 which uses system partitioning, you need to manually update files on your PSSP 1.2 boot server node prior to completing the control workstation steps. Please refer to “Updating the PSSP 1.2 Boot Server Node on PSSP 2.x Systems” on page 590.

The following steps need to be verified on each PSSP 1.2 node. Some steps can be performed by implementing user shell scripts and by using the SP distributed shell (**dsh**) services from the control workstation to the target nodes.

1. Update the NIM files and configuration

   If available, update the **/etc/niminfo** file to provide the new host name and IP address changes for the NIM client, master, and file references. If the SP node is a NIM master or a boot server, you need to remove the NIM ODM objects and configuration files from the SP node by issuing the following:

   ```
   /usr/sbin/nim -o unconfig master
   /usr/sbin/installp -u bos.sysmgt.nim.master
   ```

2. Unconfigure IBM Virtual Shared Disks (Optional)

   You need to determine if there are any IBM Virtual Shared Disk configurations available on the SP node by issuing the **/usr/lpp/csd/bin/vsdatalst -n** command. If you receive an error message indicating “not found” or your SP node is not listed in the IBM Virtual Shared Disk configuration, you do not have to be concerned about the IBM Virtual Shared Disk. If your SP node is listed as having an IBM Virtual Shared Disk configuration, you will need to suspend, stop, and unconfigure the IBM Virtual Shared Disk for this SP node.

   ```
   /usr/lpp/csd/bin/suspendvsd -a
   /usr/lpp/csd/bin/stopvsd -a
   /usr/lpp/csd/bin/ucfgvsd -a
   ```

3. Update PSSP files for updates on each PSSP 2.1 node

You need to update the following files to reflect the new host name and IP address for the SP node and servers:

a. Update the **/etc/SDR_dest_info** file using the new control workstation and SDR IP addresses

b. Update the **/etc/ssp** "cw_name", "server_hostname", and "server_name" files to reference the new boot server host name and IP address.

c. Update the **/etc/ssp/reliable_hostname** file to reference the new SP node client host name and IP address.

d. Make sure that the SP_NAME environment variable is updated or set to blank (export SP_NAME=).

e. Issue **/usr/lpp/ssp/kerberos/bin/kdestroy** to remove any active Kerberos ticket-granting-tickets. You may need to update the **/etc/krb.conf** file if the authentication server is changed. We recommend renaming the **/etc/krb-srvtab** file on the SP node.

4. Updating the PSSP 2.1 node **en0** interface

   You now need to update the **en0** interface on each SP node to allow for proper communication paths between the SP node and the control workstation and you need to change the **en0** interface to the new IP address and host name. You must also supply the proper gateway and route information for the **en0** adapter. It is important that all new IP addresses and host names be resolvable on the SP node. You do not need to modify any other adapter interfaces on the SP node. To accomplish this task, use the **mktcpip** command. You can do this using SMIT **mktcpip** or by issuing the **mktcpip** command on each SP node.

   The following examples use the **mktcpip** command:

   mktcpip -h *new node hostname* -a *new node IP addr* -m *subnet mask*
   -i en -n *name server* -d domain -g *gateway IP addr* -t N/A

   The following example changes the host name of node k22n06 to a new IP address 129.40.88.70 and host name k88n06 in domain ppd.pok.ibm.com:

   "/usr/sbin/mktcpip -h'k88n 6.ppd.pok.ibm.com' -a'129.4 .88.7 '
   -m'255.255.255.192' -i'en ' -n'129.4 .7 .1' -d'ppd.pok.ibm.com'
   -g'129.4 .88.126' -t'N/A'"

   After issuing **mktcpip**, the node loses the current network connection to the control workstation and the other SP nodes. Any further updates to the SP node will need performed using the **tty0 console** using the **s1term** command.

You have now completed the required updates for the PSSP 2.1 node. You now need to make the appropriate changes at the control workstation. Do **not** reboot the PSSP 2.1 nodes until instructed to do so. For information on making the appropriate changes at the control workstation, refer to the appropriate section (for PSSP 2.1, refer to "Execution on the PSSP 2.1 Control Workstation" on page 585; for PSSP 2.2, refer to "Execution on the PSSP 2.2 Control Workstation" on page 595).

Execution on the PSSP 2.1 Control Workstation

The following tasks need to be performed by the admin (root) user on the control workstation when changing the Ethernet IP address or host name.

If you are currently in a system partition configuration, you can save the current SDR attributes by issuing **/usr/lpp/ssp/bin/SDRArchive**. You may want to backup the current **/spdata** file system and possibly create a **mksysb** before making any control workstation IP address and host name changes.

If you are currently in an HACWS configuration, you should properly disable the backup control workstation and unconfigure HACWS prior to the changing any IP address and host names for the control workstation. Refer to "Installing and Configuring the High Availability Control Workstation" in *RS/6000 Scalable POWERparallel Systems: Installation Guide*.

You need to reference the old and the new system partition names (SP_NAME) and IP address when executing the tasks in this section. It is strongly suggested to reset the SP system into one default system partition (Syspar) prior to changing the IP address or host names in the SDR. You will be asked to reenter the system partitioning steps to reconfigure back into multiple system partitions.

The following steps 1 through 12 are required steps when there are IP address and host name changes made to the control workstation. If there are only changes being made to the SP Nodes, you may go to step 13 on page 587 in this section.

1. Reconfigure the SP system into one default system partition. Perform the appropriate steps required in Chapter 17, "Managing System Partitions" on page 229 If you currently have an SDR archive of the default system partition, you can issue the **SDRRestore** command.
2. Stop all PSSP daemons (**sdr**, **hb**, **hr**, and **hardmon**) on the control workstation.

   /bin/stopsrc -g sdr
   /bin/stopsrc -g hb
   /bin/stopsrc -g hr
   /bin/stopsrc -s hardmon
3. Remove the current source master objects for **sdr**, **hb**, and **hr** for each system partition (SP_NAME) reference.

   /usr/lpp/ssp/bin/sdr -spname *SP_NAME* **rmsrc**
   /usr/lpp/ssp/bin/hb -spname *SP_NAME* **rmsrc**
   /usr/lpp/ssp/bin/hr -spname *SP_NAME* **rmsrc**
4. Using **mktcpip** or the **chdev** command, specify IP address or host name changes required for the control workstation. This includes changes being made for any affected adapter interfaces. It is also important to make appropriate updates to the route tables, netmasks, and gateway servers.
5. If multiple system partitions exist, update the **/etc/rc.net** file to reference the new alias address used for each system partition. Issue **/etc/rc.net** to configure the new alias IP addresses. You can issue the **netstat -ni** command to validate the new alias addresses. The old alias addresses used with system partitions will be removed during the next reboot of the control workstation.
6. Manually update the **/etc/SDR_dest_info** and **/spdata/sys1/spmon/hmacls** files to the new IP address and host name for the SDR and **hardmon** interfaces.

Manually update the SDR system partition map file **/spdata/sys1/sdr/system/classes/Syspar_map** to reflect the new IP address and the new default SP_NAME.

Move the system partition directories found at location **/spdata/sys1/sdr/partitions** from the old IP address to the new IP address for each system partition being modified.

/bin/mv *old_IP_addr new_IP_addr*

7. This step should only be performed for host name and domain changes for the control workstation. It is not required for changes made to control workstation IP addresses or to updates for only SP nodes.

   Issue the **setup_authent** script to create authentication services for the new host names being used. See the step "Initialize RS/6000 SP Authentication Services" in *IBM Parallel System Support Programs for AIX: Installation and Migration Guide*.

   Manually check that the authentication **/etc/krb.conf** and **/etc/krb.realms** files reference the proper host names and domain. You can issue **/usr/lpp/ssp/kerberos/bin/ksrvutil list** to make sure that the **rcmd** and **hardmon** services reference the new host names in the **/etc/krb-srvtab** file. You may also need to recreate Kerberos principals for any users that were previously defined in the Kerberos database.

8. Create the new source master resources for the PSSP 2.1 **sdr**, **hb**, and **hr** daemons. You can issue the **/usr/lpp/ssp/install/bin/post_process** script to re-create the new source objects, then start the daemons.

   This script only creates the PSSP 2.1 source master resources and daemons for the default system partition name (SP_NAME). If you are still have multiple system partitions, you will need to create additional source master objects for the **hr** and **sdr** daemons.

   hr -spname *SP_NAME* mksrc
   sdr -spname *SP_NAME* mksrc IP Address

9. After the SDR daemon is properly activated on the SP system, manually issue **SDRChangeAttrValues** for the control workstation.

   To change the **hostname** attribute using the new host name for **SP_ports**, enter:

   SDRChangeAttrValues SP_ports hostname=*new CWS hostname*

   To change the **MACN** attribute using the new host name for **Frame**, enter:

   SDRChangeAttrValues Frame MACN=*new CWS hostname*

   To change the **control_workstation** attribute using the short host name for **SP**, enter:

   SDRChangeAttrValues SP control_workstation=*new CWS short hostname*

10. Remove the current Network Installation Manager (NIM) ODM database and configuration files on the control workstation. You need to issue:

    /usr/sbin/nim -o unconfig master
    /usr/sbin/installp -u bos.sysmgt.nim.master

    It may be necessary to update the **/etc/niminfo** file to reference any changes to the new IP address and host names.

You need to remove entries from the NFS export list that reference the old IP address and host names. You can do this by updating the **/etc/exports** file, and then stopping and starting the NFS daemons.

11. Using CMI or the **spsitenv** command, specify any host name changes that may be referenced for NTP, printing, and user management. See the step "Enter Site Environment Information" in *RS/6000 Scalable POWERparallel Systems: Installation Guide* (Optional).

12. Reboot the control workstation now. This establishes a clean system to reflect the IP address and host name changes. You should verify that all PSSP daemons are getting activated from **/etc/inittab** and the source master.

    Usage Note

    The remaining steps for the control workstation involve updating the SDR and SP system files for the SP node objects. Based on your SP configuration changes, some of the steps may be optional.

13. Update the SDR node objects by using commands or the SMIT-based SP Configuration Management Interface (CMI). You can perform the following tasks for each system partition by exporting the SP_NAME variable (for example, export SP_NAME=*syspar name*).

    a. Using CMI or the **spethernt** command, specify the new Ethernet IP address or host name changes required for the SP nodes. See the step "Enter Required Node Information" in *RS/6000 Scalable POWERparallel Systems: Installation Guide*.

       Based on your selection of long or short host names, you may need to apply the PTF fix for APAR IX56406.

    b. Using CMI or the **spadaptrs** command, reset the **css0** and other adapters that need to reference the new IP address or host names being changed. See the step "Configure Additional Adapters for Nodes" in *RS/6000 Scalable POWERparallel Systems: Installation Guide*.

    c. Using CMI or the **sphostnam** command, reset the initial host name that you want to use in your system. See the step "Configure Initial Host Names for Nodes" in *RS/6000 Scalable POWERparallel Systems: Installation Guide* (Optional).

    d. Using CMI or the **spbootins** command, reset all boot/install servers to reference the new Ethernet IP address and host names. It is important that you set your SP node boot/install servers into the proper configuration for your SP system. If you have a PSSP 1.2 system partition, you should have already designated the PSSP 1.2 boot server nodes.

       spbootins -s no -n *boot_node* -l *node_list*

       You need to set the bootp response to **customize** for all the SP nodes.

       spbootins -r **customize** -l *node_list*

       You can issue the **splstdata -b** command to verify the correct boot information for the SP nodes.

       The **spbootins** without the **-s no** flag runs the **setup_server** command which creates all the NIM-based files and resources required for installation. It also creates the authentication **rcmd** principals for the new SP node host names. You may see a NIM error message *ssp.jm*

2.1.0.0.FAILED during the NIM SPOT creation. Please disregard this error message.

You can manually verify that the following files were successfully updated after running the previous CMI steps:

/etc/ntp.conf
/tftpboot/*host***.config_info**
/tftpboot/*host***.install_info**
/tftpboot/*host***-new-srvtab**

Note to the SP Administrator

At this time, you should manually update the designated PSSP 1.2 boot server node. Refer to the "Updating the PSSP 1.2 Boot Server Node on PSSP 2.x Systems" on page 590 section.

14. If your SP system supports system partitions, you must reissue the system partitioning steps found Chapter 17, "Managing System Partitions" on page 229. The system partitioning **spapply_config** command re-creates the proper PSSP daemons and resynchronizes the SDR objects to reflect any IP address and host name changes.
15. Update any other files on the control workstation that reflect IP address or host name changes.

    You should make sure the new SP_NAME references have been updated in the SDR properly. Execute **splstdata -p** command to validate.

    a. Update any files that are involved with host name resolution. The files are **/etc/hosts**, **/etc/resolv.conf** (DNS) and **/var/yp/*** (NIS).
    b. Update the **.rhosts** or **.klogin** files.
    c. Update the **/etc/filesystems** and **/etc/jmd_config.***SP_NAME* files for your SP configuration.
    d. Make sure your **/tftpboot/script.cust** file is updated to reflect IP address and host name changes. Instead of hardcoding host names, you can reference the $SERVER and $CWS variables.
16. For each PSSP 2.1 boot server node, you should have already unconfigured NIM during the "Execution on PSSP 2.1 Nodes" on page 589 section. During the node customization, **setup_server** creates the proper install files for the nodes they are to customize. Since the SP nodes are in **customize** mode, most configuration files are updated to reflect the IP address and host name changes.

    Some files will not be updated on the nodes during the customization. You can **rcp** these files from the control workstation to the SP nodes by including them in the **/tftpboot/script.cust** file. These can include the **/etc/resolv.conf**, **/.rhosts**, and other SP customer-owned files.

    Perform a **REBOOT** for each PSSP 2.1 nodes. You can do this by using the System Monitor GUI or by issuing the **hmcmds** command. You need to follow the same install sequence by customizing the boot server nodes first, and then customizing the remaining SP nodes.

Note to the SP Administrator

For PSSP 1.2 system partitions, you must first update all required files on the PSSP 1.2 boot server node. You must then perform a network boot on each PSSP 1.2 node, except for the current PSSP 1.2 boot server node. When the PSSP 1.2 nodes are properly updated, you can then perform a network boot on the primary PSSP 1.2 boot server node.

17. Consider not performing the HACWS reconfiguration until your SP system has been properly updated and is stable.

Execution on PSSP 2.1 Nodes

The following verification steps need to be performed on each PSSP 2.1 nodes for changes made for the IP address and host name:

1. After reboot is complete and the SP nodes are customized, verify that the following have the correct IP address and host name specified:

   The updated SP Ethernet address specified for the SP node

   The updated default route and gateway are correct

   The additional adapters are correct (**en1**, **tr0**, **fi0**, **css0**)

   The host name resolution files were updated correctly (/**etc**/**hosts**)

   The /**etc**/**SDR_dest_info** file points to the control workstation and SDR Syspar IP address.

   The acct_master host name is updated for the NFS export list directory /**var**/**adm**/**acct** (SP Accounting)

   The files in the /**etc**/**ssp** directory *cw_name*, *server_hostname*, *server_name*, and *reliable_hostname* reference the correct IP address and host names.

   The /**etc**/**krb.conf**, /**etc**/**krb.realms**, and /**etc**/**krb-srvtab** files have the correct host names defined

2. You may need to modify the following files on the SP nodes to reflect the updated IP addresses and host name definitions:

   /etc/filesystems
   /etc/ntp.conf
   /etc/resolv.conf
   /.rhosts
   /.klogin

   If any of the files are incorrect, make the proper updates for the correct IP address or host name.

3. Verify that the NIM resources were built on the PSSP 2.1 boot server nodes by executing the **lsnim** command. If there were NIM problems, remove the current NIM database and configuration files.

   ```
   /usr/sbin/nim -o unconfig master
   /usr/sbin/installp -u bos.sysmgt.nim.master
   ```

   Then issue **setup_server** on the PSSP 2.1 boot server node.

4. Reconfigure and start the IBM Virtual Shared Disk for the SP nodes that support the IBM Virtual Shared Disk configuration. You can do this by issuing the following IBM Virtual Shared Disk commands:

   ```
   /usr/lpp/csd/bin/cfgvsd -a
   /usr/lpp/csd/bin/startvsd -a
   ```

   You can validate that the IP address and host names are correctly specified by issuing the following command:

   ```
   /usr/lpp/csd/bin/vsdatalst -n
   ```

5. If there were modifications made to any SP node system files, it would be best to reboot the PSSP 2.1 nodes to reflect the IP address and host name changes. When the SP nodes are initialized, your SP system should be activated using the new IP addresses and host names.

Verifying PSSP 1.2 Nodes on PSSP 2.x Systems

Since the SP node files and setup are different between PSSP 2.x and PSSP 1.2 nodes, you should follow the documentation and recommendations in the PSSP 1.2 node sections. The verification of PSSP 1.2 nodes from a PSSP 1.2 system partition should reference “Execution on PSSP 1.2 Nodes” on page 581.

Updating the PSSP 1.2 Boot Server Node on PSSP 2.x Systems

Updating of the PSSP 1.2 boot server node for a PSSP 1.2 (AIX 3.2.5) syspar can be tedious. All PSSP 1.2 system files that contain IP address and host names will need to be manually updated. Refer to “SP System Files with IP Address and Host Names” on page 576 to review the possible PSSP 1.2 files that need to be updated.

1. Update the files for the SDR, control workstation, and Kerberos servers.
   a. Update the **/etc/SDR_dest_info** file to point to the control workstation and the PSSP 1.2 (SP_NAME) SDR IP address. Set the SP_NAME variable to the correct system partition name or set it to blank.
   b. Update the **/etc/ssp** files **cw_name**, **server_hostname**, and **server_name** to reference the new control workstation and boot/install server host name and IP address.
   c. Issue **/usr/lpp/ssp/kerberos/bin/kdestroy** to remove any active Kerberos ticket-granting-tickets. You can update the **/etc/krb.conf** file to point to the correct Kerberos server. You may have to update the **/etc/krb.realms** file to match the **/etc/krb.realms** file found on the control workstation.

      You may need to extract a new **/etc/krb-srvtab** from the Kerberos database if changes are made to the Kerberos server. Issue the following command:

      ```
      /usr/lpp/ssp/kerberos/etc/ext_srvtab node
      ```

      on the control workstation. You can **ftp** the node's **srvtab** file *node***-new-srvtab** from the control workstation, and replace the **/etc/krb-srvtab** file on the boot server node.

      You need to update the **/.rhosts** and **/.klogin** files to have the proper **rcmd** support. You can reference the files from the control workstation.
2. Update the files to resolve IP address and host names.

Based on your network configuration, you need to update the following files for all resolvable host names. You can **ftp** or **rcp** the files from the control workstation to the boot server node. The following are the network-related files:

/etc/hosts Contains IP addresses and host names used with SP system

/etc/resolv.conf
Contains IP address for domain name service (Optional)

/var/yp/*nis* Contains NIS host name and IP address

Update files on the boot server to reflect its new host name and IP address. The following are some of the files:

/etc/ntp.conf Works with the IP address of the NTP server (Optional)

/etc/filesystems
Can contain the IP address or host name of NFS systems

3. Updating the PSSP 1.2 node adapters

   It may be necessary to make the appropriate updates in regard to to the proper route tables and gateway servers for the network changes. You should verify that the current route tables are correct by issuing **netstat -nr**. If they are not correct, you can *flush* the route table and remove the route stanzas from the ODM. You can do this using the SMIT TCPIP panels or by issuing the **route -n -f** and **odmdelete** commands.

   Make sure that you re-create the proper default route tables using the SMIT TCPIP or the **mktcpip** command.

   Update the **en0** interface for the boot server node by issuing SMIT **mktcpip** or the **mktcpip** command. If you have additional adapters (**en1**, **tr0**, **fi0**) in the boot server node, issue the SMIT **mktcpip** or the **mktcpip** command for each adapter.

   mktcpip -h *new node hostname* -a *new node IP addr* -m *subnet mask*
   -i *en* -n *name server* -d *domain* -g *gateway IP addr*

4. If the switch IP address was updated, you will need to update the **css0** adapter data. You need to update the ODM files to include new **css** IP address and netmask. You can do this by issuing the **odmdelete** and **odmadd** commands. A suggested path will be to customize this boot server node from another PSSP 1.2 node when the node has been properly updated.

   Make sure that the node number on this PSSP 1.2 boot server node matches the boot server node number listed in the SDR. If the node number in the SDR is not listed as a boot server node, issuing the **setup_server** command will remove required boot server files from **/usr/sys/inst.images/ssp** directory. You can validate this by issuing the following command on the PSSP 1.2 node:

   odmget -q "attribute=node_number" CuAt

   Compare the value to the node number listed in the SDR using the **splstdata -b** command.

   If the node numbers are different, you must update the SDR or the ODM to reflect the correct node number. You may need to use the **odme** editor or work with the **odmdelete** and **odmadd** commands.

5. Update the files on the **/tftpboot** directory.

You must make sure that the files created on the **/tftpboot** directory are updated to reflect the new IP address and host names for each PSSP 1.2 node being installed by the server node. You will also need to update the **/tftpboot/firstboot.cust** file to make any IP address or host name changes.

You should issue the **savebase** command to save the appropriate updates made to the ODM. We suggest issuing the **shutdown -Fr** command to reinitialize the PSSP 1.2 boot server node with the IP address and host name. When the node reboots, it executes the **setup_server** to create new PSSP 1.2 node **install_info**, **config_info**, and **srvtab** files in the **/tftpboot** directory. The following are some file examples:

*node hostname***.install_info**
*node hostname***.config_info**
*node short hostname***-new-srvtab**

6. Review all files for updates

   At this time, we suggest reviewing all the files that were listed in the "SP System Files with IP Address and Host Names" on page 576 section. After you have verified the files on the PSSP 1.2 boot server node, you should be ready to perform a network boot for the remaining PSSP 1.2 nodes in that system partition.

   You may need to open the **tty0** console using **s1term** for the PSSP 1.2 node during the node customization if the LED indicates a hang state of 231.

7. Customize the main PSSP 2.1 server

   After the PSSP 1.2 nodes have been customized successfully, you should now set up the main PSSP 1.2 boot server node to be customized by the backup PSSP 1.2 boot server. You should reference the **install**, **config**, and **srvtab** files in the **/tftpboot** directory, then perform a network boot on the main boot server node as was done in the previous step.

Changing the IP Address and Host Names for PSSP 2.2 Systems

Changing the IP address and host names on your SP system will cause changes to your control workstation and SP nodes. You may want to create a **mksysb** backup for the control workstation and backup critical file systems on the control workstation and SP nodes.

Changing a host name or IP address for PSSP nodes or control workstation requires the administrator to perform the following activities. There are network and SP configuration updates required on both the control workstation and SP nodes. We recommend that you document and understand all the required IP address and host name changes prior to performing this activity. It is also necessary to understand which SP nodes will be the boot server nodes in your SP configuration.

You will need to update the **en0** network interface and update system files on each SP node that have IP address or host name changes. You will need to know the new selected IP addresses and host names being used by the SP nodes, control workstation, and gateway servers. This is required so that the SP node can properly communicate with the control workstation during reboot activity.

You must work on the control workstation and make the necessary changes to the SDR. This includes updates to reset PSSP daemons, PSSP resource objects, and system partitions (syspar) files. You need to update files on the control workstation

to reflect the new host names and IP addresses. During this activity, the SP system should not be available to users. The SP system is not ready until the SP nodes reference the new IP address and host name changes.

You must set the SDR boot response to **customize** and then execute a reboot for the SP nodes. After the SP nodes complete the reboot and are customized, you should verify that the files on the SP nodes reflect the new IP address and host name changes.

The PSSP 2.2 installation support still requires the admin user to replace the NIM master objects. It may be necessary to update the NIM network and machine resources to match the new IP address or host name. The NIM updates are needed for the control workstation and boot/install server nodes. There are new installation commands in PSSP 2.2 that support the implemetation of NIM interfaces.

The PSSP 2.2 system now provides Availability Services which are new subsystems and interfaces that are installed on the control workstation and SP nodes. These subsystems (**hags, hats, haem, pman, sp_configd, hr, and hb**) have replaced the host responds (**hr**) and heartbeat (**hb**) interfaces from PSSP 2.1.

If you use system partitioning on your SP system, you may want to rerun the system partitioning steps on your SP system. This allows the necessary Syspar config files to reflect all updates. If you support a PSSP 1.2 (AIX 3.2.5) system partition, you must manually change the SP-based files on the PSSP 1.2 boot server node.

If your SP configuration supports HACWS, you must reconfigure the current HACWS configuration with the new IP address and host name changes. You will need to execute specific HACWS procedures to re-create the HACWS configuration. Refer “Working with HACWS Configurations” on page 600 for more information.

If your SP system supports IBM Virtual Shared Disk configurations, you must execute some IBM Virtual Shared Disk commands on the SP nodes. You may need to stop and unconfigure the IBM Virtual Shared Disk, make IP address and host name changes, and then configure and start the IBM Virtual Shared Disk on the SP nodes.

Updating the PSSP 2.2 Node Interfaces

You need to make the following updates on all PSSP 2.2 nodes prior to making updates to the control workstation.

Note to the SP Administrator

If your SP node is running at PSSP 1.2 and involves system partitioning, you can refer to “Updating the PSSP 1.2 Boot Server Node on PSSP 2.x Systems” on page 590. You will need to manually update files on your PSSP 1.2 boot server node prior to completing the control workstation steps.

If your SP node is running at PSSP 2.1 which involves system partitioning, you can refer to “Updating the PSSP 2.1 Node Interfaces” on page 583.

The following steps need to be verified on each SP node. Some of the steps can be performed by implementing user shell scripts and by using the SP distributed shell (**dsh**) services from the control workstation to the target nodes.

1. Stop/Remove the Availability Services

   You need to shutdown the daemons and source master objects that work with PSSP Availability Services. The Availability services include Topology Services, Group Services, Event Manager, Problem Management, and Performance Monitor. Execute the **syspar_ctrl** command to stop and remove these resources.

   /usr/lpp/ssp/bin/syspar_ctrl -c

2. Update the NIM files and configuration

   If available, update the **/etc/niminfo** file to provide the new host name and IP address changes for the NIM client, NIM master, and NIM file references. If the SP node is currently a NIM master (boot/install server), you need to remove the NIM ODM objects and NIM master configuration files from the SP node by issuing the **delnimmast** command:

   /usr/lpp/ssp/bin/delnimmast -l *node_number*

3. Unconfigure IBM Virtual Shared Disks (Optional)

   You need to determine if there are any IBM Virtual Shared Disk configurations available on the SP node by issuing the **/usr/lpp/csd/bin/vsdatalst -n** command. If you receive an error message indicating “not found” or your SP node is not listed in the IBM Virtual Shared Disk configuration, you do not have to be concerned about the IBM Virtual Shared Disk. If your SP node is listed as having an IBM Virtual Shared Disk configuration, you will need to suspend, stop, and unconfigure the IBM Virtual Shared Disk for this SP node.

   /usr/lpp/csd/bin/suspendvsd -a
   /usr/lpp/csd/bin/stopvsd -a
   /usr/lpp/csd/bin/ucfgvsd -a

4. Update PSSP files for updates on each PSSP 2.2 node

   You need to update the following files to reflect the new host name and IP address for the SP node and servers:

   a. Update the **/etc/SDR_dest_info** file using the new control workstation and SDR IP addresses

   b. Update the **/etc/ssp cw_name**, **server_hostname** and **server_name** files to reference the new boot server host name and IP address.

   c. Update the **/etc/ssp/reliable_hostname** file to reference the new SP node client host name and IP address.

   d. Make sure that the SP_NAME environment variable is updated or set to blank (export SP_NAME=).

   e. Issue **/usr/lpp/ssp/kerberos/bin/kdestroy** to remove any active Kerberos ticket-granting-tickets. You may need to update the **/etc/krb.conf** file if the authentication server is changed. We recommend renaming the **/etc/krb-srvtab** file on the SP node.

5. Updating the PSSP 2.2 node **en0** interface

   You now need to update the **en0** interface on each SP node to allow for proper communication paths between the SP node and the control workstation and

you need to change the **en0** interface to the new IP address and host name. You must also supply the proper gateway and route information for the **en0** adapter. Make sure that all new IP addresses and host names are resolvable on the SP node. You do not need to modify any other adapter interfaces on the SP node. To accomplish this task, use the **mktcpip** command. You can do this using SMIT **mktcpip** or by issuing the **mktcpip** command on each SP node.

The following examples use the **mktcpip** command:

mktcpip -h *new node hostname* -a *new node IP addr* -m *subnet mask*
-i en -n *name server* -d domain -g *gateway IP addr* -t bnc

The following example changes the host name of node k22n06 to a new IP address 129.40.88.70 and host name k88n06 in domain ppd.pok.ibm.com:

"/usr/sbin/mktcpip -h'k88n 6.ppd.pok.ibm.com' -a'129.4 .88.7 ' /
-m'255.255.255.192' -i'en ' -n'129.4 .7 .1' -d'ppd.pok.ibm.com' /
-g'129.4 .88.126' -t'bnc'"

After issuing **mktcpip**, the node loses the current network connection to the control workstation and the other SP nodes. Any further updates to the SP node need to be performed using the **tty0 console** using the **s1term** command. You may be able to kill the "telnet" or "rlogin" process from the control workstation or the home machine to regain the window.

You have now completed the required updates for the PSSP 2.2 node. You now need to make the appropriate changes at the control workstation. Do *not* reboot the PSSP 2.2 nodes until instructed to do so. For information on making the appropriate changes at the control workstation, refer to "Execution on the PSSP 2.2 Control Workstation."

Execution on the PSSP 2.2 Control Workstation

The following tasks need to be performed by the admin (root) user on the control workstation when changing the Ethernet IP address or host name.

You may want to backup the current **/spdata** file system, and possibly create a **mksysb** before making any IP address and host name changes to the control workstation.

If you are in a system partition configuration you may save the current SDR attributes by issuing **/usr/lpp/ssp/bin/SDRArchive**. The support in PSSP 2.2 allows the admin user to work with system partition configurations. You no longer need to default to one system partition.

If you are currently in an HACWS configuration, you should properly disable the backup control workstation and unconfigure HACWS prior to the changing any IP address and host names for the control workstation. Refer to "Working with HACWS Configurations" on page 600 for more information.

The following steps 1 through 10 are required steps when there are IP address and host name changes made to the control workstation. If there are only changes being made to the SP Nodes, you may go to step 11 on page 597 in this section.

1. Stop PSSP daemons and remove current source master objects (**sdrd**, **hardmon** and availability services) on the control workstation. The **syspar_ctrl -c** command will stop and remove all active resources that work with the PSSP

Availability Services. The **stopsrc** command will stop the other PSSP resources. Remove the SDR source master object for each system partition (SP_NAME).

/usr/lpp/ssp/bin syspar_ctrl -c
/bin/stopsrc -g sdr
/bin/stopsrc -s hardmon
/bin/stopsrc -s supfilesrv
/usr/lpp/ssp/bin/sdr -spname *SP_NAME* rmsrc

2. Using **mktcpip** or the **chdev** command, specify IP address or host name changes required for the control workstation. This includes changes being made for any affected adapter interfaces. It is also important to make appropriate updates to the route tables, netmasks, and gateway servers.
3. If multiple system partitions exist, update the **/etc/rc.net** file to reference the new alias address used for each system partition. Issue **/etc/rc.net** or execute **ifconfig** command to configure the new alias IP addresses. You can issue the **netstat -ni** command to validate the new alias addresses. The old alias addresses used with system partitions will be removed during the next reboot of the control workstation.
4. Manually update the **/etc/SDR_dest_info** and **/spdata/sys1/spmon/hmacls** files to the new IP address and host name for the SDR and hardmon interfaces.

   Manually update the SDR system partition map file **/spdata/sys1/sdr/system/classes/Syspar_map** to reflect the new IP address and their SP_NAME values with each system partition.

   Move the system partition directories found at location **/spdata/sys1/sdr/partitions** from the old IP address to the new IP address for each system partition being modified.

   /bin/mv *old_IP_addr new_IP_addr*

   Manually update the SDR system partition file **/spdata/sys1/sdr/partitions/***new_IP_addr***/classes/Syspar** to reflect the new IP address and their SP_NAME host name values with each system partition defined.
5. This step should only be performed for host name and domain changes for the control workstation. It is not required for changes made to control workstation IP addresses or to updates for SP nodes.

   Issue the **setup_authent** script to create authentication services for the new host names being used. See the step “Initialize RS/6000 SP Authentication Services” in *IBM Parallel System Support Programs for AIX: Installation and Migration Guide*.

   Manually check that the authentication **/etc/krb.conf** and **/etc/krb.realms** files reference the proper host names and domain. You can issue **/usr/lpp/ssp/kerberos/bin/ksrvutil list** to make sure that the **rcmd** and **hardmon** services reference the new host names in the **/etc/krb-srvtab** file. You may also need to recreate Kerberos principals for any users that were previously defined in the Kerberos database.
6. Create the new source master resources for each SDR object, and then start the SDR and hardmon daemons.

sdr -spname *SP_NAME* mksrc *new_IP_addr* (each Syspar)
startsrc -g sdr
startsrc -s hardmon

7. After the SDR daemon is properly activated on the SP system, manually issue **SDRChangeAttrValues** for the control workstation.

   To change the **hostname** attribute using the new host name for **SP_ports**, enter:

   SDRChangeAttrValues SP_ports hostname=*new_CWS_hostname*

   To change the **MACN** attribute using the new host name for **Frame**, enter:

   SDRChangeAttrValues Frame MACN=*new_CWS_hostname*

   To change the **control_workstation** attribute using the short host name for **SP**, enter:

   SDRChangeAttrValues SP control_workstation=*new_CWS_short_hostname*

8. Remove the current Network Installation Manager (NIM) ODM database and configuration files on the control workstation. You need to issue:

   /usr/lpp/ssp/bin/delnimmast -l 0

   It may be necessary to update the /**etc**/**niminfo** file to reference any changes to the new IP address and host names.

   You need to remove entries from the NFS export list that reference the old IP address and host names. You can do this by updating the /**etc**/**exports** file, and then stopping and starting

9. Using CMI or the **spsitenv** command, specify any host name changes that may be referenced for NTP, printing, and user management. See the step "Enter Site Environment Information" in *IBM Parallel System Support Programs for AIX: Installation and Migration Guide* (Optional).

10. Reboot the control workstation now. This establishes a clean system to reflect the IP address and host name changes. You should verify that all PSSP daemons are getting activated from /**etc**/**inittab** and the source master.

    Usage Note

    The remaining steps for the control workstation involve updating the SDR and SP system files for the SP node objects. Based on your SP configuration changes, some of the steps may be optional.

11. Update the SDR node objects by using commands or the SMIT-based SP Configuration Management Interface (CMI). You can perform the following tasks for each system partition by exporting the SP_NAME variable (for example, export SP_NAME=*syspar name*).

    a. Using CMI or the **spethernt** command, specify the new Ethernet IP address or host name changes required for the SP nodes. See the step "Enter Required Node Information" in *IBM Parallel System Support Programs for AIX: Installation and Migration Guide*.

    b. Using CMI or the **spadaptrs** command, reset the **css0** and other additional adapters that need to reference the new IP address or host names being changed. See the step "Configure Additional Adapters for Nodes" in *IBM Parallel System Support Programs for AIX: Installation and Migration Guide*.

c. Using CMI or the **sphostnam** command, reset the initial host name that you want to use in your system. See the step "Configure Initial Host Names for Nodes" in *IBM Parallel System Support Programs for AIX: Installation and Migration Guide* (Optional).

d. Using CMI or the **spbootins** command, reset all boot/install servers to reference the new Ethernet IP address and host names. It is important that you set your SP node boot/install servers into the proper configuration for your SP system. If you have system partitions, you should have already designated the proper PSSP boot server nodes.

spbootins -s no -n *boot_node* -l *node_list*

You need to set the bootp response to **customize** for all the SP nodes.

spbootins -r **customize** -l *node_list*

You can issue the **splstdata -G -b** command to verify the correct boot information for the SP nodes.

The **spbootins** without the **-s no** flag runs the **setup_server** command which creates all the NIM-based files and resources required for installation. It also creates the authentication **rcmd** principals for the new SP node host names.

You may want to validate that the following files have the proper IP adresses and host names defined.

/etc/ntp.conf
/tftpboot/*host***.config_info**
/tftpboot/*host***.install_info**
/tftpboot/*host***-new-srvtab**

Note to the SP Administrator

If this SP system supports any PSSP 1,2 nodes, you should manually update the designated PSSP 1.2 boot server node. Refer to the "Updating the PSSP 1.2 Boot Server Node on PSSP 2.x Systems" on page 590 section.

12. If your SP system supports system partitions, you may reissue the system partitioning steps found Chapter 17, "Managing System Partitions" on page 229. The system partitioning **spapply_config** command re-creates the proper PSSP daemons and resynchronizes the SDR objects to reflect any IP address and host name changes.

    If you do not execute the system partitioning step, you will then need to create the source master objects and start the daemons for the Availability Services using the **syspar_ctrl -G -A** command.

    /usr/lpp/ssp/bin/syspar_ctrl -G -A

13. Update any other files on the control workstation that reflect IP address or host name changes.

    a. Update any files that are involved with host name resolution. The files are **/etc/hosts**, **/etc/resolv.conf** (DNS) and **/var/yp/*** (NIS).

    b. Update the **.rhosts** or **.klogin** files.

    c. Update the **/etc/filesystems** and **/etc/jmd_config.***SP_NAME* files for your SP configuration.

d. Make sure your **/tftpboot/script.cust** file is updated to reflect IP address and host name changes. Instead of hardcoding host names, you can reference the $SERVER and $CWS variables.

14. For each PSSP 2.2 and PSSP 2.1 boot server node, you should have already unconfigured NIM during PSSP node update activities. During the node customization, **setup_server** creates the proper install files for the nodes they are to customize. Since the SP nodes are in **customize** mode, most configuration files are updated to reflect the IP address and host name changes.

    Some files will not be updated on the nodes during the customization. You can **rcp** these files from the control workstation to the SP nodes by including them in the **/tftpboot/script.cust** file. These can include the **/etc/resolv.conf**, **/.rhosts**, and other SP customer-owned files.

    Perform a **REBOOT** for each PSSP 2.2 and PSSP 2.1 node. You can do this by using the System Monitor GUI, or by issuing the **hmcmds** command. You need to follow the same install sequence by customizing each of the boot server nodes first, and then customizing the remaining SP nodes.

    Note to the SP Administrator

    For PSSP 1.2 system partitions, you must first update all required files on the PSSP 1.2 boot server node. You must then perform a network boot on each PSSP 1.2 node, except for the current PSSP 1.2 boot server node. When the PSSP 1.2 nodes are properly updated, you can then perform a network boot on the primary PSSP 1.2 boot server node.

    You will now need to verify that the customization was successful. For more information, refer to the appropriate section (for PSSP 1.2 nodes, see "Execution on PSSP 1.2 Nodes" on page 581; for PSSP 2.1 nodes, see "Execution on PSSP 2.1 Nodes" on page 589; and for PSSP 2.2 nodes, see "Execution on PSSP 2.2 Nodes").

15. Consider not performing the HACWS reconfiguration until your SP system has been properly updated and is stable. When you are ready to reconfigure HACWS, refer to "Working with HACWS Configurations" on page 600.

Execution on PSSP 2.2 Nodes

The following verification steps need to be performed on each PSSP 2.2 node for changes made for the IP address and host name:

1. After reboot is complete and the SP nodes are customized, verify that the following have the correct IP address and host name specified:

    The updated SP Ethernet address specified for the SP node

    The updated default route and gateway are correct

    The additional adapters are correct (**en1**, **tr0**, **fi0**, **css0**)

    The host name resolution files were updated correctly (**/etc/hosts**)

    The **/etc/SDR_dest_info** file points to the control workstation and SDR Syspar IP address.

    The acct_master host name is updated for the NFS export list directory **/var/adm/acct** (SP Accounting)

The files in the **/etc/ssp** directory *cw_name*, *server_hostname*, *server_name*, and *reliable_hostname* reference the correct IP address and host names.

The **/etc/krb.conf**, **/etc/krb.realms**, and **/etc/krb-srvtab** files have the correct host names defined

2. You may need to modify the following files on the SP nodes to reflect the updated IP addresses and host name definitions:

   /etc/filesystems
   /etc/ntp.conf
   /etc/resolv.conf
   /.rhosts
   /.klogin

   If any of the files are incorrect, make the proper updates for the correct IP address or host name.

3. Verify that the NIM resources were built on the PSSP 2.2 boot server nodes by executing the **lsnim** command. If there were NIM problems, remove the current NIM database and configuration files.

   /usr/lpp/ssp/bin/delnimmast -l *node_number*

   Then issue **setup_server** on the PSSP 2.2 boot server node.

4. Verify that the Availability Services have been activated on the PSSP 2.2 nodes. Execute **lssrc -a** command to list all the active subsystems.

   /bin/lssrc -a | grep *Availability_susbsystem*

   If you suspect that any of the expected Availability Services are inoperative or not available on the SP node, it is best to remove and then add the Availability services using the **syspar_ctrl** command.

   /usr/lpp/ssp/bin/syspar_ctrl -c
   /usr/lpp/ssp/bin/syspar_ctrl -A

5. Reconfigure and start the IBM Virtual Shared Disk for the SP nodes that support the IBM Virtual Shared Disk configuration. You can do this by issuing the following IBM Virtual Shared Disk commands:

   /usr/lpp/csd/bin/cfgvsd -a
   /usr/lpp/csd/bin/startvsd -a

   You can validate that the IP address and host names are correctly specified by issuing the following command:

   /usr/lpp/csd/bin/vsdatalst -n

6. If there were modifications made to any SP node system files, it would be best to reboot the PSSP 2.2 nodes to reflect the IP address and host name changes. When the SP nodes are initialized, your SP system should be activated using the new IP addresses and host names.

Working with HACWS Configurations

The following additional information for working with an HACWS configuration in relation to working with 1 through 10 in “Execution on the PSSP 2.2 Control Workstation” on page 595.

You may also need to reference the HACWS information “Installing and Configuring the High Availability Control Workstation” in the *IBM Parallel System Support Programs for AIX: Installation and Migration Guide*.

1. Stop daemons. Stop HACMP on both the primary and backup control workstations (this automatically stops all HACWS related daemons).
2. Use **mktcpip**. Manually issue the appropriate **ifconfig** commands to configure the control workstation service addresses (backup CWS) on the primary control workstation. Make sure the host name and IP address are configured on the backup control workstation. If you are using an alias IP address, make the required changes to the **/etc/rc.backup_cw_alias** script.

   Manually vary on the external volume group which contains the **/spdata** file system, and then mount the **/spdata** filesystem.
3. Update **rc.net**. Make updates to the **/spdata/sys1/hacws/rc.syspar_aliases** file to reference the new alias address used for system partition.
4. Update **SDR_dest_info**. No updates required.
5. Issue **setup_authent**. No updates required.
6. Create the new source master. Update the new SDR source master objects on the backup control workstation as well as the primary control workstation.

   sdr -spname *SP_NAME* mksrc *new_IP_addr* (for each Syspar)

   You can then start the SDR and hardmon daemons on the primary control workstation.
7. Update the SDR. Update the Frame object to include the hostname for the **backup_MACN** using **SDRChangeAttrValues**.

   SDRChangeAttrValues Frame backup_MACN=*backup_CWS_hostname*

   You should also now execute the HACWS installation script **install_hacws** on the primary control workstation.

   /usr/sbin/hacws/install_hacws -p *primary_name* -b *backup_name* -s
8. Remove the current NIM. No updates required.
9. Update the SDR node objects. No updates required.
10. Reboot. To reconfigure the cluster topology, follow the standard HACMP procedures documented in *HACMP/6000: Administration Guide*.

    Verify that the HACWS configuration is properly setup by executing the **/usr/sbin/hacws/hacws_verify** command on the control workstation.

    Also verify that you can properly **fail over** from the control workstation to the backup control workstation.

Glossary of Terms and Abbreviations

This glossary includes terms and definitions from:

The *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.

The *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies can be purchased from the American National Standards Institute, 1430 Broadway, New York, New York 10018. Definitions are identified by the symbol (A) after the definition.

The *ANSI/EIA Standard - 440A: Fiber Optic Terminology*, copyright 1989 by the Electronics Industries Association (EIA). Copies can be purchased from the Electronic Industries Association, 2001 Pennsylvania Avenue N.W., Washington, D.C. 20006. Definitions are identified by the symbol (E) after the definition.

The *Information Technology Vocabulary*, developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.

The following cross-references are used in this glossary:

Contrast with. This refers to a term that has an opposed or substantively different meaning.

See. This refers the reader to multiple-word terms in which this term appears.

See also. This refers the reader to terms that have a related, but not synonymous, meaning.

Synonym for. This indicates that the term has the same meaning as a preferred term, which is defined in the glossary.

This section contains some of the terms that are commonly used in the SP publications.

IBM is grateful to the American National Standards Institute (ANSI) for permission to reprint its definitions from the American National Standard *Vocabulary for Information Processing* (Copyright 1970 by American National Standards Institute, Incorporated), which was prepared by Subcommittee X3K5 on Terminology and Glossary of the American National Standards Committee X3. ANSI definitions are preceded by an asterisk (*).

Other definitions in this glossary are taken from *IBM Vocabulary for Data Processing, Telecommunications, and Office Systems* (SC20-1699) and *IBM DATABASE 2 Application Programming Guide for TSO Users* (SC26-4081).

© Copyright IBM Corp. 1996

A

address. A character or group of characters that identifies a register, a device, a particular part of storage, or some other data source or destination.

AFS. A distributed file system that provides authentication services as part of its file system creation.

AIX. Abbreviation for Advanced Interactive Executive, IBM's licensed version of the UNIX operating system. AIX is particularly suited to support technical computing applications, including high function graphics and floating point computations.

AIX Version 3. The IBM AIX Version 3 for RISC System/6000 (also known as AIX/6000). It is based on UNIX System V, conforms with POSIX IEEE Standard 1003.1, is compatible with Berkeley Software Distribution 4.3 (4.3 BSD), and is designed to meet Trusted Computing Base level C2 security.

Amd. Berkeley Software Distribution automount daemon.

API. Application Programming Interface. A set of programming functions and routines that provide access between the Application layer of the OSI seven-layer model and applications that want to use the network. It is a software interface.

application. The use to which a data processing system is put; for example, a payroll application, an airline reservation application.

application data. The data that is produced using an application program.

ARP. Address Resolution Protocol.

ATM. Asynchronous Transfer Mode. (See *TURBOWAYS 100 ATM Adapter*.)

Authentication. The process of validating the identity of a user or server.

Authorization. The process of obtaining permission to perform specific actions.

B

batch processing. * (1) The processing of data or the accomplishment of jobs accumulated in advance in such a manner that each accumulation thus formed is processed or accomplished in the same run. * (2) The processing of data accumulating over a period of time. * (3) Loosely, the execution of computer programs serially. (4) Computer programs executed in the background.

BMCA. Block Multiplexer Channel Adapter. The block multiplexer channel connection allows the RS/6000 to communicate directly with a host System/370 or System/390; the host operating system views the system unit as a control unit.

C

call home function. The ability of a system to call the IBM support center and open a PMR to have a repair scheduled.

CDE. Common Desktop Environment. A graphical user interface for UNIX.

charge feature. An optional feature for either software or hardware for which there is a charge.

CLI. Command Line Interface.

client. * (1) A function that requests services from a server and makes them available to the user. * (2) A term used in an environment to identify a machine that uses the resources of the network.

Client Input/Output Sockets (CLIO/S). A software package that enables high-speed data and tape access between SP systems, AIX systems, and ES/9000 mainframes.

CLIO/S. Client Input/Output Sockets.

CMI. Centralized Management Interface provides a series of SMIT menus and dialogues used for defining and querying the SP system configuration.

connectionless. A communication process that takes place without first establishing a connection.

connectionless network. A network in which the sending logical node must have the address of the receiving logical node before information interchange can begin. The packet is routed through nodes in the network based on the destination address in the packet. The originating source does not receive an acknowledgment that the packet was received at the destination.

control workstation. A single point of control allowing the administrator or operator to monitor and manage the SP system using the IBM AIX Parallel System Support Programs.

css. Communication subsystem.

D

daemon. A process, not associated with a particular user, that performs system-wide functions such as administration and control of networks, execution of time-dependent activities, line printer spooling and so forth.

DASD. Direct Access Storage Device. Storage for input/output data.

DCE. Distributed Computing Environment.

DFS. distributed file system. A subset of the IBM Distributed Computing Environment.

DNS. Domain Name Service. A hierarchical name service which maps high level machine names to IP addresses.

E

Error Notification Object. An object in the SDR that is matched with an error log entry. When an error log entry occurs that matches the Notification Object, a user-specified action is taken.

ESCON. Enterprise Systems Connection. The ESCON channel connection allows the RS/6000 to communicate directly with a host System/390; the host operating system views the system unit as a control unit.

Ethernet. (1) Ethernet is the standard hardware for TCP/IP local area networks in the UNIX marketplace. It is a 10-megabit per second baseband type LAN that allows multiple stations to access the transmission medium at will without prior coordination, avoids contention by using carrier sense and deference, and resolves contention by collision detection (CSMA/CD). (2) A passive coaxial cable whose interconnections contain devices or components, or both, that are all active. It uses CSMA/CD technology to provide a best-effort delivery system.

Ethernet network. A baseband LAN with a bus topology in which messages are broadcast on a coaxial cabling using the carrier sense multiple access/collision detection (CSMA/CD) transmission method.

event. In Event Management, the notification that a predicate evaluated to true. This evaluation occurs each time an instance of a resource variable is observed.

expect. Programmed dialogue with interactive programs.

F

failover. Also called fallover, the sequence of events when a primary or server machine fails and a secondary or backup machine assumes the primary workload. This is a disruptive failure with a short recovery time.

fall back. Also called fall back, the sequence of events when a primary or server machine takes back control of its workload from a secondary or backup machine.

FDDI. Fiber Distributed Data Interface.

Fiber Distributed Data Interface (FDDI). An American National Standards Institute (ANSI) standard for 100-megabit-per-second LAN using optical fiber cables. An FDDI local area network (LAN) can be up to 100 km (62 miles) and can include up to 500 system units. There can be up to 2 km (1.24 miles) between system units and/or concentrators.

File Transfer Protocol (FTP). The Internet protocol (and program) used to transfer files between hosts. It is an application layer protocol in TCP/IP that uses TELNET and TCP protocols to transfer bulk-data files between machines or hosts.

file. * A set of related records treated as a unit, for example, in stock control, a file could consist of a set of invoices.

file server. A centrally located computer that acts as a storehouse of data and applications for numerous users of a local area network.

foreign host. Any host on the network other than the local host.

FTP. File transfer protocol.

G

gateway. An intelligent electronic device interconnecting dissimilar networks and providing protocol conversion for network compatibility. A gateway provides transparent access to dissimilar networks for nodes on either network. It operates at the session presentation and application layers.

H

HACMP/6000. AIX High Availability Cluster Multi-Processing/6000.

HACWS. High Availability Control Workstation function, based on HACMP/6000, provides for a backup control workstation for the SP system.

help key. In the SP graphical interface, the key that gives you access to the SP graphical interface help facility.

High Availability Cluster Multi-Processing/6000. An IBM facility to cluster nodes or components to provide high availability by eliminating single points of failure.

High Performance Switch. An IBM multi-stage packet switch for high-performance communication between processor nodes.

HiPPI. High Performance Parallel Interface. RS/6000 units can attach to a HiPPI network as defined by the ANSI specifications. The HiPPI channel supports burst rates of 100 Mbps over dual simplex cables; connections can be up to 25 km in length as defined by the standard and can be extended using third-party HiPPI switches and fiber optic extenders.

home directory. The directory associated with an individual user.

host. A computer connected to a network, and providing an access method to that network. A host provides end-user services.

HSD. The data striping device for the IBM Virtual Shared Disk. The device driver lets application programs stripe data across physical disks in multiple IBM Virtual Shared Disks, thus reducing I/O bottlenecks and hot spots.

I

instance vector. In Event Management, a set of elements, where each element is a name/value pair of the form name=value, whose values uniquely identify the copy of the resource (and by extension, the copy of the resource variable) in the system.

Internet. A specific inter-network consisting of large national backbone networks such as APARANET, MILNET, and NSFnet, and a myriad of regional and campus networks all over the world. The network uses the TCP/IP protocol suite.

Intermediate Switch Board. Switches mounted in the High Performance Switch expansion frame.

IP address. A 32-bit address assigned to devices or hosts in an IP internet that maps to a physical address. The IP address is composed of a network and host portion.

Internet Protocol (IP). (1) A protocol that routes data through a network or interconnected networks. IP acts as an interface between the higher logical layers and the physical network. This protocol, however, does not provide error recovery, flow control, or guarantee the reliability of the physical network. IP is a connectionless protocol. (2) A protocol used to route data from its source to it destination in an Internet environment.

ISB. Intermediate Switch Board.

K

Kerberos. A service for authenticating users in a network environment.

kernel. The core portion of the UNIX operating system which controls the resources of the CPU and allocates them to the users. The kernel is memory-resident, is said to run in “kernel mode” and is protected from user tampering by the hardware.

L

LAN. (1) Acronym for Local Area Network, a data network located on the user's premises in which serial transmission is used for direct data communication among data stations. (2) Physical network technology that transfers data a high speed over short distances. (3) A network in which a set of devices is connected to another for communication and that can be connected to a larger network.

local host. The computer to which a user's terminal is directly connected.

log database. A persistent storage location for the logged information.

log event. The recording of an event.

log event type. A particular kind of log event that has a hierarchy associated with it.

logging. The writing of information to persistent storage for subsequent analysis by humans or programs.

M

mask. To use a pattern of characters to control retention or elimination of portions of another pattern of characters.

menu. A display of a list of available functions for selection by the user.

Motif. The graphical user interface for OSF, incorporating the X Window System. Also called OSF/Motif.

MTBF. Mean time between failure. This is a measure of reliability.

MTTR. Mean time to repair. This is a measure of serviceability.

N

naive application. An application with no knowledge of a server that fails over to another server. Client to server retry methods are used to reconnect.

network. An interconnected group of nodes, lines, and terminals. A network provides the ability to transmit data to and receive data from other systems and users.

NFS. Network File System. NFS allows different systems (UNIX or non-UNIX), different architectures, or vendors connected to the same network, to access remote files in a LAN environment as though they were local files.

NIM. Network Installation Management is provided with AIX 4.1 to install AIX on the nodes.

NIM client. An AIX system installed and managed by a NIM master. NIM supports three types of clients:

Standalone
Diskless
Dataless

NIM master. An AIX system that can install one or more NIM clients. An AIX system must be defined as a NIM master before defining any NIM clients on that system. A NIM master managers the configuration database containing the information for the NIM clients.

NIM object. A representation of information about the NIM environment. NIM stores this information as objects in the NIM database. The types of objects are:

Network
Machine
Resource

NIS. Network Information System.

node. In a network, the point where one or more functional units interconnect transmission lines. A computer location defined in a network.

Node Switch Board. Switches mounted on frames that contain nodes.

NSB. Node Switch Board.

NTP. Network Time Protocol.

O

ODM. Object Data Manager. In AIX, a hierarchical object-oriented database for configuration data.

P

parallel environment. A system environment where message passing or SP resource manager services are used by they application.

Parallel Environment. A licensed IBM program used for message passing applications on the SP or RS/6000 platforms.

parallel processing. A multiprocessor architecture which allows processes to be allocated to tightly coupled multiple processors in a cooperative processing environment, allowing concurrent execution of tasks.

parameter. * (1) A variable that is given a constant value for a specified application and that may denote the application. * (2) An item in a menu for which the operator specifies a value or for which the system provides a value when the menu is interpreted. * (3) A name in a procedure that is used to refer to an argument that is passed to the procedure. * (4) A particular piece of information that a system or application program needs to process a request.

partition. See system partition.

Perl. Practical Extraction and Report Language.

pipe. A UNIX utility allowing the output of one command to be the input of another. Represented by the | symbol. It is also referred to as filtering output.

perspective. The primary window for each SP Perspectives application, so called because it provides a unique view of an SP system.

POE. Formerly Parallel Operating Environment, now Parallel Environment for AIX.

port. (1) An end point for communication between devices, generally referring to physical connection. (2) A 16-bit number identifying a particular TCP or UDP resource within a given TCP/IP node.

predicate. In Event Management, the relational expression between a resource variable and other elements (such as constants or the previous value of an instance of the variable) that, when true, generates an event. An example of a predicate is $X < 1$ where X represents the resource variable IBM.PSSP.aixos.PagSp.%totalfree (the percentage of total free paging space). When the predicate is true, that is, when the total free paging space is observed to

be less than 10%, the Event Management subsystem generates an event to notify the appropriate application.

process. * (1) A unique, finite course of events defined by its purpose or by its effect, achieved under defined conditions. * (2) Any operation or combination of operations on data. * (3) A function being performed or waiting to be performed. * (4) A program in operation. For example, a daemon is a system process that is always running on the system.

protocol. A set of semantic and syntactic rules that defines the behavior of functional units in achieving communication.

PMR. Problem Management Report.

Problem Management Report. The number in the IBM support mechanism that represents a service incident with a customer.

Primary node or machine. (1) A device that runs a workload and has a standby device ready to assume the primary workload if that primary node fails or is taken out of service. (2) A node on the High Performance Switch that initializes, provides diagnosis and recovery services, and performs other operations to the switch network. (3) In IBM Virtual Shared Disk function, the node at which the logical volume is actually local and that acts as server node to I/O requests from other nodes.

R

RAID. Redundant array of independent disks.

rearm predicate. In Event Management, a predicate used to generate an event that alternates with an original event predicate in the following way: the event predicate is used until it is true, then the rearm predicate is used until it is true, then the event predicate is used, and so on. The rearm predicate is commonly the inverse of the event predicate (for example, a resource variable is on or off). It can also be used with the event predicate to define an upper and lower boundary for a condition of interest.

remote host. *See foreign host*.

resource. In Event Management, an entity in the system that provides a set of services. Examples of resources include hardware entities such as processors, disk drives, memory, and adapters, and software entities such as database applications, processes, and file systems. Each resource in the system has one or more attributes that define the state of the resource.

resource monitor. A program that supplies information about resources in the system. It can be a command, a daemon, or part of an application or subsystem that manages any type of system resource.

resource variable. In Event Management, the representation of an attribute of a resource. An example of a resource variable is IBM.AIX.PagSp.%totalfree, which represents the percentage of total free paging space. IBM.AIX.PagSp specifies the resource name and %totalfree specifies the resource attribute.

RISC. Reduced Instruction Set Computing (RISC), the technology for today's high performance personal computers and workstations, was invented in 1975. Uses a small simplified set of frequently used instructions for rapid execution.

rlogin (remote LOGIN). A service offered by Berkeley UNIX systems that allows authorized users of one machine to connect to other UNIX systems across a network and interact as if their terminals were connected directly. The rlogin software passes information about the user's environment (for example, terminal type) to the remote machine.

RPC. Acronym for Remote Procedure Call, a facility that a client uses to have a server execute a procedure call. This facility is composed of a library of procedures plus an XDR.

RSH. A variant of RLOGIN command that invokes a command interpreter on a remote UNIX machine and passes the command line arguments to the command interpreter, skipping the LOGIN step completely. See also *rlogin*.

S

SCSI. Small Computer System Interface.

server. (1) A function that provides services for users. A machine may run client and server processes at the same time. (2) A machine that provides resources to the network. It provides a network service, such as disk storage and file transfer, or a program that uses such a service. (3) A device, program, or code module on a network dedicated to providing a specific service to a network. (4) On a LAN, a data station that provides facilities to other data stations. Examples are file server, print server, and mail server.

Small Computer System Interface (SCSI). An input and output bus that provides a standard interface for the attachment of various direct access storage devices (DASD) and tape drives to the RS/6000.

shell. The shell is the primary user interface for the UNIX operating system. It serves as command language interpreter, programming language, and allows foreground and background processing. There

are three different implementations of the shell concept: Bourne, C and Korn.

Small Computer Systems Interface Adapter (SCSI Adapter). An adapter that supports the attachment of various direct-access storage devices (DASD) and tape drives to the RS/6000.

SMIT. The System Management Interface Toolkit is a set of menu driven utilities for AIX that provides functions such as transaction login, shell script creation, automatic updates of object database, and so forth.

SNMP. Simple Network Management Protocol. (1) An IP network management protocol that is used to monitor attached networks and routers. (2) A TCP/IP-based protocol for exchanging network management information and outlining the structure for communications among network devices.

socket. (1) An abstraction used by Berkeley UNIX that allows an application to access TCP/IP protocol functions. (2) An IP address and port number pairing. (3) In TCP/IP, the Internet address of the host computer on which the application runs, and the port number it uses. A TCP/IP application is identified by its socket.

standby node or machine. A device that waits for a failure of a primary node in order to assume the identity of the primary node. The standby machine then runs the primary's workload until the primary is back in service.

subnet. Shortened form of subnetwork.

subnet mask. A bit template that identifies to the TCP/IP protocol code the bits of the host address that are to be used for routing for specific subnetworks.

subnetwork. Any group of nodes that have a set of common characteristics, such as the same network ID.

SUP. Software Update Protocol.

Sysctl. Secure System Command Execution Tool. An authenticated client/server system for running commands remotely and in parallel.

System Administrator. The user who is responsible for setting up, modifying, and maintaining the SP system.

subsystem. A software component that is not usually associated with a user command. It is usually a daemon process. A subsystem will perform work or provide services on behalf of a user request or operating system request.

syslog. A BSD logging system used to collect and manage other subsystem's logging data.

system partition. A group of nonoverlapping nodes on a switch chip boundary that act as a logical SP system.

T

tar. Tape ARchive, is a standard UNIX data archive utility for storing data on tape media.

Tcl. Tool Command Language.

TclX. Tool Command Language Extended.

TCP. Acronym for Transmission Control Protocol, a stream communication protocol that includes error recovery and flow control.

TCP/IP. Acronym for Transmission Control Protocol/Internet Protocol, a suite of protocols designed to allow communication between networks regardless of the technologies implemented in each network. TCP provides a reliable host-to-host protocol between hosts in packet-switched communications networks and in interconnected systems of such networks. It assumes that the underlying protocol is the Internet Protocol.

Telnet. Terminal Emulation Protocol, a TCP/IP application protocol that allows interactive access to foreign hosts.

Tk. Tcl-based Tool Kit for X Windows.

TMPCP. Tape Management Program Control Point.

token-ring. (1) Network technology that controls media access by passing a token (special packet or frame) between media-attached machines. (2) A network with a ring topology that passes tokens from one attaching device (node) to another. (3) The IBM Token-Ring LAN connection allows the RS/6000 system unit to participate in a LAN adhering to the IEEE 802.5 Token-Passing Ring standard or the ECMA standard 89 for Token-Ring, baseband LANs.

transaction. An exchange between the user and the system. Each activity the system performs for the user is considered a transaction.

transceiver (transmitter-receiver). A physical device that connects a host interface to a local area network, such as Ethernet. Ethernet transceivers contain electronics that apply signals to the cable and sense collisions.

transfer. To send data from one place and to receive the data at another place. Synonymous with move.

transmission. * The sending of data from one place for reception elsewhere.

TURBOWAYS 100 ATM Adapter. An IBM high-performance, high-function intelligent adapter that provides dedicated 100 Mbps ATM (asynchronous transfer mode) connection for high-performance servers and workstations.

U

UDP. User Datagram Protocol.

User Datagram Protocol (UDP). (1) In TCP/IP, a packet-level protocol built directly on the Internet Protocol layer. UDP is used for application-to-application programs between TCP/IP host systems. (2) A transport protocol in the Internet suite of protocols that provides unreliable, connectionless datagram service. (3) The Internet Protocol that enables an application programmer on one machine or process to send a datagram to an application program on another machine or process.

UNIX operating system. An operating system developed by Bell Laboratories that features multiprogramming in a multiuser environment. The UNIX operating system was originally developed for use on minicomputers, but has been adapted for mainframes and microcomputers. **Note:** The AIX operating system is IBM's implementation of the UNIX operating system.

user. Anyone who requires the services of a computing system.

user ID. A nonnegative integer, contained in an object of type *uid_t*, that is used to uniquely identify a system user.

V

Virtual Shared Disk, IBM. The function that allows application programs executing at different nodes of a system partition to access a raw logical volume as if it were local at each of the nodes. In actuality, the logical volume is local at only one of the nodes (the server node).

W

workstation. * (1) A configuration of input/output equipment at which an operator works. * (2) A terminal or microcomputer, usually one that is connected to a mainframe or to a network, at which a user can perform applications.

X

X Window System. A graphical user interface product.

Index

Special Characters

/etc/.profile file 53
/etc/aliases file 124
/etc/amd/amd-maps directory 106, 108
/etc/amd/amd-maps/amd.u map 108
/etc/cshutSeq
  definition of 20
  example 28
  if empty 20
  if none exists 20
  syntax of 26
/etc/cstartSeq
  definition of 20
  example 28
  if empty 20
  if none exists 20
  syntax of 26
/etc/environment file 54, 55
/etc/ha/cfg directory
  EMCDB version string files 359
  run-time EMCDB files 359
/etc/inittab file
  use by Event Management 361
  use by Group Services 343
/etc/krb.conf file 169
/etc/krb.realms file 169
/etc/ntp.conf file 101
/etc/rc.tcpip file 124
/etc/SDR_dest_info file 544
/etc/sendmail.cf file 122
/etc/services file 168
  use by Event Management 353, 354
  use by Group Services 339
/etc/subsysSeq
  definition of 20
  example 29
  if none exists or if empty 20
  syntax of 26
/etc/syslog.conf file 285
/spdata/sys1/ha/cfg directory
  staging area for EMCDB files 357
/usr/lib/X11/app-defaults/SPmon file 512
/usr/local/fonts file 58
/usr/lpp/ssp/bin directory 112
/usr/lpp/ssp/bin/newpass.log file 72
/usr/lpp/ssp/config/cron_template 142
/usr/lpp/ssp/install/config/haemloadlist file
  listing resource variable information 535
/var file system
  and Event Management tracing 362
  and Group Services tracing 343
/var/adm/acct/nite directory 144
/var/adm/cacct/ directory 145
/var/adm/cacct/fiscal directory 145
/var/adm/cacct/sum directory 145
/var/adm/SPlogs/amd.log file 107
/var/adm/SPlogs/cs/cshut.*MMDDhhmmss.pid* 25
/var/adm/SPlogs/cs/cstart.*MMDDhhmmss.pid* 25
/var/ha/lck directory
  Event Management lock files 358
  Group Services lock files 340
/var/ha/log directory
  Event Management log files 358
  Group Services log files 340
/var/ha/run directory
  Event Manager daemon working files 358
  Group Services daemon working files 341
  registration cache 358
  resource monitor working directories 358
/var/ha/soc directory
  Event Management socket files 357
  Group Services socket files 340
/var/spool/lpd/qdir directory 114
/var/spool/qdaemon directory 114
/var/sysman/file.collections file 42
/var/sysman/sup/.resident file 42
/var/sysman/sup/lists file 42
/var/sysman/sup/node.root/prefix file 55
/var/sysman/sup/user.admin/list file 55
/var/sysman/sup/user.admin/prefix file 55
/var/sysman/sup.resident file 59
.forward file 124, 128
.klogin file 165
.profile 63
.resident file 59
.rhosts file 165

Numerics

604 High Node 2
  time to power on 21
604 High Nodes 6

A

access to the configuration data 543
accounting 10
  accounting
  additional setup tasks 141
  by node or by class 133
  differences between SP and standard AIX 133
  entering node accounting information 139
  entering system information 136

© Copyright IBM Corp. 1996

accounting *(continued)*
files 142
for exclusive use of node jobs 134
HACWS and 331
installation 141
report and summary files 143
setting up SP 135
system partitioning 136
template for crontabs 142
adapter failures 323
Adapter Membership
resource monitor 353
adapters, network connectivity 8
adding and deleting files in a file collection 54
adding subsystems
Event Management (haemctrl) 361
Group Services (hagsctrl) 342
adding user accounts 70
administrative tasks, scheduling 407
ADSM
tuning for 425
AFS
adding principals to 176
administering AFS authentication services 183
changing passwords in 164
AFS authentication services 182
afsclient.cust 475
AIX 8
AIX error log
Group Services entries 345
use by Event Manager daemon 358
AIX error log facility
configuring 395
SMIT, from 395
creating SNMP traps from 384
generating reports 394
managing 393
SMIT, from 393
trimming logs 394
SMIT, from 394
AIX operating system resources
resource monitor 353
AIX operating system statistics
obtained from SPMI 359
aixos resource monitor 353
alternate boot system image
enabling 467
using 467
Amd 105
amd.u map 107
diagnostics 107
HACWS and 331
maps 106
start and stop
amd.u map 107
applications
authentication
adding IBM RS/6000 workstations to your realm 167
adding principals 175
administering SP authentication services
adding principals 175
changing admin instance password 178
changing passwords 177
setting passwords 175
administration roles 166
AFS
adding principals 176
administering 183
changing passwords in 164
AFS services 182
authentication client system 172
authorization as an authentication database administrator 176
backing up database 173
changing admin instance password 178
changing passwords 177
changing your password 164
configuration files 168
credentials 160
destroying tickets 163
directory paths for services 172
HACWS and 326
identifying yourself 160
implementations 167
listing tickets 163
managing information for authenticated services
changing service keys 181
managing the database 173
primary server 170
protecting your database 166
realms 158
remote commands 165
secondary servers 167
service keys 179
services on the SP 151, 164
services, using 160
setting passwords 175
setting up a secondary server 171
setting up an authentication client system 172
terminology 152
understanding 151
using the SP authentication services
changing your password 164
destroying tickets 163
identifying yourself 160
listing tickets 163
authentication, SDR 543
authority levels for SP System Monitor
control authority 283
monitor authority 283

authorization, SDR 543
availability 10
availablity services 3

B

backing up authentication database 173
basecode mode hardware variables for the SP Switch 532
battery, SP virtual 467
boot/install server
  sequencing of startup and shutdown 19
bootlist, setting 468, 469
bootptab 480
BSD syslog facility
  generating reports 398
  managing 397
building a file collection 56

C

cache, registration
  Event Management 358
cardtype 17 and 20 unique frame hardware variables 514
change management 10
changing amd defaults 105
changing tape device attributes 453, 458
changing user accounts 72
changing your password 75
characteristics of SP system
cleaning subsystems
  Event Management (haemctrl) 362
  Group Services (hagsctrl) 343
client
  type of resource monitor 352
client communication
  with Event Management subsystem 351
  with Group Services subsystem 338
client, Group Services
  definition 337
CLIO/S, tuning 428
closing windows to end the display manager session 287
cluster
  and Event Management 351
cluster workstation
  and Event Management 351
cmonacct 134
collecting system data for IBM service 402
command
  type of resource monitor 352
command files, nrunacct 143
command line, from the 395
command line, using the 239, 241, 243, 246, 247, 249, 250, 251
commands
  and system partitioning 230
  cshutdown 23
  cstartup command 20
  df 548
  dsh 31
  dspcat
    displaying event management descriptions 538
  dspmsg
    displaying event management descriptions 538
  Eprimary
  Estart
  hostlist 33
  jm_config 91
  jm_start 94
  jm_status 98
  jm_stop 97
  jm_verify 90
  locate_jm 98
  lscfg 548
  mkamdent 108
  netstat 548
  omit 64
  omitany 64
  p_cat 35
  passwd 75
  pcp 35
  pdf 37
  pexec 34
  pfind 35
  pfps 37
  pls 35
  pmv 35
  ppred 36
  pps 35
  prm 35
  SDRGetObjects
    listing resource variable information 535
    output example 537
  spchuser 67
  spevent
    listing resource variable information 538
    output example 539
  splsuser 67
  spmkuser 67
  sprmuser 67
  supper activate 46
  supper install 48
  supper offline 61
  supper reset 47
  supper scan 48
  supper status 43
  supper update 48, 110
  symlinkall 65
  yppasswd 75
  zcat 108

communication subsystem support 10
communication, client
  with Event Management subsystem 351
  with Group Services subsystem 338
communications, Event Management
  between Event Manager daemons 353
  local EM clients 354
  remote EM clients 354
  resource monitors 354
communications, Group Services
  between Group Services daemons 339
  local GS clients 340
configuration data
  activating 368
  changing in the SDR 367
  for third-party resource monitors 355, 357, 367
configuration data (SDR)
  component of Event Management 354
configuration database
  component of Event Management 356
configuration load utility
  component of Event Management 357
configuration management 9
configuration utility
  component of Event Management 355
configuring
  tape device 448
  tape management pseudo control point 438
    using SMIT 438
configuring resource manager 88
configuring the system monitor 284
configuring the user home environment 285
connectivity adapters, network 8
connectivity, switch 222
control script
  component of Event Management 356
  component of Group Services 340
control workstation 7
  booting for HACWS 322
  failovers 324
  high availability 317
  preparing
controlling login 77
core file
  Event Manager daemon 358
  Group Services daemon 341
cprdaily 134
create class command 193
create proc command 193
create var command 193
creating custom layouts
credentials
  deleting 163
  displaying 163
  managing 160
  obtaining 160
crontabs file 45, 112, 147
crontabs, accounting template for 142
cshutdown command
  log file 25
  using 23
CSS MTU size, tuning 416
CSS subsystem
  resource monitor 352
cssMembership group 353
  use by Event Management 360
cstartup command
  log file 25
  safety warning 20
customizing
  system partitions 244

D

daemon
  type of resource monitor 352
daemons 471
  lpd 111
  sendmail 120, 123, 124
  srcmstr 131
data files 142
deconfiguring a tape management pseudo control point 442
default configuration data
  component of Event Management 357
  loading into the SDR with haemloadcfg 357
  location for Event Management 357
default predicate
  changing 355
default system partition 234
defining
  tape device 448
  tape management pseudo control point 438
    using SMIT 438
defining and configuring a tape device 448
defining and configuring a tape management pseudo control point 438
defining and configuring a tape management pseudo control point using SMIT 438
deleting subsystems
  Event Management (haemctrl) 362
  Group Services (hagsctrl) 343
deleting user accounts 73
descriptions, displaying event management
  using the dspcat command 538
  using the dspmsg command 538
determining if nodes are operating 306
determining switch connectivity 222
device driver
  pseudo 434
df command 548

dialog boxes of the SP Perspectives GUI 273
disks for HACWS 319
displaying event management descriptions
  using the dspcat command 538
  using the dspmsg command 538
displaying frame and node layouts 291
displaying the configuration data 547
distributed shell 31
distributed tape management system (DTMS) enablement
  magnetic tape extension (MTE) pseudo device driver 434
  overview 433
  site-defined exit facility 436
  tape management pseudo control point (TMPCP) 434
distribution of mail-related files 131
domain, operational
  for Event Management 350
  for Group Services 338
dsh 31
dsh based 34
dsh command 31
dspcat command
  displaying event management descriptions 538
dspmsg command
  displaying event management descriptions 538
DTMS
  magnetic tape extension (MTE) pseudo device driver 434
  overview 433
  site-defined exit facility 436
  tape management pseudo control point (TMPCP) 434

E

EM client (Event Management client)
  definition 349
  registration request cache 358
  restrictions on number 364
EM_Instance_Vector SDR class
  key attributes for loading 357
  use by Event Management 354
EM_Resource_Class SDR class
  key attributes for loading 357
  use by Event Management 355
EM_Resource_Monitor SDR class
  key attributes for loading 357
  use by Event Management 355
EM_Resource_Variable SDR class
  key attributes for loading 357
  use by Event Management 354
EM_Structured_Byte_String SDR class
  key attributes for loading 357
  use by Event Management 354
EMAPI (Event Management Application Programming Interface)
  component of Event Management 351
EMAPI libraries
  location 351
embedded resource monitor
  type of resource monitor 352
EMCDB (Event Management Configuration Database)
  and Event Manager daemon initialization 364
  compiling with haemcfg 355
  placing a new version in production 364
  run-time directory 359
  run-time file 356
  staging directory 357
  staging file 355
  version string 355
  version string and joining peer group 363
  version string and peer group state 360
  version string directory 359
Emonitor 227
Emonitor subsystem
  and system partitioning 253
en_pend method flow 396
enMembership group 353
  use by Event Management 360
Eprimary command
Equiesce 227
error log management
  archiving error logs 400
  collecting system data for IBM service 402
  creating archives 401
  creating service collections 403
  gathering archives 402
  gathering service collections 404
  installing and configuring 392
  removing archives 401
  removing service collections 404
  viewing error logs 400
error logs
  archiving 400
  creating archives 401
  gathering archives 402
  managing 391
  removing archives 401
  viewing 400
error message output log
  Event Management 358
error notification objects, managing 395
error templates
  managing 397
ESCON
  tuning for 427
Estart command
Event Management
  abnormal termination of haemctrl add 362
  disk space and tracing 362

Event Management *(continued)*
performance and tracing 362
shared variables with performance monitoring 359
Event Management API (EMAPI)
component of Event Management 351
Event Management client (EM client)
definition 349
Event Management communications
between Event Manager daemons 353
local EM clients 354
remote EM clients 354
resource monitors 354
Event Management Conditions subsystem
and system partitioning 253
Event Management configuration
changing 355
SDR classes 354
Event Management Configuration Database (EMCDB)
and Event Manager daemon initialization 364
compiling with haemcfg 355
creating 356
placing a new version in production 364
run-time directory 359
staging directory 357
version string and joining peer group 363
version string and peer group state 360
version string directory 359
Event Management directories
/etc/ha/cfg 359
/spdata/sys1/ha/cfg 357
/var/ha/lck 358
/var/ha/log 358
/var/ha/run 358
/var/ha/soc 357
Event Management SDR classes
key attributes for loading 357
Event Management subsystem
activating configuration data 368
adding with haemctrl command 361
and system partitioning 253
changing configuration data in the SDR 367
cleaning with haemctrl command 362
client communication 351
component summary 349
components 349—359
configuration 361—362
configuring and operating 349—368
deleting with haemctrl command 362
dependencies 359—360
Event Manager daemon initialization 362—364
Event Manager daemon operation 364—365
getting subsystem status 365—367
installation 361
introducing 349
loading non-PSSP data into the SDR 367
operational domain 350
Event Management subsystem *(continued)*
recovery from failure (automatic) 364
starting with haemctrl command 362
stopping with haemctrl command 362
tracing with haemctrl command 362
Event Manager daemon
abnormal termination core file 358
communications 353
component of Event Management 350
current working directory 358
definition 349
error message log file 358
getting status 365—367
group membership 360
group subscriptions 360
haemctrl control script 356
initialization 362—364
message trace output log file 358
operation 364—365
recovery from failure (automatic) 364
trace output log file 358
use of AIX error log 358
event processing 332
examples
resource data
output from the SDRGetObjects command 537
using SP Perspectives 539
exit facility, site-defined 436
expression
synonymous with predicate 538

F

failovers of control workstation 324
fencing nodes 223
file collection
adding and deleting files 54
building 56
HACWS and 331
installing 59
from a specific server
managing 41
modifying the hierarchy 61
organization 45
refusing files 61
removing 63
structure 45
tasks
updating 52
using scan to verify 51
file management 9
files
/etc/.profile 53
/etc/aliases 124
/etc/environment 54, 55
/etc/ntp.conf 101

files *(continued)*
/etc/rc.tcpip 124
/etc/sendmail.cf 122
/etc/syslog.conf 285
/usr/lib/X11/app-defaults/SPmon 512
/usr/local/fonts 58
/usr/lpp/ssp/bin/newpass.log 72
/usr/lpp/ssp/config/cron_template 142
/var/adm/SPlogs/amd.log 107
/var/sysman/file.collections 42
/var/sysman/sup/.resident 42
/var/sysman/sup/lists 42
/var/sysman/sup/node.root/prefix 55
/var/sysman/sup/user.admin/list 55
/var/sysman/sup/user.admin/prefix 55
/var/sysman/sup.resident 59
.forward 124, 128
.profile 63
.resident 59
accounting report and summary 143
afsclient.cust 475
crontabs 45, 112
ibmSPMIB.my 481
jmd_config 502
last 48
list 54
logs 53
prefix 54
refuse 61
sendmail.cf 119
when 48
files and directories
component of Event Management 357
component of Group Services 340
files, accounting 142
files, data 142
files, nrunacct command 143
filesets, Event Management
ssp.clients 361
ssp.ha 361
filesets, Group Services
ssp.ha 342
first boot customization file (script.cust)
sample file 505
first boot customization file (tuning.cust)
sample file 510
frame hardware variables 513, 526
frame supervisor cables 334
frame supervisor cards for HACWS 320
frames 7
from SMIT 395, 397
from the command line 395

G

group membership list
definition 337
Group Services
abnormal termination of hagsctrl add 343
disk space and tracing 343
performance and tracing 343
Group Services API (GSAPI)
component of Group Services 339
Group Services client
definition 337
Group Services communications
between Group Services daemons 339
local GS clients 340
Group Services daemon
abnormal termination core file 341
communications 339
component of Group Services 338
current working directory 341
getting status 346—347
hagsctrl control script 340
initialization 343—345
operation 345
recovery from failure (automatic) 345
trace output log file 340
Group Services directories
/var/ha/lck 340
/var/ha/log 340
/var/ha/run 341
/var/ha/soc 340
Group Services subsystem
adding with hagsctrl command 342
and Event Manager daemon initialization 363
and system partitioning 252
cleaning with hagsctrl command 343
client communication 338
component summary 338
components 338—341
configuration 342—343
configuring and operating 337—347
deleting with hagsctrl command 343
dependencies 341
dependency by Event Management 360
getting subsystem status 346—347
Group Services daemon initialization 343—345
Group Services daemon operation 345
initialization errors 345
installation 342
introducing 337
operational domain 338
recovery from failure (automatic) 345
starting with hagsctrl command 343
stopping with hagsctrl command 343
system groups 353
tracing with hagsctrl command 343

group state
  and joining the ha_em_peers group 363
  EMCDB version string 360
group state value
  definition 337
group, Group Services
  definition 337
groups
  Group Services
    restrictions on number per client 345
    restrictions on number per domain 345
GS client (Group Services client)
  restrictions on number 345
GS nameserver
  establishing 344
GSAPI (Group Services Application Programming Interface)
  component of Group Services 339
GSAPI libraries
  location 339

H

ha_em_peers group
  joining 363
  use by Event Management 360
HACMP/6000
  configuring resources for 332
  customizing for HACWS 332
  event processing 332
  HACWS foundation 320
  operating for the control workstation 322
HACWS
  See also High Availability Control Workstation (HACWS)
  managing 317
HACWS (High Availability Control Workstation)
  and Event Management 361
  and Group Services 343
haemcfg command
  compiling the EMCDB 355
haemctrl command
  adding the Event Management subsystem 361
  cleaning the Event Management subsystem 362
  control script for Event Management 356
  deleting the Event Management subsystem 362
  starting the Event Management subsystem 362
  stopping the Event Management subsystem 362
  summary of functions 361
  tracing the Event Management subsystem 362
haemd daemon
  location 350
haemloadcfg command
  differences from SDRCreateObjects 357
  Event Management configuration load utility 357
  use of loadsdr program 357
haemloadlist file
  default Event Management configuration data 357
haemrcpcdb script
  use by Event Management 364
hagsctrl command
  adding the Group Services subsystem 342
  cleaning the Group Services subsystem 343
  control script for Group Services 340
  deleting the Group Services subsystem 343
  starting the Group Services subsystem 343
  stopping the Group Services subsystem 343
  summary of functions 342
  tracing the Group Services subsystem 343
hagsd daemon
  location 338
halting the SP 23
hardware monitor resource monitor 352
hardware overview 5
Heartbeat subsystem
  and system partitioning 252
help for the SP Perspectives GUI 277
High Availability Control Workstation (HACWS)
  and Event Management 361
  and Group Services 343
  configuration changes 325
  customizing HACMP/6000 for 332
  failovers of control workstations 324
  hardware 319
  managing subsystems for 326, 327, 328, 329, 330, 331
    accounting 331
    Amd 331
    file collections 331
    High Performance Switch 328
    mail service 331
    NTP 330
    parallel management commands 330
    print service 331
    Resource Manager 330
    security features 326
    SP Switch 329
    SP System Monitor 327
    starting up and shutting down system 328
    Sysctl 327
    system partitions 327
    user accounts 330
  software 320
  terminology 318
  understanding 318
high availability services
  Event Management subsystem 349
  Group Services subsystem 337, 349
  Topology Services subsystem 337, 349
high performance switch 7
  HACWS and 328

High Performance Switch variables 528
high-performance switch
  configuring
highlights of PSSP software 9
Host Membership
  resource monitor 353
host name changes for SP systems 575
host_responds
  updating 550
Host_Responds daemon
  dependency by Event Management 360
host_responds SDR class
  and Event Management 360
  resource monitor 353
Host_Responds subsystem
  and system partitioning 253
hostlist command 33
HostMembership group 353
  use by Event Management 360
hr script
  Host_Responds daemon 360

I

IBM parallel system support programs for AIX (PSSP) 9
IBM products, other 15
IBM.PSSP.harmld resource monitor 352
IBM.PSSP.harmpd resource monitor 352
IBM.PSSP.hmrmd resource monitor 352
IBM.PSSP.Membership.LANAdapter.state resource variable 360
IBM.PSSP.Membership.Node.state resource variable 360
IBM.PSSP.pm.User_state resource variables 377
IBM.PSSP.pmanrmd resource monitor 352
IBM.PSSP.Response.Host.state resource variable 360
IBM.PSSP.Response.Switch.state resource variable 360
ibmSPMIB.my 481
include command 192
independent software vendor products 15
inetd.conf file 168
installation 9
  Event Management subsystem 361
  Group Services subsystem 342
installation support 2
installing
  enhanced physical tape device driver 447
  MTE pseudo device driver 436
installing a file collection 59
installing a file collection from a specific server
installing resource manager 86
installing the enhanced physical tape device driver 447
instance vector
  configuration in SDR 354
integrating mail services 119
  using /etc/hosts 128
  using DNS 120
  using NIS 124
integrating print services 111
internal resource monitor
  definition 352
  supplied by PSSP 352
introduction 1
IP address changes for SP systems 575
isolating nodes 306

J

jm_config command 91
jm_start command 94
jm_status 98
jm_stop 97
jm_verify command 90
jmd_config file 502
job management 9
joining a Group Services group
  Event Management subsystem 363

K

Kerberos 154
  client and server interaction 154
  DES 156
  password, master 173
  principals 156
  tickets 158
key attributes
  Event Management SDR objects 357
keys
  changing 181
  creating 180
  displaying information for 180
  distributing to client systems 182
  managing 179

L

large data transfers, tuning for 423
last file 48
Launch Pad
  bringing up 267
  customizing 268
  description of 268
  elements of the 268
line, from the command 395
line, using the command 239, 241, 243, 246, 247, 249, 250, 251
list file 54
list file keywords and operands 64

list node file system information 548
list node hardware information 548
list node network information 548
listing a user account 75
listing resource variable information
  from SP Perspectives 538
  using load list files 535
  using the SDRGetObjects command 535
load command 194
LoadLeveler
  resource monitor 352
loadsdr program
  use by haemloadcfg command 357
local EM clients
  Event Management communications 354
local GS clients
  Group Services communications 340
locate_jm 98
lock file
  Event Management 358
  Group Services 340
log file
  Event Management 358
  Group Services 340
log_event 375
login control 77
logs file 53
lowclust, tuning 414
lowmbuf, tuning 415
lpd daemon 111
lscfg command 548
lssrc command
  getting Event Management status 365
  getting Group Services status 346

M

magnetic tape extension (MTE) pseudo device driver 434
  installing 436
  site-defined exit facility 436
  tape management pseudo control point (TMPCP) 434
mail 119
mail services 119
  HACWS and 331
mail.aliases map 124
maintaining the crontabs file 147
Management Information Base
  SP 386
Management Protocol Interface (MPI) 443
management, configuration 9
management, user 9
managing
  file collection 41
  user accounts 67
managing error notification objects 395
managing mail 119
managing print 111
managing user information 67
manual pages for public code xxvi
mb_cl_hiwat, tuning 414
Membership resource monitor 353
message boxes of the SP Perspectives GUI 276
message trace output log
  Event Management 358
MIB (Management Information Base)
  understanding 382
mkamdent command 108
modifying the amd.u map 107
modifying the file collection hierarchy 61
monitoring and operating the SP system 281
monitoring and operating the system
  HACWS and 327
MPI 443
MTE pseudo device driver 434
  installing 436
  site-defined exit facility 436
  tape management pseudo control point (TMPCP) 434

N

Nagle algorithm 418
nameserver, Group Services
  establishing 344
netstat command 548
Netview for AIX
  configuring for the SP system 389
network boot 304
network connections for HACWS 320
network connectivity adapters 8
Network File Sustem (NFS)
  tuning 424
Network File System (NFS) 105
network tuning options
  changing on control workstation 408
  changing on selected nodes 409
  changing on the node 408
new features of PSSP
NFS
  See also Network File System (NFS)
  mounts over a switch 228
node groups
  commands for creating, modifying, and deleting 259
  commands for examining 259
  commands that take node groups as arguments 261
  creating 258
  creating, modifying, and deleting 259
  definition list 257
  deleting 260

node groups *(continued)*
examining 259, 260
example of resolving 258
flattening 258
global 257
managing 257
modes 257
names 257
partition-bound 257
resolving 258
SDR information 257
SMIT panels for creating, modifying, and deleting 259
SMIT panels for deleting 260
SMIT panels for examining 260
understanding 257
node hardware variables 517, 519, 521, 524
node isolation 223, 306
node jobs, accounting for exclusive use of 134
node.root file collection 45
nodes
Group Services
restrictions on number per domain 345
monitoring on a switch 227
overview of sequencing 19
nodes, 604 High 6
nodes, primary 215
nodes, primary backup 215
nodes, processor 6
nodes, thin 6
nodes, wide 6
notebooks of the SP Perspectives GUI 274
notifcation object, installing a 396
notification objects, managing error 395
notify_event 375
notifying users of impending shutdown 24
nrunacct command files 143
NTP 101
HACWS and 330
NTP options 101

O

objects, managing error notification 395
observation interval
changing 355
omit command 64
omitany command 64
operating frame environment controls 297
operating the front panel controls 297
operating the global controls 304
operation 9
ordering additional copies of this book xxvi
overview of RS/6000 Scalable POWERparallel Systems
overview, hardware 5
overview, software 8

P

p_cat 35
packaging 10
parallel management commands 31
HACWS and 330
sample 34
Parallel System Support Programs (PSSP)
new features of 1
Parallel System Support Programs for AIX (PSSP), IBM 9
partitioning, system
aid for creating configurations 2
and commands 230
applying a configuration 248
configuration 2, 239, 241
aid for 2
displaying information for 241
selecting 239
HACWS and 327
layouts 243
selecting 243
overview 235
steps for 238
understanding 229
partitions, system
changed 249
rebooting 249
changing 246
customizing 244
default system partition 234
displaying configuration information 249
managing system partition-sensitive subsystems 252
verifying configuration information 251
partititons, system
managing 229
passwd command 75
passwork, Kerberos master 173
PCA card, tuning 428
pcp 35
pdf 37
peer group, Event Management
joining 363
Performance Aide for AIX
dependency by Event Management 359
performance monitoring
shared variables with Event Management 359
Performance Toolbox for AIX
dependency by Event Management 359
use of shared resource variables 359
Performance Toolbox Parallel Extensions (PTPE)
use of shared resource variables 359

Performance Toolbox Parallel Extensions for AIX (PTPE) 3
perspective
customizing a 272
description of 270
elements of 270
perspectives 267
pexec 34
pexscr 35
pfck 38
pfind 35
pfps 37
physical tape device driver 442
pls 35
pman 369
pmand
controlling 371
understanding 370
pmandef
issuing an SNMP trap for an event 375
logging an event 375
purpose 372
running a command for an event 373
security considerations 372
pmandefaults 376
pmanquery 376
pmanrmd
configuring 378
controlling 377
SDR information 379
understanding 376
pmanrmd.conf 377, 378
pmanrmdloadSDR 379
pmanrminput 377
pmv 35
port numbers
component of Event Management 353
component of Group Services 339
power_system file collection 44
powering on the SP
safety warning 20
pps 35
predicate
definition 349
prefix file 54
primary authentication server 170
primary node takeover 212
primary windows of the SP Perspectives GUI 269
principals
adding 165
adding and assigning passwords 175
deleting 179
displaying information 174
managing 174
service 157
types 156
principals *(continued)*
user 157
print 111
print management 10
HACWS and 331
print management command support 112, 113
print management programs
pmbec 116
pmfe 116
pmswitch 116
print management security modes 111
print services 111
printer subsystem requirements 114
prm 35
problem determination
Event Management subsystem
abnormal termination core file 358
abnormal termination of haemctrl add 362
getting subsystem status 365
tracing 362
Group Services subsystem
abnormal termination core file 341
abnormal termination of hagsctrl add 343
getting subsystem status 346
tracing 343
problem management 10
resource monitor 352
Problem Management subsystem
and system partitioning 253
components of 370
creating resource monitors 377
creating subscriptions 372
default events 376
environment variables 374
IBM.PSSP.pm.User_state resource variables 377
log_event 375
logging an event 375
notify_event 375
pmand 370
pmandef 372
pmandefaults 376
pmanquery 376
pmanrmd 376
pmanrmd.conf 377
pmanrmdloadSDR 379
pmanrminput 377
requesting information about a subscription 376
resource monitor daemon 376
resource variables and values 377
running a command for an event 373
SNMP trap, issuing 375
understanding 369
process resource monitor 352
processor nodes 6
protocol, Group Services
definition 337

protocol, switch communication 211
provider
  definition 337
pseudo device driver 434
  installing 436
PSSP 9, 11

Q

querying resource manager 98
quiescing the SP Switch 227

R

rcp 165
realms 158
reboot
  and Event Management 361
  and Group Services 343
rebooting the SP 23
reconfiguring
  Event Management subsystem 367
reconfiguring resource manager 96
recovery from failure
  Event Management 364
  Group Services 345
refuse file 61
refusing files in a file collection 61
registration cache
  and Event Manager daemon 363
  Event Management 358
remote EM clients
  Event Management communications 354
removing a file collection 63
repartitioning, SP Switch 226
report and summary files, accounting 143
reporting file collection information 49
required disk space on Print Host(s) 114
required file collections 43
resource class
  configuration in SDR 355
resource data, listing
  using the SDRGetObjects command 535
resource identifier
  synonymous with instance vector 538
resource manager 85
  configuring 88
  HACWS and 330
  installing 86
  querying 98
  reconfiguring 96
  recovery 99
  starting 90
  stopping 97
  understanding 85
resource monitor
  component of Event Management 352
  configuration in SDR 355
  connection types 352
  current working directory 358
  definition 349
  Event Management communications 354
Resource Monitor Application Programming Interface (RMAPI)
  component of Event Management 351
  use of EMCDB version string directory 359
  use of shared memory key 358
resource monitor, non-PSSP
  loading configuration data for 355, 357, 367
resource monitor, third-party
  loading configuration data for 355, 357, 367
resource monitors
  creating 377
resource monitors supplied by PSSP
  aixos (internal) 353
  IBM.PSSP.harmld 352
  IBM.PSSP.harmpd 352
  IBM.PSSP.hmrmd 352
  IBM.PSSP.pmanrmd 352
  Membership (internal) 353
  Response (internal) 353
resource variable
  configuration in SDR 354
  definition 349
  IBM.PSSP.Membership.LANAdapter.state 360
  IBM.PSSP.Membership.Node.state 360
  IBM.PSSP.Response.Host.state 360
  IBM.PSSP.Response.Switch.state 360
resource variable information, displaying
  using the spevent command 538
resource variable, PSSP-supplied
  listing 535
Response resource monitor 353
restarting the SP 20, 23
restrictions
  Event Management 364
    number of EM clients 364
    observation of resource variable values 365
  Group Services 345
    groups per client 345
    groups per domain 345
    nodes per domain 345
    number of GS clients 345
rfc1323, tuning 415
RMAPI (Resource Monitor Application Programming Interface)
  component of Event Management 351
  use of EMCDB version string directory 359
  use of shared memory key 358
RMAPI libraries
  location 351

router nodes, tuning 424
RS/6000 Scalable POWERparallel Systems
  characteristics of 1
  overview 5
rsh 165
rsh privileges 111
run-time EMCDB
  creating 356, 364

S

sample files 475
  script.cust 505
  tuning.cust 510
sb_max, tuning 414
SBS (structured byte string)
  configuration in SDR 354
scheduling administrative tasks 407
script.cust
  sample file 505
SDR 541, 552
  archiving 238
  pmanrmd information 379
  shadow files 551
SDR (System Data Repository)
  and haemctrl clean 362
  and haemctrl delete 362
  and hagsctrl clean 343
  and hagsctrl delete 343
  dependency by Event Management 360
  dependency by Group Services 341
  key attributes for Event Management objects 357
  use by Event Management 349
SDR authentication 543
SDR authorization 543
SDR daemon log 544
SDR server 543
SDRCreateObjects command
  differences from haemloadcfg command 357
SDRDeleteObjects command
  changing Event Management configuration 367
SDRGetObjects command
  example of output 537
  listing resource variable information 535
secondary authentication server 171
secondary servers 167
security 10, 151
selecting nodes 215
  primary 215
  primary backup 215
selecting primary and primary backup nodes 215
sendmail daemon 120, 123, 124
sendmail.cf file 119
SEPBU unique frame hardware variables 515
sequencing nodes
  about 19
sequencing nodes *(continued)*
  default 20, 27
  defining 26
sequencing of nodes
  overview 19
server
  type of resource monitor 352
service collections
  creating 403
  gathering 404
  removing 404
service, collecting system data for IBM 402
set command 194
setting time 103
setting up a primary authentication server 170
setting up a secondary authentication server 171
setting up SP accounting 135
shadow files 551
shared memory key
  use by RMAPI 358
shutting down and rebooting the SP 23
shutting down the SP 23
  checking the status of 25
  notifying users before 24
  overview 19
  phases of shutdown processing 24
single point of control. 31
site-defined exit facility 436
SMIT
  See System Management Interface Tool (SMIT)
SMIT interface 70
SMIT interface access 544
SMIT, from 394, 395, 397
SMIT, using 236, 238, 240, 241, 243, 244, 247, 248, 249, 251
SNMP
  contents of traps 385
  traps from AIX Error Log events 383
  traps from Event Management events 384
  understanding 382
SNMP Support subsystem
  and system partitioning 253
SNMP traps
  pmandef to specify issuing 375
snmp_trap_gen 383
sockets
  component of Event Management 353
  component of Group Services 339
  names used by Event Management 354
software overview 8
SP
  overview of starting up and shutting down 19
  powering on 20
  restarting 20
  starting 20

SP Ethernet
  tuning 410
SP hardware state
  resource monitor 352
SP Perspectives
  example of displaying resource data 539
  listing resource variable information 538
  terminology differences from APIs 538
SP Perspectives GUI
  administrative tasks 278
  applications 267
  authorizing users for 266
  customizing primary windows 272
  dialog boxes 273
  getting started 265
  graphical elements 267
  help for 277
  Launch Pad 268
  message boxes 276
  notebooks 274
  perspectives 269
  primary windows 269
  starting 267
  tasks that can be performed with 265
  understanding the interface 267
  Using 265
SP switch
  HACWS and 329
SP Switch repartitioning 226
SP Switch variables 529
SP system
  tuning 407, 410
SP System Monitor GUI 286
SP user management options 68
sp_configd 381, 383
SP_NAME environment variable
  and haemcfg command 355
  and haemctrl command 356
  and haemloadcfg command 357
  and hagsctrl command 340
SP_ports SDR class
  use by Event Management 354
spchuser command 67, 69, 72
  home attribute 69
  SMIT interface 70
special subsystems
  creating 29
spevent command
  example of output 539
  listing resource variable information 538
spimg 13
splsuser command 67, 75
  SMIT interface 70
SPMI (System Performance Monitor Interface)
  dependency by Event Management 359
SPMI library
  shared memory key 358
spmkuser command 67, 69, 70
  home attribute 69
  SMIT interface 70
sprmuser command 67, 73
  SMIT interface 70
SRC (System Resource Controller)
  and Event Manager daemon 363
  and Group Services daemon 344
  dependency by Group Services 341
srcmstr daemon 131
ssp.authent 12
ssp.basic 11
ssp.clients 12
ssp.clients fileset, PSSP
  and Event Management 351, 361
  and Group Services 342
ssp.csd 13
ssp.css 12
ssp.docs 13
ssp.gui 11
ssp.ha 12
ssp.ha fileset, PSSP
  and Event Management 361
  and Group Services 342
ssp.hacws 14
ssp.jm 11
ssp.mte 14
ssp.perlpkg 13
ssp.pman 12
ssp.public 13
ssp.st 11
ssp.sysctl 12
ssp.sysman 11
ssp.top 12
staging file for EMCDB
starting resource manager 90
starting subsystems
  Event Management (haemctrl) 362
  Group Services (hagsctrl) 343
starting the display manager interface 287
starting the SP 20
  overview 19
starting the switch 221
starting up the SP
  checking the status of 25
status, Event Management
  output of lssrc command 365—367
status, Group Services
  output of lssrc command 346—347
stopping resource manager 97
stopping subsystems
  Event Management (haemctrl) 362
  Group Services (hagsctrl) 343

structured byte string (SBS)
  configuration in SDR 354
structured field
  configuration in SDR 354
subscriber
  definition 337
subsystem
  Event Management 349—368
  Group Services 337—347
  system partition-sensitive 340, 356
subsystem status
  for Event Management 365—367
  for Group Services 346—347
subsystems
  special 29
  tuning 423
subsystems, system partition-sensitive
  Emonitor subsystem 253
  Event Management Conditions subsystem 253
  Event Management subsystem 253
  general structure 253
  Group Services subsystem 252
  Heartbeat subsystem 252
  Host_Responds subsystem 253
  listing 252
  managing 252
  Problem Management subsystem 253
  SNMP Support subsystem 253
  summary 253
summary and report files, accounting 143
sup.admin file collection 44
supper activate command 46
supper install command 48
supper offline command 61
supper program 41
supper reset command 47
supper scan command 48
supper status command 43
supper subcommands table 49
supper update command 48, 110
support, installation 2
switch 211
  clocks 214
  setting switch clock source 214
    SMIT panels 214
  switch topology files 218, 219
switch clocks 214
switch connectivity 222
switch protocol 211
switch queue, tuning 420
switch topology files
  annotating 219
  managing 218
switch variables 527
switch_responds SDR class
  and Event Management 360
switch_responds SDR class *(continued)*
  resource monitor 353
switch, high performance 7
switch, starting 221
switch, system partitioning 213
switch, using 211
switches
  monitoring nodes on 227
  quiescing 227
  tuning 417
symlinkall command 65
synchronization, time 103
syntax rules 64
sysctl
  application development
    passing parameters to a script 204
    sending a script 202
    specifying server options 203
  architecture 187
  Authentication Services, relationship to 187
  authorization
    adding new callbacks 195
    components 188
    levels 188
    variables 201
  built-in commands 206
  commands
    built-in sysctl commands 206
    Extended Tcl command summary 205
    Tcl command summary 205
  components, illustration of 187
  configuration
    debugging problems 191
    procedure 190
  configuration commands
    create class 193
    create proc 193
    create var 193
    include 192
    load 194
    set 194
  description of 185
  executables
    sysctl 189
    sysctld 189
  Extended Tcl command summary 205
  files
    sysctl.conf 189
    sysctld.log 190
  HACWS and 327
  installation 190
  Tcl command summary 205
  using
    directing commands to remote hosts 197
    interactive mode 196
    non-interactive mode 197
    prerequisites 196

sysctl *(continued)*
using *(continued)*
sending commands to a collection of nodes 198
sending commands to a server 197
Sysctl-based 36
Syspar Controller
managing system partition-sensitive subsystems 252
Syspar SDR class 355
syspar_ctrl command
and Event Management subsystem 356
and Group Services subsystem 340
listing system partition-sensitive subsystems 252
managing system partition-sensitive subsystems 252
Syspar_ports SDR class
use by Event Management 353
use by Group Services 339
System Data Repository 541, 552
default sequencing based on 20
System Data Repository (SDR)
and haemctrl clean 362
and haemctrl delete 362
and hagsctrl clean 343
and hagsctrl delete 343
archiving 238
configuration data for Event Management 354
dependency by Event Management 360
dependency by Group Services 341
key attributes for Event Management objects 357
use by Event Management 349
System Management Interface Tool (SMIT)
system partition
and Event Management 350
and Event Management SDR classes 355
and Group Services 338
specifying for haemcfg command 355
specifying for haemctrl command 356
specifying for haemloadcfg command 357
specifying for hagsctrl command 340
system partitioning aid 2
system partitioning and the switch 213
System Performance Monitor Interface (SPMI)
dependency by Event Management 359
System Resource Controller (SRC)
and Event Manager daemon 363
and Group Services daemon 344
dependency by Group Services 341
system resources
listing PSSP-supplied resource variables 535

T

takeover, primary node 212
tape device attributes, changing 453, 458
tape management control point (TMCP) 446
tape management enhancement 442
Management Protocol Interface (MPI) 443
tape management control point (TMCP) 446
User Protocol Interface (UPI) 445
tape management pseudo control point (TMPCP) 434
tar files, reading 108
TCP port
use by Event Management 353
TCP socket
Event Management client communication 351
tcp_mssdflt, tuning 415
tcp_recvspace, tuning 415
tcp_sendspace, tuning 415
terminology differences
between APIs and SP Perspectives 538
thewall, tuning 414
thin nodes 6
third-party resource monitor
loading configuration data for 355, 357, 367
tickets
cache files 159
lifetimes 159
understanding 158
time limits
Event Management
connection to Group Services 363
observation intervals 365
peer group joining 363
reconnection to resource monitors 364
Group Services
connection to Topology Services 344
time setting 103
time synchronization 103
TMCP 446
TMPCP 434
Topology Services subsystem
and Group Services daemon initialization 344
and system partitioning 252
dependency by Group Services 341
trace output log
Event Management 358
Group Services 340
tracing subsystems
Event Management (haemctrl) 362
Group Services (hagsctrl) 343
trademarks xv
trimming log files 399
troubleshooting
Event Management subsystem
abnormal termination core file 358
abnormal termination of haemctrl add 362
getting subsystem status 365
tracing 362
Group Services subsystem
abnormal termination core file 341
abnormal termination of hagsctrl add 343

troubleshooting *(continued)*
Group Services subsystem *(continued)*
getting subsystem status 346
iniitalization errors 345
tracing 343
tuning
ADSM 425
AIX 3.2.5 considerations 411
CLIO/S 428
consolidated systems 423
ESCON 427
external server considerations 423
general recommendations 407
large data transfers 423
Network File System (NFS) 424
network options 408
PCA card 428
problems 418
router nodes 424
single parent–multiple child applications 421
SP Ethernet 410
SP system-specific 410
specific workloads 416
subsystems 423
switch queue lengths 420
switches 417
system topology considerations 409
the SP system 407
user environments 416
tuning.cust
sample file 510

U

UDP port
use by Event Management 353
use by Group Services 339
udp_recvspace, tuning 415
udp_sendspace, tuning 415
understanding file collections 41
understanding resource manager 85
understanding the file collection structure 45
directory and master files 45
understanding the NFS and Amd implementation 105
understanding the system data repository 541
unfencing nodes 223
Unix domain socket
Event Management client communication 351
Group Services client communication 338
resource monitor communication 351
use by Event Management 353
use by Group Services 339
UPI 445
user accounts
adding 70
changing 72
user accounts *(continued)*
deleting 73
HACWS and 330
list 75
manage 67
user environments, tuning 416
user management 9
User Protocol Interface (UPI) 445
user.admin file collection 44
using a switch 211
using Amd for other file systems 108
using SMIT 238, 240, 241, 243, 244, 247, 248, 249, 251
using the command line 239, 241, 243, 246, 247, 249, 250, 251
using the display manager interface 286

V

verifying file collections using scan 51
version string for EMCDB 355
VSD subsystem
resource monitor 352

W

when file 48
wide nodes 6
window elements 287
workstation outside SP
and Event Management 351
workstation, control 7

Y

yppasswd command 75

Z

zcat command 108

Communicating Your Comments to IBM

IBM Parallel System Support
Programs for AIX
Administration Guide

Publication No. GC23-3897-02

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply.

Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. However, the comments you send should pertain to only the information in this manual and the way in which the information is presented. To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

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

If you are mailing an RCF from a country other than the United States, you can give the RCF to the local IBM branch office or IBM representative for postage-paid mailing.

If you prefer to send comments by mail, use the RCF at the back of this book.

If you prefer to send comments by FAX, use this number:

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

If you prefer to send comments electronically, use this network ID:

- IBMLink: (United States customers only): KGNVMC(MHVRCS)
- IBM Mail Exchange: USIB6TC9 at IBMMAIL
- Internet e-mail: mhvrcfs@vnet.ibm.com
- World Wide Web: http://www.s390.ibm.com/os390

Make sure to include the following in your note:

Title and publication number of this book
Page number or topic to which your comment applies

Optionally, if you include your telephone number, we will be able to respond to your comments by phone.

Reader's Comments — We'd Like to Hear from You

IBM Parallel System Support
Programs for AIX
Administration Guide

Publication No. GC23-3897-02

You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

Note: Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

Today's date: _____

What is your occupation?

Newsletter number of latest Technical Newsletter (if any) concerning this publication:

How did you use this publication?

[ ]	As an introduction	[ ]	As a text (student)
[ ]	As a reference manual	[ ]	As a text (instructor)
[ ]	For another purpose (explain)		

Is there anything you especially like or dislike about the organization, presentation, or writing in this manual? Helpful comments include general usefulness of the book; possible additions, deletions, and clarifications; specific errors and omissions.

Page Number: Comment:

Name

Address

Company or Organization

Phone No.

Reader's Comments — We'd Like to Hear from You
GC23-3897-02

IBM®

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

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

Fold and Tape | **Please do not staple** | Fold and Tape

Cut or Fold
Along Line

IBM®

Program Number: 5765-529

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

Drop in
Back Cover
Image Here.

GC23-3897- 2