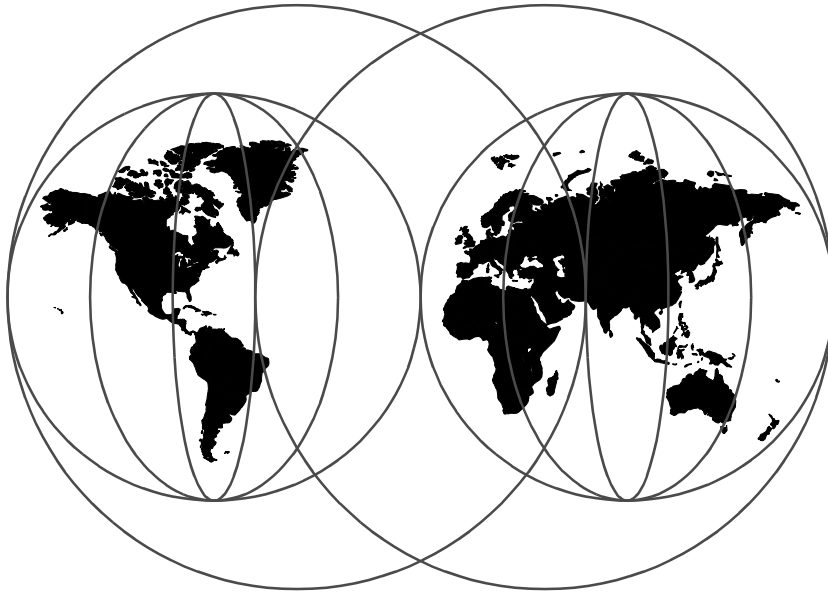


## **PSSP 3.1 Announcement**

*Marcelo R. Barrios, Rami Alfalahi, Jean-Michel Berail, Robin Findlay  
Alan Foster, Theeraphong Thitayanun, Hasan Hakan Yardim*



**International Technical Support Organization**

<http://www.redbooks.ibm.com>

SG24-5332-00





International Technical Support Organization

**PSSP 3.1 Announcement**

November 1998

**Take Note!**

Before using this information and the product it supports, be sure to read the general information in Appendix C, "Special Notices" on page 325.

First Edition (November 1998)

This edition applies to PSSP Version 3, Release 1 (5765-D51) for use with the AIX Version 4, Release 3 Modification 2 Operating System.

**Note**

This book is based on a pre-GA version of a product and may not apply when the product becomes generally available. We recommend that you consult the product documentation or follow-on versions of this redbook for more current information.

Comments may be addressed to:  
IBM Corporation, International Technical Support Organization  
Dept. HYJ Mail Station P099  
522 South Road  
Poughkeepsie, New York 12601-5400

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

© Copyright International Business Machines Corporation 1998. 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

<b>Contents</b> .....	iii
<b>Figures</b> .....	.xi
<b>Tables</b> .....	xv
<b>Preface</b> .....	xvii
The Team That Wrote This Redbook .....	xviii
Comments Welcome .....	xx
<b>Chapter 1. Announcement Overview</b> .....	1
1.1 New in PSSP .....	2
1.1.1 New Hardware Support .....	3
1.1.2 Alternate and Mirroring rootvg Volume Group Support .....	5
1.1.3 External SSA/SCSI Boot .....	6
1.1.4 Improved Network Adapter Support .....	6
1.1.5 Switch Improvements .....	7
1.1.6 New Packaging .....	7
1.1.7 TME 10 Integration .....	8
1.1.8 SP Perspectives .....	11
1.1.9 SP Resource Center .....	12
1.1.10 Software Requirements .....	14
1.1.11 Cluster Technology (RSCT) .....	14
1.1.12 Security .....	15
1.2 New Product Releases .....	16
1.2.1 HACMP/ES 4.3.0 .....	16
1.2.2 GPFS 1.2 .....	16
1.2.3 LoadLeveler 2.1 .....	16
1.2.4 Parallel Environment 2.4 .....	16
<b>Chapter 2. Installation Management Enhancements</b> .....	17
2.1 Installation Process .....	17
2.1.1 An RS/6000 as a Control Workstation .....	18
2.1.2 Main Components of the Installation Process .....	25
2.1.3 Installation .....	29
2.1.4 New in PSSP 3.1: More Details .....	33
2.1.5 New in PSSP 3.1: The Commands .....	44
2.2 Migration .....	52
2.2.1 Definitions - Overview - Limitations .....	53
2.2.2 Reasons .....	55
2.2.3 Planning .....	55

2.2.4	CWS Migration . . . . .	56
2.2.5	Node Migration . . . . .	59
2.3	Coexistence . . . . .	62
2.3.1	Definition . . . . .	62
2.3.2	Limitations . . . . .	63
2.4	New Features . . . . .	63
2.4.1	Multiple rootvg Support . . . . .	63
2.4.2	Booting from External Disks . . . . .	65
<b>Chapter 3. SP Perspectives . . . . .</b>		<b>71</b>
3.1	Overview . . . . .	71
3.1.1	New Features . . . . .	71
3.1.2	SPMON Equivalence . . . . .	73
3.1.3	New Filesets . . . . .	73
3.2	Launch Pad . . . . .	74
3.2.1	Profiles . . . . .	75
3.2.2	Adding More Applications to the Launch Pad . . . . .	76
3.2.3	Predefined Icons . . . . .	78
3.3	The Hardware Perspective . . . . .	79
3.3.1	Controlling Hardware . . . . .	84
3.3.2	Monitoring Hardware . . . . .	88
3.3.3	Filtering by Monitored State . . . . .	93
3.3.4	Viewing LED Values . . . . .	95
3.4	Event Perspectives . . . . .	95
3.4.1	Integration with RSCT . . . . .	98
3.4.2	Activating a Preset Monitored Event . . . . .	98
3.4.3	Modifying a Monitored Event . . . . .	103
3.4.4	Creating A New Condition and Event Definition . . . . .	106
3.4.5	Tivoli Integration . . . . .	113
3.5	Recoverable Virtual Shared Disk Perspectives . . . . .	113
3.5.1	Enhancements . . . . .	113
3.5.2	Configuration and Control . . . . .	114
3.5.3	Monitoring . . . . .	121
<b>Chapter 4. SP-Attached Server Support . . . . .</b>		<b>125</b>
4.1	Hardware Attachment . . . . .	125
4.1.1	SP-Attached Server . . . . .	126
4.1.2	SP-Attached Server Attachment . . . . .	127
4.2	Installation and Configuration . . . . .	136
4.2.1	Pre-Installation Checklist . . . . .	141
4.3	Software Changes . . . . .	142
4.3.1	SDR Changes . . . . .	142
4.3.2	Hardmon . . . . .	146

4.4	Changes in the User Interface	151
4.4.1	Perspectives	151
4.5	Attachment Scenarios	156
<b>Chapter 5. Switch Support Enhancements</b>		<b>161</b>
5.1	Automatic Node Unfence	161
5.1.1	Implementation Overview	161
5.1.2	Example Scenarios	162
5.1.3	Coexistence Consideration	163
5.2	Startup of Switch-Dependent Applications at Boot Time	164
5.2.1	Implementation Overview	165
5.2.2	Coexistence Consideration	166
5.3	Switch Admin Daemon	167
5.3.1	Implementation Overview	167
5.3.2	Example Scenarios	170
5.3.3	Coexistence Consideration	172
5.4	Centralized Error Logging	173
5.4.1	Implementation Overview	173
5.4.2	Example Scenario	179
5.4.3	Coexistence Consideration	181
<b>Chapter 6. RS/6000 Cluster Technology</b>		<b>183</b>
6.1	RSCT Packaging	185
6.1.1	Coexistence of rsct and ssp.ha/ssp.topsvcs on a Node	185
6.1.2	Directory Structure of RSCT	186
6.1.3	Prerequisites	186
6.2	Topology Services	186
6.2.1	Heartbeat on Additional Networks	189
6.2.2	Dynamic Update (Refresh)	190
6.2.3	New SDR Classes	191
6.2.4	Topology Services in HACMP Domain	191
6.2.5	Node/Adapter Numbering	192
6.2.6	Large Systems Improvements	193
6.3	Group Services	193
6.3.1	New GS Protocols	196
6.3.2	RSCT Enhancements to Support HACMP/ES in GS	199
6.4	Event Management	199
6.4.1	Shared Memory Segment	201
6.4.2	Resource Monitors	202
6.4.3	Event Management Enhancements to Support HACMP/ES	207
6.4.4	Event Management Configuration Database	210
6.4.5	EMAPI and RMAPI Changes	211
6.4.6	New Command - haemqvar	213

6.4.7	Event Registration Acknowledgment . . . . .	216
6.4.8	New SDR Classes and Attributes . . . . .	216
6.4.9	What is New in EM Security? . . . . .	216
<b>Chapter 7. Recoverable/Virtual Shared Disk 3.1 . . . . .</b>		<b>219</b>
7.1	R/VSD Concepts . . . . .	219
7.1.1	VSD Overview . . . . .	219
7.1.2	RVSD Overview . . . . .	219
7.2	R/VSD 3.1 Enhancements . . . . .	220
7.2.1	Packaging Changes . . . . .	220
7.2.2	Dynamic Node and Device Changes . . . . .	221
7.2.3	Separate File System for VSD Configuration and Log Files . . . . .	222
7.3	Migration and Coexistence Considerations . . . . .	222
7.3.1	The rvsdrestrict Command . . . . .	223
7.3.2	PTFs for Coexistence . . . . .	225
7.4	Recommendation . . . . .	225
<b>Chapter 8. HACMP/ES 4.3.0 . . . . .</b>		<b>227</b>
8.1	Overview . . . . .	227
8.1.1	Terminology . . . . .	227
8.1.2	Additional Support . . . . .	228
8.1.3	Software Enhancements . . . . .	231
8.2	HACMP ES Release 2 LPP . . . . .	231
8.2.1	Packaging . . . . .	231
8.2.2	Dependencies . . . . .	231
8.2.3	Changes and Restrictions . . . . .	232
8.3	Migration and Coexistence . . . . .	232
8.3.1	Migration from HACMP/6000 . . . . .	232
8.3.2	Migration from HACMP/ES 4.2.1 and 4.2.2 . . . . .	233
8.4	New Functionality . . . . .	234
8.4.1	Global Network Support . . . . .	234
8.4.2	Dynamic Reconfiguration Event (DARE) . . . . .	237
8.4.3	Security . . . . .	238
8.4.4	Global ODM . . . . .	239
8.4.5	Concurrent Access . . . . .	239
8.4.6	Supported Networks . . . . .	241
8.4.7	Tunable Heartbeat . . . . .	241
8.5	New Commands . . . . .	242
8.5.1	clhandle . . . . .	242
8.5.2	cldomain . . . . .	242
8.5.3	clmixver . . . . .	243
8.5.4	claddnetwork . . . . .	243



<b>Chapter 9. GPFS 1.2</b> . . . . .	245
9.1 Why GPFS? . . . . .	245
9.2 GPFS Overview . . . . .	245
9.2.1 Implementation Overview . . . . .	245
9.2.2 GPFS Components . . . . .	247
9.3 Hardware and Software Requirements . . . . .	250
9.3.1 Hardware Requirements . . . . .	250
9.3.2 Software Requirements . . . . .	251
9.4 GPFS 1.2 Enhancements . . . . .	251
9.4.1 Scalability Enhancements . . . . .	251
9.4.2 Usability and System Management Enhancements . . . . .	253
9.4.3 Performance Enhancements . . . . .	255
9.5 Migration Considerations . . . . .	256
9.5.1 GPFS Configuration For Migration . . . . .	256
9.5.2 Migration Approach . . . . .	256
9.5.3 A Full Migration . . . . .	257
9.5.4 A Staged Migration . . . . .	257
9.6 Coexistence Considerations . . . . .	257
9.6.1 Within a Partition . . . . .	257
9.6.2 Multiple Partitions . . . . .	257
9.7 Compatibility . . . . .	258
<b>Chapter 10. LoadLeveler Version 2.1</b> . . . . .	259
10.1 LoadLeveler Introductory Concepts . . . . .	259
10.1.1 LoadLeveler: A Breakdown of How It Works . . . . .	261
10.1.2 LoadLeveler Daemons . . . . .	263
10.1.3 Checkpointing . . . . .	265
10.1.4 Scheduling . . . . .	266
10.1.5 Parallel Jobs . . . . .	266
10.2 LoadLeveler Jobs . . . . .	267
10.2.1 Writing a Job Command File . . . . .	268
10.2.2 Submitting a Job Command File . . . . .	271
10.2.3 Managing a Job . . . . .	272
10.3 Installing and Configuring LoadLeveler . . . . .	273
10.3.1 Installation . . . . .	274
10.3.2 LoadLeveler Administration File . . . . .	277
10.3.3 LoadLeveler Configuration File . . . . .	280
10.4 Controlling LoadLeveler . . . . .	282
10.4.1 Using the LoadLeveler GUI . . . . .	282
10.4.2 Submitting a Job . . . . .	286
10.4.3 Building a New Job . . . . .	287
10.5 New Features in LoadLeveler Version 2.1 . . . . .	289
10.5.1 Enhanced Parallel Environment Support in LoadLeveler . . . . .	289

10.5.2	Integration of Resource Manager Functions . . . . .	290
10.5.3	Changes and Enhancements to Checkpointing . . . . .	291
10.5.4	New Scheduling Algorithm . . . . .	293
10.5.5	Migration from Version 1.3 of LoadLeveler . . . . .	293
10.5.6	Interactive Session Support (ISS) . . . . .	293
<b>Chapter 11.</b>	<b>Parallel Environment 2.4 . . . . .</b>	<b>295</b>
11.1	Increased Tasks Limits Per Job . . . . .	296
11.2	Multiple User Space Tasks Per Node . . . . .	296
11.3	POE and Job Management . . . . .	301
11.3.1	Differences Between LoadLeveler and Resource Manager . . . . .	302
11.4	User-Initiated Parallel Checkpoint/Restart . . . . .	303
11.4.1	Limitations . . . . .	303
11.4.2	How Checkpointing Works . . . . .	304
11.5	MPI Thread Compatibility . . . . .	305
11.5.1	Responder Threads . . . . .	307
11.5.2	Threaded MPI Library Compatibility . . . . .	308
11.5.3	AIX Thread Structure . . . . .	309
11.6	New Environment Variables . . . . .	310
11.7	MPI I/O Subset . . . . .	312
11.8	MUSPPA-lite . . . . .	315
11.9	Xprofiler Enhancements . . . . .	316
11.10	Message Queue Debugging . . . . .	318
<b>Appendix A.</b>	<b>Changes to the SDR . . . . .</b>	<b>319</b>
<b>Appendix B.</b>	<b>New Commands and Changes to Old Commands . . . . .</b>	<b>323</b>
B.1	New Commands in PSSP 3.1 . . . . .	323
B.2	Changes to Old Commands in PSSP 3.1 . . . . .	324
<b>Appendix C.</b>	<b>Special Notices . . . . .</b>	<b>325</b>
<b>Appendix D.</b>	<b>Related Publications . . . . .</b>	<b>329</b>
D.1	International Technical Support Organization Publications . . . . .	329
D.2	Redbooks on CD-ROMs . . . . .	329
D.3	Other Publications . . . . .	329
<b>How to Get ITSO Redbooks . . . . .</b>		<b>331</b>
How IBM Employees Can Get ITSO Redbooks . . . . .		331
How Customers Can Get ITSO Redbooks . . . . .		332
IBM Redbook Order Form . . . . .		333

<b>List of Abbreviations</b> . . . . .	335
<b>Index</b> . . . . .	337
<b>ITSO Redbook Evaluation</b> . . . . .	343

**x** PSSP 3.1 Announcement

---

## Figures

1. Resource Center Welcome Screen . . . . .	12
2. Sample Section of the Resource Center Index Frame . . . . .	13
3. NIM Objects . . . . .	18
4. /spdata Directory Structure . . . . .	22
5. SMIT Panel to Enter Data into the SDR. . . . .	24
6. Installation Process . . . . .	25
7. SMIT Panel Related to Node Information . . . . .	26
8. NIM Basic Uses . . . . .	27
9. Perspectives Window to Network Boot a Node . . . . .	31
10. Hardware Perspective to Monitor a Node . . . . .	33
11. PSSP Filesets . . . . .	34
12. RSCT Filesets . . . . .	34
13. Additional Filesets Included in PSSP Software . . . . .	35
14. Remote Shell Structure Before PSSP 3.1 . . . . .	36
15. Remote Shell Structure in PSSP 3.1 or Later . . . . .	37
16. Partial List of Attributes from the Syspar SDR Class. . . . .	39
17. Main SMIT SP Security Panel . . . . .	39
18. SMIT Panel for Selecting Authorization Methods for Root Access . . . . .	40
19. SMIT Panel for Enabling Authentication Methods . . . . .	41
20. New SMIT Panel to Create a Volume Group. . . . .	44
21. New SMIT Panel to Modify a Volume Group. . . . .	46
22. New SMIT Panel to Delete a Volume Group . . . . .	47
23. New SMIT Panel to Issue the spbootins Command . . . . .	49
24. New SMIT Panel to Initiate the spmirrorvg Command. . . . .	50
25. New SMIT Panel to Initiate the spunmirrorvg Command. . . . .	51
26. Example of splstdata -v . . . . .	52
27. Migration Considerations for PSSP 3.1 . . . . .	53
28. SMIT Panel for the spbootlist Command. . . . .	64
29. Cabling SSA Disks to RS/6000 SP Nodes. . . . .	65
30. Connections on the SSA Disks . . . . .	66
31. SMIT Panel to Specify an External Disk for SP Node Installation . . . . .	68
32. Output of the splstdata -b Command. . . . .	69
33. bosinst.data File with the New CONNECTION Attribute . . . . .	70
34. Fly-Over Help Displayed for Power-On Icon . . . . .	73
35. SP Perspectives Launch Pad . . . . .	74
36. Save Preferences Dialog Box . . . . .	76
37. Adding a New Application to the Launch Pad . . . . .	77
38. New Icon Added to the Launch Pad . . . . .	78
39. Launch Pad Application Details View . . . . .	79
40. Hardware Perspective . . . . .	81

41. The Add Pane Dialog . . . . .	82
42. Hardware Perspective Showing All Pane Types . . . . .	83
43. Selecting Node Objects in a Pane . . . . .	85
44. Change Key Switch Nodes Dialog . . . . .	86
45. Notebook for Node 5 . . . . .	87
46. The Hardware Perspective View Menu . . . . .	88
47. Set Monitoring Dialog Box . . . . .	89
48. A Nodes Pane Showing Monitoring of hostResponds . . . . .	90
49. Monitoring Nodes for Three Important Conditions . . . . .	91
50. Monitoring Nodes in Table View . . . . .	92
51. Set Table Attributes Dialog Box . . . . .	92
52. Filter Nodes Dialog . . . . .	94
53. Hardware Perspective-Filtering by Monitored State . . . . .	95
54. Node LCD and LED Displays . . . . .	95
55. The /etc/sysctl.pman.acl File . . . . .	96
56. Icon Color Table for Event Definitions . . . . .	97
57. Initial Event Perspective Display . . . . .	98
58. Event Perspective with Condition Pane in Table View . . . . .	100
59. keyNotNormal Row Event Definition Registered . . . . .	102
60. keyNotNormal Event Triggered . . . . .	102
61. keyNotNormal Event Definition Notebook Actions Page . . . . .	104
62. Problem Management Error Log Entry keyNotNormal Trigger . . . . .	105
63. Problem Management Error Log Entry keyNotNormal Rearm . . . . .	106
64. Creating New Condition for SSA Power Cooling Event Monitor . . . . .	107
65. Event Definition for New SSA Power Cooling Event Monitor . . . . .	109
66. Event Definition Notebook Notification Page . . . . .	110
67. Notebook Actions Page for New SSA Power Cooling Event Monitor . . . . .	111
68. Warning Window Popup for New SSA Power Cooling Event Monitor . . . . .	112
69. Virtual Shared Disk Perspective Initial Window . . . . .	115
70. Virtual Shared Disk Perspective View Pull -Down . . . . .	116
71. Add Pane to Show Virtual Shared Disks . . . . .	117
72. Creating a Virtual Shared Disk . . . . .	118
73. Output of lsvg -l extvg . . . . .	118
74. Virtual Shared Disk Pane . . . . .	119
75. Configuring Virtual Shared Disks on Nodes . . . . .	119
76. Configured Virtual Shared Disks . . . . .	120
77. Virtual Shared Disk Perspective Monitoring hasInactiveIBMVSDs . . . . .	121
78. Filter to Show Related Objects . . . . .	123
79. State Changes to the Virtual Shared Disk Perspective Icon . . . . .	124
80. The S70 Components . . . . .	127
81. The S70 Attachment to the SP . . . . .	128
82. RS-232 Connections to the S70 . . . . .	129
83. Node Numbering . . . . .	130

84. S70 Switch Adapter Attachment Slot . . . . .	134
85. S70 Floor Placement . . . . .	135
86. Non-SP Frame Information . . . . .	137
87. Example of a Frame Class with an SP-Attached Server . . . . .	143
88. Entries of the Node Class for SP Nodes and SP-Attached Server . . . . .	144
89. Example of the Syspar_map Class with SP-Attached Server . . . . .	145
90. Example of the NodeControl Class with the SP-Attached Server . . . . .	145
91. The Relationship Between Node and NodeControl Class . . . . .	146
92. Hardmon Flow of Control . . . . .	148
93. S70 Daemon Internal Flow . . . . .	150
94. Example of Perspectives with SP-Attached Server . . . . .	152
95. The Output of the spon Command . . . . .	154
96. splstdata -n Output . . . . .	155
97. splstdata -f Output . . . . .	155
98. spgetdesc -u -a Output . . . . .	156
99. Scenario 1: SP-Attached Server and One SP Frame . . . . .	156
100.Scenario 2: SP-Attached Server to Two SP Frames . . . . .	157
101.Scenario 3: SP Frame and Multiple SP-Attached Servers . . . . .	158
102.Scenario 4: Non-Contiguous SP-Attached Server . . . . .	159
103.A Node with the Autojoin Bit On Automatically Joins the Switch Network	163
104.A Rebooted Node Automatically Joins the Switch Network . . . . .	163
105.Changes in the /usr/lpp/ssp/css/rc.switch Script in PSSP 3.1 . . . . .	165
106.A Portion of /etc/inittab in PSSP 3.1 . . . . .	166
107.Cases When a Node Went Down on the css0 Interface . . . . .	168
108.Cases When a Node Came Up on host_respond . . . . .	169
109.Example of the cssadm.debug File on the CWS . . . . .	170
110.Example of the cssadm Daemon Response . . . . .	171
111.Another Example of the cssadm Daemon Response . . . . .	172
112.Example of the summlog File on the CWS . . . . .	178
113.Example of the summlog.out File on the CWS . . . . .	179
114.The summlog File When a Primary Node Takeover Occurs . . . . .	180
115.RSCT Infrastructure . . . . .	184
116.TS and GS Interfaces . . . . .	187
117.Issrc -ls hats.sp4en0 Command . . . . .	188
118.TS Process Flow . . . . .	189
119.HACMPtopsvcs GODM Class . . . . .	190
120.Machines List File in HACMP Domain . . . . .	192
121.hats_node_number Utility . . . . .	193
122.Group Services Structure . . . . .	194
123.GS Functional Flow . . . . .	196
124.ha_gs_change_attribute Sample Protocol Execution . . . . .	197
125.ha_gs_goodbye Sample Protocol Execution . . . . .	198
126.EM Functional Design . . . . .	201

127.New Shared Memory Architecture . . . . .	202
128.SP Resource Monitors . . . . .	204
129.Issrc -ls haem.sp4en0 Command . . . . .	205
130.Issrc -ls haemaixos.sp4en0 Command . . . . .	206
131.haemaixos SRC Subsystem . . . . .	207
132.haem SRC Subsystem . . . . .	209
133.emsvcs SRC Subsystem. . . . .	210
134.haemqvar Command. . . . .	214
135.haemqvar Output. . . . .	215
136.The rvsdrestrict Command Usage. . . . .	224
137.ATM Adapters Using Classic IP . . . . .	229
138.ATM Adapters Configured for LAN Emulation. . . . .	230
139.Example of Subnets on an SP Private Ethernet . . . . .	235
140.SMIT Menu for Global Network Addition. . . . .	236
141.HACMPnetwork ODM with Global Network Defined. . . . .	237
142.HACMPnim Class Example. . . . .	242
143.Example LoadLeveler Configuration . . . . .	259
144.A LoadLeveler Job . . . . .	261
145.LoadLeveler Job Flow . . . . .	262
146.Job Command File Using Shell Command Statements . . . . .	270
147.Job Command File Using Dependencies . . . . .	271
148.Standard Listing of the llstatus Command. . . . .	273
149.Example LoadLeveler Administration File. . . . .	279
150.Specifying Machine Names in the Administration File . . . . .	280
151.Running Jobs at a Specific Time of Day . . . . .	281
152.LoadLeveler GUI Main Window. . . . .	283
153.LoadLeveler GUI Jobs Pane . . . . .	285
154.LoadLeveler GUI Machines Pane . . . . .	286
155.Submit a Job Dialog . . . . .	287
156.Choosing a Job Type for Building . . . . .	288
157.Build a Job Dialog . . . . .	289
158.Multiple User Space Tasks Per Node . . . . .	297
159.How the Adapter Handles These Communication Windows. . . . .	299
160.Getting the Switch Port Numbers from the SDR . . . . .	300
161.Thread Structure of a MPI-POE Task . . . . .	306
162.MPI Structure . . . . .	307
163.Structure of AIX Thread Library. . . . .	309
164.File Access Through MPI-IO . . . . .	314
165.Functional Flow of MPI-IO Implementation . . . . .	315



---

## Tables

1. Supported IBM LPPs Per Supported PSSP and AIX Release. . . . .	19
2. Mandatory PSSP File Set . . . . .	23
3. Available Values for the bootp_response Parameter. . . . .	28
4. Resources Allocated by NIM According to the bootp_response Value . . . . .	28
5. General Uses of script.cust and firstboot.cust . . . . .	30
6. Supported Migration Paths to PSSP 3.1 . . . . .	54
7. Performance Aide for AIX (PAIDE) File Sets. . . . .	57
8. Command to Issue to Stop the Daemons . . . . .	58
9. Possible AIX or PSSP Combinations in a Partition . . . . .	62
10. Supported Adapters for Nodes with Full SSA Boot. . . . .	66
11. Supported Adapters for Nodes with SCSI Boot . . . . .	67
12. Column Description for Event Definitions . . . . .	101
13. Column Description for keyNotNormal Event Condition . . . . .	103
14. Required ssp.css Level for Coexistence . . . . .	164
15. IBM.PSSP.CSSlog.errlog is a Structured Byte String with These Fields. . . . .	174
16. AIX Error Log Entries. . . . .	174
17. RSCT Install Images . . . . .	185
18. Dual Daemons in SP and HACMP Domain . . . . .	210
19. Changes to VSD Fileset Names . . . . .	221
20. Changes to VSD Perspective Fileset Names . . . . .	221
21. Changes to RVSD Fileset Names . . . . .	221
22. Which R/VSD Level is Supported in Which PSSP Level. . . . .	223
23. PTFs for Coexistence with RVSD 3.1 . . . . .	225
24. Network Support in HACMP . . . . .	241
25. Example LoadLeveler Directory Tree . . . . .	274
26. Location of Configuration and Administration Files . . . . .	275
27. Resource Manager Functions Now in LoadLeveler. . . . .	291
28. Adapter/CPU Default Settings . . . . .	311
29. Adapter/CPU Use Under LoadLeveler. . . . .	311
30. New Attributes in Adapter Class . . . . .	320
31. New Attributes in Frame Class . . . . .	320
32. New Attributes in SP Class . . . . .	320
33. New Attributes in Node Class . . . . .	320
34. New Attribute in EM_Condition Class . . . . .	321
35. New Attributes in Syspar Class . . . . .	321
36. EM_Resource_ID New Attributes . . . . .	321
37. EM_Resource_Monitor Class . . . . .	321
38. EM_Resource_Class . . . . .	322
39. EM_Resource_Variable. . . . .	322
40. New Commands in PSSP 3.1 . . . . .	323

41. Changes to Existing Commands in PSSP 3.1 . . . . . 324

---

## Preface

In October 1998, IBM announced enhancements to the RS/6000 SP that include a new version of the Parallel System Support Programs, Version 3, Release 1 (PSSP 3.1) and support for externally-attached nodes.

After an overview of the announcement, this redbook goes into detail about the various hardware and software enhancements, which include:

- The RS/6000 SP Enterprise Server attachment (7017-S70 and 7017-S7A), which is the first 64-bit node available for SP systems.
- Alternate and mirroring root volume groups support that greatly increases availability and flexibility for SP configurations.
- External SSA and SCSI disk booting support that enables SP nodes to boot from a pool of external disks.
- Switch enhancements including autounfence, a new switch admin daemon, and log consolidation.
- RS/6000 Cluster Technology (RSCT) for highly available SP and HACMP environments.
- SP Perspectives enhancements including multipane and multiwindow support for better manageability.
- A new PSSP packaging that includes additional products at no extra charge, such as: High Availability Control Workstation (HACWS), IBM Recoverable Virtual Shared Disks, and Performance Toolbox Parallel Extension (PTPE).
- A new Tivoli adapter for forwarding SP events to a Tivoli console (T/EC).
- A new resource center for online documentation.

Other SP-related products have also announced new releases. This redbook discusses the following:

- High Availability Cluster Multi-Processing (HACMP) 4.3.0
- General Parallel File System 1.2
- LoadLeveler 2.1
- Parallel Environment for AIX 2.4

This redbook is for IBM customers, Business Partners, IBM technical and marketing professionals and anyone seeking an understanding of the new hardware and software components and improvements included in this RS/6000 SP announcement.

---

## The Team That Wrote This Redbook

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

**Marcelo R. Barrios** is a project leader at the International Technical Support Organization, Poughkeepsie Center. He has been with IBM for five years working in different areas related to RS/6000. Currently, he focuses on RS/6000 SP technology by writing redbooks and teaching IBM classes worldwide.

**Rami Alfalahi** is an IT Senior Specialist with IBM Sweden. He has worked for IBM for eight years. His areas of expertise include SAP, RDBMS and SP support. Before joining the IBM support organization, he was a C developer. Rami holds an MS degree in Computer Science from the University of Salford, England.

**Jean-Michel Berail** is an AIX Specialist at the Parallel System Support Center (PSSC) in IBM Montpellier, France. He has worked for IBM for 20 years and has been with the RS/6000 SP group for one and a half years. He organizes, writes, and teaches AIX and RS/6000 SP courses in IBM Education Centers or customer areas. Jean-Michel is a graduate of the E.N.S.E.R.B. (Bordeaux, France) engineering school.

**Robin Findlay** is an AIX Specialist working in the AIX Support Centre at IBM in Basingstoke, UK. He has worked at IBM for two years. His areas of expertise include performance monitoring, SP, and network support. Before joining IBM, he worked at Southampton Institute on the implementation of computer conferencing systems for distance learning. He holds a Ph.D. in Zoology from the University of Aberdeen.

**Alan Foster** is Lead Technical Consultant for AnIX Computers Ltd, the UK's leading SP reseller and IBM business partner. He has worked at AnIX since its formation in 1990. His areas of expertise include SP planning, installation and support plus the integration of HACMP and NetView in an SP environment. He has worked with AIX for the last 12 years and prior to that with Unix System III.

**Theeraphong Thitayanun** is an Advisory IT specialist with IBM Thailand. He has been with IBM for 10 years. His main responsibility is to provide services and support in all areas of RS/6000 SP. His areas of expertise include RS/6000 SP, ADSM and HACMP. Theeraphong holds a degree in Computer Engineering from Chulalongkorn University and, as a Monbuscho student, a

Master Degree in Information Technology from Nagoya Institute of Technology, Japan.

**Hasan Hakan Yardim** is an I/T Specialist at IBM Global Services in Turkey. After joining IBM in 1996, he concentrated mainly on service support for RS/6000 SP, HACMP for AIX, and Internet Solutions on the AIX platform. He has been working in the UNIX area for five years. Hasan holds a B.S. degree in Electronics Engineering from the Bosphorus University in Istanbul.

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

***International Technical Support Organization***

Abbas Farazdel  
Hans-Juergen Kitzhoefer  
Yoshimichi Kosuge

***PPS Poughkeepsie Lab***

Jan Badovinat  
Peter Badovinat  
Diane Brent  
Michael Chase-Salerno  
Peter Chenevert  
Endy Chiakpo  
Michael Coffey  
Robert Curran  
Janet Ellsworth  
Zina Ferro  
Brian Herr  
Felipe Knop  
Linda Mellor  
Norman Nott  
Peter Pagerey  
Richard Treumann  
William Tuel  
Kevin Reilly  
Sean Safron  
Gina Yuan

***Education and Training***

Theodore Sullivan

**IBM Australia**

Darren Gilchrist

We would like to extend also our gratitude to Yann Guerin from IBM France for his great contribution to the success of this project.

Finally, one word of appreciation to Terry Barthel, Carol Dixon and Al Schwab, our editing team, for their hard work and commitment to get this book done on time.

---

**Comments Welcome**

**Your comments are important to us!**

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

- Fax the evaluation form found in "ITSO Redbook Evaluation" on page 343 to the fax number shown on the form.
- Use the electronic evaluation form found on the Redbooks Web sites:

For Internet users <http://www.redbooks.ibm.com>

For IBM Intranet users <http://w3.itso.ibm.com>

- Send us a note at the following address:

[redbook@us.ibm.com](mailto:redbook@us.ibm.com)

---

## Chapter 1. Announcement Overview

The announcement of the RS/6000 SP family includes new features, improvements to existing features, and removal of some restrictions existing in previous versions of the Parallel System Support Programs (PSSP). The new features include:

- PSSP 3.1 and support for AIX 4.3.2
- Support for RS/6000 Enterprise Servers (7017-S70/S7A) attachment
- New product releases:
  - HACMP ES 4.3.0
  - GPFS 1.2
  - LoadLeveler 2.1
  - Parallel Environment 2.4

Enhancements to existing features include:

- Support for up to four User Space tasks per node, enabling Message Passing Interface (MPI) applications to exploit SMP nodes and servers for significant performance improvements.
- Providing a more highly available environment by:
  - Support for booting from an external disk for selected nodes.
  - Support for mirroring root volume group (rootvg), which prevents a single disk from becoming a single point of failure.
  - Support for alternate rootvg volume groups, which provides booting of a single node with different software versions.
- An improved and consistent graphical user interface (SP Perspectives).
- National language support for the graphical user interface.
- Repackaging of PSSP to include formerly priced functions such as High Availability Control Workstation (HACWS), Performance Toolbox Parallel Extensions (PTPE), and the Recoverable Virtual Shared Disks (RVSD). The job management function has been included in the new version of LoadLeveler (version 2.1).
- A TME 10 Enterprise Console (T/EC) adapter for existing Tivoli customers, to forward event notifications from the Event Management subsystem of PSSP to the Tivoli Enterprise Console.
- A new SP Resource Center, providing one single interface for all softcopy SP documentation and information resources.

- New security enhancements, including the use of the AIX 4.3.2 authenticated remote commands.

Restrictions that have been removed in PSSP 3.1 include:

- Support for single SMP thin node configurations
- Support for multiple SP security configurations

The remainder of this chapter briefly describes these topics and provides a summary of the new hardware and software components that are part of this RS/6000 SP announcement.

---

## 1.1 New in PSSP

PSSP 3.1 is a new version of the Parallel System Support Programs that can run on RS/6000 SP (SP) and RS/6000 Enterprise Servers that are attached to SP Systems. The RS/6000 Enterprise Servers (7017-S70 and 7017-S7A) can be attached and run in either a switched or switchless environment.

The RS/6000 Enterprise Server uses RS/6000 feature codes, not RS/6000 SP feature codes. PSSP software must be licensed for the SP attached servers. For more information about attached RS/6000 Enterprise Servers, refer to Chapter 4, “SP-Attached Server Support” on page 125.

The new version of PSSP requires AIX 4.3.2. The control workstation must be running at the highest level of PSSP and AIX on the system. PSSP 3.1 can coexist with previous levels of PSSP in the same partition (PSSP version 2.2, 2.3, and 2.4). Older PSSP versions (1.2 and 2.1) are not supported in coexistence mode with PSSP 3.1.

PSSP 3.1 does not support the High Performance switch (HiPS), so systems with this type of switch need to be upgraded to the SP Switch prior to the installation of this version of PSSP.

Along with the support for RS/6000 Enterprise Server attachments, PSSP 3.1 includes support for future node enhancements, such as POWER3-based nodes (available 1Q/99). It also includes many features that were part of separate and priced Licensed Program Product (LPP) in previous versions of PSSP. Features such as High Availability Control Workstation (HACWS), Performance Toolbox Parallel Extension (PTPE) and Recoverable Virtual Shared Disks (RVSD) were priced LPPs in past PSSP versions. However, these features are shipped with PSSP 3.1 at no extra charge<sup>1</sup>.

<sup>1</sup> It may not apply to all geographies



### 1.1.1 New Hardware Support

This RS/6000 SP announcement includes several improvements in hardware components, such as support for SP-attached servers and support for additional PCI adapters. Details for the SP-attached servers can be found in Chapter 4, "SP-Attached Server Support" on page 125.

The new PCI adapters are:

- **S/390 ESCON Channel PCI Adapter** (Feature code 2751)

This adapter enables PCI nodes to attach to IBM Enterprise Systems Connection (ESCON) channels of the System/390 mainframe. The S/390 ESCON Channel PCI adapter attaches directly to an ESCON channel, providing fiber optic links using LED technology, and also attaches to ESCON Directors (fiber optic switches), permitting connections to multiple channels. Host operating systems supported include: VM/ESA, MVS/ESA, and OS/390.

The adapter has restrictions regarding the slots in which it may be installed. It requires one full-length PCI slot.

The adapter is supported in AIX 4.3.1 or later with a separately orderable device driver in licensed program 5765-D49.

Attributes provided by the adapter:

- Supports 3088, CLAW (TCP/IP), and MPC (SNA and TCP/IP) Emulations.
  - Supports CLIO/S.
  - Supports attachments to either 10MB or 17MB ESCON channels.
  - Supports ESCON Multiple Image Facility (EMIF).
  - Maximum distance supported using a combination of LED and XDF ESCON links is 43km.
- **8-Port Asynchronous PCI Adapter** (Feature code 2943)

The 8-Port Asynchronous Adapter provides direct connection for up to eight asynchronous EIA-232 or RS-422 devices from a single PCI bus slot. The adapter adheres to the Peripheral Component Interconnect (PCI) Revision 2.1 standard EIA-232 and RS-422. The adapter features a low-cost, high-performance 32-bit card, 33MHz bus speed, yielding a PCI bus transfer rate of 132MB/s. The adapter provides a single DB78 output connector, which connects directly to the 8-port DB25 connector box. All eight ports are software programmable to support either protocol, at up to 230K baud. The full set of modem control lines for asynchronous

communications is provided for each port. Devices such as terminals, modems, processors, printers, and controllers may be attached.

Attributes provided by the adapter:

- 8-port asynchronous device connections
- 32-bit Bus Master PCI bus - 132MB/s
- PCI short form factor adapter
- EIA-232 maximum distance of 62 m, dependent on baud rate
- RS-422 maximum distance of 1,200 m, dependent on baud rate
- 230K baud maximum rate
- **2-Port Multiprotocol PCI Adapter** (Feature code 2962)

The 2-Port Multiprotocol Adapter is used to make high-speed connections between stand-alone system units on a Wide Area Network (WAN). The adapter hardware supports SDLC and X.25. The adapter connects to WAN lines through externally attached data communication equipment, including Channel Service Units (CSU), Data Service Units (DSU), and asynchronous modems, at speeds up to 2,048Mbps. The adapter provides two ports that will accommodate one of four selectable interfaces: EIA-232D/V.24, V.35, V.36/EIA-449, or X.21. The interface is selected by the cable, which is separately orderable. A wrap plug is included with each adapter.

The 2-Port Multiprotocol Adapter requires software for SDLC and X.25 protocol services. IBM provides SDLC support as part of the AIX base operating system. IBM AIXLINK/X.25 provides a V.24, V.35, or X.21 port connection to X.25 packet switched networks. IBM AIXLINK/X.25 is a separately orderable AIX licensed program (5696-926), which should be reviewed for details on services supported.

- **TURBOWAYS 155 PCI UTP ATM Adapter** (Feature code 2963)

This adapter provides dedicated, 155Mbps full-duplex connection to ATM networks over either permanent virtual circuits (PVC) or ATM Forum compliant switched virtual circuits (SVC) UNI 3.1 signaling. The adapter supports the AAL-5 adaptation layer interface. It enables TCP/IP to run over an ATM network. It also supports communication with devices located on an ATM network or bridged to a token-ring, Ethernet, or other LAN. LAN Emulation (LANE) is provided by the AIX operating system.

The adapter requires customer-provided CAT5 High-Speed Unshielded Twisted Pair (UTP) or Shielded Twisted Pair (STP) cable certified for ATM operation with a maximum length of 100 meters, terminated with an RJ45 connector.

Attributes provided:

- Best effort service
- Signaling channel setup
- Virtual connection setup and tear-down
- Support for point-to-point and point-to-multipoint switched Virtual circuits (maximum 1024)
- Support for classical IP and ARP over ATM (RFC 1577)
- Support for Token Ring and Ethernet LAN Emulation
- Support for ATM SNMP
- **SCSI-2 Ultra/Wide PCI Adapter** (Feature code 6206)

The PCI Single-Ended Ultra SCSI Adapter is an ideal solution for applications requiring large block data transfers (>64KB block size) in a multi-disk-drive environment utilizing SCSI-2 protocol. The adapter has a maximum data transfer rate of 40MB/s, which is twice the maximum data transfer rate of the SCSI-2 Fast/Wide adapters (20MB/s). It occupies one PCI slot, and it conforms to the SCSI-2 standard and Fast-20 (Ultra) documentation. It uses a standard 68-pin connector.

- **SCSI-2 Differential Ultra/Wide PCI Adapter** (Feature code 6207)

This adapter is the next generation of SCSI-2 performance with maximum data transfer of 40MB/s. It allows connections to external SCSI/2 F/W or Ultra SCSI type devices up to 25 meters away.

The adapter will negotiate with each external device and transfer data at the fastest SCSI data transfer rate capable by the external device. It uses a standard 68-pin connector.

## **1.1.2 Alternate and Mirroring rootvg Volume Group Support**

PSSP 3.1 provides support for alternate root volume group (rootvg) and mirrored rootvg. In previous versions, the only way to have multiple bootable images or copies of root volume groups was by maintaining the disks and boot list manually. Now in this new version of PSSP, the SDR has been modified to support this feature. Nodes can have multiple rootvg definitions and mirroring can be initiated from the control workstation at any time.

### **1.1.2.1 Alternate rootvg Volume Group**

The alternate rootvg volume group feature allows you to configure multiple rootvg volume groups for nodes or group of nodes. You may select the rootvg volume group for the next boot by changing the boot list in the nonvolatile random access memory (NVRAM) and booting the node or control

workstation. PSSP provides the necessary commands to configure and select multiple rootvg volume groups for nodes or the control workstation. For more details refer to 2.4.1, “Multiple rootvg Support” on page 63.

#### **1.1.2.2 Mirrored rootvg Volume Group**

PSSP also offers the possibility to create copies of the current rootvg volume group, eliminating the single point of failure represented by the disk containing the volume group. Mirroring can be initiated at any time from the control workstation by using a set of commands provided for that purpose. The mirroring facility uses the standard AIX mirroring facility, which means it is fully supported for AIX. More details about this can be found in 2.4.1, “Multiple rootvg Support” on page 63.

### **1.1.3 External SSA/SCSI Boot**

Along with support for multiple rootvg volume groups, PSSP 3.1 provides the ability for nodes to boot from external disk (SSA or SCSI), making it possible to configure a highly available system by combining these new features.

Some SP nodes can now be booted from external Serial Storage Architecture (SSA) disk storage devices or an external SCSI-2 Fast/Wide disk 7027-HSD storage device, providing SP customers with increased capacity and higher availability. The option to have an SP node without internal disk is also supported.

Users can now configure a pool of disks from which nodes can boot. Only P2SC, Wide 77Mhz, and High nodes support the external boot disk feature.

For more information refer to 2.4.2, “Booting from External Disks” on page 65.

### **1.1.4 Improved Network Adapter Support**

In previous versions of PSSP, if you attempt to configure an adapter other than css0, en0, en1, tr0, tr1, fi0, or fi1, you would have received an error message indicating that the adapter name was invalid. Previous versions of PSSP only supported a limited number of adapter instances.

Now with PSSP 3.1 you can define unlimited instances of PSSP supported adapters per node. The adapter definitions are stored in the SDR, so when the node is installed those adapters defined in the SDR will be configured as part of the customization phase at the end of the installation process.

### 1.1.5 Switch Improvements

PSSP 3.1 includes several improvements for the switch subsystem. The improvements are intended to provide a more reliable and stable switch subsystem with less human intervention. They include:

**Autounfence** - Nodes are now automatically unfenced when they have been fenced by the primary node. This allows nodes to rejoin the switch after a failure occurs and the node had to be rebooted.

**Error Log Consolidation** - Although the control workstation is not part of the switch network, it is the single point of control for the RS/6000 SP. This new feature allows you to consolidate switch error log information on the control workstation for further analysis and actions.

**Switch Admin Daemon** - This new daemon runs on the control workstation and monitors Node and Adapter events and responds with automatic Estarts when appropriate.

The High Performance Switch (HiPS) is not supported by PSSP 3.1.

For more details about these features refer to Chapter 5, "Switch Support Enhancements" on page 161.

### 1.1.6 New Packaging

Several packaging changes were made in this PSSP version. Most of the features that were separately orderable and priced are now part of the base PSSP code. They include:

- High Availability Control Workstation (HACWS)
- Performance Toolbox Parallel Extensions (PTPE)
- Recoverable Virtual Shared Disks (RVSD)

Also, some features have been "moved" outside the PSSP main fileset as the new RS/6000 Cluster Technology (RSCT), which now features a new package of its own:

- RSCT Basic
- RSCT Clients

Each of these components offers different installable images depending on the environment you want to work with. The environments supported are:

- HACMP Domain (RS/6000 family)
- SP Domain (SP only)

For more information about these changes, refer to Chapter 6, “RS/6000 Cluster Technology” on page 183.

For more information about PSSP packaging refer to 2.1.4.1, “New Packaging” on page 34.

### 1.1.7 TME 10 Integration

Tivoli Management Environment (TME 10) can be used to monitor SP-generated events. PSSP 3.1 includes a TME 10 Enterprise Console (T/EC) adapter that allows you to forward events generated by the Event Management subsystem to the TME 10 Enterprise Console, which acts as a centralized point of control for the TME 10-managed environment. To define which events are forwarded by Event Management, you can use either the Event Management Perspective or the `pmandef` command.

The adapter consists of the `tecad_pssp` command and the `rvclasses.cfg` and `pssp_classes.baroc` configuration files. This product is offered as an optional fileset, `ssp.tecad`, and is installed using the `installp` command. It can be installed on any node, or the control workstation.

After installing the T/EC adapter, you have to run the `install_agent` command which will include the event definition in the SDR. If you are installing this adapter on the control workstation, you have to run `install_agent` on every partition, since this command is partition-sensitive. You can do it by changing the `SP_NAME` variable to match the different partitions.

The `install_agent` command must be run on every node where you want to forward events to the TME 10 Enterprise Console. If you plan to have similar configuration files on all your nodes in a partition, you may consider having a master file on the control workstation and distributing this file across nodes by using file collections.

The `install_agent` command is included in the `/usr/lpp/ssp/tecad` installation directory. It requires the name of the configuration file to be installed, which in the standard installation case is `rvclasses.cfg` (also in the installation directory). For example, the following command will install the SDR classes for the PSSP T/EC adapter:

```
/usr/lpp/ssp/tecad/install_agent /usr/lpp/ssp/tecad/rvclasses.cfg
```

The class added to the SDR is `Tec_Agent_Class` and it contains the definition for all the resource variables known by Event Management.

Once the `install_agent` command has succeeded in loading the SDR with the proper information, the `tecad_pssp` command can be utilized. This command can reside anywhere in your system, but it requires a configuration file. You can provide the name and path of the configuration file using the `-l` flag, or you can use the default path/name. The default name is `/usr/lpp/pssp/config/tecad_pssp.cfg`. Check the Tivoli EIF Manual for information on configuration files. The only mandatory information is the location of the T/EC server, in the form:

```
ServerLocation="your.server.name.domain"
```

If you have a T/EC installation on the node where you are installing the `tecad_pssp` command, it is probably a good idea to put the `tecad_pssp` command and the `tecad_pssp.cfg` files where the other adapters reside (`/etc/Tivoli/tecadap`).

We highly recommend that the `tecad_pssp` command be installed on the control workstation, which has access to all system partitions. This way you can create event subscriptions to any node, and select the "action" (the `tecad_pssp` command) to run on node 0 (the control workstation).

The syntax for the `tecad_pssp` command is:

```
tecad_pssp [-f path/filename] [-Cc] [-m "text"] [-a "admin"] [-s severity] [-p port]
```

where:

- `-f`: is the path/filename of the configuration file. The default value is `./tecad_pssp.cfg`.
- `-C`: for connection-oriented protocol.
- `-c`: for connectionless protocol (default).
- `-m`: adds "text" to the message field of the event.
- `-a`: adds "admin" in the `tec_administrator` field of the event.
- `-s`: sets the severity of the event to "severity". This should one of the following strings:
  - FATAL
  - CRITICAL
  - WARNING
  - MINOR
  - HARMLESS
  - UNKNOWN

`-p:` sets the communication port number to "port". Note that you can also set the port number in the configuration file.

The `tecad_pssp` command is designed to be called by the problem management subsystem (PMAN). It will not work without the environment variables supplied by the `pman` daemon. In order to use the `tecad_pssp` command, you need to make a subscription to the Problem Management subsystem, either using the `pmandef` command or the Event Perspective GUI.

#### 1.1.7.1 Loading the PSSP Classes on the Tivoli Server

In order to receive the events generated by the `tecad_pssp` command in a TEC server, you will need to load the `pssp_classes.baroc` file in the TEC rulebase. Consult the *TEC User's Guide* for details on this procedure.

The `tecad_pssp` command is distributed as an executable, and also in source form. If you want to modify the source code so that the adapter can handle custom classes, you will need to recompile the source file. You need to compile the `tecad_pssp.c` file using the `gcc` compiler to ensure compatibility with the TEC/EIF libraries.

#### Important

The `tecad_pssp.c` file is sample source code, and as such is not a supported IBM product.

You can compile this command configured to use secure communications (in which case the node where `tecad_pssp` runs needs to be a Tivoli-managed node), or using unsecure communications (in which case the node where the `tecad_pssp` command runs does not need to have any Tivoli software in it). The use of an unsecure channel is not a severe penalty, since it is used for event notification only. See the TEC/EIF manuals for a description of the communication channel modes. The package supplied with PSSP contains a little script for compiling the `tecad_pssp` command, called `makeit`. You can build the secure/unsecure `tecad_pssp` by typing:

```
makeit secure
```

or

```
makeit unsecure
```

(If you do not specify the channel option, unsecure is assumed.)



### 1.1.7.2 Extending `tecad_pssp` to Handle User Defined Events

If you want to extend the `tecad_pssp` command to handle user-defined events, you need to:

1. Create the new resource variable and make it known to the Event Management subsystem (see details on the *Event Management Programming Guide and Reference*, SA72-7354).
2. Map this new resource variable into some T/EC event class. You can use one of the classes created in the `pssp_classes.baroc` file, or create a new one. If you create a new class, you need to load it in the Tivoli console.

More details on this procedure can be found in the README file located in the `/usr/lpp/ssp/tecad` directory.

### 1.1.7.3 Using the `tecad_pssp` Command

You can use the `tecad_pssp` command as an action in the `pmandef` command. This means that you have to be authorized to use this command (your Kerberos principal must be listed in the `/etc/sysctl.pman.acl` file).

To define and forward an event to the T/EC adapter, use the following syntax:

```
pmandef -s example1
-e "Any Resource Variable;Any Instance Vector;Any Predicate"
-c "$AGENT_PATH/tecad_pssp -l $CONF_PATH/tecad_pssp.cfg"
-r "Any Rearm Predicate"
-C "$AGENT_PATH/tecad_pssp -l $CONF_PATH/tecad_pssp.cfg"
-n 0
```

You should run this command from the control workstation to save installation efforts and keep the management of the system easier. This is done by setting the `-n` flag to 0, indicating that the command needs to be run on Node 0, the control workstation.

## 1.1.8 SP Perspectives

The SP Perspectives graphical interface has been greatly improved in this new PSSP release. Major rework was done at the framework level, giving consistency to all the perspectives windows and menus.

Changes in the SDR are now refreshed automatically in the interface, making it possible to keep track of changes in the system.

Users have the ability to display multiple panes in the same or different windows. Also, views can be customized to display lists, icons, or tables.

The System Monitor (spmon) graphical interface (flag -g) has been removed from this release of PSSP, but the functionality has been moved into Hardware perspective and enhanced. Users can now monitor multiple hardware and software conditions from a single icon.

For details about the improvements in the interface, see Chapter 3, “SP Perspectives” on page 71. For a complete user’s guide, refer to *SP Perspectives Comprehensive Guide*, SG24-5180.

### 1.1.9 SP Resource Center

The new SP Resource Center is a browser-based application utilizing Java and JavaScript technology. It is available on the PSSP distribution tape and CD-ROM. The CD-ROM can be installed to an AIX platform or run from the CD. on AIX/NT/Win95 platforms.

Online books are shipped on the CD-ROM in HTML and PDF formats.

The Resource Center is accessed from the Launch Pad or by running the `/usr/lpp/ssp/bin/resource_center` command. The initial screen display is shown in Figure 1.



Figure 1. Resource Center Welcome Screen

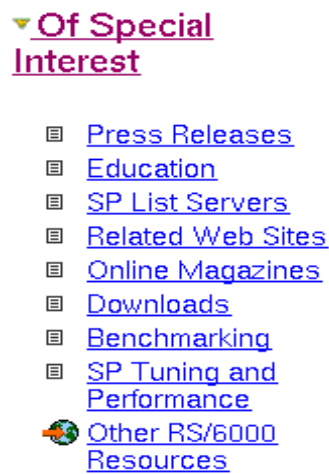
### 1.1.9.1 Links to Local Information

The Resource Center will automatically create links to installed publications on the platform it is installed on. The CWS is an obvious choice as this would normally have the PSSP and any related LPP documentation installed.

The resource center dynamically determines which resources are locally available. Links are maintained for

- Publications (Locally installed)
- README files (Locally installed)
- SP Performance Information
- RS/6000 Software and Hardware Information

The Netscape index frame for the *Of Special Interest* section is shown in *Figure 2*.



*Figure 2. Sample Section of the Resource Center Index Frame*

Note the link information, text pages adjacent to items which are local, and the *Other RS/60000 Resources* available out on the Web.

### 1.1.9.2 Links to Information on the Web

Where information is not locally available, links are made for:

- Publications: Browse, Search, Download or Order
- Readme files not locally installed

- Redbook and White Papers
- Education
- Up-to-date service information
- Up-to-date performance information
- Product information

### **1.1.10 Software Requirements**

PSSP 3.1 is supported on AIX 4.3.2. It is compiled on AIX 4.3.1 and is supported by binary compatibility on AIX 4.3.2. Your installation's current operational requirements should give you a good understanding of the software requirements that will exist in your RS/6000 SP after it has been migrated to PSSP 3.1. For more information refer to Chapter 2, "Installation Management Enhancements" on page 17.

### **1.1.11 Cluster Technology (RSCT)**

PSSP 2.2, announced in 1996, was one of the major PSSP announcements. It included a completely new way to provide high availability to SP systems. The core of this new functionality consists of the three subsystems that provide connectivity and availability information:

- Topology Services
- Group Services
- Event Management

Great improvements have been made since this initial release of PSSP. Along with PSSP 2.3, announced in August 1997, HACMP Enhanced Scalability 4.2.2 was announced too. Based on the HACMP for AIX interface, but using the high availability infrastructure, this new product provides a way to create HACMP clusters on the SP, but does not limit them to eight nodes per cluster, but to 32, with a statement of direction (SoD) to get to the number of nodes supported by PSSP (128 at that time).

PSSP 3.1 now brings another set of improvements to both the initial high availability infrastructure and the HACMP Enhanced Scalability by making it possible to integrate RS/6000 workstations and servers, and RS/6000 SP nodes, in one or multiple clusters, even including overlapping. In order to provide this new functionality, the former high availability infrastructure had to be repackaged and renamed to become an independent filesset.

The new RS/6000 Cluster Technology (RSCT) is now shipped as a separate fileset; it provides the "basic plumbing" for both PSSP and HACMP Enhanced Scalability 4.3.0.

Each of the three basic components has been enhanced to provide support for a heterogeneous environment that includes SP nodes and standard RS/6000 workstations and servers. All the dependencies on PSSP code have been removed to allow these components to run in a pure AIX environment.

For more information on this RS/6000 Cluster Technology, refer to Chapter 6, "RS/6000 Cluster Technology" on page 183.

For more information about changes in HACMP/ES, refer to Chapter 8, "HACMP/ES 4.3.0" on page 227.

### 1.1.12 Security

Since the announcement of PSSP 2.4 support for AIX 4.3 in April 1998, SP security has started to slowly move from PSSP-provided services to be part of the standard AIX services. PSSP 3.1 is the first of several future releases that will exploit the new and enriched security features provided by the AIX operating system.

In PSSP releases prior to PSSP 3.1, security settings were global, Kerberos V4 the default, and only authentication and authorization mechanisms were supported on the RS/6000 SP. PSSP 3.1 now brings new choices to the security settings on your SP:

- Security settings are no longer global but partition-sensitive.
- Authentication and authorization methods can be:
  - Standard AIX
  - Kerberos V4
  - Distributed Computing Environment (DCE), which means Kerberos V5
- Remote commands (`rsh`, `rccp`, `rlogin`, `telnet`, and `ftp`) are provided by the AIX operating system, and they support multiple authentication methods.
- New administrative commands are offered to manage the security setting from the control workstation.

For more information about SP, security refer to 2.1.4.2, "Security" on page 35.

---

## **1.2 New Product Releases**

Along with PSSP, this announcement includes new releases for several licensed program products.

### **1.2.1 HACMP/ES 4.3.0**

This new release of HACMP includes a set of new features that include the ability to run in multiple clusters, including a RS/6000 cluster, outside the SP. For more details about this new HACMP release, refer to Chapter 8, “HACMP/ES 4.3.0” on page 227.

### **1.2.2 GPFS 1.2**

The first release of GPFS was announced early this year. This second release includes several improvements such as dynamic node and disk addition to the GPFS pool, dynamic inode allocation for file systems and more. For further information, refer to Chapter 9, “GPFS 1.2” on page 245.

### **1.2.3 LoadLeveler 2.1**

This new version of LoadLeveler includes the support for interactive parallel jobs. This functionality was delivered by Resource Manager in previous PSSP releases. LoadLeveler allows you now to schedule your interactive parallel jobs using the graphical user interface or through the POE command line interface. Refer to Chapter 10, “LoadLeveler Version 2.1” on page 259 for more information.

### **1.2.4 Parallel Environment 2.4**

The two major enhancements in this new release of Parallel Environment are the ability to run multiple user space job in a single node, and the support for part of the MPI-2 standard, including the MPI-IO subset. For more information, refer to Chapter 11, “Parallel Environment 2.4” on page 295.

---

## Chapter 2. Installation Management Enhancements

This chapter explains the installation process, the migration rules, the coexistence rules, and the new features used for installing PSSP 3.1. Although detailed information about these topics is provided in other publications, we describe here what is new or different for each of these factors in PSSP 3.1.

---

### 2.1 Installation Process

The Network Installation Management (NIM) facility in AIX is used to install the nodes. Not all the capabilities of NIM are used in the RS/6000 SP environment, only the minimum needed to install a stand-alone machine. For further information about NIM, refer to *Network Installation Management Guide and Reference*, SC23-2627.

To run properly, NIM needs information regarding the network, the machines (master, client) and the resources (spot, lppsource, mkysyb, bosinst\_data, and so forth) to be managed.

NIM organizes the install environment into object classes, object types, and object attributes. As shown by Figure 3 on page 18, there are three classes of objects:

- Machines (including the master and the clients)
- Networks
- Resources

Within each class, there are object types:

- Machine types: master, stand-alone, diskless, dataless
- Network types: ent, tok, Fiber Distributed Data Interface (FDDI)
- Resource types: files and directories such as lppsource, mkysyb, script, bosinst\_data, spot

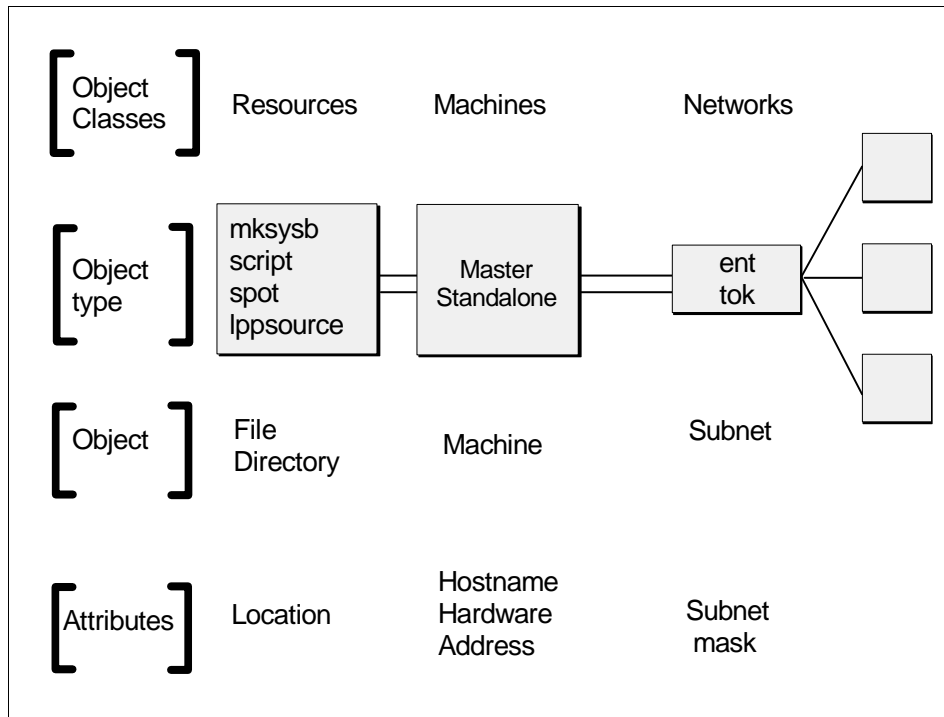


Figure 3. NIM Objects

In the RS/6000 SP environment, all this information is stored in the System Data Repository (SDR). The `setup_server` script uses the information stored in the SDR to create the NIM environment.

Once `setup_server` creates the NIM objects, you can start the installation by using either Node Condition or Manual Node Condition. Refer to Figure 6 on page 25 for an overview of the mechanism.

In this chapter, we only point out the changes brought by PSSP 3.1 during the installation process. For the installation details, refer to *IBM Parallel System Support Install & Migration Guide Version 3 Release 1, GA22-7347*.

### 2.1.1 An RS/6000 as a Control Workstation

This section is divided into four parts corresponding to the four major steps that occur during the whole RS/6000 SP installation:

- 2.1.1.1, "Prepare the Control Workstation." on page 20.
- 2.1.1.2, "Adapt the AIX Environment" on page 22.



- 2.1.1.3, “Install the PSSP” on page 23.
- 2.1.1.4, “Enter Data into the SDR” on page 24.

In addition to the operational requirements placed on your system software, IBM RS/6000 software products also have PSSP release level dependencies. Table 1 summarizes those dependencies.

Table 1. Supported IBM LPPs Per Supported PSSP and AIX Release

PSSP and AIX	IBM LPPs
PSSP 3.1 (5765-D51) and AIX 4.3.2 (or later) (5765-C34)	<ul style="list-style-type: none"> <li>•LoadLeveler 2.1 (5765-D61)</li> <li>•Parallel Environment 2.4, 2.3 (5765-543)</li> <li>•Engineering and Scientific Subroutine Library (ESSL) 3.1 or later (5765-C42)</li> <li>•Parallel ESSL 2.1 or later (5765-C41)</li> <li>•General Parallel File System 1.2 (5765-B95)</li> <li>•CLIO/S 2.2</li> <li>•Network Tape Access and Control System 1.2 (5765-637)</li> <li>•NetTAPE Tape Library Connection 1.2 (5765-643)</li> <li>•HACMP/ES and HACMP 4.3 (5765-D28)</li> </ul>
PSSP 2.4 (5765-529) and AIX 4.2.1 (or later) or AIX 4.3 (5765-655 or 5765-C34)	<ul style="list-style-type: none"> <li>•LoadLeveler 2.1 (5765-145), LoadLeveler 1.3 (5765-145)</li> <li>•Parallel Environment 2.3 (5765-543)</li> <li>•Parallel ESSL 2.1 (5765-C41)</li> <li>•General Parallel File System 1.1 (5765-B95)</li> <li>•Recoverable Virtual Shared Disk 2.1 (5765-646)</li> <li>•PIOFS 1.2 (5765-297)</li> <li>•Performance Toolbox Parallel Extensions (priced feature of PSSP)</li> <li>•CLIO/S 2.2</li> <li>•Network Tape Access and Control System 1.2 (5765-637)</li> <li>•NetTAPE Tape Library Connection 1.2 (5765-643)</li> <li>•HACMP/ES and HACMP 4.2 (5765-A86)</li> <li>•HACWS (priced feature of PSSP)</li> </ul>

PSSP and AIX	IBM LPPs
PSSP 2.2 (5765-529) and AIX 4.2.1 or AIX 4.2.0 (5765-655 or 5765-C34)	<ul style="list-style-type: none"> <li>•LoadLeveler 1.3</li> <li>•Parallel Environment 2.2</li> <li>•PVMe 2.2</li> <li>•Parallel ESSL 1.2 (5765-422)</li> <li>•PIOFS 1.2</li> <li>•Performance Toolbox Parallel Extensions (priced feature of PSSP)</li> <li>•Recoverable Virtual Shared Disk 1.2 (5765-444) CLIO/S 2.2</li> <li>•NetTAPE 1.1.1</li> <li>•HACMP 4.2</li> </ul>
PSSP 2.2 (5765-529) and AIX 4.1.5 or AIX 4.1.4 (5765-393 or 5765-C34)	LoadLeveler 1.2.1 and 1.3 Parallel Environment 2.2 PVMe 2.2 Parallel ESSL 1.2 PIOFS 1.2 Performance Toolbox Parallel Extensions (priced feature of PSSP) Recoverable Virtual Shared Disk 1.2 CLIO/S 2.2 NetTAPE 1.1.1 HACWS (priced feature of PSSP) HACMP 4.2

PSSP 2.1 is no longer supported in the same system partition. Also migration to this new level of PSSP is only available from certain levels of PSSP and AIX only. For a complete reference, see 2.2, "Migration" on page 52. For coexistence issues, refer to 2.3, "Coexistence" on page 62.

The PSSP 3.1 enhancement details have been gathered in section 2.1.4, "New in PSSP 3.1: More Details" on page 33.

#### **2.1.1.1 Prepare the Control Workstation.**

To properly install a CWS with PSSP 3.1, you must check first the hardware connections. A new feature of PSSP 3.1 is the support for SP-attached servers like RS/6000 Enterprise Servers (7017-S70/S7A). They require two serial port connections: one to provide tty support, and the other to be used for hardware connections. Thus, in certain cases, you may need an asynchronous adapter card to provide the extra ports. Chapter 4, "SP-Attached Server Support" on page 125 gives more information about the S70/S7A attachment.

Because the CWS is a boot/install server, it must have enough disk space to save all the data required by NIM: in case of migration or coexistence, different AIX versions or PSSP levels are used and must be stored in the CWS.

See *IBM RS/6000 SP Planning Volume 2, Control Workstation and Software Environment*, GA22-7281 for detailed information about recommended disk space.

Once the hardware connections requirements are done, you must install the requisite software:

- AIX Version 4.3.2
- bos.net (TCP/IP and NFS)
- perfagent.tools Version 2.2.32.x

**Note**

For coexistence only with older versions of PSSP (2.4, 2.3, or 2.2), you must also install perfagent.server Version 2.2.32.x.

See 2.2.4, "CWS Migration" on page 56 for more detail about perfagent.sever versions.

Now we can go to the PSSP environment itself. The first change that is apparent to you is related to the /spdata directory structure. A new directory named PSSP-3.1 must be created to store the PSSP 3.1 file sets as shown by the Figure 4 on page 22.

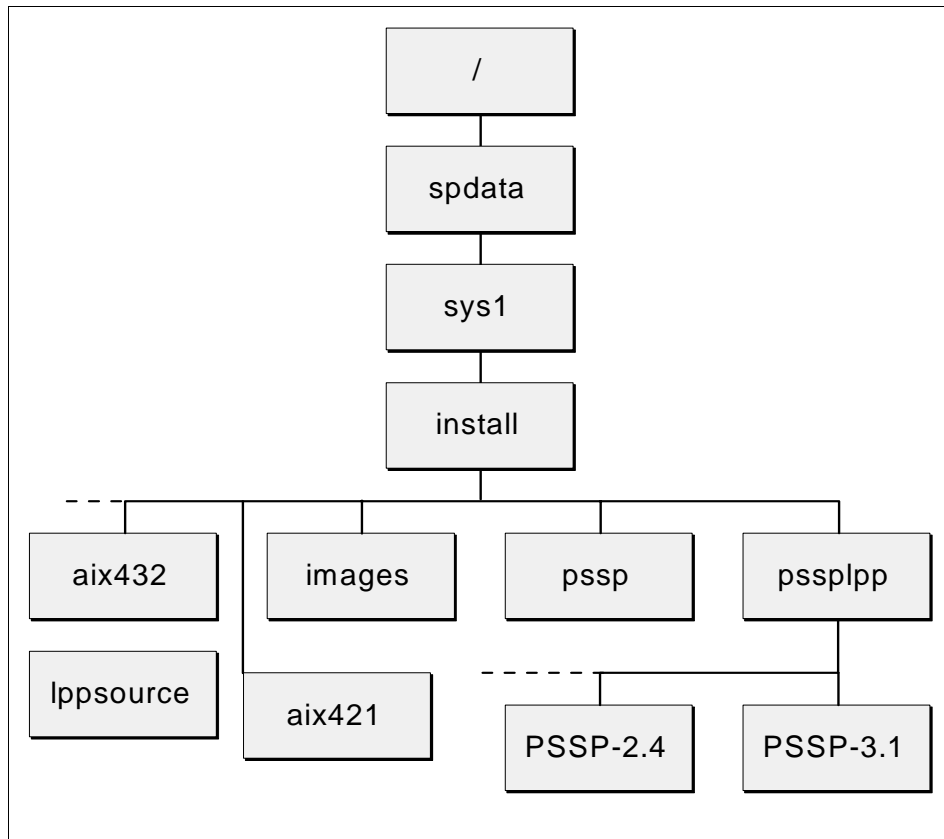


Figure 4. /spdata Directory Structure

Once the /spdata structure has been created, you have to fill it up. The following sections help you decide what to put in the right place.

### 2.1.1.2 Adapt the AIX Environment

Before you install the PSSP images on the CWS, you first need to copy the images at the right levels (4.3.2 for AIX and 3.1 for the PSSP) of the installable file sets or mksysb to the appropriate directories. To do that, follow these three steps:

- Copy the AIX LPP image in the directory

/spdata/sys1/install/<name>/lppsource

where <name> represents the AIX level, such as aix432.

Do not forget the perfagent.tools that must be copied in the lppsource directory.

- Copy the PSSP filesets in the directory  
`/spdata/sys1/install/pssplpp/<code_version>`

where <code\_version> is a reserved name for the PSSP version (PSSP-3.1 in our case).

Some modifications on the packaging of the PSSP file sets have occurred; section 2.1.4.1, “New Packaging” on page 34 shows the details.

- Copy the installable image (or images) for the nodes (mksysb format) into the directory

`/spdata/sys1/install/images`

Although you can choose any name for the mksysb files, we recommend you choose one that represents the related AIX level.

### 2.1.1.3 Install the PSSP

PSSP 3.1 comes with a new packaging.

The PSSP images are made up of one or more file sets. You do not need to install all those file sets to your CWS: only some are mandatory while the others are optional depending on how your RS/6000 SP is configured (Is a switch installed? Do we use the CWS as an authentication server?). In the same way, some of them are installed on the nodes later in the installation process.

Table 2 shows the PSSP file sets that must be installed on the CWS.

*Table 2. Mandatory PSSP File Set*

File set	Required on CWS	Comments
rsct.basic	Yes	
rsct.clients	Yes	
ssp.authent	Yes	If using PSSP Kerberos server code
ssp.basic	Yes	
ssp.clients	Yes	
ssp.css	Yes	If switch is installed
ssp.ha_topsvcs.compat	Yes	
ssp.sysctl	Yes	
ssp.top	Yes	If switch is installed

For a complete list of file sets, refer to *RS/6000 SP: Planning, Volume 2, Control Workstation and Software Environment*, GA22-7281.

As described in 1.1.7, "TME 10 Integration" on page 8, PSSP 3.1 includes a TME 10 Enterprise Console (T/EC) adapter used to monitor SP-generated events. Thus, you can optionally add the PSSP T/EC adapter to your system. For further information on that subject, refer to *IBM Parallel System Support Programs for AIX Installation and Migration Guide Version 3 Release 1*, GA22-7347.

#### 2.1.1.4 Enter Data into the SDR

After you have collected the information you need (see the worksheets provided in *IBM RS/6000 SP Planning Volume 2, Control Workstation and Software Environment*, GA22-7281, for guidance), you must enter the data into the SDR. To do that, a list of SMIT panels is available.

The first SMIT panel (fastpath enter\_data) which is the starting point to enter information into the SDR (see Figure 5), has changed to accept information from a non-SP Frame (S70/S7A).

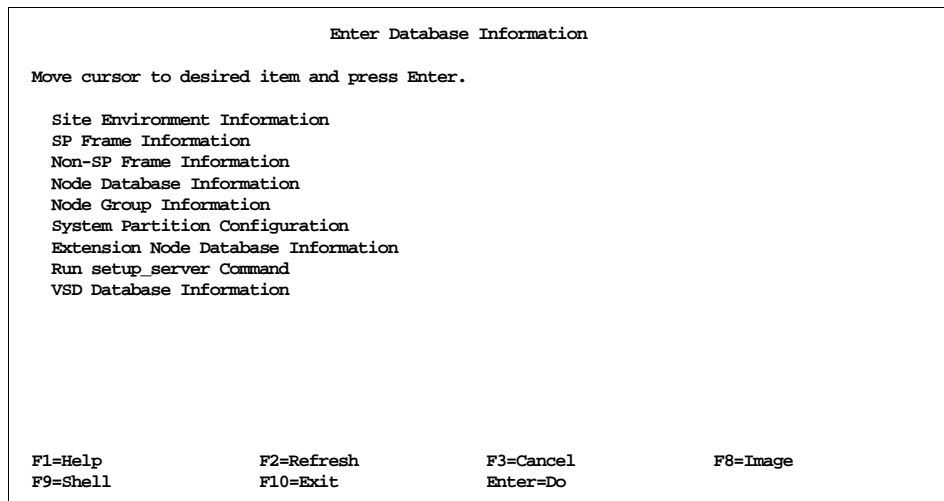


Figure 5. SMIT Panel to Enter Data into the SDR

The SDR modifications driven by this PSSP version are mainly related to the nodes. The Node Database Information sub-menu, as shown by Figure 7 on page 26, has been changed to allow you to enter the information generated by the PSSP 3.1 enhancements.

## 2.1.2 Main Components of the Installation Process

As discussed, SDR and NIM are the two major components used for the installation process. However, a third important actor is used to transfer data from the SDR to the NIM database: the script `setup_server`, as shown in Figure 6.

This section, after discussing what information is needed and how to enter it into the SDR, describes the two ways of installing a client given by NIM, and finally, gives information related to the `setup_server` script.

### Note

Although some of the information in this section is not new, we provide it so you will have a better understanding of the enhancements included in PSSP 3.1.

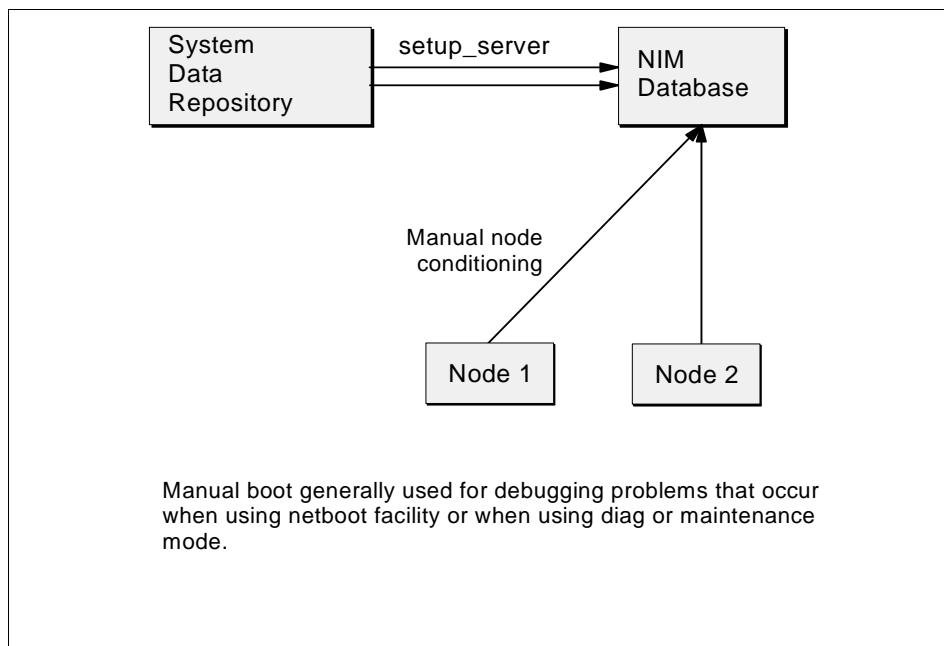


Figure 6. Installation Process

### 2.1.2.1 SDR

Because the SDR changes driven by the S70/S7A attachment are discussed in other section, we describe here the improvements related to the management of the nodes.

PSSP 3.1 allows us to manage new concepts directly from the CWS: multiple rootvg, capability to mirror Volume Groups, and capability to change the bootlist on a node. To do that, a new SDR class has been created, the Volume\_Group class. Therefore, new commands and new SMIT panels are provided to enter this new information into the SDR. This section describes only what is new about how to enter this node information. Details and explanations related to the new commands are found in 2.1.4, “New in PSSP 3.1: More Details” on page 33.

```
Node Database Information

Move cursor to desired item and press Enter.

SP Ethernet Information
Get Hardware Ethernet Addresses
Additional Adapter Information
Hostname Information
Create Volume Group Information
Change Volume Group Information
Start Disk Mirroring
Discontinue Disk Mirroring
Set Bootlist
Boot/Install Server Information
Accounting Information
Get Node Description Information

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

Figure 7. SMIT Panel Related to Node Information

The new SMIT panel (fastpath node\_data), as shown by Figure 7, gives five new options:

- Create Volume\_Group Information
- Change Volume\_Group Information
- Start Disk Mirroring
- Discontinue Disk Mirroring
- Set Bootlist

The first two correspond to the newly created Volume\_Group class, the next two options correspond to the Volume\_Group mirroring, and the last one corresponds to the ability to setup the node bootlist.

These new options are discussed in Part 2.1.4, “New in PSSP 3.1: More Details” on page 33.



### 2.1.2.2 NIM

As shown in Figure 8, NIM gives you two modes to install a client:

- Push mode: the master initiates the installation process.
- Pull mode: the client initiates the installation process.

In the RS/6000 SP environment, only the Pull mode is supported. If the Push mode is activated (manually only), the installation will not finish properly.

The Pull mode itself can be activated in two ways:

- Node Conditioning (an automatic way)
- Manual Node Conditioning

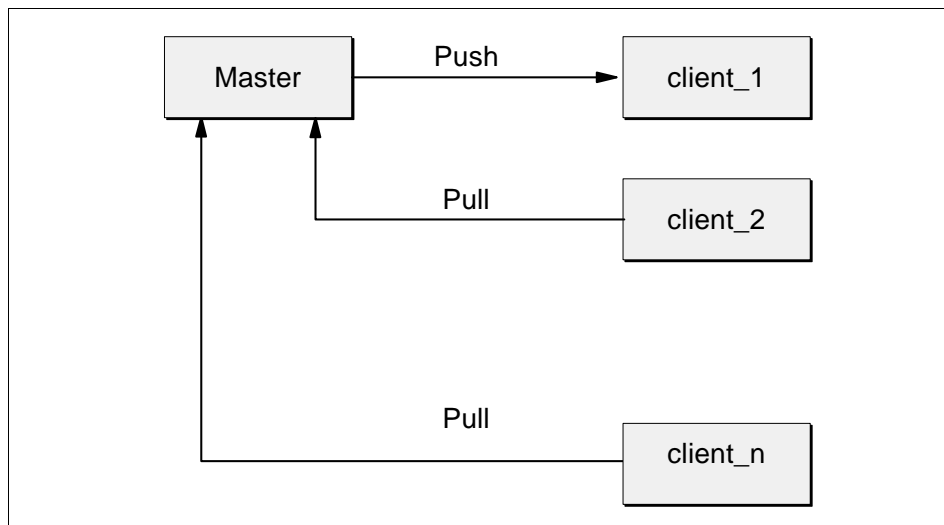


Figure 8. NIM Basic Uses

### 2.1.2.3 The setup\_server Script

Regarding NIM, the most important purpose of this script is that it takes information from the SDR to put into the NIM database. This data is then used to properly install the nodes.

The setup\_server script is divided into parts named *wrappers*. Each wrapper executes a specific task:

- Creation of the machine master with its network
- Creation of the resources (spot, lppsource, mkysyb, bosinst\_data, script)

- Creation of the machine client
- Allocation of resources to a client: the value of the bootp\_response parameter determines the list of the resources allocated to the node.

Figure 23 on page 49 gives the new SMIT panel which allows you to enter this parameter

Table 3 gives the different values of the bootp\_response parameter.

*Table 3. Available Values for the bootp\_response Parameter*

<b>Value</b>	<b>Action</b>
install	The installation of the node is enabled
disk	None
customize	The node will be customized
diag	Go into diagnostic mode
maintenance	Go into maintenance mode
migrate	The migration of the node will occur

Table 4 shows the resources allocated by NIM according to the value of the bootp\_response parameter.

*Table 4. Resources Allocated by NIM According to the bootp\_response Value*

<b>bootp_response Value</b>	<b>Allocated Resources</b>
install	bosinst_data lpp_source mksysb script spot
customize	None
disk	None
maintenance	bosinst_data lpp_source spot
diag	bosinst_data spot

bootp_response Value	Allocated Resources
migrate	bosinst_data lpp_source script spot

## 2.1.3 Installation

During the installation, after the mksysb image is downloaded, you may want to automate additional customization such as:

- Adding installp images
- Configuring host names
- Configuring adapters that are not configured automatically
- Running specific scripts

Two different customer-supplied scripts are provided to achieve these customizations. Before starting the installation process, you must modify these scripts according to the final statement you want (see 2.1.3.1, “Node Customization” on page 29 for some additional information). Once this is done, you can start the installation.

### 2.1.3.1 Node Customization

In PSSP releases prior to PSSP 2.4, the script.cust file was used exclusively for node customization. This is now divided between script.cust, which is executed before the node reboots, and firstboot.cust, which is executed at the end of the reboot.

To work with script.cust and firstboot.cust, you have to do the following steps:

- Make a copy of a sample located in /usr/lpp/ssp/samples and put it in the /tftpboot directory by issuing the commands:

```
cp /usr/lpp/ssp/samples/script.cust /tftpboot
cp /usr/lpp/ssp/samples/firstboot.cust /tftpboot
```

- Make these scripts executable by issuing the commands:

```
chmod +x /tftpboot/script.cust
chmod +x /tftpboot/firstboot.cust
```

Table 5 shows the general uses of script.cust and firstboot.cust.

Table 5. General Uses of script.cust and firstboot.cust

Name	Uses
script.cust	<ul style="list-style-type: none"><li>•Set time zone</li><li>•Modify page space</li><li>•Install LPPs that need a system reboot</li><li>•Change max users</li><li>•Change system environment</li></ul>
firstboot.cust	<ul style="list-style-type: none"><li>•Import Volume Groups</li><li>•Set up your method of name resolution</li><li>•Install LPPs that do not need a system reboot</li><li>•Increase LV or file system space</li><li>•Copy security files</li><li>•Copy your personal scripts and run them</li></ul>

### 2.1.3.2 Network Boot: Node Conditioning

One of the enhancements of PSSP 3.1 is related to the graphical user interface Perspectives (for more information, refer to Chapter 3, “SP Perspectives” on page 71). Figure 9 on page 31 shows one of the new windows which allows you to control a node. To start a network boot, you only need to use the push button labeled **Network Boot**.

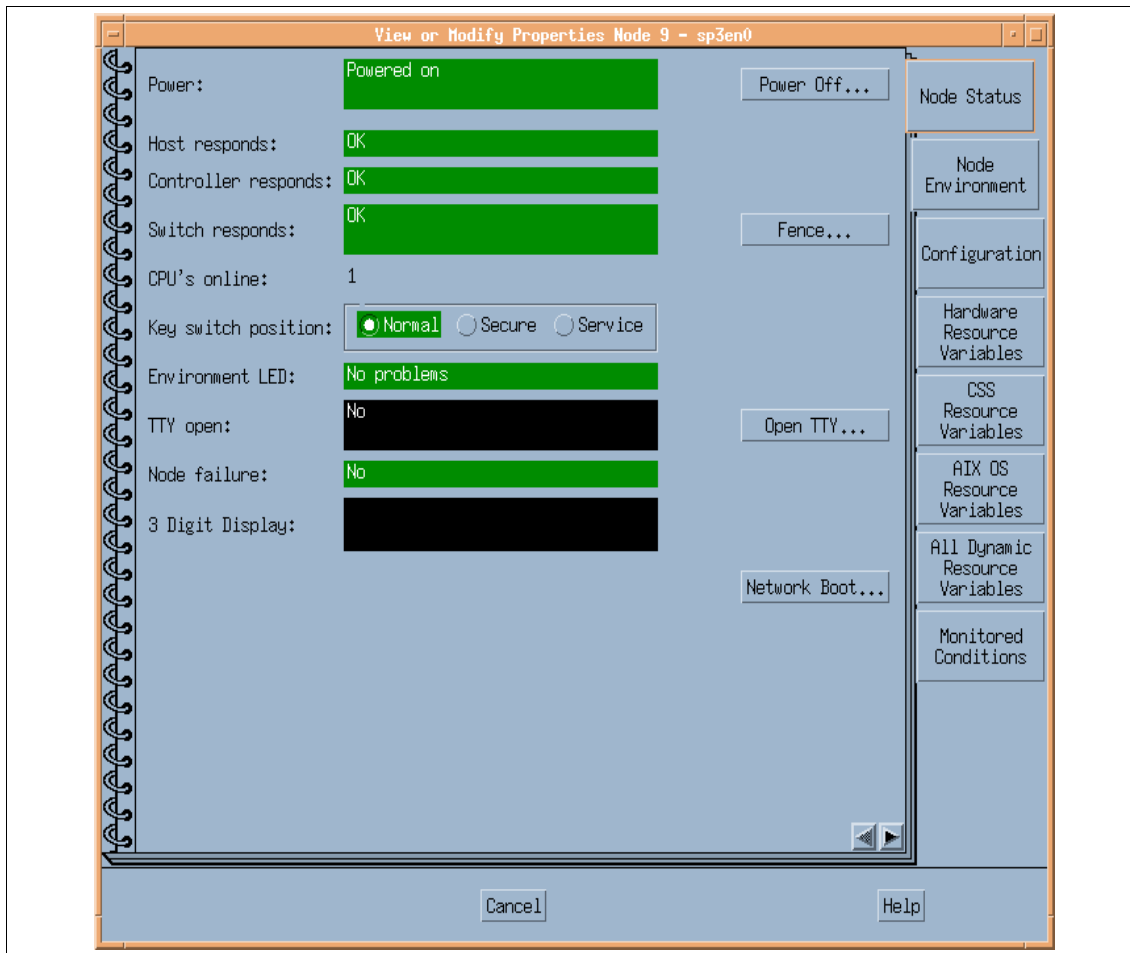


Figure 9. Perspectives Window to Network Boot a Node

### 2.1.3.3 Network Boot: Manual Conditioning

A more complicated way of network booting a node is by doing a list of activities manually. Nevertheless, this is very useful when a problem occurs in the installation process: you can follow step by step what is happening and know where the installation fails. With PSSP 3.1, a unique Perspective window shown by Figure 9 on page 31 allows you to do all the necessary steps as follows:

Start Perspectives (by issuing the command `perspectives &`)

Select **Hardware Perspective**

Select the node in the Node's Pane (Figure 10 on page 33 shows the SP Perspective window that you are in and the result is the Perspective window shown by Figure 9 on page 31).

**Actions ---> Change Key Switch**

**Actions ---> LCD and LED Display**

**Actions ---> Power off the node**

**Actions ---> Power on the node**

When the LED for node stops at 200, set the key to service:

**Actions ---> Change Key Switch ---> Service ---> Apply**

**Actions ---> Power/off Reset ---> Reset ---> Apply**

**Actions ---> Open TTY**

Press Enter in the TTY window.

A menu appears in the TTY window; select **Boot Startup**

Select the Ethernet choice according to your SP Ethernet cabling.

Verify if the IP addresses are correct for the Client and Server. If not, select each one and fill in all 0s (000.000.000.000) or the correct IP address for each.

Press 99 to return.

Select **Start System Reboot**

**Actions ---> Change Key Switch ---> Normal ---> Apply**

Return to the TTY window and press Enter to activate.

The LED displays the progress of the Install Base Operating System.

**Note**

This example of Manual Conditioning is only relevant to UP nodes. Some differences exist when using Manual Conditioning on 332 MHz SMP nodes and High Nodes.

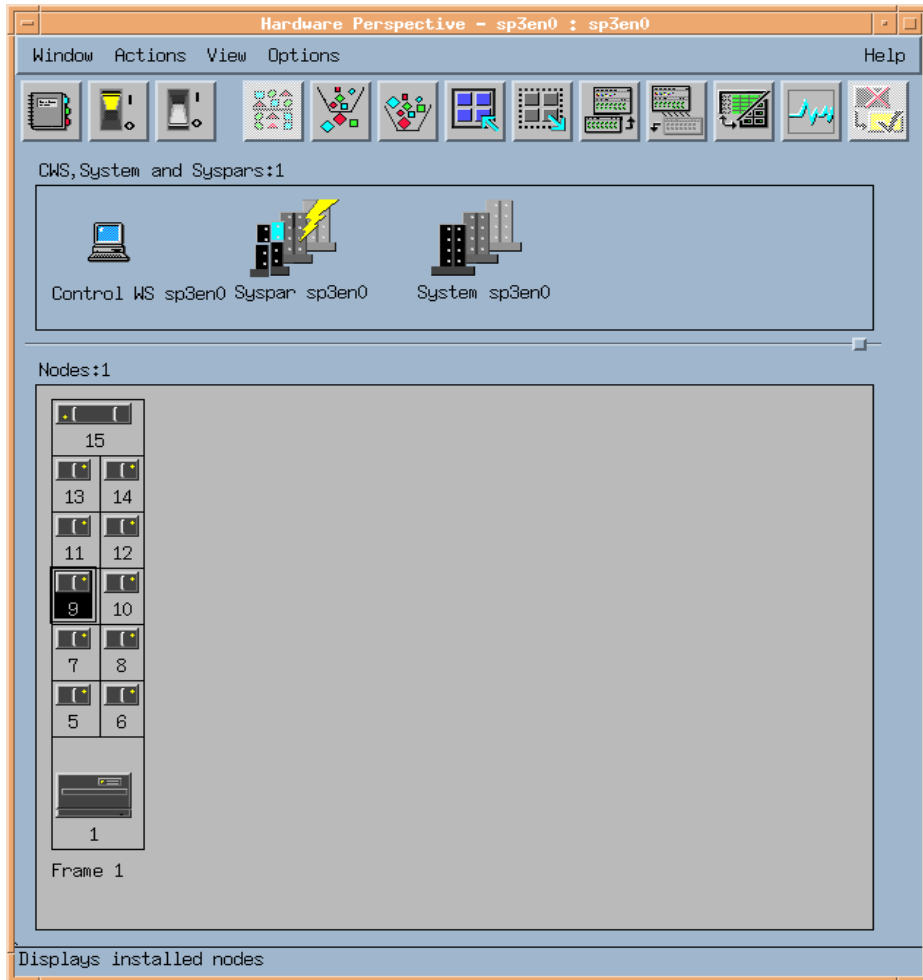


Figure 10. Hardware Perspective to Monitor a Node

### 2.1.4 New in PSSP 3.1: More Details

You know now the new concepts included in PSSP 3.1 to accomplish the installation such as multiple rootvg, mirroring rootvg volume groups, and setting the node bootlist.

This sections provides details about the new packaging and what is changed in SP security, and also gives information on how to use these concepts.

### 2.1.4.1 New Packaging

Figure 11 shows the filesets included in the main installable image (pssp.installp).

<b>ssp</b>		
ssp.ha_topsvcs.compat	3.1.0.0	Compatability for ssp.ha and ssp.topsvcs clients
ssp.st	3.1.0.0	Job Switch Resource Table Services Package
ssp.sysman	3.1.0.0	Optional System Management programs
ssp.public	3.1.0.0	Public Code Compressed Tarfiles
ssp.clients	3.1.0.0	SP Authenticated Client Commands
ssp.authent	3.1.0.0	SP Authentication Server
ssp.css	3.1.0.0	SP Communication Subsystem Package
ssp.top	3.1.0.0	SP Communication Subsystem Topology Package
ssp.spmgr	3.1.0.0	SP Extension Node SNMP Manager
ssp.jm	3.1.0.0	SP Job Manager Package
ssp.perlpkg	3.1.0.0	SP PERL Distribution Package
ssp.help.ma_RP.gui	3.1.0.0	SP Perspectives GUI Help Information
ssp.pman	3.1.0.0	SP Problem Management
ssp.ucode	3.1.0.0	SP Supervisor Microcode Package
ssp.sysctl	3.1.0.0	SP Sysctl Package
ssp.gui	3.1.0.0	SP System Monitor Graphical User Interface
ssp.top.gui	3.1.0.0	SP System Partitioning Aid
ssp.basic	3.1.0.0	SP System Support Package
ssp.docs	3.1.0.0	SP man and info files
ssp.tecad	3.1.0.0	SP HA TEC Event Adapter Package

Figure 11. PSSP Filesets

The RSCT filesets are shown in Figure 12.

<b>rsct.basic</b>		
rsct.basic.hacmp	1.1.0.0RS/6000	Cluster Technology basic function (HACMP domain)
rsct.basic.sp	1.1.0.0RS/6000	Cluster Technology basic function (SP domain)
rsct.basic.rte	1.1.0.0RS/6000	Cluster Technology basic function (all domains)
<b>rsct.clients</b>		
rsct.clients.hacmp	1.1.0.0RS/6000	Cluster Technology client function (HACMP domain)
rsct.clients.sp	1.1.0.0RS/6000	Cluster Technology client function (SP domain)
rsct.clients.rte	1.1.0.0RS/6000	Cluster Technology client function (all domain)

Figure 12. RSCT Filesets

The RS/6000 Cluster technology requires the System Measurement Performance Interface (SMPI) libraries to obtain the AIX resource variables. Before AIX 4.3.2, the SPMI library was part of the perfagent.server option, which was a fileset part of the Performance Aid for AIX (PAIDE). As a result, in previous versions of PSSP there was a



prerequisite of PAIDE being installed in the system in order to install PSSP.

In AIX 4.3.2 and later the SPMI library is included in the perfagent.tool fileset which is no longer part of the PAIDE option but included as part of the base AIX software. This means that PAIDE is no longer a prerequisite for PSSP, only perfagent.tools. However, if you plan on installing the PTPE option, you will be requested to install PAIDE since it is still a prerequisite for PTPE.

Figure 13 shows additional filesets included in PSSP software.

<b>ssp.ptpegui</b>	
ssp.ptpegui	3.1.0.0 SP Performance Monitor Graphical User Interface
<b>ssp.resctr</b>	
ssp.resctr.rte	3.1.0.0 SP Resource Center
<b>ssp.vsdgui</b>	
ssp.vsdgui	3.1.0.0 VSD Graphical User Interface (Perspectives)
<b>vsd</b>	
vsd.cmi	3.1.0.0 VSD Centralized Management Interface (SMIT)
vsd.vsd	3.1.0.0 VSD Device Driver
vsd.hsd	3.1.0.0 VSD Hashed Shared Disk
vsd.sysctl	3.1.0.0 VSD sysctl commands
vsd.rvsd.hc	3.1.0.0 Recoverable VSD Connection Manager
vsd.rvsd.rvsd	3.1.0.0 Recoverable VSD Daemon
vsd.rvsd.scripts	3.1.0.0 Recoverable VSD Recovery Scripts
<b>ptpe</b>	
ptpe.program	3.1.0.0 Performance Toolbox Parallel Extensions Program
ptpe.docs	3.1.0.0 Performance Toolbox Parallel Extensions References
<b>ssp.hacws</b>	
ssp.hacws	3.1.0.0 SP High Availability Control Workstation

Figure 13. Additional Filesets Included in PSSP Software

For details about the Recoverable Virtual Shared Disks (RVSD) filesets refer to Chapter 7, “Recoverable/Virtual Shared Disk 3.1” on page 219.

For details about the different graphical interfaces available in PSSP, refer to Chapter 3, “SP Perspectives” on page 71.

#### 2.1.4.2 Security

AIX 4.3.1 and later releases provide remote commands that can be customized to call various authentication methods including Kerberos 4 (K4). Thus the remote commands (`rsh` and `rcp`) have been removed from PSSP, and the standard AIX commands are now used for system management on the RS/6000 SP. To configure a DCE cell in which Kerberos 5 (K5) can work, DCE has to be installed. In the same way to configure a realm where K4 can work, the PSSP authentication fileset has

to be installed. The later is still mandatory in RS/6000 SP as it is used by some components in the RS/6000 SP, such as the hardmon daemon and the sysctl facility.

Figure 14 shows the remote shell (rsh) structure before PSSP 3.1. In previous releases of PSSP, all the user-issued remote commands were based either on the standard AIX `rsh` command, or the PSSP-supplied `rsh` command which uses Kerberos V4 as its authentication and authorization mechanism. All the PSSP-related commands were based on the PSSP-supplied `rsh` command.

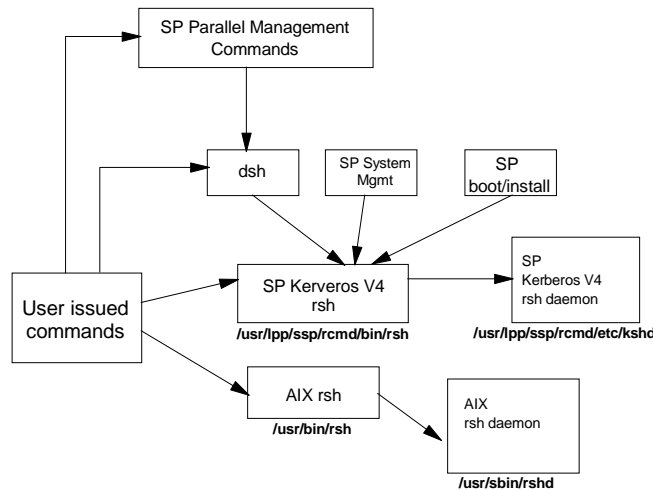


Figure 14. Remote Shell Structure Before PSSP 3.1

In PSSP 3.1, and with the support of AIX 4.3.1 and later, the remote shell structure has changed, as shown in Figure 15 on page 37.

In this new structure, PSSP no longer provides special versions of remote commands (`rsh` and `rcp` using Kerberos V4), but the AIX operating system does. However, in order to keep backward compatibility with previous versions of PSSP, there is a link from the original PSSP location for these commands to the standard AIX version of the `rsh` command.

PSSP commands will continue calling the PSSP `rsh`, which is now linked to the AIX `rsh` command. The `rsh` and `rcp` commands in AIX can be configured to use multiple authentication and authorization methods.

There are three authentication services that you can use on your SP system:

- Kerberos V4

- Standard AIX
- Kerberos V5 used by IBM AIX Distributed Computing Environment (DCE)

PSSP 3.1 requires Kerberos V4 to be configured on the control workstation and nodes, although you may install additional authentication services, such as Kerberos V5 provided by DCE.

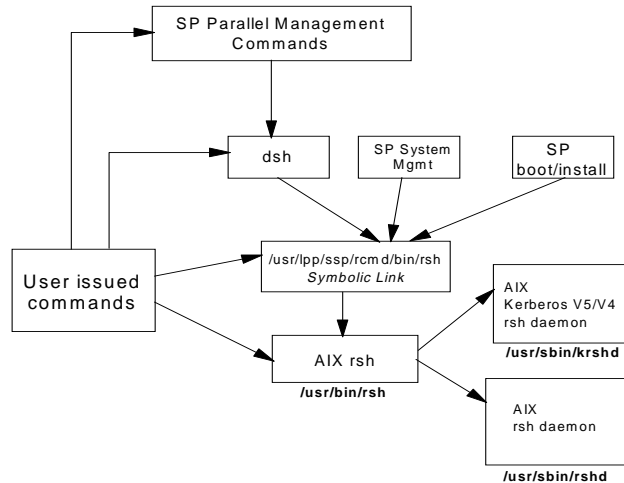


Figure 15. Remote Shell Structure in PSSP 3.1 or Later

If you decide to use Kerberos V5 from DCE as one of the authentication mechanisms, you will need to install DCE client code on the nodes, and the DCE server options (DCE Cell Directory Services and DCE Security) somewhere in the system, either on the control workstation or off the SP, but not on an SP node. Time synchronization provided by the PSSP is optional, but you have to make sure that time is kept synchronized between the nodes and the control workstation.

For PSSP 3.1, you must install and configure Kerberos V4 for all partitions within the SP system. In addition to Kerberos V4, you may also select to install and configure DCE for each system partition.

You can enable authentication methods for a system partition, but you cannot enable authentication methods for individual nodes or the control

workstation. All nodes within a single partition will have the same set of authentication methods enabled. The authentication methods enabled on the control workstation will be the union of all authentication methods enabled for all system partitions.

There are restrictions on the set of authentication methods that may be enabled for a partition containing releases of AIX previous to AIX 4.3.1. There are no restrictions on the setting of authentication methods between different partitions within a single SP system; different partitions may have different sets of authentication methods enabled.

In order to use the AIX authenticated remote commands within the SP and allow the system use of `rsh` and `rcp` commands, select:

1. The type of authorization to use for root user access using the authenticated remote commands (`rcmds`) within each system partition. The options are:
  - Kerberos V4 (required)
  - Standard AIX
2. The authentication methods to enable for each system partition. The options are:
  - Kerberos V5
  - Kerberos V4 (required)
  - Standard AIX

**Note**

If you do not enable K5 or standard AIX, FTP, Telnet and rlogin will not work.

The use of Kerberos V4 authentication and Kerberos V4 authorization for root user `rsh` is required in PSSP 3.1. Although the `rsh` and `rcp` commands are capable of supporting other authentication methods, the `sysctl` and `hardmon` services require the use of Kerberos V4.

When determining which authentication method to use for remote commands, AIX examines the order of precedence set by the AIX `chauthent` command. This order determines which authentication method is used when the remote commands are issued. This means that if the first method fails authorization, the second method is tried, and so on. The

order of precedence, defined as being from the highest to the lowest level of security, is DCE (if installed), Kerberos V4, and Standard AIX.

Since the authentication method is set per partition, nodes get the information about the authentication settings from the SDR. At boot time, the rc.sp script will call the `spauthconfig` script to set up the right authentication method for that node. You should not use the AIX `chauthent` command, since any settings with this command will be overridden in the boot.

The `spauthconfig` script will check which authorization method is to be set up via the `auth_root_rcmd` attribute in the Syspar class, and call the `updauthfiles` script as appropriate.

Finally, it will check what authentication method is to be enabled via the `auth_methods` attribute in the Syspar class.

Figure 16 shows a partial list of attributes from the Syspar SDR class.

```
[sp4en0]# SDRGetObjects Syspar auth_install auth_root_rcmd auth_methods
auth_install auth_root_rcmd auth_methods
k4          k4          k4:std
```

Figure 16. Partial List of Attributes from the Syspar SDR Class

### The Two Steps

PSSP 3.1 provides a new set of SMIT panels for configuring the authorization methods and enabling the authentication services. You can access the main security panel by using the SMIT fastpath command `smitty spauth_config`. The main panel is shown in Figure 17.

```
RS/6000 SP Security

Move cursor to desired item and press Enter.

Select Authorization Methods
Enable Authentication Methods

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

Figure 17. Main SMIT SP Security Panel

This can be summarized in two steps:

1. Create the authorization files (.rhosts for Standard AIX) using the SMIT fastpath command `smitty spauth_rcmd` or the PSSP command `spsetauth`. These files contain hostnames of nodes which are allowed root access using the remote commands.

**Note**

The .klogin files for Kerberos V4 are transferred from the control workstation during the customization phase. The .klogin file on the control workstation is created by the `setup_authent` script.

Figure 18 shows the SMIT panel that allows you to set the authorization methods for root access and create the corresponding authorization files.

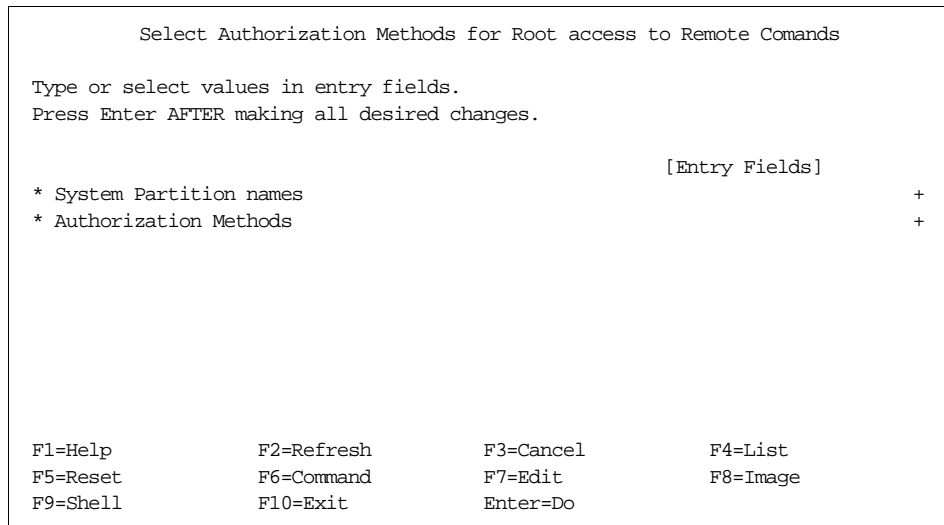


Figure 18. SMIT Panel for Selecting Authorization Methods for Root Access

2. Enable the selected authentication methods by using the SMIT fastpath command, `smitty spauth_methods`, or the PSSP command `chauthpar`. This step updates the SDR attribute `auth_methods` in the SDR Syspar class and enables these authentication methods in addition to any method already set. Figure 19 on page 41 shows the SMIT panel for this step. The change made to the SDR will be picked up by the nodes in the next reboot or customization. The `force` option may be used to do the changes on the nodes immediately.

```

Enable Authentication Methods

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

                                     [Entry Fields]
Enable on Control Workstation Only      no      +
Force change on nodes                   no      +
* System Partition names                +
* Authentication Methods                 +

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

```

Figure 19. SMIT Panel for Enabling Authentication Methods

For more information about RS/6000 SP security, refer to the chapter entitled "Security Features on the SP" in the *Parallel System Support Programs for AIX: Administration Guide, SA72-7348*.

### 2.1.4.3 Some Definitions

As mentioned, new concepts like multiple rootvg, mirroring rootvg, and changing the node bootlist are now available with PSSP 3.1.

Before going into detail about each of these, some definitions or explanations on what is behind them would be useful.

The concept called "Multiple Rootvg" or "Alternate Root Volume Group" provides the ability to boot a separate Volume Group on a node. To do that, a new SDR class called Volume\_Group has been created to store the data. These additional Volume Groups allow booting of a separate version of the operating system on the node. Obviously, before using this alternative, you must do as many installations as you need. Each installation uses a different Volume\_Group name created at the SDR level.

Although the name of these Volume Groups must be different in the SDR because they are different objects in the same class (the first one can be rootvg and the following othervg, for example), this name stays in the SDR and is not used directly by NIM to install the node. Only the attribute Destination Disks is used to create the rootvg node Volume Group.

If your node has two (or more) available rootvgs, only one is used to boot: it is determined by the bootlist of the node. Because the user determines which version of the operating system to boot, another concept appears

with PSSP 3.1: the possibility to change the bootlist of a node directly from the CWS by using the new command `spbootlist` (see 2.1.5.8, “New Command: `spbootlist`” on page 52).

Another enhancement in PSSP 3.1 is the possibility of mirroring the Root Volume Group directly from the CWS. Mirroring is writing simultaneous copies of the operating system logical volumes to provide redundancy. Either two or three copies (one or two mirrors) are allowed in AIX.

The operating system determines which copy of each operating system logical volume is active based on availability.

Prior to PSSP 3.1, on the RS/6000 SP, attributes such as operating system level, PSSP level, installation time and date were associated with the Node object in the SDR.

Now, with PSSP 3.1, these attributes are more correctly associated with a Volume Group: a node is not at AIX 4.3.2, for example; a Volume Group of the node is at 4.3.2. To display this information, a new option (-v) has been added in the `splstdata` command.

So, part of this feature is to break the connection between nodes and attributes more properly belonging to a Volume Group. For this reason, some information has been moved from the SMIT panel Boot/Install Server Information to Create Volume Group Information or Change Volume Group Information panel.

So, after these definitions, we now describe these enhancements and the related commands in more detail.

#### **2.1.4.4 SDR Changes on the Volume\_Group Class**

As explained, a new `Volume_Group` class has been created. The following lists its attributes:

- `node_number`
- `vg_name` (Volume Group name)
- `pv_list` (one or more physical volumes)
- `quorum` (quorum is true or false)
- `copies` (1, 2, or 3)
- `install_image` (name of the `mksysb`)
- `code_version` (PSSP level)
- `lppsource_name` (which `lppsource`)
- `boot_server` (which node serves this Volume Group)



- last\_install\_time (time of last install of this Volume Group)
- last\_install\_image (last mksysb installed on Volume Group)
- last\_bootdisk (which physical volume to boot from)

The attributes pv\_list, install\_image, code\_version, lppsource\_name, boot\_server have been duplicated from the Node class to the Volume\_Group class. New SMIT panels associated with these changes are detailed in the following sections.

#### 2.1.4.5 SDR Changes on the Node Object

The new Volume\_Group class uses some attributes from the old Node class. The following list describes the changes made to the Node Object:

- A new attribute is created: selected\_vg.
- selected\_vg points to the current Volume\_Group object.
- The Node object retains all attributes.
- Now the Node attributes common to the Volume\_Group object reflect the current Volume Group of the node.
- The Volume\_Group objects associated with a node reflect all the possible Volume Group states of the node.

#### Note

All applications using the Node Object remain unchanged, with the exception of some SP installation code.

#### 2.1.4.6 Volume\_Group Default Values

When the SDR is initialized, a Volume\_Group object for every node is created.

By default, the vg\_name attribute of the Volume\_Group object is set to rootvg and the selected\_vg of the Node object is set to rootvg.

The following are the other default values:

- The default install\_disk is hdisk0.
- Quorum is true.
- Mirroring is off, copies set to 1.
- There are no bootable alternate root Volume Groups.
- All other attributes of the Volume\_Group are initialized according to the same rules as the Node object.

## 2.1.5 New in PSSP 3.1: The Commands

After describing the improved PSSP 3.1 installation process, let us now go to the new commands used to create, change, delete, mirror, and unmirror Volume\_Group objects. Also, changes to existing commands are described.

### 2.1.5.1 New Volume\_Group Command: `spmkvgobj`

All information needed by NIM, such as `lppsource`, physical disk, server, `mksysb`, and so forth, is now moved from Boot/Install server Information to a new panel accessible by the fast path `createvg_dialog` as shown by Figure 20.

```

                                Create Volume Group Information

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

                                [Entry Fields]
Start Frame                      [] #
Start Slot                       [] #
Node Count                       [] #

OR

Node List                        [10]

Volume Group Name                 [rootvg]
Physical Volume List              [hdisk0,hdisk1]
Number of Copies of Volume Group  1 +
Boot/Install Server Node         [0] #
Network Install Image Name       [bos.obj.mksysb.aix432.090898]
LPP Source Name                  [aix432]
PSSP Code Version                PSSP-3.1 +
Set Quorum on the Node                               +

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

```

Figure 20. New SMIT Panel to Create a Volume Group

The associated command of this SMIT panel is `spmkvgobj`, whose options are:

```
-r vg_name
-l node_list
-h pv_list
-i install_image
-v lppsource_name
-p code_version
-n boot_server
```

```
-q quorum
-c copies
```

The following command built by the previous SMIT panel is a good example of the use of `spmkvgobj`:

```
/usr/lpp/ssp/bin/spmkvgobj -l '10' -r 'rootvg' -h 'hdisk0,hdisk1' -n
'0' -i 'bos. obj.mkysyb.aix432.090898' -v 'aix432' -p 'PSSP-3.1'
```

More information about the `-h` option: for PSSP levels prior the PSSP 3.1, two formats were supported to specify the SCSI disk drive and are always usable:

- Hardware location format  
00-00-00-0,0 to specify a single SCSI disk drive  
or 00-00-00-0,0:00-00-00-1,0 to specify multiple hardware locations (in that case, colon is the separator).
- Device name format  
hdisk0 to specify a single SCSI disk drive  
or hdisk0,hdisk1 to specify multiple hardware locations (in that case, comma is the separator).  
  
You must not use this format when specifying an external disk because the relative location of hdisks can change depending on what hardware is currently installed. It is possible to overwrite valuable data by accident.

A third format is now supported to be able to boot on SSA external disks: a combination of the parent and connwhere attributes for SSA disks from the Object Data Management (ODM) CuDv. In the case of SSA disks, the parent always equals `ssar`. The connwhere value is the 15-character unique serial number of the SSA drive (the last three digits are always 00D for a disk). This value is appended as a suffix to the last 12 digits of the disk ID stamped on the side of the drive. If the disk drive has already been defined, the unique identity may be determined using SMIT panels, or by following these two steps:

- Issue the command:  

```
lsdev -Ccpdisk -r connwhere
```
- Select the 15-character unique identifier for which characters 5 to 12 match those on the front of the disk drive.

For example, to specify the parent-connwhere attribute, you can enter:

ssar//0123456789AB00D

Or, to specify multiple disks, separate using colons, as follows:

ssar//0123456789AB00D:ssar//0123456789FG00D

**Important**

The ssar identifier must have a length of 21 characters.

Installation on SSA disks is now supported. For more information, refer to 2.4.2, "Booting from External Disks" on page 65.

### 2.1.5.2 New Volume\_Group Command: spchvgobj

After a Volume\_Group has been created by the `spmkvgobj` command, you may want to change some information: use the `spchvgobj` command or the new SMIT panel (fastpath is `changevg_dialog`) shown by Figure 21.

This command uses the same options as the `spmkvgobj` command. The following is an example built by the SMIT panel:

```
/usr/lpp/ssp/bin/spchvgobj -l '1' -r 'rootvg' -h  
'hdisk0,hdisk1,hdisk2' -c '2' -p 'PSSP-3.1'
```

```
Change Volume Group Information  
  
Type or select values in entry fields.  
Press Enter AFTER making all desired changes.  
  
[Entry Fields]  
Start Frame          [] #  
Start Slot          [] #  
Node Count          [] #  
  
OR  
  
Node List           [1]  
  
Volume Group Name   [rootvg]  
Physical Volume List [hdisk0,hdisk1,hdisk2]  
Number of Copies of Volume Group 2 +  
Set Quorum on the Node +  
Boot/Install Server Node [] #  
Network Install Image Name []  
LPP Source Name     []  
PSSP Code Version   PSSP-3.1 +  
  
F1=Help             F2=Refresh          F3=Cancel           F4=List  
F5=Reset            F6=Command          F7=Edit             F8=Image  
F9=Shell            F10=Exit            Enter=Do
```

Figure 21. New SMIT Panel to Modify a Volume Group

**Note**

To verify the content of the Volume\_Group class of node 1, you can issue the following SDR command:

```
SDRGetObjects Volume_Group node_number==1 vg_name pv_list copies
```

**2.1.5.3 New Volume\_Group Command: sprmvobj**

To be able to manage the Volume\_Group class, a third command to remove a Volume\_Group object that is not the current one has been added: `sprmvobj`.

This command accepts the following options:

```
-r vg_name  
-l node_list
```

Regarding SMIT: the Delete Database Information SMIT panel (fastpath is `delete_data`; Figure 7 on page 26) has been changed to access the new SMIT panel named Delete Volume Group Information (fastpath is `deletevg_dialog`).

Refer to Figure 22 for details.

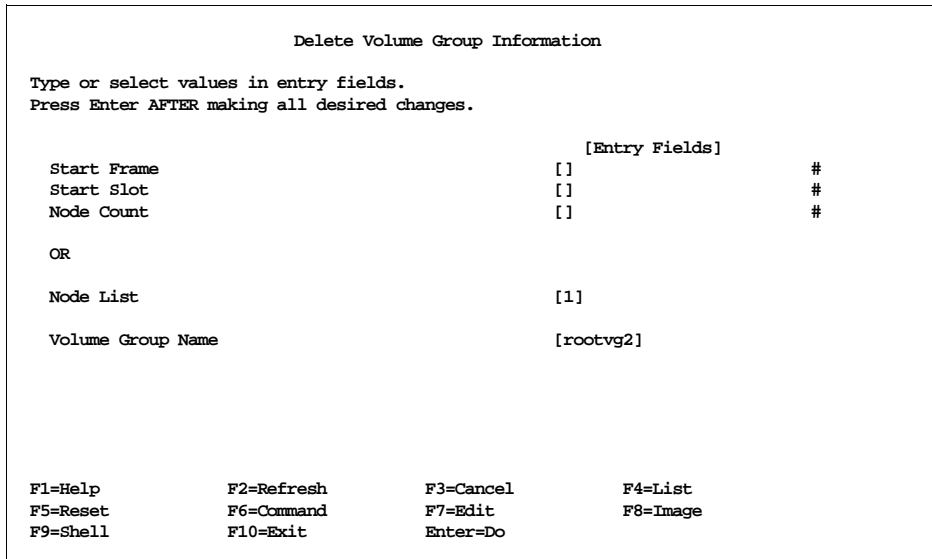


Figure 22. New SMIT Panel to Delete a Volume Group

The following is an example built by the SMIT panel used in Figure 22:

```
/usr/lpp/ssp/bin/sprmvobj -l '1' -r 'rootvg2'
```

#### 2.1.5.4 Command Changes for spbootins

`spbootins` is the command to set various node attributes in the SDR (code\_version, lppsource\_name, and so forth).

By using the `spbootins` command now, you can select a Volume Group from all the possible Volume Groups for this node in the Volume\_Group class.

Attributes shared between the Node and Volume\_Group objects are changed using a new set of Volume\_Group commands, not using `spbootins`.

The new `spbootins` is as follows:

```
spbootins
```

```
-r <install|diag|maintenance|migrate|disk|customize>  
-l <node_list>  
-c <selected_vg>  
-s <yes|no>
```

`spbootins` no longer have the following flags:

```
-h <install_disk>  
-n <boot_server>  
-v <lppsource_name>  
-i <install_image_name>  
-p <PSSP_level>  
-u <usr_server_id>  
-g <usr_gateway_id>  
-a <interface name>
```

#### Note

-u, -g and -a flags were dropped because PSSP 3.1 no longer support /usr servers.

Figure 23 on page 49 shows the new SMIT panel to issue `spbootins` (fastpath is `server_dialog`).

```

                                Boot/Install Server Information

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

                                [Entry Fields]
Start Frame                      [] #
Start Slot                       [] #
Node Count                       [] #

OR

Node List                        [10]

Response from Server to bootp Request      install +
Volume Group Name                         [rootvg]
Run setup_server?                         yes +

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

```

Figure 23. New SMIT Panel to Issue the `spbootins` Command

You get the same result by issuing the following from the command line:

```
spbootins -l 10 -r install -c rootvg -s yes
```

Note that the value `yes` is the default for the `-s` option; in that case, the script `setup_server` is run automatically.

### 2.1.5.5 New Volume\_Group Command: `spmirrorvg`

This command enables mirroring on a set of nodes given by the option

```
-l node_list
```

You can force (or not force) the extension of the Volume Group by using the `-f` option (available values are: `yes` or `no`).

This command takes the Volume Group information from the SDR updated by the last `spchvgobj` and `spbootins` commands.

Note:

You can add a new physical volume to the node `rootvg` by using the `spmirrorvg` command; the following steps give the detail:

- Add a physical disk to the actual `rootvg` in the SDR by using `spchvgobj` without changing the number of copies.
- Run `spmirrorvg`

Figure 24 on page 50 shows the new SMIT panel to issue `spmirrorvg` (fastpath is `start_mirroring`).

```

Initiate Mirroring on a Node

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

                                [Entry Fields]
Start Frame                      []          #
Start Slot                       []          #
Node Count                       []          #

OR

Node List                        [1]
Force Extending the Volume Group? no          +

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

```

Figure 24. New SMIT Panel to Initiate the `spmirrorvg` Command

The following is an example built by the SMIT panel used in Figure 24:

```
/usr/lpp/ssp/bin/spmirrorvg -l '1'
```

For more detail regarding the implementation of mirroring root volume groups, refer to the manual *Parallel System Support Programs for AIX: Administration Guide*, SA72-7348 Appendix B.

**Note**

This command uses the `dsh` command to run the AIX-related commands on the nodes.

**2.1.5.6 New Volume\_Group Command: `spunmirrorvg`**

This command disables mirroring on a set of nodes given by the option

```
-l node_list
```

Figure 25 on page 51 shows the new SMIT panel to issue `spunmirrorvg` (fastpath is `stop_mirroring`).



```

                                Discontinue Mirroring on a Node

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

                                [Entry Fields]
Start Frame                      []                #
Start Slot                       []                #
Node Count                       []                #

OR

Node List                        []

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

```

Figure 25. New SMIT Panel to Initiate the `spunmirrorvg` Command

The following is the example built by the SMIT panel used in Figure 25:

```
/usr/lpp/ssp/bin/spunmirrorvg -l '1'
```

**Note**

This command uses the `dsh` command to run the AIX related commands on the nodes.

**2.1.5.7 Command Changes for `splstdata`**

`splstdata` can now display information about Volume\_Groups using the new option

```
-v
```

Figure 26 on page 52 shows the information related to node 1 in the result of the command `splstdata -v -l 1`.

List Volume Group Information						
node#	name	boot_server	quorum	copies	code_version	lppsource_name
	last_install_image			last_install_time	last_bootdisk	
	pv_list					
1	rootvg	0	true	1	PSSP-3.1	aix432
	default			Thu_Sep_24_16:47:50_EDT_1998	hdisk0	
	hdisk0					
1	rootvg2	0	true	1	PSSP-3.1	aix432
	default			Fri_Sep_25_09:16:44_EDT_1998	hdisk3	
	ssar//0004AC50532100D:ssar//0004AC50616A00D					
1	jmbvg	0	true	1	PSSP-3.1	aix432
	default			Fri_Sep_25_11:50:47_EDT_1998	hdisk0	
	ssar//0004AC5150BA00D					

Figure 26. Example of `splstdata -v`

### 2.1.5.8 New Command: `spbootlist`

`spbootlist` sets the bootlist on a set of nodes by using the option

`-l node_list`

This command takes the Volume Group information from the SDR updated by the last `spchvgobj` and `spbootins` commands.

Section 2.4.1, “Multiple rootvg Support” on page 63 gives information on how to use this new command.

### 2.1.5.9 AutoJoin Automatic:

All nodes that are not explicitly fenced with the `Efence` command will rejoin the switch when they are powered on or rebooted. 5.1, “Automatic Node Unfence” on page 161 gives more details.

---

## 2.2 Migration

After some definitions or other general considerations about migration (see Figure 27 on page 53), this chapter describes what are the main changes driven by PSSP 3.1 when you migrate your system to PSSP 3.1 and to your target level of AIX.

The migration itself is divided into two parts:

- One related to the CWS
- Another related to the nodes

Because migration of your CWS, your nodes, or both, is a complex task, you must do careful planning before you attempt to migrate. Thus, a full migration plan involves breaking your migration tasks down into distinct, verifiable (and recoverable) steps and planning the requirements for each step. A well-planned migration has the added benefit of minimizing system downtime.

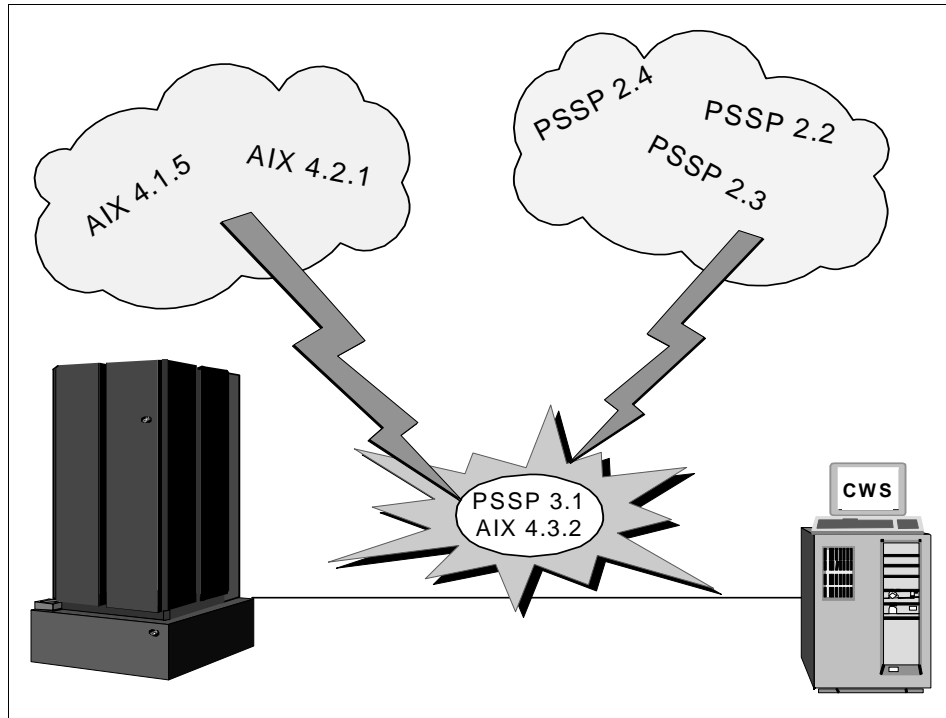


Figure 27. Migration Considerations for PSSP 3.1

### 2.2.1 Definitions - Overview - Limitations

An AIX level is made of three parts: VV.RR.MM

V for version

R for release

M for modification

A version of AIX is composed by the two first parts.

A migration is an evolution to a newer version: 4.2 to 4.3, for example.

An update is when the change occurs only on the third part.

For PSSP, all level changes are updates.

In PSSP 3.1, the only supported paths are shown in Table 6.

Table 6. Supported Migration Paths to PSSP 3.1

From		To	
PSSP Level	AIX Level	PSSP Level	AIX Level
2.2	4.1.5 4.2.1	3.1	4.3.2
2.3	4.2.1 4.3.2	3.1	4.3.2
2.4	4.2.1 4.3.2	3.1	4.3.2

If your current system, CWS, or any node is running at a PSSP or AIX level not listed in the **From** column of Table 6, you must migrate to one of the listed combinations before you can migrate to PSSP 3.1. How to actually migrate is documented in the manual *IBM Parallel System Support Install & Migration Guide Version 3 Release 1, GA22-7347*.

Here an example: your CWS is currently installed with AIX 4.2.1 and PSSP 2.3 and you want to migrate to AIX 4.3.2 and PSSP 3.1. To do that, and to minimize the service window, we suggest the following steps:

1. Migrate from AIX 4.2.1 to AIX 4.3.2 (do not change the PSSP level)
2. Verify system stability and functionality
3. Migrate from PSSP 2.3 to PSSP 3.1 (do not change AIX level)

However, even you have found your migration path, some products or components of PSSP have limitations that might restrict your ability to migrate:

- Switch Management
- RS/6000 Cluster Technology
- Performance Toolbox Parallel Extensions
- High Availability Cluster Multi-Processing
- IBM Virtual Shared Disk
- IBM Recoverable Virtual Shared Disk
- General Parallel File System
- Parallel Environment

- LoadLeveler
- Parallel Tools
- PIOFS, CLIO/S and NetTAPE
- Extension node support

For more information about these limitations, refer to Table 1 on page 19 or to the document *IBM RS/6000 SP Planning Volume 2, Control Workstation and Software Environment*, GA22-7281.

## 2.2.2 Reasons

Why migration and not installation?

Whereas the installation does a complete overwrite of your system, the migration preserves all local system changes you made before, such as:

- Users and groups: to preserve the settings for the users, like passwords, profiles, login shells.
- File systems and Volume Groups (names, parameters, sizes, directories are kept).
- RS/6000 SP setup (AMD, file collections).
- Network setup (TCP/IP, SNA).

## 2.2.3 Planning

Before migrating, you may want to create one or more system partitions. First, as an option, you can create a production system partition with your current AIX and PSSP level software and a test system partition with your target level of AIX and PSSP 3.1 level software.

Second, you may want to partition your system due to coexistence limitations. The detailed information about coexistence is given in 2.3, "Coexistence" on page 62.

Before you migrate any of your nodes, you must migrate your CWS to the latest level of AIX and PSSP of any node you wish to serve. For example, if you plan to migrate any node to AIX 4.3.2 and PSSP 3.1, the CWS must first be migrated to AIX 4.3.2 and PSSP 3.1.

Before starting the migration, make an archive of the SDR, a backup of the CWS and the nodes (those you are using for the migration).

Remember, if you are using a Boot/Install Server, this node must be at the *highest* level of AIX and PSSP which it will serve.

After these general considerations, we give now some details of the migration process at the CWS level, and then at the node level.

## 2.2.4 CWS Migration

This section briefly describes what is new in PSSP 3.1 for migrating the control workstation. For further information refer to *IBM Parallel System Support Programs for AIX: Installation and Migration Guide, GA22-7347*.

We describe the main steps in the installation process, but with the migration goal in mind. We assume the migration of the CWS to AIX 4.3.2 has been done successfully.

### 2.2.4.1 Create the Required /spdata Directory

PSSP 3.1 file sets must reside on the CWS. You have to do a change of the existing /spdata directory structure. PSSP 3.1 supports multiple levels of AIX and PSSP: a directory hierarchy was designed to contain different levels of images.

Refer to Figure 4 on page 22 to see the related directory structure.

### 2.2.4.2 Copy the AIX LPP Images and Others Required AIX LPPs

If you have not done so already, you must copy the AIX filesets into the /spdata/sys1/install/<name>/lppsource directory on your hard disk on the CWS

<name> is the name of your AIX level; for example, aix432.

One of these fileset (perfagent) has a particular role as detailed in the following section.

### 2.2.4.3 Verify Correct Level of PAIDE (perfagent)

The perfagent.server file is part of the Performance Aide for AIX (PAIDE) feature of the Performance Toolbox for AIX (PTX), which must be copied into the lppsource directory.

The perfagent.tools fileset is part of AIX 4.3.2. This product provides the capability to monitor the performance of your SP system, collects and display statistical data for SP hardware and software, and simplifies run-time performance monitoring of a large number of nodes.

PAIDE (perfagent.server) must be installed and copied to all of the lppsource directories on the CWS of any SP that has one or more nodes at PSSP 2.2 or later. The level of PAIDE installed or copied to the lppsource directory on the CWS must coordinate with the level of AIX installed on the CWS or the level of AIX in that directory.

There are four PAIDE features of PTX: one for AIX 4.1, one for AIX 4.2, one for AIX 4.3.1 and one for AIX 4.3.2. The level of perfagent required is dependent upon the level of AIX, as shown in Table 7.

Table 7. Performance Aide for AIX (PAIDE) File Sets

AIX Level	PSSP Level	Required PAIDE File Set
AIX 4.1.5	PSSP 2.2	perfagent.server 2.1.5.x
AIX 4.2.1	PSSP 2.2	perfagent.server 2.2.1.2 or greater
AIX 4.2.1	PSSP 2.3	perfagent.server 2.2.1.2 or greater
AIX 4.3.1	PSSP 2.3	perfagent.server 2.2.31.x
AIX 4.3.1	PSSP 2.4	perfagent.server 2.2.31.x
AIX 4.3.2	PSSP 2.2	perfagent.tools and perfagent.server 2.2.32.x
AIX 4.3.2	PSSP 2.3	perfagent.tools and perfagent.server 2.2.32.x
AIX 4.3.2	PSSP 2.4	perfagent.tools and perfagent.server 2.2.32.x
AIX 4.3.2	PSSP 3.1	perfagent.tools 2.2.32.x

#### 2.2.4.4 Copy the PSSP Images for PSSP 3.1

The RS/6000 SP package consists of several filesets that must be copied into the /spdata/sys1/install/pssplpp/PSSP-3.1 directory. Then you must rename the PSSP package to pssp.installp and, finally, create the .toc file. The following lists an example of the related commands:

```
bffcreate -qvx -t /spdata/sys1/install/pssplpp/PSSP-3.1 -d /dev/rmt0 all
cd /spdata/sys1/install/pssplpp/PSSP-3.1
mv ssp.usr.3.1.0.0 pssp.installp
inutoc .
```

#### 2.2.4.5 Copy an Installable Image (mkysb Format) for the Node

The target directory of this file must be:

```
/spdata/sys1/install/images
```

### 2.2.4.6 Stop Daemons on the CWS and Verify

You must stop the daemons in the order given by Table 8.

Table 8. Command to Issue to Stop the Daemons

To stop this daemon	Issue this command
Partition-sensitive daemons	<code>syspar_ctrl -G -k</code>
sysctld daemon	<code>stopsrc -s sysctld</code>
Amd	<code>/etc/amd/amq</code> (PSSP 2.2 users only) (see note)
splogd daemon	<code>stopsrc -s splogd</code>
hardmon daemon	<code>stopsrc -s hardmon</code>
sdrd daemon	<code>stopsrc -g sdr</code>

You can issue the `lssrc -a` command to verify that the daemons are no longer running on the CWS.

Note for PSSP 2.2 users:

If the `amq` command returns a response similar to:

```
/ root "root" k22n04:(pid12128)
```

```
/u toplvl /etc/amd/amd-maps/amd.u /u
```

the **Amd** daemon is running. Make sure that no processes are using any directories controlled by **Amd** (only two entries such as those previously listed appear in the `amq` output). If there is additional `amq` output indicating that directories are in use, stop all processes using those directories and then either wait five minutes to allow Amd to time out and unmount the directories, or force an unmount of the directories with the `amp -u` option.

Stop the daemon by issuing the following command:

```
kill -term <process_id>
```

where `<process_id>` is the pid value listed in the `amq` output, or which can be determined by issuing the following command:

```
ps -ef | grep amd
```

Do not use `kill -9` to stop the daemon as this will not allow the Amd daemon to properly shut down its control and may cause the file system to hang.



#### 2.2.4.7 Install PSSP on the CWS

The PSSP 3.1 file sets are packaged to be installed on top of previously supported releases. You should install all file sets available in the PSSP 3.1 package.

To properly set up the PSSP 3.1 on the CWS for the SDR, Hardmon, and other SP-related services, issue the following command:

```
install_cw
```

#### 2.2.4.8 Automounter Migration Note for PSSP 2.2 Users

Since the use of **Amd** was replaced with the AIX **automount** daemon as of PSSP 2.3, `services_config` will automatically create a new **auto.u** file from the existing `/etc/amd/amd-maps/amd.u` map file using the `mkautomap` installation script.

#### 2.2.4.9 Update the State of the Supervisor Microcode

Check which supervisors need to be updated by using SMIT panels or by issuing the `spsvrmgr` command:

```
spsvrmgr -G -r status all
```

In case of an action is required, you can update the microcode by issuing the command:

```
spsvrmgr -G -u <frame_number>:<slot_number>
```

#### 2.2.4.10 Refresh the pmand Daemons

The `pmand` daemons on the nodes need to be refreshed in order to recognize changes in the SDR. To perform this, you can issue the following command:

#### 2.2.4.11 Migrating Shared Disk

If you already use Virtual Shared Disks, you have some preparation to do. The migration of Virtual Shared Disk software (VSD or R/VSD) is detailed in 7.3, "Migration and Coexistence Considerations" on page 222.

### 2.2.5 Node Migration

You cannot migrate the nodes until you have migrated the CWS to your target AIX level (4.3.2) and PSSP 3.1.

You can migrate the nodes to your AIX level and PSSP 3.1 in one of three ways:

1. Migration Install

This method preserves all the file systems except /tmp, as well as the root Volume Group, logical volumes, and system configuration files. This method requires the setup of AIX NIM on the new PSSP 3.1 CWS. This applies only to migrations when an AIX version or release is changing.

See 2.2.5.1, "Migration Install of Nodes to PSSP 3.1" on page 60.

## 2. mksysb Install

This method erases all existence of current rootvg and installs your target AIX level and PSSP 3.1 using an AIX 4.3.2 mksysb image for the node. This installation requires the setup of AIX NIM on the new PSSP 3.1 CWS.

See 2.2.5.2, "mksysb Install of Nodes" on page 61.

## 3. Upgrade

This method preserves all occurrences of the current rootvg and installs AIX PTF updates using the `installp` command. This method applies to AIX modification level changes or when the AIX level is not changing, but you are migrating to a new level of PSSP.

See 2.2.5.3, "Migrate or Upgrade Nodes to AIX 4.2.1 or Later" on page 61.

To identify the appropriate method, you must use the table named Paths to Migrate the Nodes in the document *IBM Parallel System Support Install & Migration Guide Version 3 Release 1, GA22-7347*.

Although the way to migrate a node has not changed with PSSP 3.1, we point out here how the PSSP 3.1 enhancements can be used when you want to migrate.

### 2.2.5.1 Migration Install of Nodes to PSSP 3.1

The `bootp_response` parameter related to the node you want to migrate must be set to the value "migrate". To do that, the new PSSP 3.1 commands (`spchvgobj`, `spbootins`) are used as shown by the following example:

To migrate nodes 1 and 2 to AIX 4.3.2 and PSSP 3.1, issue the following commands:

```
spchvgobj -r rootvg -p PSSP-3.1 -v aix432 -l 1,2
spbootins -s no -r migrate -l 1,2
```

In this example, we assume the `lppsource` name directory is `/spdata/sys1/install/aix432/lppsource`.

The SDR is now updated. After a safe verification with the command

```
splstdata -G -b -l <node_list>
```

the information is transferred to the NIM database by running the script `setup_serverf`.

In case of using a node as boot/install server, you must also run the `setup_server` script at this node.

Finally, a shutdown followed by a network boot will migrate the node: the AIX part will be done by NIM, whereas the script `pssp_script` does the PSSP part.

### 2.2.5.2 mksysb Install of Nodes

In fact, this is a node installation that has been described in section 2.1.3, "Installation" on page 29.

### 2.2.5.3 Migrate or Upgrade Nodes to AIX 4.2.1 or Later

This method is used primarily when you need to:

- Migrate to a new level of PSSP without changing AIX levels.
- Migrate to a new level of PSSP and upgrade to a new modification level of AIX (for example, if you are on AIX 4.3.1 and PSSP 2.4 and you want to go to AIX 4.3.2 and PSSP 3.1).

We describe here only the second type of migration (the first one is included in the second).

You must first update the AIX level of the node by mounting the AIX 432 `lppsource` file system from the CWS on your node and running the `installp` command.

Then, after you have the right AIX level installed on your node, you must use the value "customize" of the `bootp_response` parameter. To do that, the new PSSP 3.1 commands (`spchvgobj`, `spbootins`) are used as shown by the following example.

To migrate nodes 1 and 2 to AIX 4.3.2 and PSSP 3.1, where `lppsource` was placed in `/spdata/sys1/install/aix432/lppsource`, issue the following commands:

```
spchvgobj -r rootvg -p PSSP-3.1 -v aix432 -l 1,2  
spbootins -s no -r customize -l 1,2
```

After verification that the SDR is now updated, you can run the `setup_server` command to have the NIM database updated also.

The value "customize" of the bootp\_response parameter is used by the script pssp\_script. In a stable node configuration, you only need to run pssp\_script on the node to have your customization done. But, at this step, the version of this script is not that related to PSSP 3.1. Therefore, before executing pssp\_script, you must first copy the one related to PSSP 3.1 into the node. The following command copies the pssp\_script file from the CWS to the node:

```
pcp -w <node> /spdata/sys1/install/pssp/pssp_script \
/tmp/pssp_script
```

After the copy is done, execute the pssp\_script which updates the PSSP 3.1 node environment.

---

## 2.3 Coexistence

PSSP 3.1 can coexist with PSSP version 2.2 and later.

### 2.3.1 Definition

Coexistence is the ability to have multiple levels of AIX and PSSP in the same partition.

**Note**

An unpartitioned system is viewed as having one default partition.

Table 9 shows what AIX levels and PSSP levels are supported by PSSP 3.1 in the same partition. Any combination of PSSP levels listed in this table can coexist in a system partition. So, you can migrate to a new level of PSSP or AIX one node at a time.

*Table 9. Possible AIX or PSSP Combinations in a Partition*

AIX Levels	PSSP Levels
AIX 4.1.5 or AIX 4.2.1	PSSP 2.2
AIX 4.2.1 or AIX 4.3.2	PSSP 2.3
AIX 4.2.1 or AIX 4.3.2	PSSP 2.4
AIX 4.3.2	PSSP 3.1

## 2.3.2 Limitations

Some PSSP components and related LPPs still have some limitations. Also, many software products have PSSP and AIX dependencies.

The products or components of PSSP that can limit your coexistence are the same as for a migration. Therefore, for details, refer to 2.2.1, “Definitions - Overview - Limitations” on page 53.

---

## 2.4 New Features

This section gives other details on how to use the PSSP 3.1 enhancements, like multiple rootvg, and use of external disks to boot a node.

### 2.4.1 Multiple rootvg Support

#### 2.4.1.1 Definition

PSSP 3.1 supports Multiple Rootvg: you can now have various rootvg's on your nodes at different AIX levels and choose those to be used at the next reboot.

#### 2.4.1.2 How to Declare a New rootvg

Several steps must be done in the right order; they are the same as for an installation. The only difference is that you must enter an unused Volume Group name.

The related SMIT panel or commands are given by Figure 20 on page 44 and Figure 23 on page 49.

At this point, the new Volume Group is declared but it is not usable. You must now install it using a Network Boot, for example.

#### 2.4.1.3 How to Activate a New rootvg

Several rootvg's are available on your node. To activate one of them, the bootlist has to be changed by using the `spbootlist` command or the related SMIT panel (the fastpath is `bootlist_dialog`) as shown in Figure 28 on page 64. Because the `spbootlist` command takes information from the node boot information given by `splstdata -b`, this information has to be changed by issuing the `spbootins` command. Once the change is effective, you can issue the `spbootlist` command.

Verify your node bootlist by issuing the command:

```
dsh -w <node> 'bootlist -m normal -o'
```

Then reboot the node.

The following example gives the steps to follow to activate a new rootvg on node 1 (hostname is node01). We assume two Volume Groups (rootvg1, and rootvg2) already have been installed on the node. rootvg1 is the active rootvg.

1. Change the node boot information:

2. `spbootins -l 1 -c rootvg2 -s no`

3. Note: it is not necessary to run `setup_server`.

4. Verify:

5. `splstdata -b`

6. Change the node bootlist:

7. `spbootlist -l 1`

8. Verify:

9. `dsh -w node01 'bootlist -m normal -o'`

10. Reboot the node:

`dsh -w node01 'shutdown -Fr'`

### Important

The key switch must be in the normal position

```

                                Set Bootlist on Nodes

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

                                [Entry Fields]
Start Frame                       [] #
Start Slot                         [] #
Node Count                         [] #

OR

Node List                          []

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

```

Figure 28. SMIT Panel for the `spbootlist` Command

## 2.4.2 Booting from External Disks

Support has been included in PSSP 3.1 for booting an SP node from an external disk. The disk subsystem can be either external Serial Storage Architecture (SSA) or external Small Computer Systems Interface (SCSI). The option to have an SP node without an internal disk storage device is now supported.

### 2.4.2.1 SSA Disk Requirements

Figure 29 and Figure 30 show the SSA disk connections to a node.

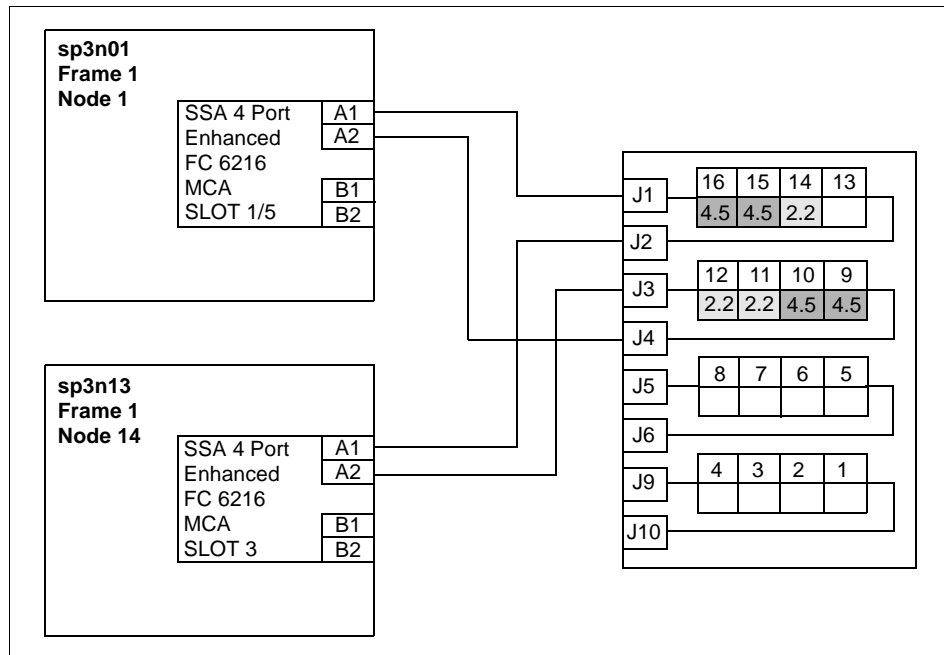


Figure 29. Cabling SSA Disks to RS/6000 SP Nodes

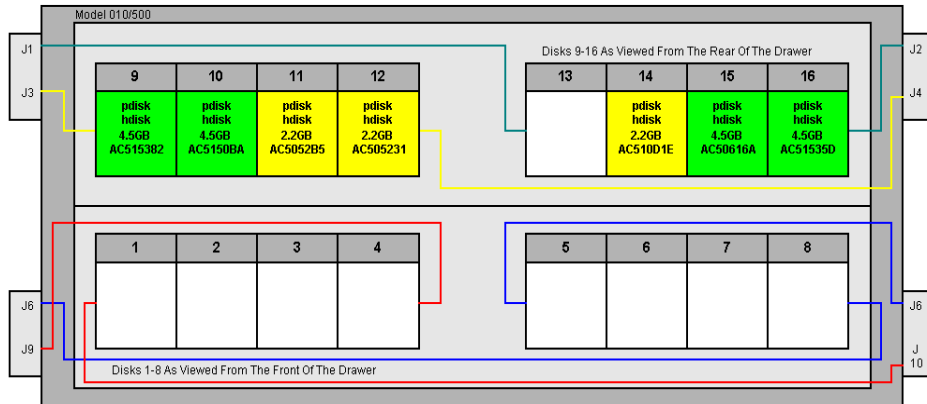


Figure 30. Connections on the SSA Disks

Not all node types can support SSA boot. Table 10 shows the node types that support an SSA boot.

Table 10. Supported Adapters for Nodes with Full SSA Boot

Node Code	Feature	Node Type	Feature Code Numbers of Supported SSA Adapters
2005		77 MHz Wide	#6214 SSA 4-Port Adapter #6216 Enhanced SSA 4-Port Adapter #6217 SSA RAID Adapter #6219 Enhanced SSA RAID Adapter
2006		604 High	Same as above
2007		120 MHz Thin	Same as above
2008		135 MHz Wide	Same as above
2009		604e High	Same as above
2022		160 MHz Thin	Same as above

The SP-supported external SSA disk subsystems are:

7133 IBM Serial Storage Architecture Disk Subsystems Models 010, 020, 500, and 600.



### 2.4.2.2 SCSI Disk Requirements

Some nodes can now be booted from external SCSI-2 Fast/Wide disk 7027-HSD storage device. Not all nodes can support an SCSI boot. Table 11 lists the nodes and the adapters for external disk booting.

Table 11. Supported Adapters for Nodes with SCSI Boot

Node Code	Feature	Node Type	Feature Code Numbers of Supported SCSI Adapters
2002		66 MHz Thin	#2412, #2416
2003		66 MHz Wide	Same as above
2004		66 MHz Thin 2	Same as above
2005		77 MHz Wide	Same as above
2006		604 High	Same as above
2007		120 MHz Thin	Same as above
2008		135 MHz Wide	Same as above
2009		604e High	Same as above
2022		160 MHz Thin	Same as above
2050		332 MHz SMP Thin	#6207, #6209
2051		332 MHz SMP Wide	#6207, #6209

The SP-supported external SCSI disk subsystems are:

7027-HSD IBM High Capacity Drawer with an SP SCSI-DE/FW adapter for Micro Channel machines, or SP Ultra-SCSI adapter for PCI machines.

### 2.4.2.3 Specifying an External Installation Disk

During the node installation process, external disk information may be entered in the SDR by first typing the SMIT fastpath `smitty node_data` (refer to Figure 7 on page 26). Depending on whether you have already created the Volume\_Group, you must then choose **Create Volume Group Information** or **Change Volume Group Information** from the Node Database Information Window (related commands are `spmkvgobj` or `spchvgobj`). Alternatively, you may use the SMIT fastpath `smitty changevg_dialog` (refer to Figure 21 on page 46) to get straight there.

Figure 31 shows the Change Volume Group Information window. In this the user is specifying an external SSA disk as the destination for rootvg on node1. Note that you may specify several disks in the Physical Volume List field (refer to 2.1.5.1, “New Volume\_Group Command: spmkvgobj” on page 44 for more information on how to enter the information).

```

Change Volume Group Information

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

[TOP]                                [Entry Fields]
Start Frame                          []                #
Start Slot                            []                #
Node Count                            []                #

OR

Node List                             [1]

Volume Group Name                     [rootvg]
Physical Volume List                  [ssar//0004AC50532100D]
Number of Copies of Volume Group      1                +
Set Quorum on the Node                 +
Boot/Install Server Node              []                #
Network Install Image Name            []

[MORE...2]

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

```

Figure 31. SMIT Panel to Specify an External Disk for SP Node Installation

When you press the Enter key in the Change Volume Group Information window, the external disk information is entered in the Node class in the SDR. This can be verified by running the `splstdata -b` command as shown in Figure 32 on page 69. This shows that the install disk for node 1 has been changed to `ssar//0004AC50532100D`.

Under the covers, `smitty changevg_dialog` runs the `spchvgobj` command. This is a new command in PSSP 3.1 that recognizes the new external disk address formats. It may be run directly from the command line using this syntax:

```
spchvgobj -r rootvg -h ssar//0004AC50532100D -l 1
```

```

sp3en0{ / } splstdata -b -l 1

List Node Boot/Install Information

node#      hostname  hdw_enet_addr  srvr  response  install_disk
last_install_image  last_install_time  next_install_image  lppsource_name
pssp_ver          selected_vg
-----
1 sp3n01.msc.itso.  02608CE8D2E1  0    install  ssar//0004AC510D1E00D
                default          initial      default    aix432
                PSSP-3.1          rootvg

```

Figure 32. Output of the splstdata -b Command

#### 2.4.2.4 Changes to the bosinst.data File

When the changes have been made to the Node class in the SDR to specify an external boot disk, the node can be set to "install" with the `spbootins` command:

```
spbootins -s yes -r install -l 1
```

The `setup_server` command will cause the network install manager (NIM) wrappers to build a new `bosinst.data` resource for the node, which will be used by AIX to install the node.

The format of `bosinst.data` has been changed in AIX 4.3.2 to include a new member to the `target_disk` stanza specified as `CONNECTION=`. This is shown in Figure 33 on page 70 for node 1's `bosinst.data` file (node 1 was used as an example node in Figure 31 on page 68 and Figure 33 on page 70). NIM puts in the new `CONNECTION=` member when it builds the file.

```
control_flow:
  CONSOLE = /dev/tty0
  INSTALL_METHOD = overwrite
  PROMPT = no
  EXISTING_SYSTEM_OVERWRITE = yes
  INSTALL_X_IF_ADAPTER = no
  RUN_STARTUP = no
  RM_INST_ROOTS = no
  ERROR_EXIT =
  CUSTOMIZATION_FILE =
  TCB = no
  INSTALL_TYPE = full
  BUNDLES =

target_disk_data:
  LOCATION =
  SIZE_MB =
  CONNECTION = ssar//0004AC50532100D

locale:
  BOSINST_LANG = en_US
  CULTURAL_CONVENTION = en_US
  MESSAGES = en_US
  KEYBOARD = en_US
```

Figure 33. *bosinst.data* File with the New CONNECTION Attribute

---

## Chapter 3. SP Perspectives

The purpose of this chapter is to cover the new functionality of the *SP Perspectives* element of PSSP 3.1. The SP Perspectives graphical user interface, *GUI*, is now the focal point for managing your SP.

For more detailed information on the use of SP Perspectives in PSSP 3.1 refer *PSSP Administration Guide*, SA22-7348. This guide is included with the PSSP 3.1 product.

---

### 3.1 Overview

Prior to this release of PSSP, there was a second GUI in common use, which was invoked by the `spmon -g` command. The `-g` flag is now removed and the functionality it provided is available in SP Perspectives. All other flags for `spmon` are still used. You are still, for example, able to issue the command `spmon -G -d`

The AIX command `perspectives` starts the *Launch Pad*, from which you can launch these applications:

- Hardware perspective, `/usr/lpp/ssp/bin/sphardware`
- Event perspective, `/usr/lpp/ssp/bin/spevent`
- IBM Virtual Shared Disk perspective, `/usr/lpp/ssp/bin/spvsd`
- Performance monitor, `/usr/lpp/ssp/bin/spperfmon`
- Partitioning aid, `/usr/lpp/ssp/bin/spsyspar`
- SP resource center, `/usr/lpp/ssp/bin/resource_center`
- Various SMIT fastpaths
- Four sample hardware perspective configurations
- SP Perspectives online help

#### 3.1.1 New Features

A number of new features are added to SP Perspectives with this release of PSSP.

##### 3.1.1.1 Multiple Panes and Multiple Windows

The capability to have multiple instances of the same type of pane inside one window has been added to the Hardware Perspective and the IBM Virtual

Shared Disk Perspective. For instance, you can display three separate node panes, each monitoring a different condition.

The add/delete pane combo box has been removed and replaced with the add pane and delete pane toolbar icons. These functions are also available under the View pull down menu.

#### **3.1.1.2 Displaying Panes In A Table View**

A pane can have another view to it, a tabular or details view. This shows the objects in the pane as rows in a table. The attributes displayed in the columns of the table are user selectable. In this way you can display a great deal of information in a concise way.

#### **3.1.1.3 SDR Refresh**

The perspective applications are now aware of changes taking place to the SDR. If, for example, you are working in the hardware perspective and have a pane showing node groups, any changes made to the NodeGroups SDR Class are reflected in the running perspective.

#### **3.1.1.4 Notebooks**

New notebook status pages are added for nodes, frames and switches. For examples, refer to "The Hardware Perspective" on page 79.

#### **3.1.1.5 Filters**

The capability to filter nodes by monitored state is available in the Hardware Perspective and the IBM Virtual Shared Disk Perspective. You can monitor a node pane for one or more conditions you are interested in. By filtering on triggered conditions you display only nodes which have in effect failed any of these monitored conditions. Any nodes displaying because a monitored condition has triggered can be acknowledged to remove them from the filtered display.

#### **3.1.1.6 Preferences**

Preferences can be saved from the application and loaded from the command line on invocation. The capability to load preferences interactively from a dialog has been removed.

#### **3.1.1.7 Fly-Over Help**

Fly-over help has been added to all views, as shown by Figure 34.

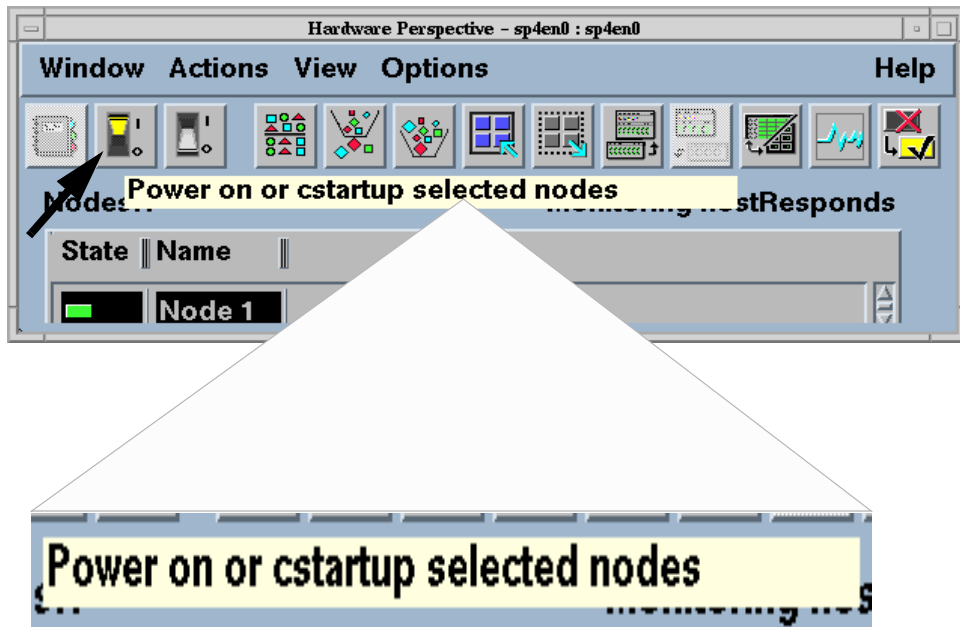


Figure 34. Fly-Over Help Displayed for Power-On Icon

### 3.1.2 SPMON Equivalence

The spmon GUI has been used extensively in the SP world for control and basic monitoring of an SP system. The spmon functionality for this hardware control and monitoring is now available in SP Perspectives.

The hardware perspective provides you with equivalence for the following commonly used spmon functions:

- Front panel display for a single node
- Global controls for nodes
- All node summary displays
- Frame environment layout
- Switch environment layout

### 3.1.3 New Filesets

The only new fileset is ssp.resctr, the SP Resource Center, a separately installable package, which is also available on CD-ROM. It installs on AIX and will run from CD-ROM on both AIX and Windows 95/NT systems. This is a

browser-based application using HTML pages, JavaScript and Java. For more information, see 1.1.9, “SP Resource Center” on page 12.

### 3.2 Launch Pad

SP Perspectives can be described as a suite of applications for system management which are available from a common launch pad. The primary function of the launch pad is to provide a common interface for invoking all five perspectives:

- Hardware Perspective for monitoring and controlling hardware
- Event Perspective for managing system events and taking actions when events occur
- IBM Virtual Shared Disk Perspective for managing shared disks
- The System Partitioning Aid
- The Performance Monitor Perspective

The launch pad also provides access to other applications such as SMIT, the partition-sensitive subsystems command `syspar_ctrl`, and the SP Resource Center which gains access to the SP documentation through a Web browser interface. The launch pad is shown in Figure 35.

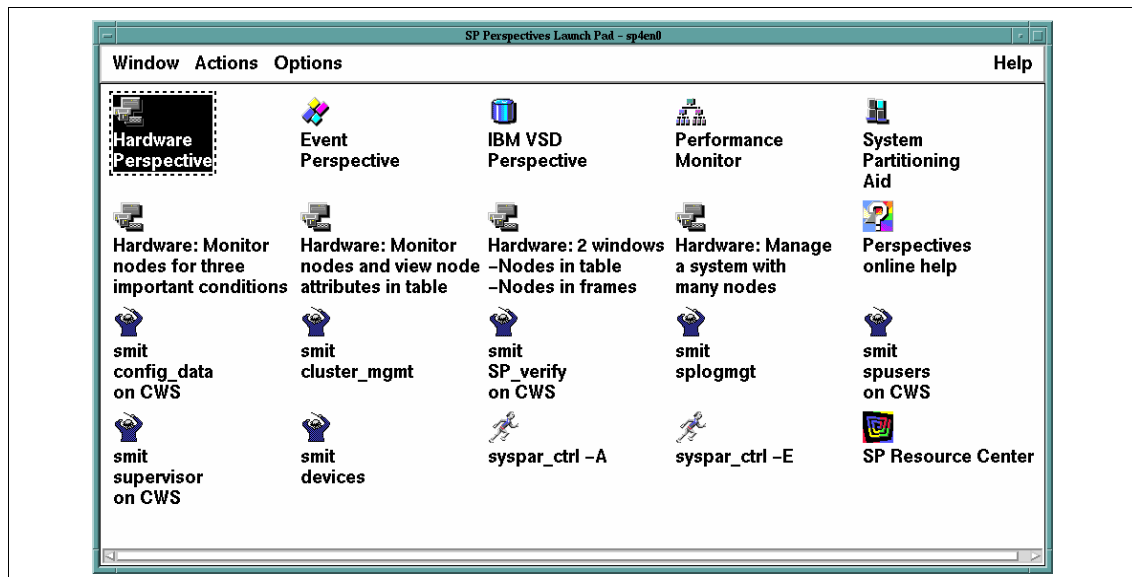


Figure 35. SP Perspectives Launch Pad



### 3.2.1 Profiles

You can alter the look and feel of the launch pad by opening the **Options** pull down menu and selecting either the **Set Colors** or **Set Fonts** dialog boxes from the menu. It is also possible to add new icons to the launch pad by selecting **Customize Applications** from the Options pull down menu. Any changes to the launch pad are not remembered the next time SP Perspectives is started unless these preferences are first stored in a profile. In order to "save preferences access the **Options->Save Preferences** dialog menu.

SP Perspectives can save a profile as one of two types:

- **User.** User profiles are stored in the user's home directory. The profile is named with the filename that the user supplied using the **Options->Save Preferences** dialog with .perspectives prepended to it. For example, if you save a user profile to a file called "robin", then the file will appear in your home directory as a file called ".perspectivesrobin".
- **System.** System profiles are stored in the PSSP directory tree in /usr/lpp/ssp/perspectives/\$LANG/profiles. They use the same naming convention as user profiles.

The launch pad Save Preferences dialog box is shown in Figure 36. In this case you save launch pad colors, font, layouts and applications to a user profile called "robin" which will be stored in your home directory.

You specify the user profile or system profile to use when you start SP Perspectives. For example, to get SP Perspectives to load the user profile "robin" when it starts, you would type on the command line:

```
$ perspectives -userProfile robin
```

System profiles are loaded with the -systemProfile flag. For a system profile called system.robin, you would type:

```
$ perspectives -systemProfile system.robin
```

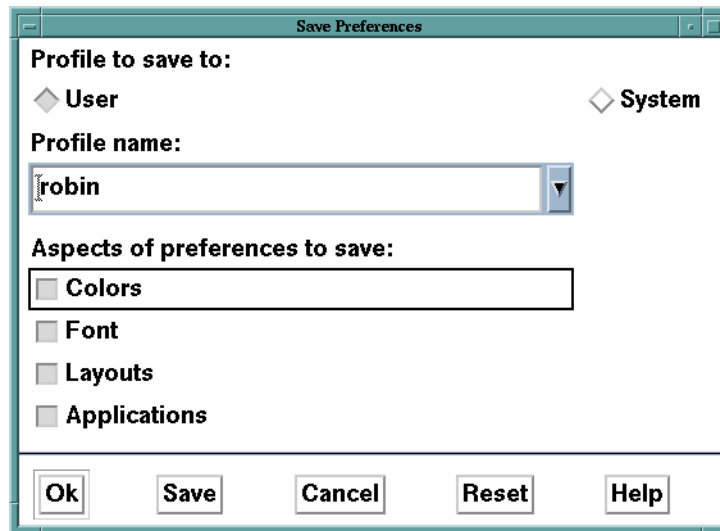


Figure 36. Save Preferences Dialog Box

### 3.2.2 Adding More Applications to the Launch Pad

Additional applications can be placed on the launch pad by selecting **Options->Customize Applications** from the Options pull down menu. This opens an additional pane on the bottom of the Launch Pad which is used to define the new application. The easiest way to define a new application is to highlight one of the existing ones and then change the name, the description of what the new application will do and the name of the program that it will execute. Clicking on the **Add** button will then put an icon on the desktop which represents the new application.

Figure 37 shows the launch pad in customize mode. It illustrates adding a new application called "My App" which runs the shell script `/home/robin/robin.sh` in an `aixterm` window.

When the new icon has been added, you can click on the **Leave Customize Mode** button.

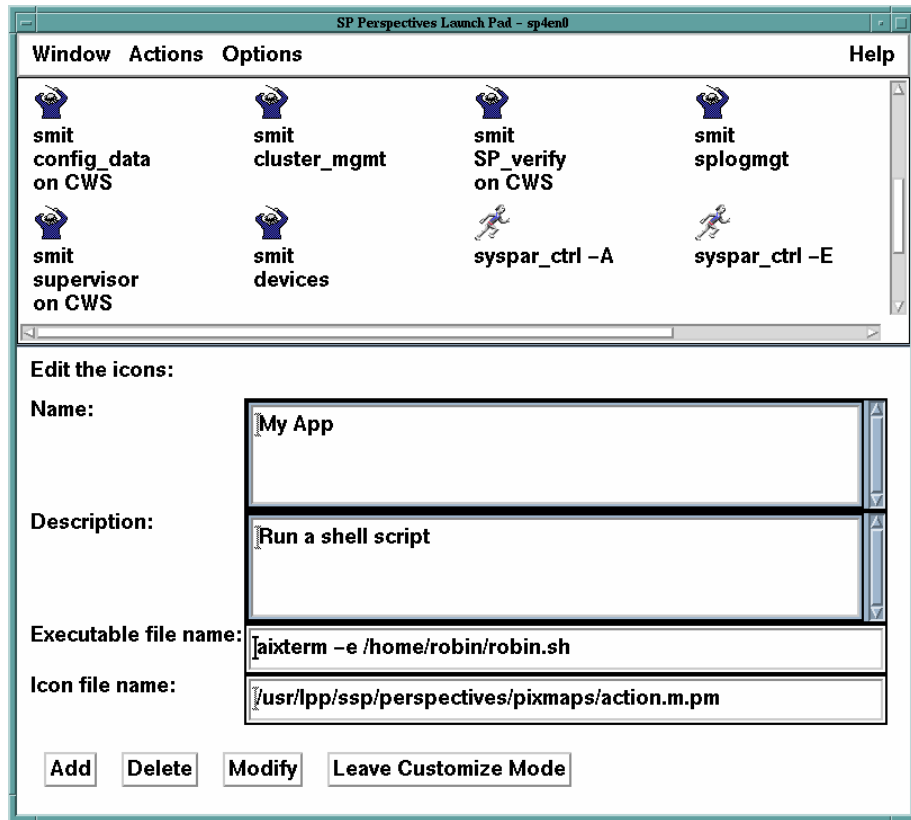


Figure 37. Adding a New Application to the Launch Pad

Figure 38 shows the new icon "My App" added to the Launch Pad.

The Launch Pad is then saved in a profile so that the icon appears the next time the Launch Pad is run. Saving a profile is described in 3.2.1, "Profiles" on page 75.

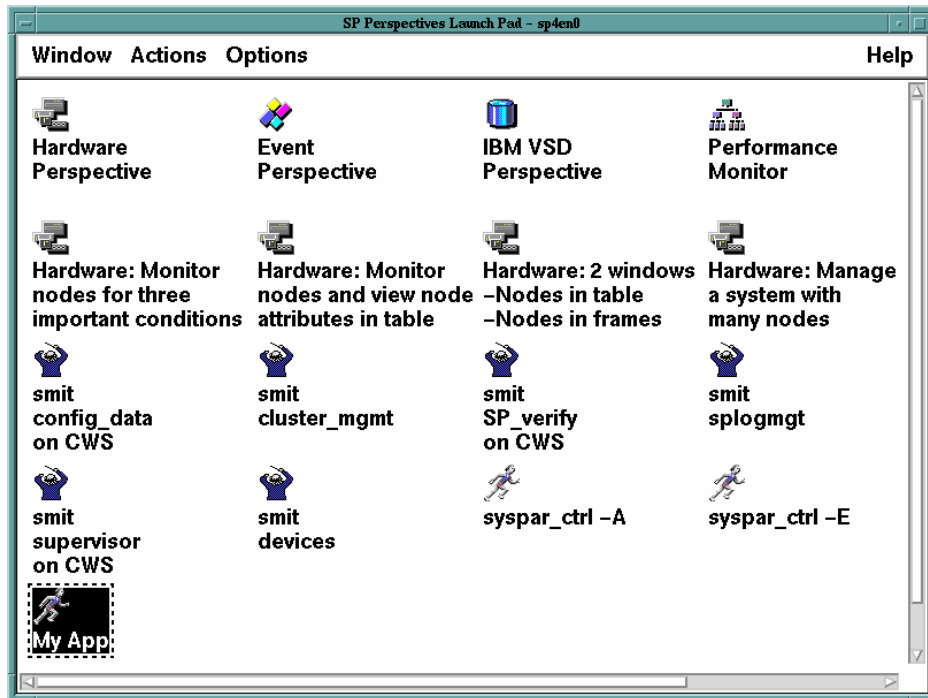


Figure 38. New Icon Added to the Launch Pad

### 3.2.3 Predefined Icons

The Launch Pad comes with sixteen predefined icons which launch the different SP Perspectives, SMIT screens and the partition-sensitive subsystem commands as discussed in 3.2, "Launch Pad" on page 74. You can view the extended descriptions of all the predefined icons by pulling down the **Options** menu and selecting **Show Application Details**. The application detail view is illustrated in Figure 39.

Referring to Figure 39 and using the hardware perspective as an example, we can see that there are five hardware perspectives shown on the launch pad. Starting at the top of the figure and moving from left to right, we see that the first icon simply launches the hardware perspective. The fifth, sixth, seventh and eighth icons launch the hardware perspective too, but they also load a different profile, specific to the hardware perspective, which defines how the hardware perspective looks on the screen and what system objects are displayed. For example, the fifth icon ("Hardware: Monitor nodes for three important conditions") will launch the hardware perspective with a profile which shows host responds, switch responds and node power LED in

different panes in the same window. The sixth, seventh and eighth icons load different profiles for the hardware perspective as described in the same figure.

The hardware profiles are discussed in more detail in “The Hardware Perspective” on page 79

The profiles that the predefined icons load are system profiles. They are stored in the /usr/lpp/spp/perspectives/\$LANG/profiles directory.

Icon	Application Name	Description
	Hardware Perspective	Manage, control, and monitor hardware.
	Event Perspective	Setup and manage events.
	IBM VSD Perspective	Manage and monitor IBM VSDs and IBM HSDs
	Performance Monitor	Setup and configure performance monitoring.
	System Partitioning Aid	Define system partitions.
	Hardware: Monitor nodes for three important conditions	Hardware Perspective in global view with a profile to monitor the hostResponds, switchResponds, and nodePowerLED conditions in separate nodes panes.
	Hardware: Monitor nodes and view node attributes in table	Hardware Perspective in global view with a profile to monitor the hostResponds, switchResponds, and nodePowerLED conditions and view the matching attributes in a table view.
	Hardware: 2 windows - Nodes in table - Nodes in frames	Hardware Perspective in global view with a profile to bring up two windows. View nodes in a table view in one window, and view nodes in frames in the second window.

Figure 39. Launch Pad Application Details View

### 3.3 The Hardware Perspective

The hardware perspective has two functions:

- Controlling hardware. For example, selecting a node, or several nodes, and performing an action on them such as power on/off, fence/unfence or network boot.
- Monitoring hardware. For example, opening a window which watches important SP conditions such as host responds, switch responds and node power LEDs.

Before we review these functions and how to perform them, we will look at the structure of the hardware perspective window as it appears when it is run either from the command line with the `"sphardware` command or by double clicking the left mouse button on the top leftmost icon **Hardware Perspective** on the launch pad. The hardware perspective is illustrated in Figure 40.

The hardware perspective uses a concept of system objects. The system objects are defined as:

- Control workstation
- System
- System Partitions
- Nodes
- Frames and Switches
- Node Groups

These objects are displayed by default as icons which are placed inside panes in the perspective window (icon view).

The hardware perspective allows four different kinds of pane. It refers to them as:

- CWS, System and Syspars (contains the control workstation, system and system partition objects)
- Nodes (contains node objects)
- Frames and Switches (contains frame and switch objects)
- Node Groups (contains node group objects)

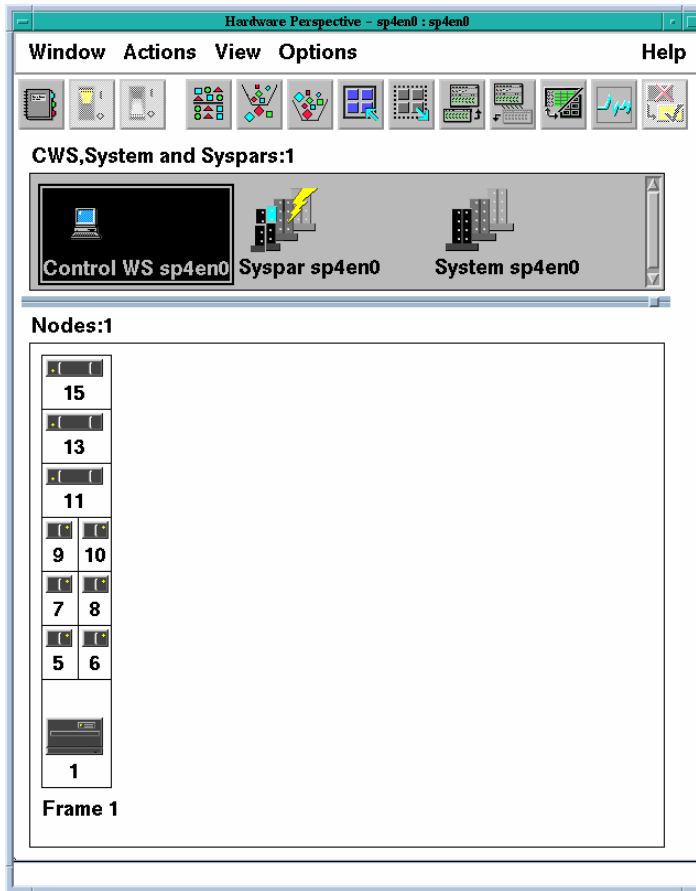
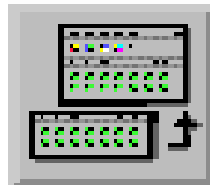


Figure 40. Hardware Perspective

Figure 40 shows the hardware perspective displaying only the "CWS, System and Syspars" and "Nodes" panes. The extra pane types can be added to the window by selecting **View-Add pane** or by single clicking on the **Add Pane** icon on the task bar. The Add Pane icon looks like this:



Clicking on **Add Pane** brings up the dialog as shown in Figure 41. This figure illustrates that there is a drop-down list box which will allow you to choose the pane type. You can then pick whether to add the new pane to the current perspective window or to a new window. Panes can similarly be deleted by selecting **View->Delete Pane** from the menu bar:

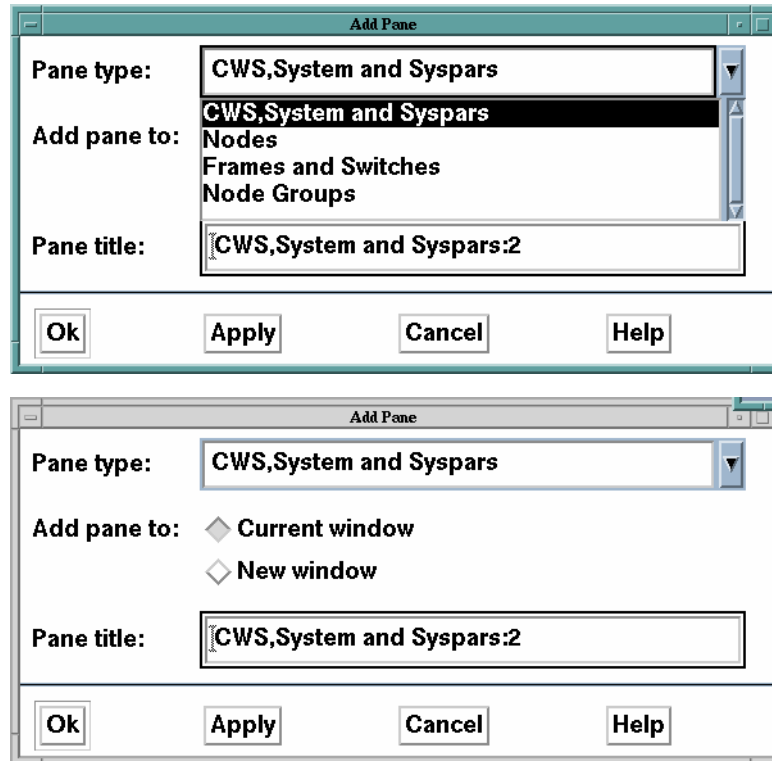
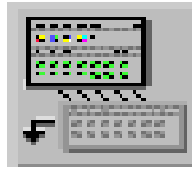


Figure 41. The Add Pane Dialog

Figure 42 shows all four types of pane added to the perspective.



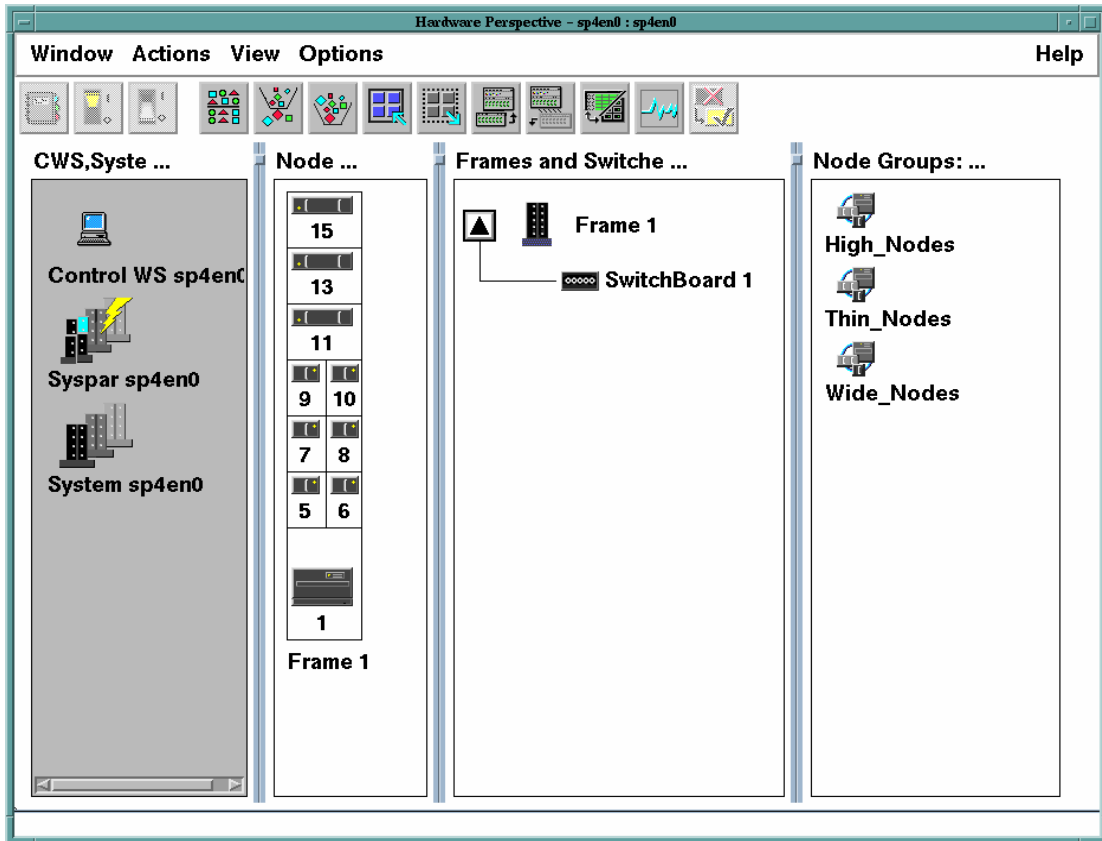


Figure 42. Hardware Perspective Showing All Pane Types

The hardware perspective also has a concept called the current (active) pane. When you click in a pane with the mouse, the background of that pane turns gray and it becomes the current pane. Only objects in the current pane can have actions performed against them. You may also perform actions on the pane itself such as changing its title or deleting it (these actions are found on the "View" menu).

Figure 42 shows that the CWS, System and System Partitions pane is the current pane.

The objects inside a pane may also be displayed in "table view" instead of as icons. This is described in more detail in 3.3.2, "Monitoring Hardware" on page 88.

Once you have arranged the panes as you like them in the window then it is probably a good idea to save them in a profile, otherwise the layout will not be restored the next time the hardware perspective is started. As described already in 3.2.1, "Profiles" on page 75, the "Save Preferences" is accessed from the "Options" menu. If the profile is saved as a user profile then it is saved in the user's home directory with sphardware prepended to the profile name. System profiles are stored as before in /usr/lpp/ssp/perspectives/profiles also with sphardware prepended to the name.

### 3.3.1 Controlling Hardware

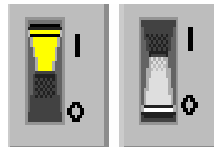
A control operation on an object may be initiated in one of three ways. These are:

- Through a pull-down menu on the hardware perspective window
- Through an object's notebook
- From the tool bar

To perform any of the control actions it is first necessary to select the object or objects that you want to perform the operation on. Select the object you want with a single click of the left mouse button, this also activates the pane if it is not the currently selected pane. There are a number of ways to select multiple objects:

- Hold down the <Ctrl> key while clicking the left mouse button.
- Rubberband selection.
- In the **View** menu, select **Select All**.
- Click on the **Select All** icon on the tool bar.

Once your nodes are selected, if the action required is power off or power on, you can select the power-off icon or power-on icon from the tool bar, shown here:



Otherwise a controlling action can be performed by dropping down the **Actions** menu as shown in Figure 43. This figure shows that the action will be performed on node 1, 5 and 6 as these nodes have been selected.

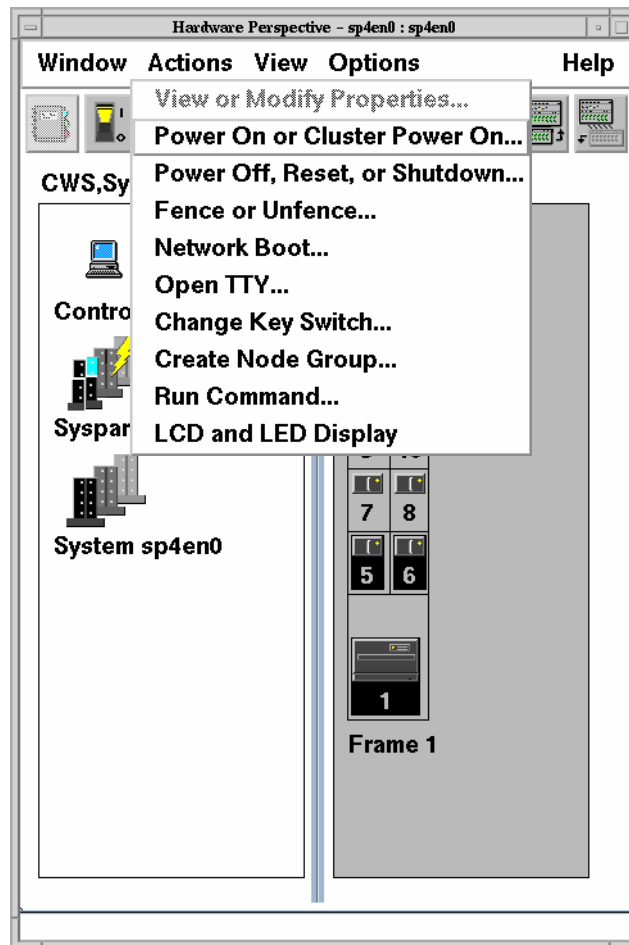


Figure 43. Selecting Node Objects in a Pane

If you wish to change the key switch position, then the "Change Key Switch Nodes" dialog will appear as shown in Figure 44. This shows that the key has been put in to service mode for nodes 1,5 and 6. It is clear that the "Apply" button has been clicked and the control has been run as the command output text box reads: "All key switch actions have been issued".

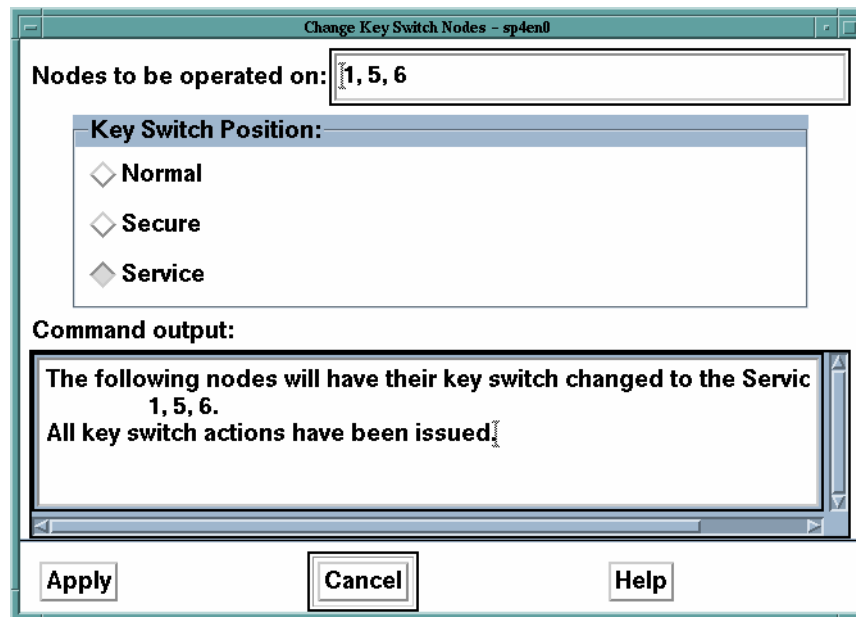


Figure 44. Change Key Switch Nodes Dialog

If a control action is only to be performed on one object, then it can be done from that object's notebook. To access the notebook, first select the object in the pane. The notebook is opened by **Actions->View or Modify Properties** in the Actions menu or by single clicking the left mouse button on the notebook icon on the tool bar. The notebook icon looks like this:



Figure 45 shows the notebook belonging to node5. The node name is displayed in the title bar of the notebook window. The notebook window has been designed to look like a spiral-bound notebook with separator tabs to go to other pages in the notebook. Figure 45 shows the notebook open at the "Node status" tab where the node control buttons are located. From this tab, the node can be powered on or off, fenced or unfenced, a terminal session (tty) can be opened or a network boot performed.

The node status tab is also useful for checking some of the attributes (more properly called resources) of the node such as, controller responds, host

responds, switch responds, whether the node has power and the key switch position.

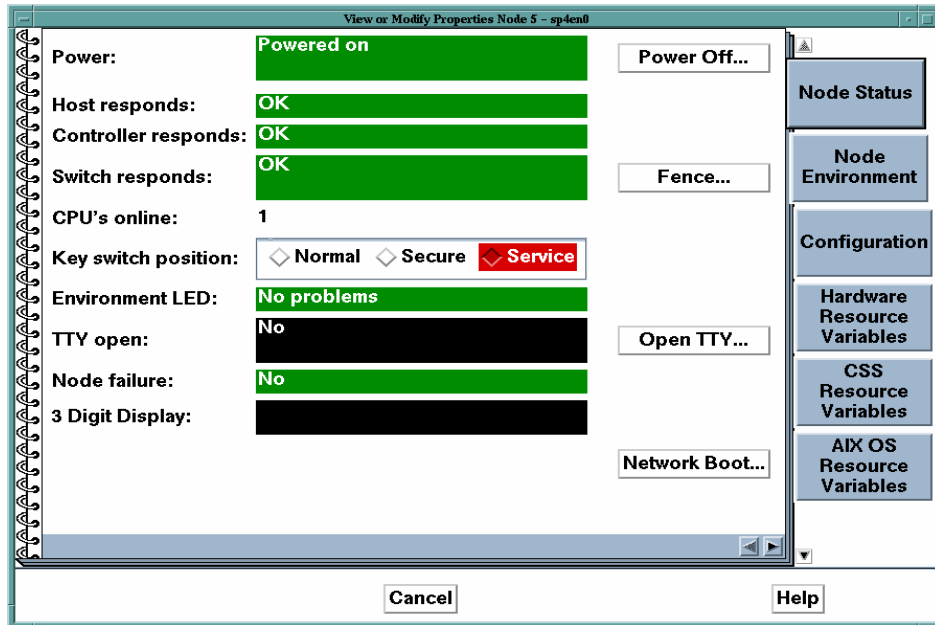


Figure 45. Notebook for Node 5

The complete list of tabs are:

- Node status. Control the node and view some node resources.
- Node environment. Displays power supply and temperature information for the node.
- Configuration. Displays information from the node SDR object class.
- Hardware Resource Variables. Lists the names of hardware variables associated with nodes and the current values of those variables.
- CSS Resource Variables. Lists the names of switch specific variables associated with nodes and the current values of those variables.
- AIX OS Resource Variables. Lists the names of AIX operating system variables associated with nodes and the current values of those variables.
- All dynamic resource variables. The complete list hardware, switch and AIX OS variables and their current values.

- **Monitored Conditions.** A list of variables that are currently being monitored for this node. Monitoring nodes is described in 3.3.2, “Monitoring Hardware” on page 88.

It is useful to know the names of the resource variables when configuring the event perspective for performing event monitoring, see 3.4, “Event Perspectives” on page 95. Also, an experienced user who is familiar with the resource names would be able to quickly scan the variable lists in a notebook to check for problems.

The complete list of tabs is not shown in Figure 45; clicking on the small down arrow above the "Help" button allows the rest of the tabs to be scrolled.

### 3.3.2 Monitoring Hardware

Initiating monitoring of the hardware resources is done by selecting **View->Set Monitoring** from the "View" pull down menu. Before doing this, make sure that your current pane is the correct one. For example, if you want to start monitoring the nodes, then you need to ensure that the nodes pane is the current pane. The "View" menu is illustrated in Figure 46.

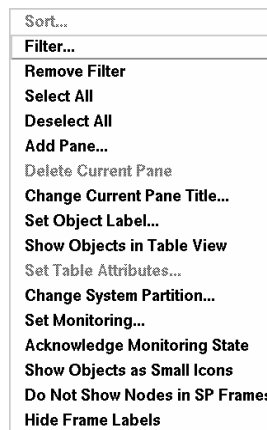


Figure 46. The Hardware Perspective View Menu

The "Set Monitoring" dialog is shown in Figure 47. In the top pane of the window there is a list of conditions which can be monitored. In the figure, hostResponds has been selected for monitoring. More than one condition may be selected by holding down the <Ctrl> key and clicking the left mouse button. Clicking on the **Ok** button will then start the monitoring for all nodes. When monitoring is started in a node or node group pane, all objects in the pane are monitored.

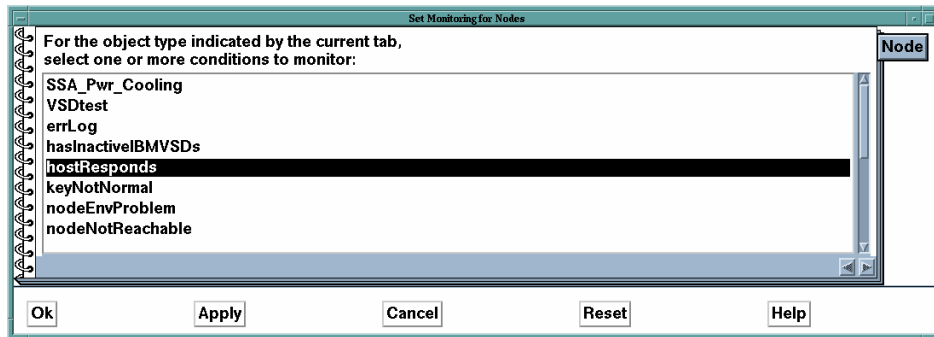


Figure 47. Set Monitoring Dialog Box

Figure 48 shows the hardware perspective with a single Nodes pane which is monitoring the hostResponds condition. There are three features of the display that should be noted:

- The condition(s) that the pane is monitoring is displayed above the top right hand corner of the pane.
- If the condition is true, in this case if the node has host responds, then the icon will be green. If the condition is not met, then the icon will be black and have a small red cross through it. You can see from Figure 48 that node 13 does not have host responds. A monitored condition which has failed and is red can also be acknowledged by clicking on the "acknowledge monitored or unknown state" icon or selecting **View->Acknowledge Monitoring State**. If this is selected the icon will turn yellow with a black tick mark.
- If the perspective window is minimized, then the icon background color will be the aggregate of all the conditions on all the objects as listed below:
  - A mix of red, green and yellow, icon will be red
  - A mix of yellow and green, icon will be yellow
  - All green, icon will be green

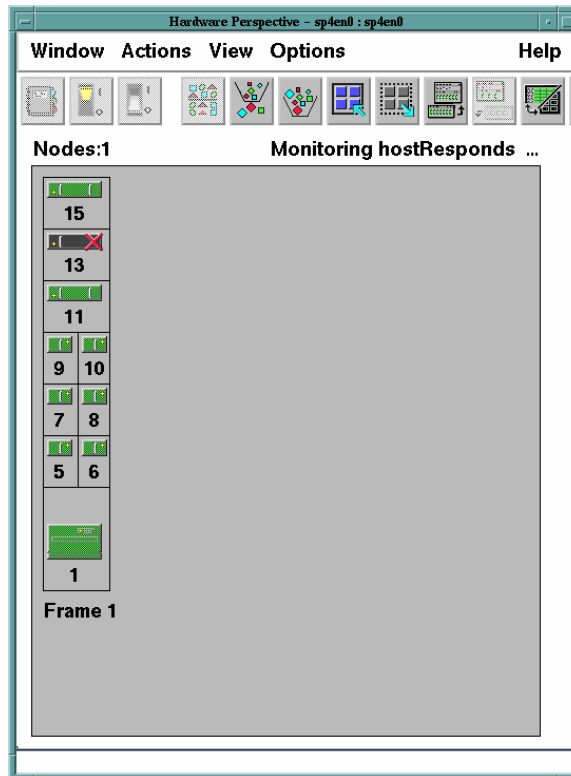


Figure 48. A Nodes Pane Showing Monitoring of hostResponds

If you need to monitor more than one condition at once, then it is possible to open a new pane for each monitored condition, otherwise the aggregate for all the monitored conditions is displayed. For example, if an icon object is displayed as black with a red cross, then it is not clear which monitored condition is causing the problem. You can see an example of a three-pane window monitoring three conditions by clicking on the predefined icon, "Monitor nodes for three important conditions", on the launch pad:



**Hardware: Monitor nodes for three important conditions**

Figure 49 shows the resulting hardware perspective.



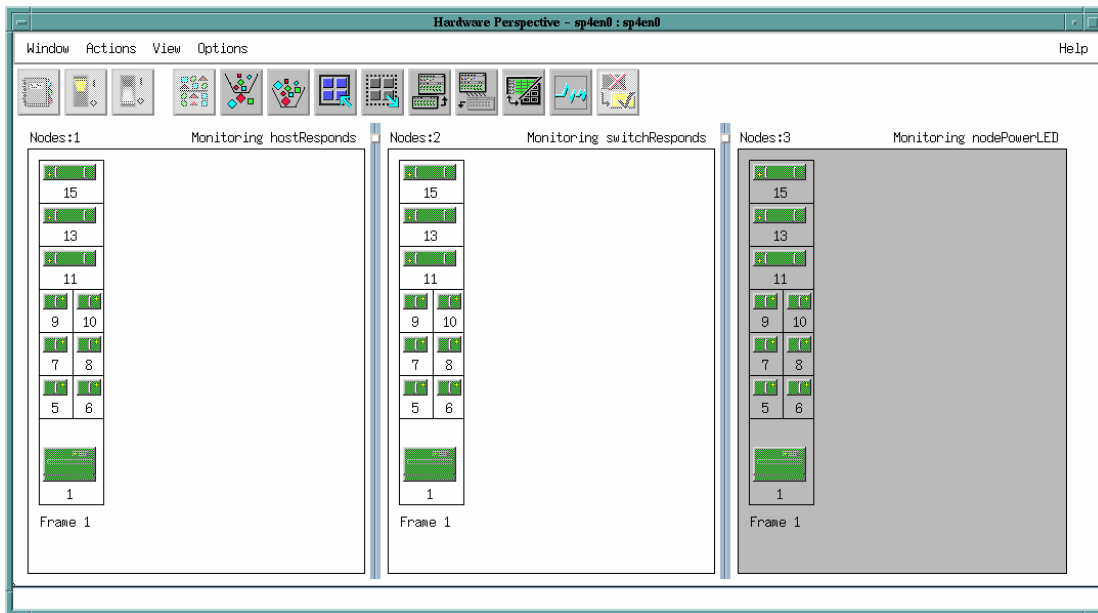


Figure 49. Monitoring Nodes for Three Important Conditions

If multiple panes are displayed in a window, this can take up a lot of screen space. Although the panes can be made smaller by resizing the pane drag handles, this could still be a problem if you need to monitor multiple conditions or your SP environment has multiple frames. There is a more compact way to display the same information contained in Figure 49. This is referred to as "Table View". You can change the display to table view by selecting **View->Show Objects in Table View** from the "View" menu. The same information in Figure 49 is shown in table view in Figure 50. Further space may be saved in the window by choosing not to display the tool bar. This can be turned off from the "Options" menu.

State	Name	Host responds	Controller responds	Switch responds
	Node 1	OK	OK	OK
	Node 5	OK	OK	OK
	Node 6	OK	OK	OK
	Node 7	OK	OK	OK
	Node 8	OK	OK	OK
	Node 9	OK	OK	OK
	Node 10	OK	OK	OK
	Node 11	OK	OK	OK
	Node 13	OK	OK	OK
	Node 15	OK	OK	OK

Figure 50. Monitoring Nodes in Table View

The columns that are displayed in the table can be altered by selecting **View->Set Table Attributes** from the "View" menu. The Set Table Attributes dialog is shown in Figure 51 where Host Responds, Controller Responds and Switch Responds have been selected for display as in Figure 50.

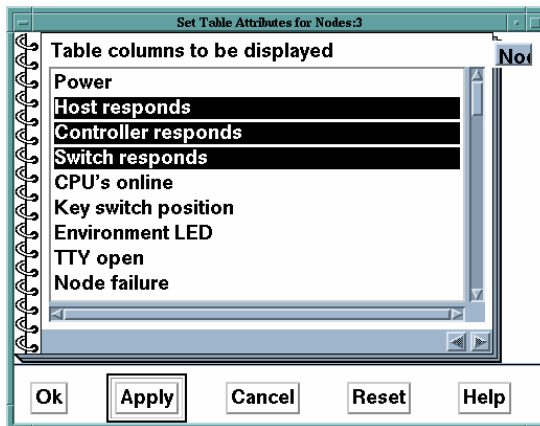


Figure 51. Set Table Attributes Dialog Box

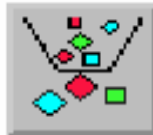
Any monitored condition that has been set up in icon view will be inherited when you switch to table view and displayed as an icon in the "State" column. If the monitored condition is met, then the state icon will be green. If it is not met, then it will be black with a red cross through it. If there is no monitored condition, then the state icons will all be black. Setting up monitoring is accomplished just as it is in icon view, by selecting the **View->Set Monitoring** option from the "View" menu.

Two things should be remembered when you are working in table view:

- The color of the state icon is the aggregate of *all* the monitored conditions for that object. If there is more than one monitored condition, then the state icon will only be green if all the monitored conditions are met.
- If the table view window is minimized, then the background color of the iconified window is the aggregate of *all* the monitored conditions of the objects. Iconifying the window can be useful, especially if the table is large, as it makes it very easy to detect state changes in monitored conditions.

### 3.3.3 Filtering by Monitored State

This is useful for displaying only those monitored objects which are in a certain state. For example, if you are monitoring switch responds, then you might only want to display those nodes which are not OK. To access the filtering functions, select **View->Filter** option from the "View" menu or click on the filter icon on the tool bar:



This will make the filter dialog box appear, as shown in Figure 52. Assuming that we have already set up monitoring for switch responds, then we may display only those nodes which do not have switch responds by depressing the following radio buttons in the dialog and clicking on the OK button:

- Filter objects by monitored state
- Not triggered
- Hide objects that match the filtering criteria

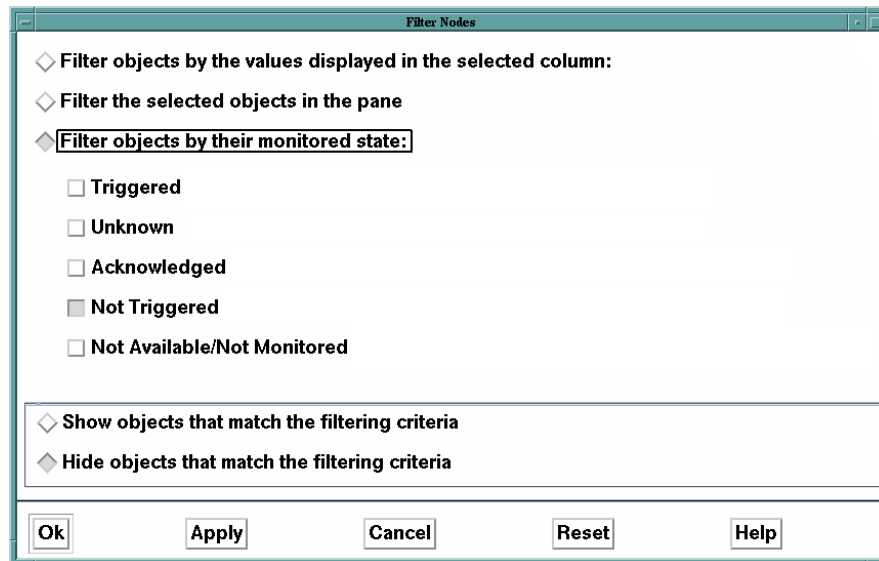


Figure 52. Filter Nodes Dialog

By selecting these three buttons, we hide all those nodes which have not triggered the switch responds monitored condition. The hardware perspective will then appear as in Figure 53. To achieve the same effect, we might have also have done:

- Filter objects by their monitored state
- Triggered
- Show objects that match the filtering criteria

In other words, we show only those objects which have triggered the switch responds monitored condition. This is just another way of doing the same thing.

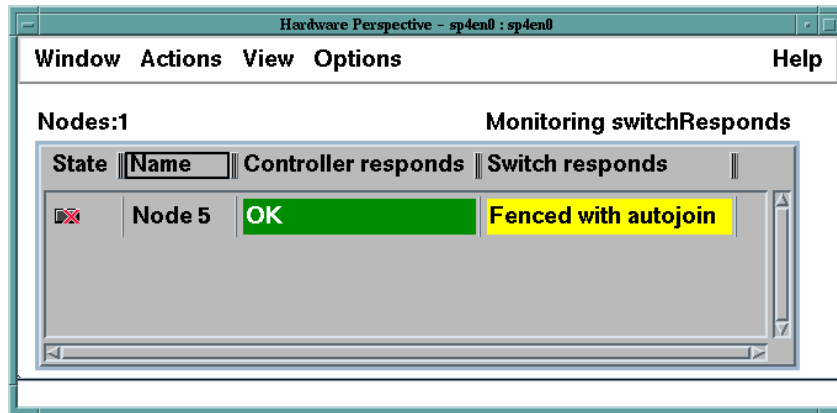


Figure 53. Hardware Perspective-Filtering by Monitored State

### 3.3.4 Viewing LED Values

There is one further useful feature of the hardware perspective. This is the LCD and LED display, which can be accessed by **Actions->LCD and LED Display**; see Figure 43 on page 85. This brings up a small window which shows the LED or LCD display dependent on the node type; see Figure 54.

Left-clicking the mouse button inside the window displays the node numbers.



Figure 54. Node LCD and LED Displays

---

## 3.4 Event Perspectives

For an overview of the event management system, refer to 6.4, “Event Management” on page 199. In this section we focus on the use of the event perspective to define and take action on events.

To show the new functions available in the event perspective, we have provided simple examples of registering, modifying and adding event definitions and conditions using the event perspective. These examples show the new conditions pane for the event perspective and make use of the options to use multiple panes and display objects in a table view.

To use the event perspective to register event definitions with the problem management system, you must add your kerberos principal to the problem management access control list file `/etc/sysctl.pman.acl`. This is a text file and can be edited using your favorite system editor. The entry for our file is shown in Figure 55. If you do not have an entry in this file, the event perspective will issue a warning as it initializes and certain actions cannot be performed. The file must exist on all nodes where an event definition is registered to a principal other than `root.admin`. It is most easily distributed to nodes from the CWS using the `pcp` command.

```
#acl#

# These are the kerberos principals for the users that can configure
# Problem Management on this node. They must be of the form as
# indicated
# in the commented out records below. The pound sign (#) is the comment
# character, and the underscore (_) is part of the "_PRINCIPAL" keyword,
# so do not delete the underscore.

_PRINCIPAL root.admin@MSC.ITSO.IBM.COM
```

Figure 55. The `/etc/sysctl.pman.acl` File

Once the event perspective has initialized, the screen should appear similar to Figure 55. Colors and fonts in these figures have been changed from the default to make the screen images more readable.

The definitions for the colors assigned to icons are shown in Figure 56.













Notification Option	Rearm Expression	Unregistered	Registered No Event	Registered Event Occurs
Y	Y	 All Gray	 MultiColor	 White Envelope
Y	N	 All Gray	 All Blue	 Blue Envelope
N	Y	 All Gray	 2 Color 2 Gray	 2 Color 2 Gray
N	N	 All Gray	 2 Color 2 Gray	 2 Color 2 Gray

Figure 56. Icon Color Table for Event Definitions

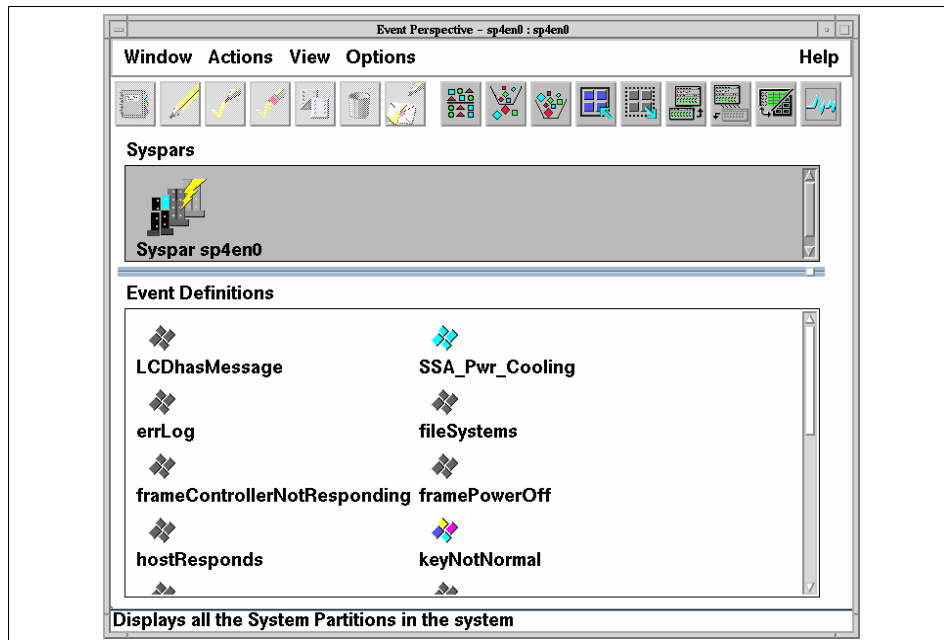


Figure 57. Initial Event Perspective Display

### 3.4.1 Integration with RSCT

RSCT is RS/6000 Cluster Technology. For more information about this product, refer to Chapter 6, “RS/6000 Cluster Technology” on page 183. Prior to the release of PSSP 3.1, different terminology was used by the event management system and event perspective application. These differences have now been removed and terminology is consistent between the different subsystems. The terminology changes are as follows:

- Use *expression* instead of *predicate*.
- Use *rearm expression* instead of *rearm predicate*.
- Use *resource identifier* instead of *instance vector*.
- Use *element name* instead of *field name*.
- Use *element value* instead of *field value*.

### 3.4.2 Activating a Preset Monitored Event

There are 19 pre-defined Event Definitions that display when the event perspective is started. These definitions can be seen in Figure 57. They can be loaded through a menu item, **Load Defaults**, from the **Actions** pull-down



of the Event Definitions pane. In order to explain the steps required in activating, modifying and creating events, we modify the views of the event perspective. The actions to carry out this modification are as follows:

- Highlight the Syspar Pane, select
  - **View->Delete The Current Pane**
- Highlight the Event Definitions Pane, select
  - **View->Show Object in Table View**
    - Select all table columns for display
- Select **View->Add Pane**, select **Conditions**
- Highlight the Conditions Pane, select
  - **View->Show Object in Table View**
    - Select all table columns for display
- Select **Options->Hide Information Area**
- Select **Options->Hide Tool Bar**

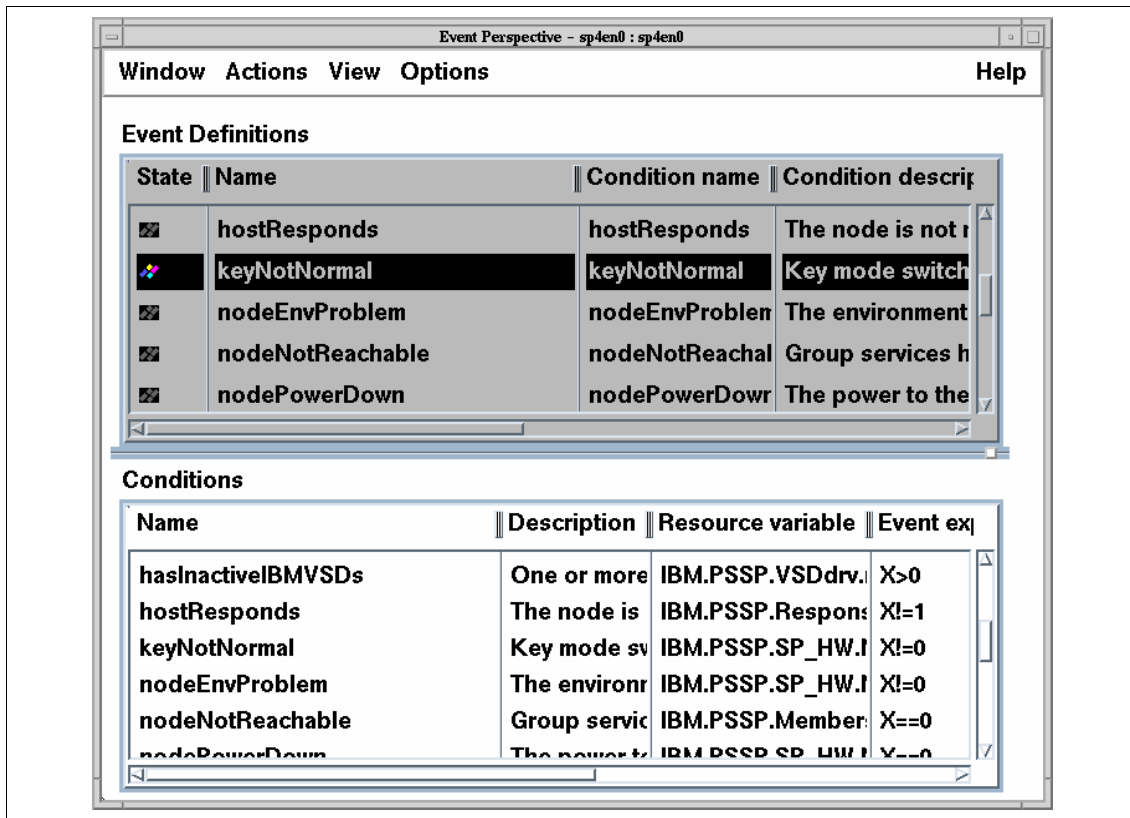


Figure 58. Event Perspective with Condition Pane in Table View

The perspective now shows the Event Definition and the Conditions panes in table view, as shown in Figure 58.

From the predefined Event Definitions, we choose keyNotNormal. This definition is used because it is very simple to test from the CWS, using the hmcnds command. Table 12 describes the attributes for an event definition.

Table 12. Column Description for Event Definitions

Column Name	Description
State	Icon that identifies the state of the event <i>Gray</i> = Not registered <i>Colored</i> = Registered, event is armed <i>Envelope</i> = Event has occurred and not rearmed See also Table 56 on page 97
Name	The event definition name
Condition name	The condition defined for the event, see Table 13
Condition description	Textual description of the event definition
Specified Resource ID elements	The elements of the resource ID to use For a list of resource IDs use SDRGetObjects EM_Resource_ID
Initial evaluation	True or false, performs an initial evaluation when the event is registered.
Registration	True or false, registers the event. If the Resource_ID element is NodeNum=*, the event is registered on all nodes. Check the riResource_name in the EM_Resource_ID class for valid element names
Notification	True or false, if true and the registered event takes place while the event perspective is running a notification window will appear with the event logged. If the event perspective is not running notification will be displayed on the next invocation.
Event resources	
Kerberos principal	The principal associated with this event.
Actions	True or false, if true an action is defined to run when the event occurs. An optional action runs if a rearm condition exist.

We register the event keyNotNormal by highlighting the keyNotNormal row and either selecting **Actions->Register** from the command bar or, if the tool

bar is present, click the left mouse button on the register button. Alternatively, double-click the left mouse button on the keyNotNormal row and select register from the notebook.

Once registered, the Registration field should change to True and the icon will become colored as shown in Figure 59. We are now monitoring the keyNotNormal event for all nodes, with notification switched on. To test the event, we issue the command `hmcmds secure 5`; this changes the key on node 5 to secure.

**Event Definitions**


State	Name	Condition name	Condition descrip
	keyNotNormal	keyNotNormal	Key mode switch



Figure 59. keyNotNormal Row Event Definition Registered

With the event activated, the State icon changes and the event notification log window opens to show the event. The icon change is shown in Figure 60

**Event Definitions**


State	Name	Condition name	Condition descrip
	keyNotNormal	keyNotNormal	Key mode switch



Figure 60. keyNotNormal Event Triggered

Change the key back to normal using the command `hmcmds normal 5`. Observe the following, the icon changes back to its colored monitor state and a rearm entry is written to the notification log.

The conditions for activating and rearming this definition are defined in the `keyNotNormal` Condition. The attributes defined for this condition are shown in Table 13.

Table 13. Column Description for `keyNotNormal` Event Condition

Column Name	Description
Name	<code>keyNotNormal</code>
Description	Text description of the <code>keyNotNormal</code> condition
Resource variable	<code>IBM.PSSP.SP_HW.Node.keyModeSwitch</code>
Event expression	<code>X!=0</code>
Rearm expression	<code>X==0</code>
Fixed resource ID elements	

The resource variable `IBM.PSSP.SP_HW.Node.keyModeSwitch` can be one of the following three values:

- 0, the key is in normal mode
- 1, the key is in secure mode
- 2, the key is in service mode

Within the expression, X refers to the variable `IBM.PSSP.SP_HW.Node.keyModeSwitch`. When `X!=0`, the key is in secure or service mode and the event is set. When `X==0`, the key is in normal mode and the event rearms. By modifying the event definition notebook actions page, you can run your own action whenever the event occurs and optionally, whenever the rearm event occurs.

### 3.4.3 Modifying a Monitored Event

As an example, we modify the `keyNotNormal` event to carry out an action for both the event and the rearm. We open the event description notebook action page as shown in Figure 61.

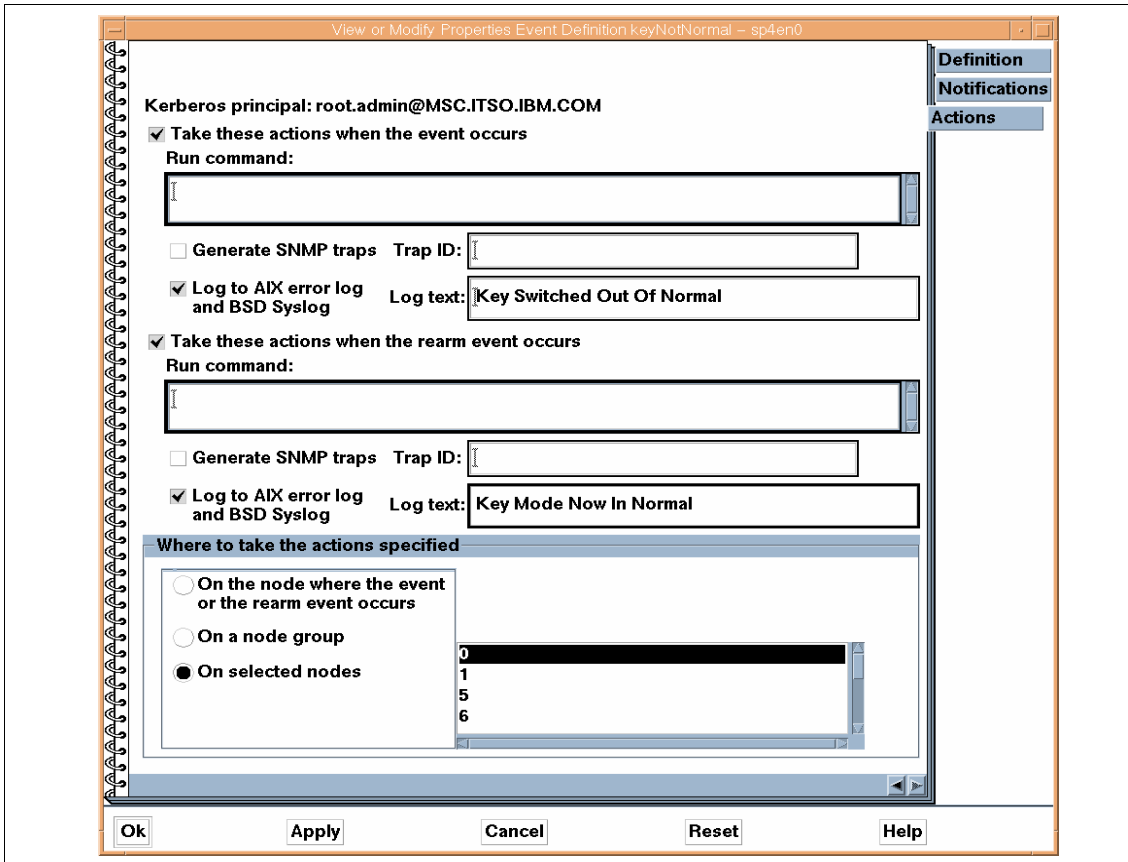


Figure 61. keyNotNormal Event Definition Notebook Actions Page

We click on the buttons to **Take these actions when the event occurs** and **Take these actions when the rearm event occurs**. For each of these actions, we select **Log to AIX error log and BSD Syslog**. In the panel alongside the radio buttons, we insert the text we want to appear in the error log. Finally, we specify where the actions take place by selecting the **On Selected Nodes** button and highlighting node **0**, the CWS. We click on the **Apply** button and the event description is modified for the above actions. After running the `hmcnds` command again, we examine the error log on the CWS to verify the modification was successful, by running the command `errpt -N pmand`.

The error log summary is as follows:

### Error Log Summary

IDENTIFIER	TIMESTAMP	T	C	RESOURCE_NAME	DESCRIPTION
6B1EC00C	0907144798	U	S	pmand	MONITORED SITUATION CLEARED
E460E36E	0907144598	U	S	pmand	MONITORED SITUATION EXISTS

The detail entry section for each of these entries is shown in Figure 62 and Figure 63. A detail entry can be displayed using the `-a` flag with the `errpt` command.

```
Detail Data
NAME
keyNotNormal
Node Number
0
RESOURCE TYPE
IBM.PSSP.SP_HW.Node.keyModeSwitch
RESOURCE NAME
NodeNum=5
SURPASSED THRESHOLD VALUE
X!=0
DESCRIPTION
Key Switched Out Of Normal
```

Figure 62. Problem Management Error Log Entry `keyNotNormal` Trigger

```
Detail Data
NAME
keyNotNormal
Node Number
0
RESOURCE TYPE
IBM.PSSP.SP_HW.Node.keyModeSwitch
RESOURCE NAME
NodeNum=5
SURPASSED THRESHOLD VALUE
X==0
DESCRIPTION
Key Mode Now In Normal
```

Figure 63. Problem Management Error Log Entry keyNotNormal Rearm

### 3.4.4 Creating A New Condition and Event Definition

This release of event perspective provides a simple interface for defining an Event Definition and creating a Condition. As an example, we create a new Condition and Event Definition and register it.

In a large SP system, there is often a large number of SSA disk drawers. We define an error log monitor to watch for a specific AIX error log entry appearing. If an SSA drawer loses a redundant power supply or cooling fan, the error is logged to the AIX error log.

The monitor we define watches the error log for entries with a resource name of *SSA\_DETECTED\_ERROR*.

From the Event perspective with the Conditions Pane highlighted, we choose the **Actions->Create** from the command bar. The create condition notebook appears as shown in Figure 64.



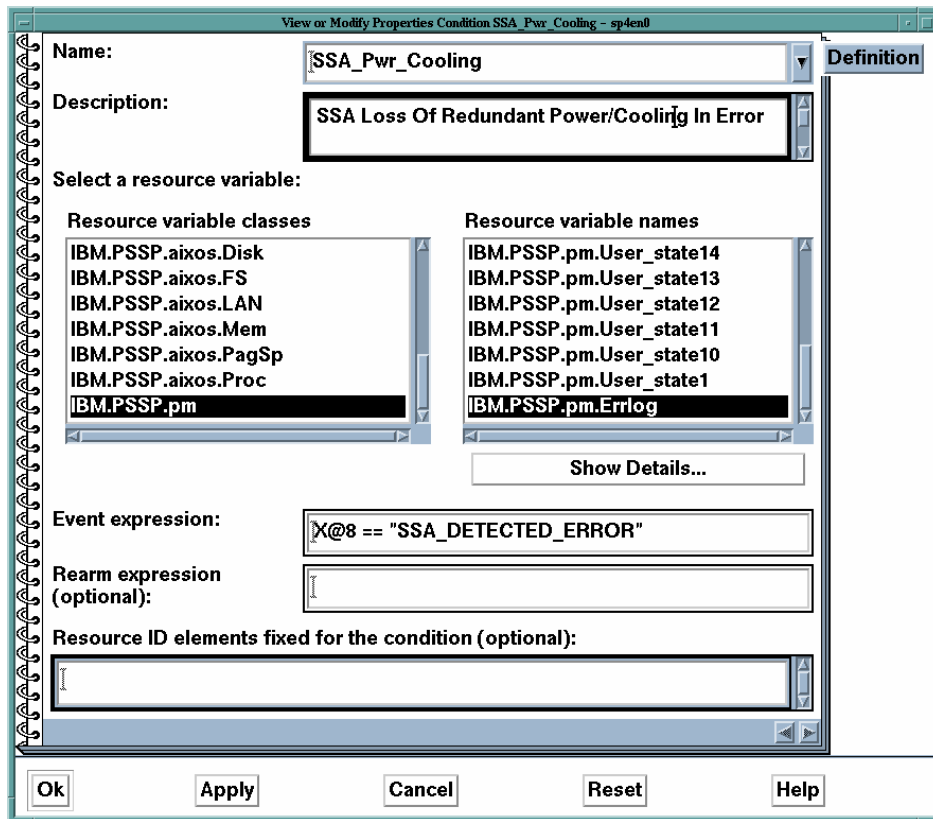


Figure 64. Creating New Condition for SSA Power Cooling Event Monitor

We fill in the name of our condition, SSA\_Pwr\_Cooling, and give it a description. The resource variable class for use with the AIX error log is IBM.PSSP.pm and the variable name IBM.PSSP.pm.Errlog. Our Event expression field is X@8=="SSA\_DETECTED\_ERROR". We do not need to define a rearm condition. The problem management system will inform us when an entry appears in the error log. Should a second entry appear, the problem management system will again react. The rearm expression allows you to monitor for a subsequent change of state and, if it occurs, to take a separate action for it. In the case of SSA\_DETECTED\_ERROR, there would never be an indication in the error log of the problem being resolved. Also, for this error, all we care about is having a custom notification of the problem. We are using the event perspective as a front end to the problem management system. Once our event definition has been registered, we no longer need the event perspective running.

By selecting the **Show Details** button, an information window appears which describes the Resource Variable. Within the Resource Variable Description pane you find a full description of the resource variable and examples of its use. These descriptions can also be shown by using the command `haemqvar`. To see the description for the resource variable in Figure 64 run the command:

```
haemqvar IBM.PSSP.pm IBM.PSSP.pm.Errlog "*" | pg
```

In this instance, the `IBM.PSSP.pm.Errlog` is a structured byte string.

These fields, taken from the error log, are held within this structured byte string and as follows:

- Sequence number (field number 0: character string data type)
- Error ID (field number 1: character string data type)
- Error class (field number 2: character string data type)
- Error type (field number 3: character string data type)
- Alert flags value (field number 4: character string data type)
- Resource name (field number 5: character string data type)
- Resource type (field number 6: character string data type)
- Resource class (field number 7: character string data type)
- Error label (field number 8: character string data type)

`X@8` refers to field 8, the Error label. Using the Error ID is also possible, but the ID may change with different AIX releases. Selecting the OK button defines our Condition and closes the window.

The Event Definition is now created and customized as shown in Figure 65 and Figure 67.

On the Definition page of the notebook:

- Enter the Event Definition name. Keep this the same as the condition name; in this instance, `SSA_Pwr_Cooling`.
- In the Condition pane we choose `SSA_Pwr_Cooling`, from the already defined conditions, by selecting the right-hand arrow of the Name field. The description is automatically filled in from the Condition definition.
- We select the **Wild-card Element Values** button so that we are monitoring all nodes.

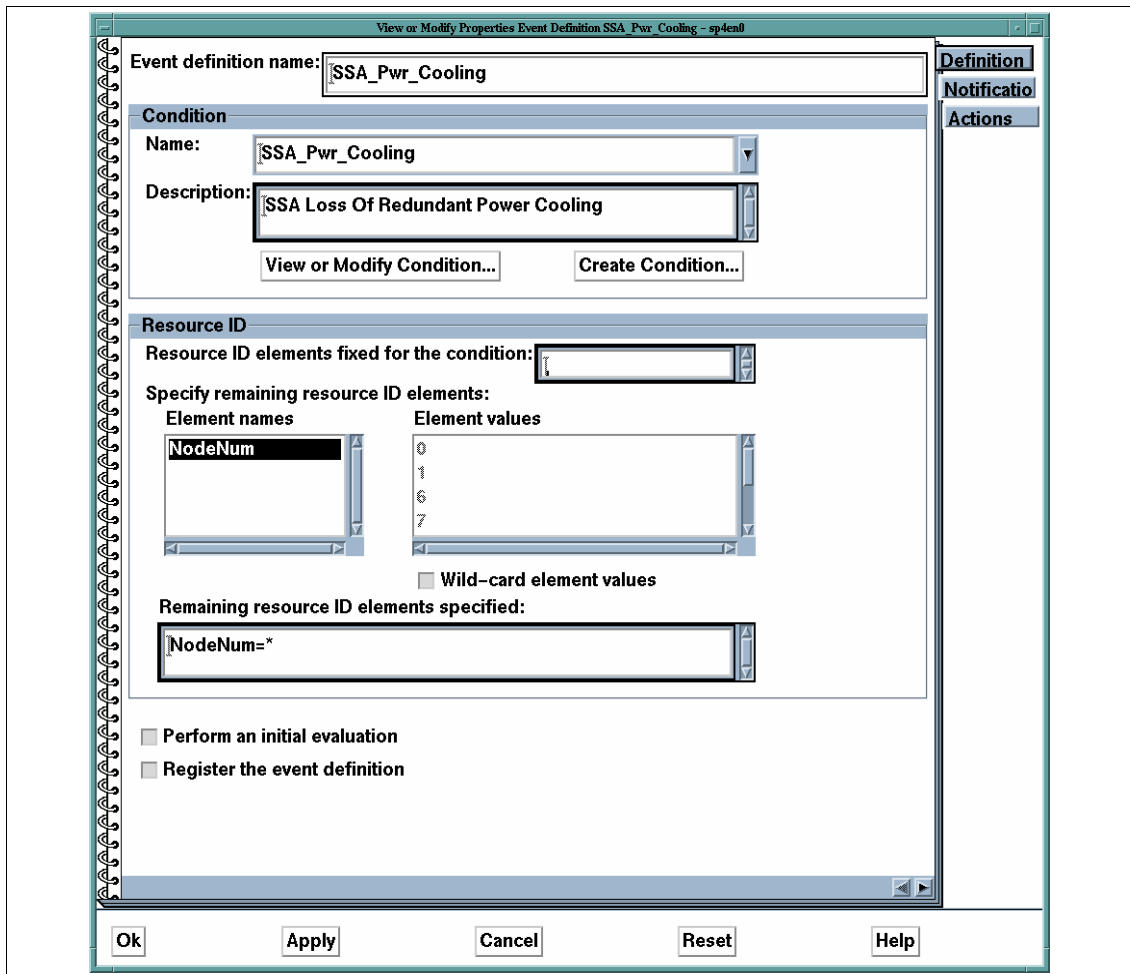


Figure 65. Event Definition for New SSA Power Cooling Event Monitor

We now turn to the notification page of the notebook. Notification occurs while we are using the event perspective. If the event perspective is not running, this built-in notification will not occur. What we require is notification to take place in a way, which we define, that has no reliance on the event perspective. Therefore, in the notification page, we deselect the notification button as shown in Figure 66.

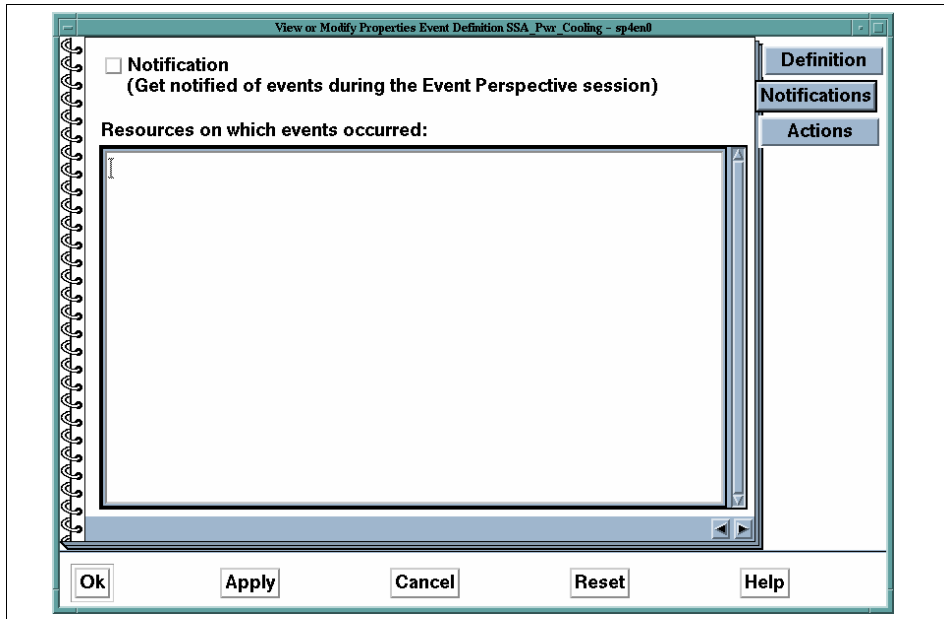


Figure 66. Event Definition Notebook Notification Page

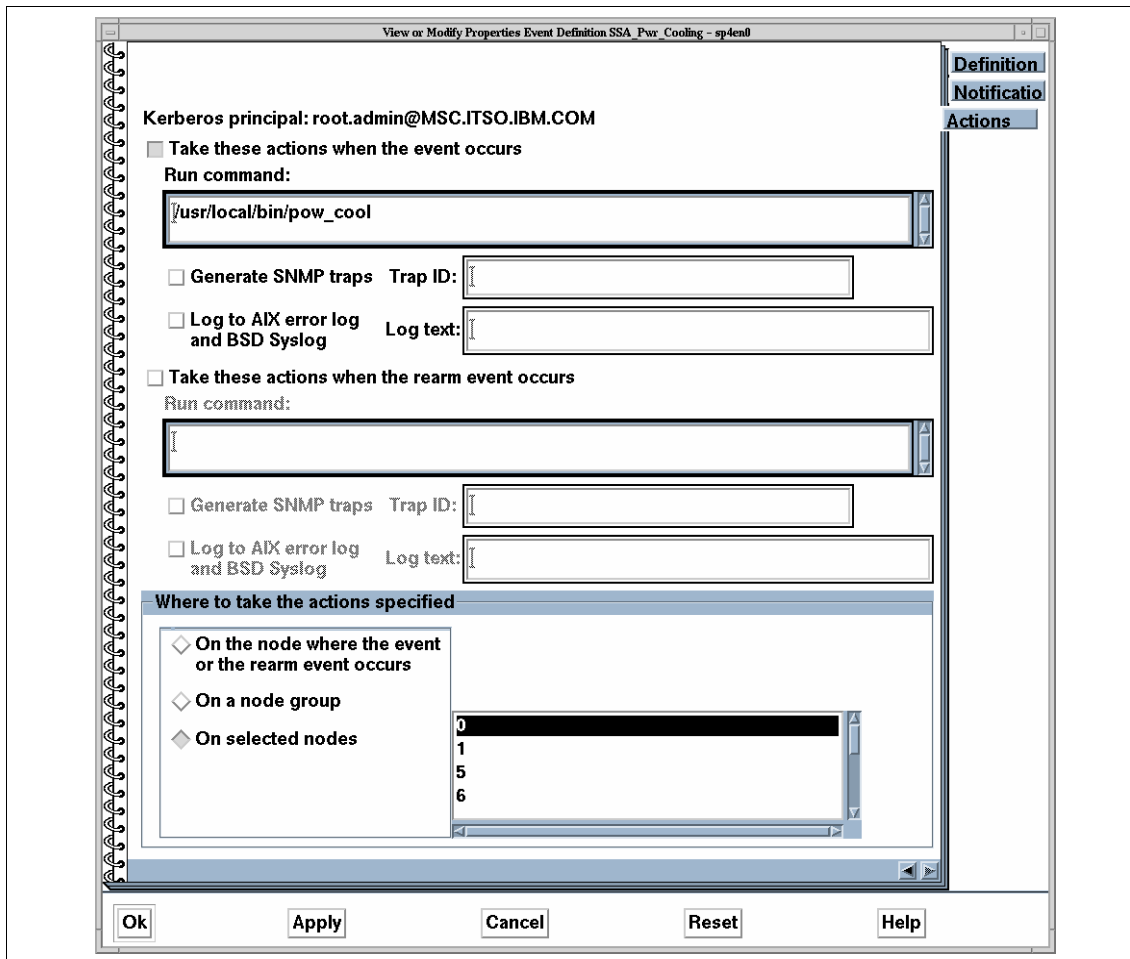


Figure 67. Notebook Actions Page for New SSA Power Cooling Event Monitor

Moving to the Action page of the notebook:

- Select the **Take these actions when the event occurs** button.
- In the **Run command** panel, we enter the name of a command on the CWS, /usr/local/bin/pow\_cool. This is a simple shell script which will pop up a window on the CWS when the event is triggered.
- In the **Where to take the actions specified** panel, select the **On selected nodes** button and highlight node **0**, the CWS

After selecting **OK** or **Apply**, we register the event in the main event perspectives window. If the loss of power/cooling is logged in the error log of

any node, we pop up a red window on the CWS with relevant data. The shell script `/usr/local/bin/pow_cool` is shown below.

```
/usr/local/bin/pow_cool
#!/bin/ksh
NODE="${PMAN_IVECTOR###*}"
trap "rm -f /tmp/node${NODE}" 0 1 2 3 15
#Create file
print "A ${PMAN_HANDLE} event occurred
Node: ${NODE}
Time: ${PMAN_TIME}
Check the error log for node ${NODE}" >/tmp/node${NODE}
# Popup the message
aixterm -display 9.12.2.176:0 -geometry 35x6+0+0 -T ${PMAN_HANDLE} \
-bg red -fg white -e pg /tmp/node${NODE}
```

Figure 68 shows the window displayed when the `/usr/local/bin/pow_cool` script runs. The script uses environment variables supplied by the problem management system.

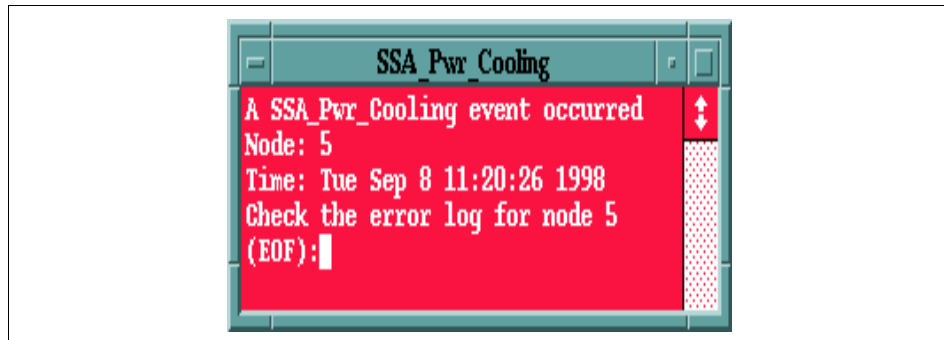


Figure 68. Warning Window Popup for New SSA Power Cooling Event Monitor

### 3.4.5 Tivoli Integration

The `tecad_pssp` is a utility command which can be executed by the problem management subsystem (pman), so that the event information will be forwarded to the Tivoli enterprise console. You will need to install it *once* on the SP system, so that it populates the System Data Repository with the necessary classes. Once installed; these classes are accessible by all nodes in the SP system.

The files required to use this utility reside in the `/usr/lpp/ssp/tecad` directory. There is a README file that describes in detail setting up and testing the link to the Tivoli enterprise console. Once the `tecad_pssp` utility is functioning correctly, it is a simple matter of using it in the **Take these actions when the event occurs** section of the Event Definitions notebook Actions page. The event definitions you have registered can forward notification of a condition trigger to the Tivoli enterprise console.

For more information on using the `tecad_pssp` command refer to the Redbook *Integrating TME 10 on the RS/6000 SP*, SG24-2071

---

## 3.5 Recoverable Virtual Shared Disk Perspectives

The tree-type display and the concept of direct and indirect clients in the Nodes pane in previous releases have been removed.

Most of the new functionality allowing Virtual Shared Disk management from the IBM Virtual Shared Disk perspectives is not available on any nodes which are running earlier releases. Nodes with PSSP versions earlier than 3.1 *cannot* use the Virtual Shared Disk perspective to do the following tasks:

- Monitor conditions related to IBM Virtual Shared Disks.
- Obtain configuration information for IBM Virtual Shared Disks and IBM HSDs in the notebooks or in table view. These pages will be blank and so will the table cells.
- Obtain the state of IBM Virtual Shared Disks or the IBM Recoverable Virtual Shared Disk subsystem.
- Use the Filter to Show Related functions.

In order to carry out these functions, the Virtual Shared Disk command line must be used.

### 3.5.1 Enhancements

- Separate Virtual Shared Disk and HSD panes and windows.

- Apply filters that limit the information displayed in each pane.
- Display object attributes in a table format.
- Support for Virtual Shared Disk commands from the Actions menu.
- New monitor conditions: `hasInactiveBMVSDs` and `rvsdInRecovery`.
- Simplified IBM Virtual Shared Disk Node properties notebook
- Refresh of Recoverable Virtual Shared Disk subsystem from the Control Recoverable Virtual Shared Disk subsystem dialog in the Actions menu. This refresh is a new option in the Recoverable Virtual Shared Disk subsystem.

### 3.5.2 Configuration and Control

In order to show the enhancements we configure a number of Virtual Shared Disks using the Virtual Shared Disk perspective. Using the Virtual Shared Disk perspective requires your Kerberos principal to be defined in the Virtual Shared Disk `sysctl` access control lists, `/etc/sysctl.vsd.acl` and the `sysctl` access control lists, `/etc/sysctl.acl`. The system administrator must run the command `sysctl svcrestart` for these changes to take effect.

Once you have the correct authorization you start the Virtual Shared Disk perspective either from the launch pad or by issuing the `spvsd` command. The initial screen should appear similar to Figure 69.



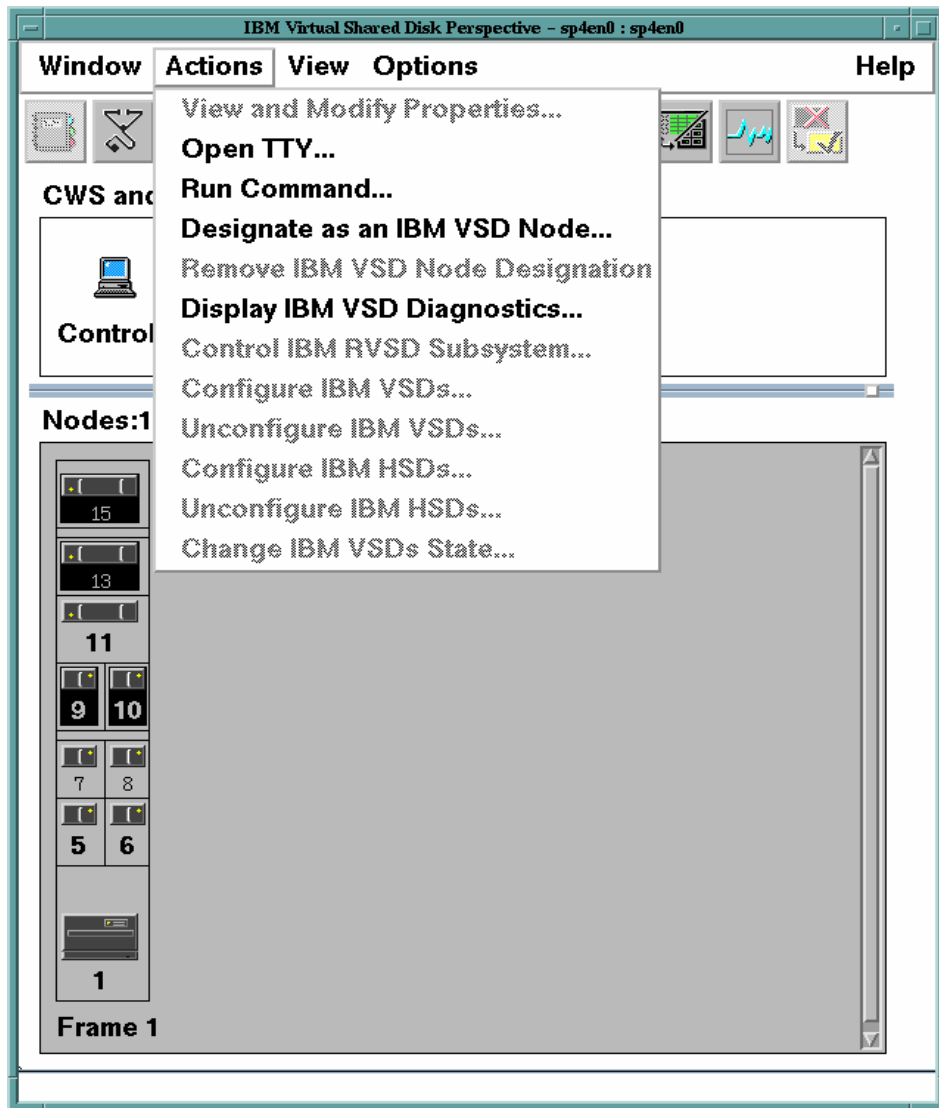


Figure 69. Virtual Shared Disk Perspective Initial Window

For any nodes to become Recoverable Virtual Shared Disk servers or clients, they must have the Virtual Shared Disk LPP installed. The Recoverable Virtual Shared Disk extension is now packaged with the standard Virtual Shared Disk code. Once the code has been installed on nodes, all control of Virtual Shared Disks can be carried out from the Virtual Shared Disk perspective. Referring to Figure 69, we select nodes 7, 8, 13 and 15 and from

the command bar select **Actions->Designate as an IBM VSD Node**. After confirming the action, those nodes will have a blue disk attached to their icon.

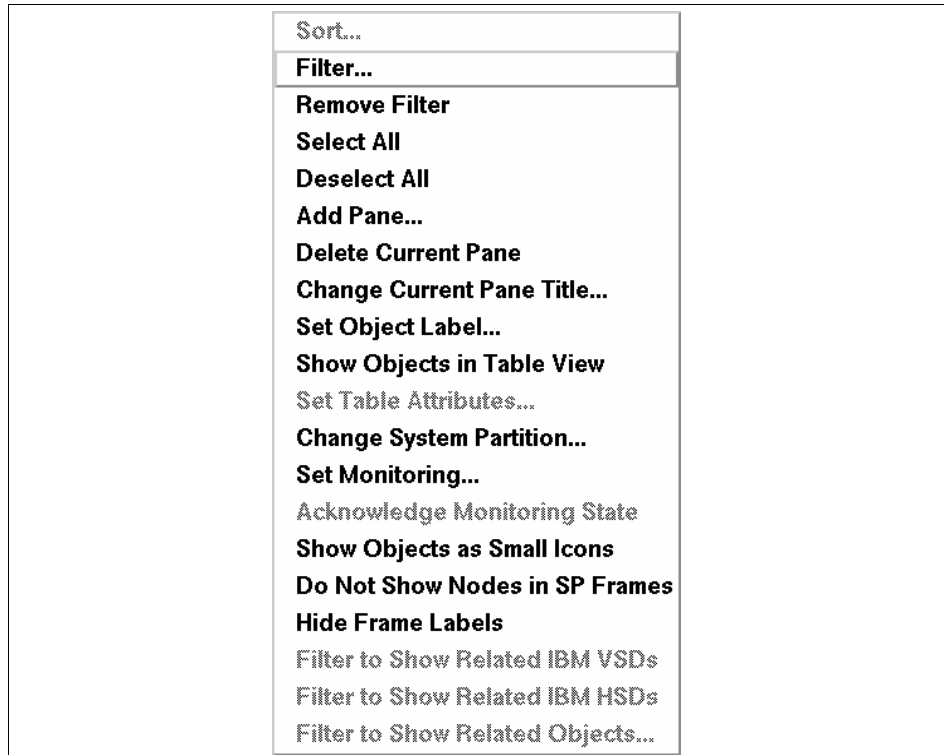


Figure 70. Virtual Shared Disk Perspective View Pull -Down

Our next step is to create Virtual Shared Disks for use by these nodes. We select **View** on the command bar, the window shown Figure 70 displays. To create new Virtual Shared Disks, you need to select a Virtual Shared Disk pane. Use the left-hand mouse button and select **Add Pane**; the window shown in Figure 71 is displayed.

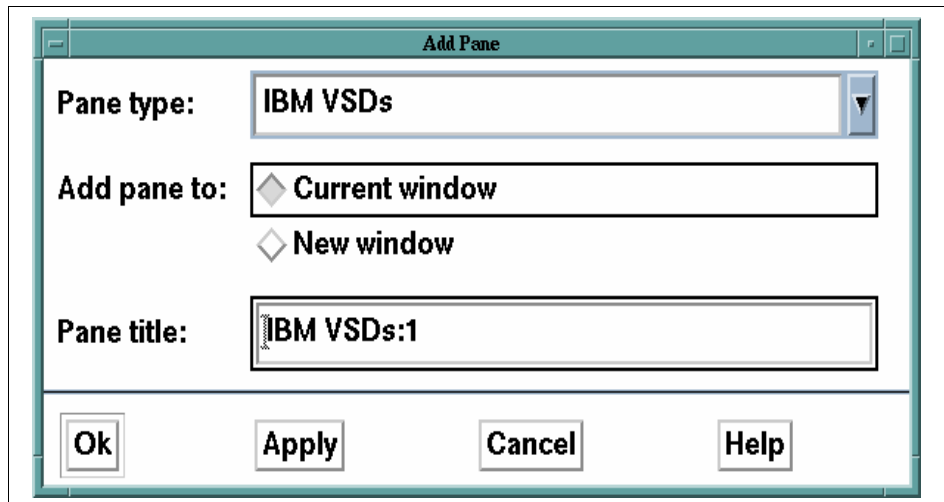


Figure 71. Add Pane to Show Virtual Shared Disks

The options available for selecting a pane are CWS and Syspars, Nodes, IBM Virtual Shared Disks and IBM HSDs. Select **IBM VSDs** as shown and click the **OK** button. The new pane is created. Make this pane, IBM VSDs:1, the current pane and select **Actions->Create** from the command bar. A new window opens for the creation of Virtual Shared Disks, as shown in Figure 72. Fill in your required parameters to create new Virtual Shared Disks.

In our example we select node 15 and hdisk1. The disk is not allocated to any volume group. We define 4 Virtual Shared Disks of 20MBs and a volume group called extvg. If the disk was in a shared disk subsystem, for example in an SSA loop, with node 13, we could use Recoverable Virtual Shared Disks by using node 13 as the backup node. We select the **OK** button. Be patient; this action now creates the volume group and logical volumes on node 15 for Virtual Shared Disk use.

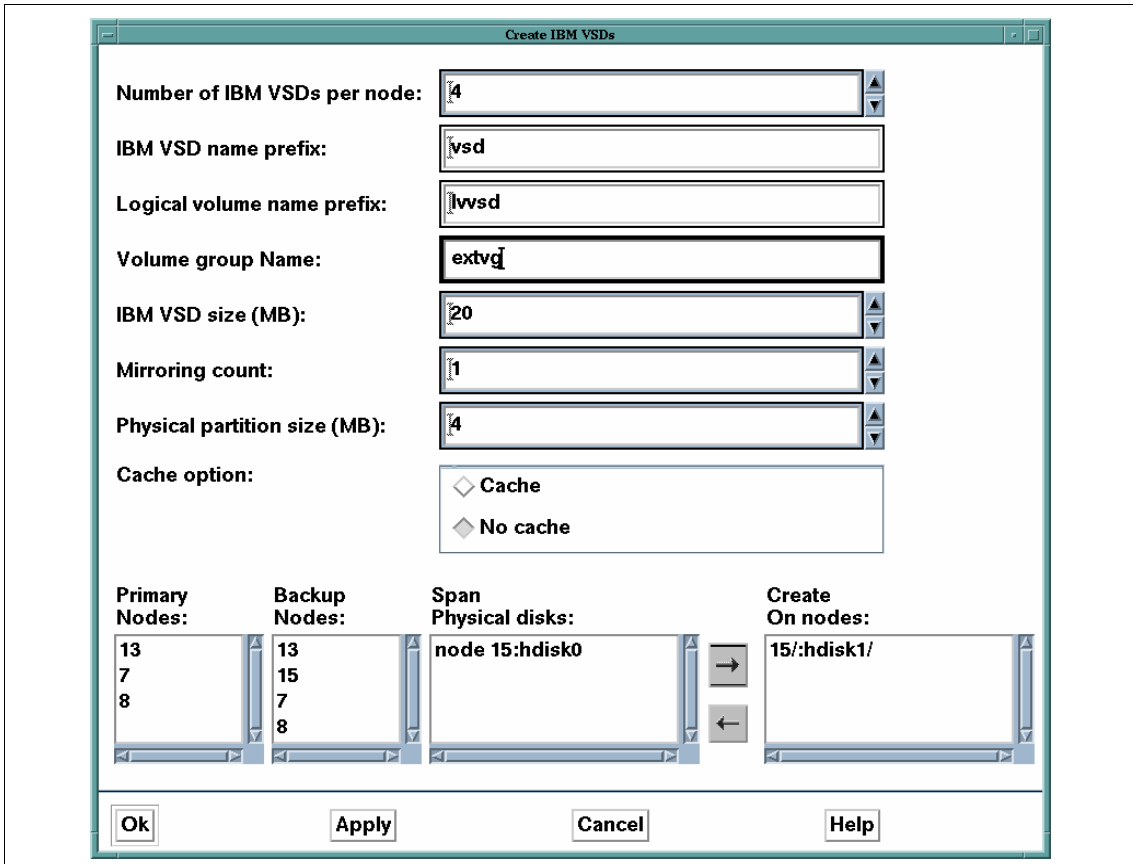


Figure 72. Creating a Virtual Shared Disk

We can see the addition of these logical volumes by running the AIX command `lsvg -l extvg`; the output is shown in Figure 73.

```

sp4n15:/ >lsvg -l extvg
extvg:
LV NAME          TYPE      LPs  PPs  PVs  LV STATE    MOUNT POINT
lvvsd1n15       jfs       5    5    1    closed/syncd N/A
lvvsd2n15       jfs       5    5    1    closed/syncd N/A
lvvsd3n15       jfs       5    5    1    closed/syncd N/A
lvvsd4n15       jfs       5    5    1    closed/syncd N/A

```

Figure 73. Output of `lsvg -l extvg`

The Virtual Shared Disk pane in Figure 74 now shows the newly created Virtual Shared Disks:

### IBM VSDs:1



Figure 74. Virtual Shared Disk Pane

We now configure the Virtual Shared Disks on different nodes. Refer to Figure 70, the Actions pull-down. Select nodes 13 and 15 and then run the **Action->Configure IBM VSDs**; a new window opens:

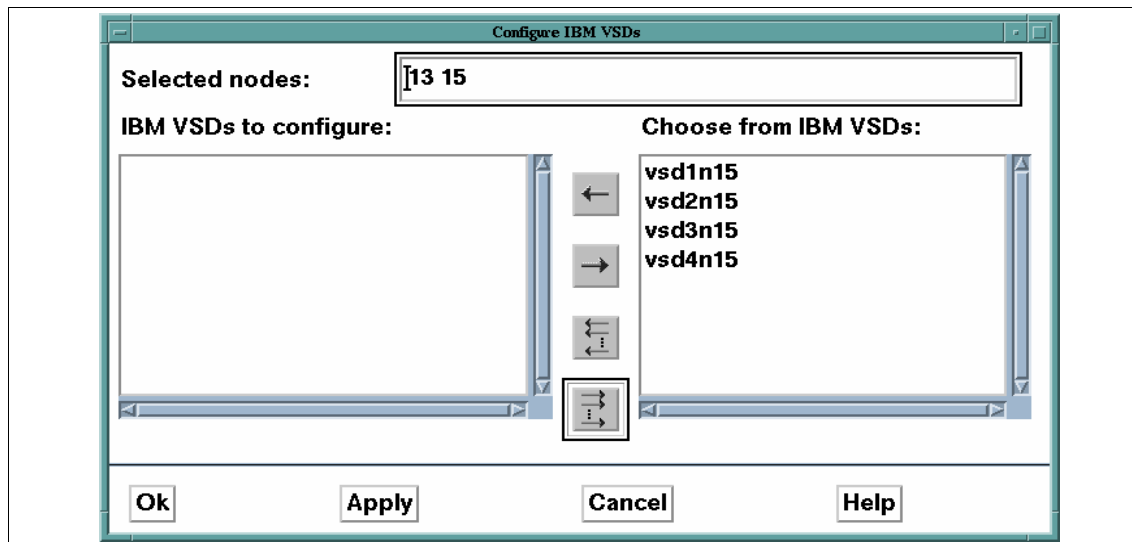


Figure 75. Configuring Virtual Shared Disks on Nodes

We select all the Virtual Shared Disks and press **OK**. All the Virtual Shared Disks are now configured on nodes 13 and 15. We also configure vsd1n15 and vsd3n15 on node 7, plus vsd2n15 and vsd4n15 on node 8. After modifying the Virtual Shared Disk perspective using the following selections, we have the display shown in Figure 76.

- Highlight the CWS and Syspars pane, select **View->Delete Current Pane**.
- Use the left mouse button; select nodes 7, 8, 13 and 15.

- Select **View->Filter->Filter the selected objects in the pane->OK.**
- Select **View->Show Objects in Table View.**
- From the Select Table Attributes window, choose:
  - **Active IBM VSDs count.**
  - **Suspended IBM VSDs count.**
  - **Stopped IBM VSDs count.**

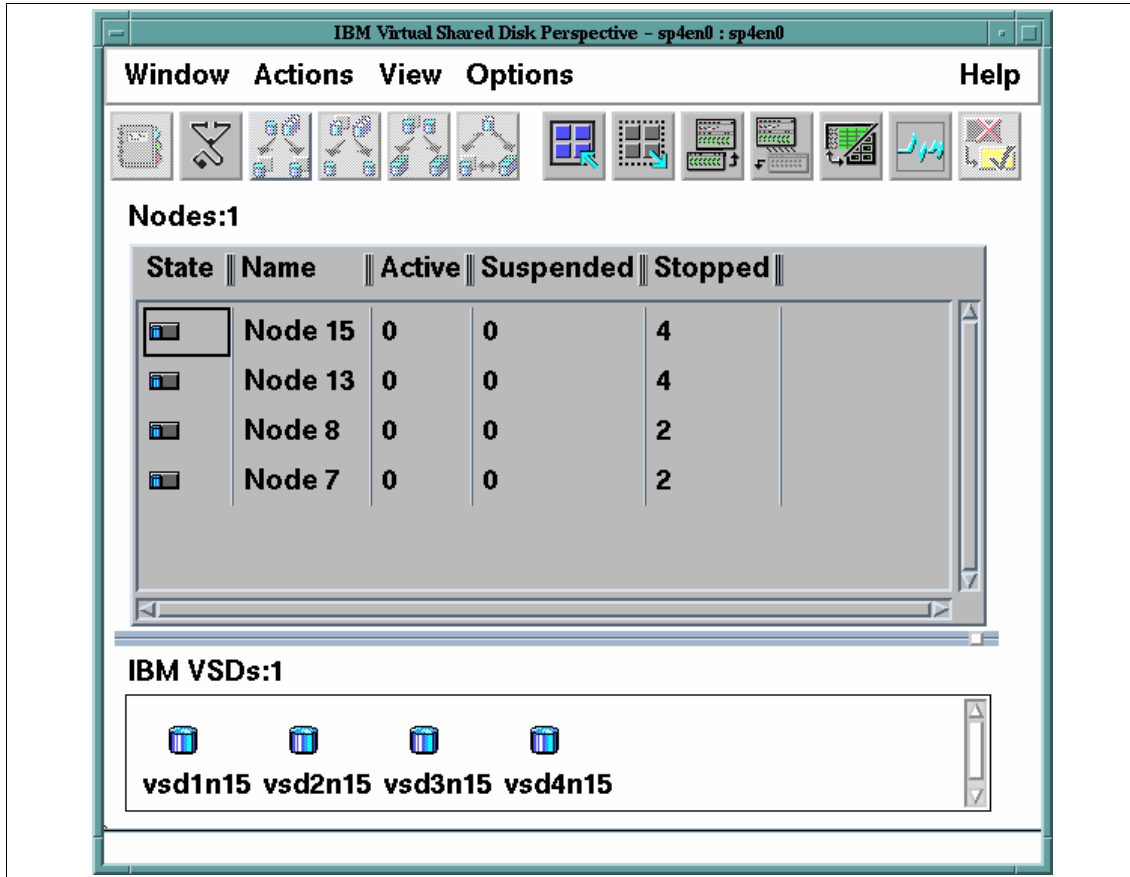


Figure 76. Configured Virtual Shared Disks

Control of the Virtual Shared Disks on the nodes can now be undertaken from the Actions pull-down; refer to Figure 70 on page 116.

### 3.5.3 Monitoring

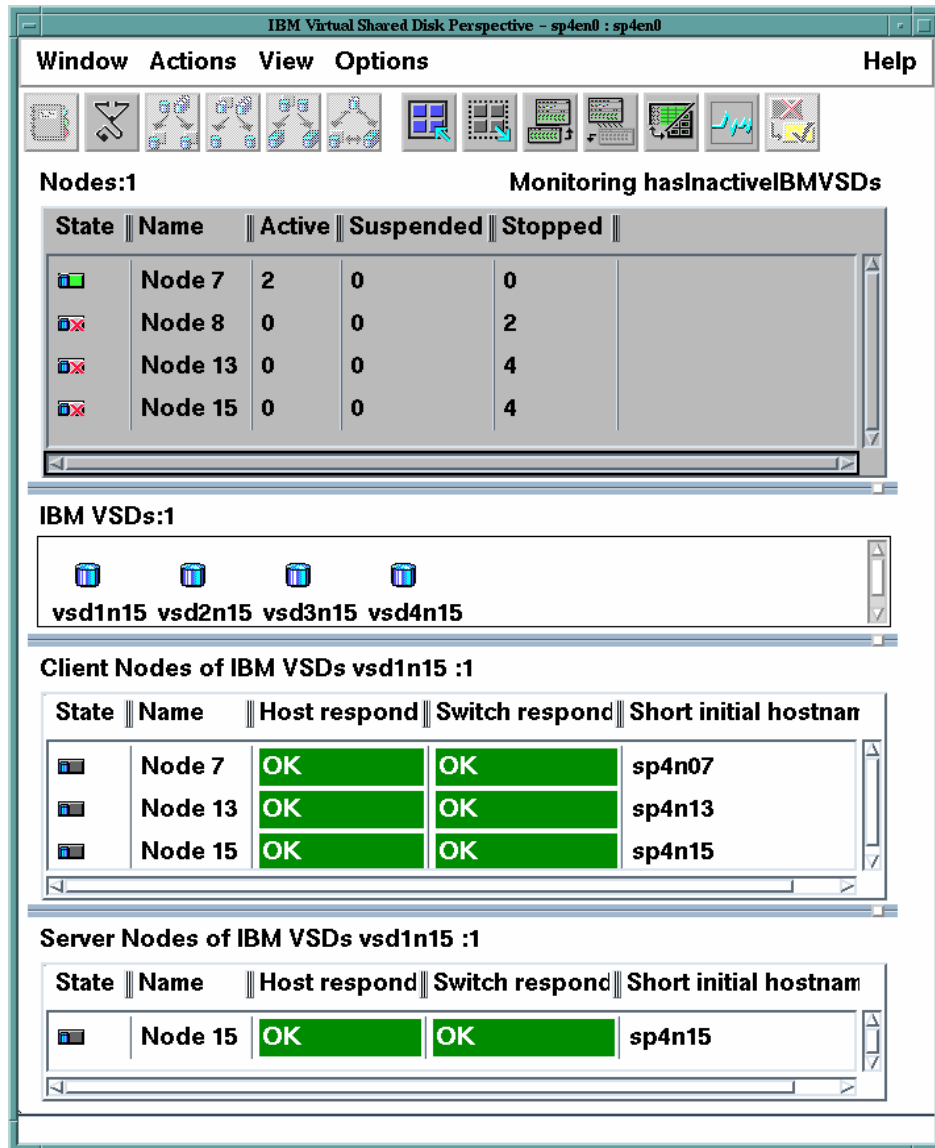


Figure 77. Virtual Shared Disk Perspective Monitoring hasInactiveIBMVSDs

Figure 77 shows a Virtual Shared Disk perspective window for our current system. We have four defined panes:

- The first two panes are as we defined in Figure 76, our configured Virtual Shared Disk display. We make just a single change to this display: activate monitoring through **Actions->Set Monitoring** or by pushing the Set monitoring button on the tool bar. From the set monitoring window, we select **hasInactiveBMVSDs**. This updates the state column and also allows us to acknowledge the state of any node where the condition is true.
- The third pane is activated by selecting **Actions->Filter to Show Related Objects**, and selecting client nodes. The related objects window is shown in Figure 78.
- The fourth pane is activated by selecting **Actions->Filter to Show Related Objects**, and selecting server nodes. The related objects window is shown in Figure 78.



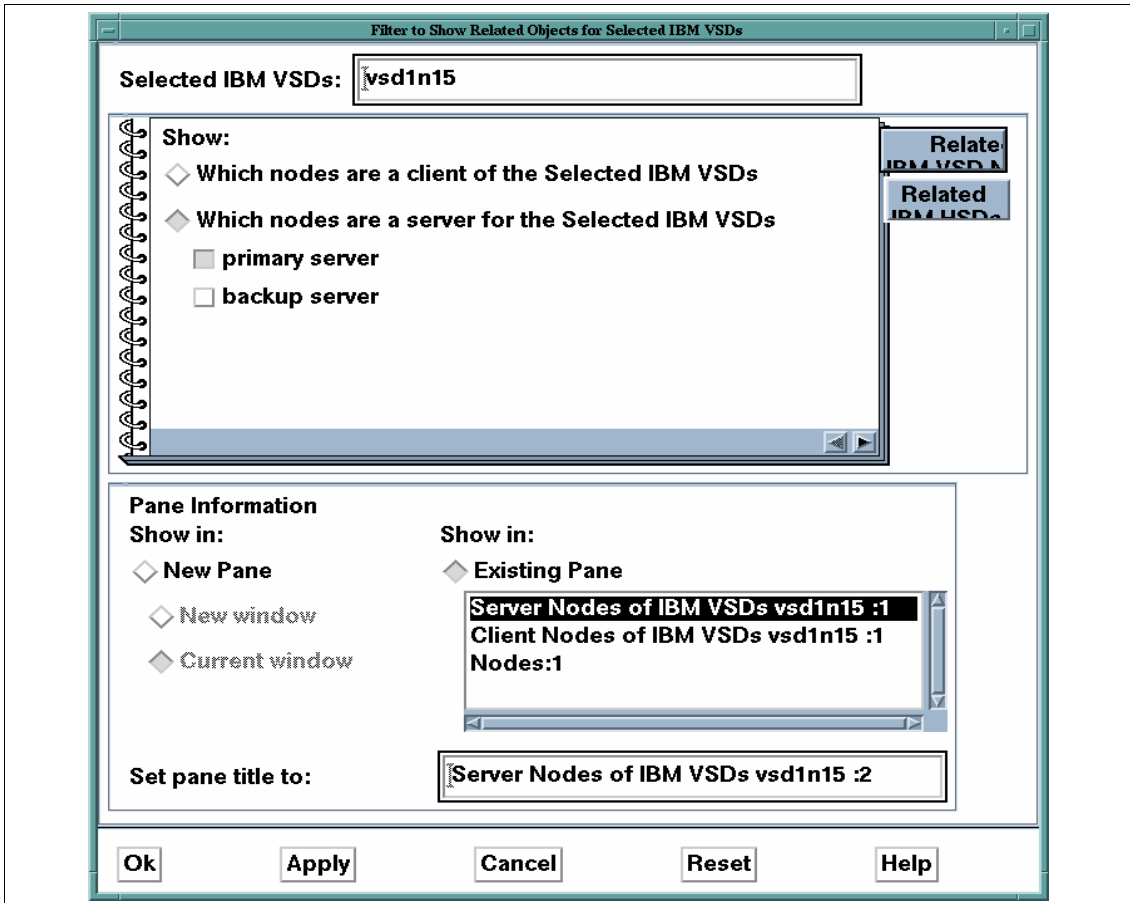


Figure 78. Filter to Show Related Objects

With monitoring enabled, we can iconify the Virtual Shared Disk perspective. The icon will change color as the monitored state changes. See Figure 79 for the state changes of the Virtual Shared Disk perspective icon.

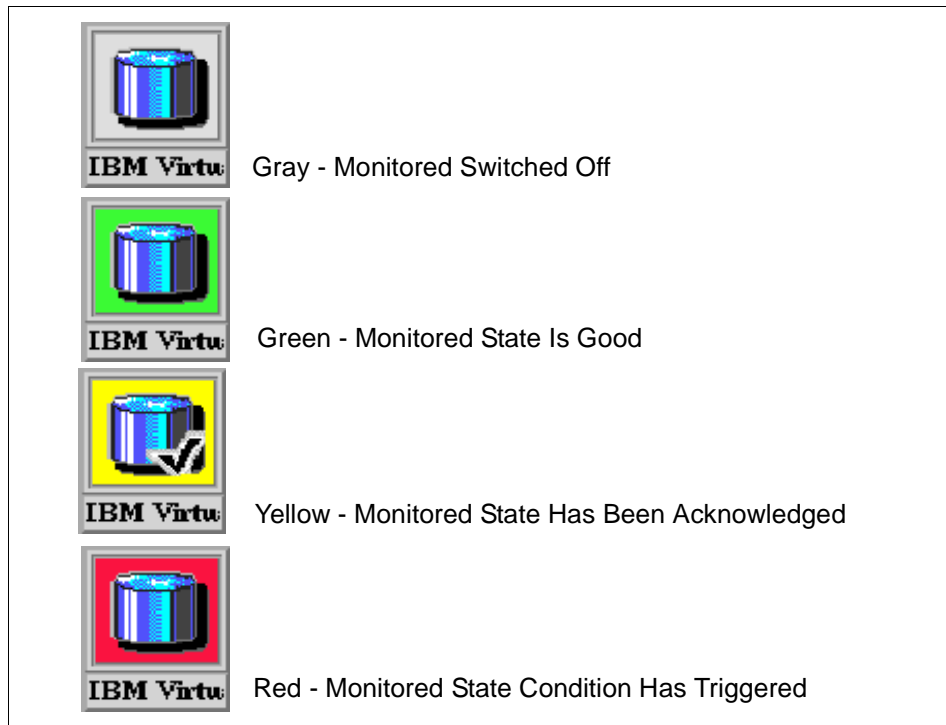


Figure 79. State Changes to the Virtual Shared Disk Perspective Icon

---

## Chapter 4. SP-Attached Server Support

PSSP 3.1 provides support for newly announced hardware, the RS/6000 Enterprise Server Models S70 and S7A, known as SP-attached servers. These are high-end RS/6000 PCI-based, and are the first 64-bit SMP architecture nodes that attach independently to the SP, as they are simply too large to physically reside in an SP frame.

These servers excel in capacity and scalability in on-line transaction processing (OLTP), Enterprise Resource Planning (ERP), server consolidation, Supply Chain Management, and large database server applications such as SAP.

A seamless integration of these servers into the SP system was the design requirement for PSSP 3.1. Therefore, some changes to PSSP are necessary to attach these servers to your SP environment, and to treat them as standard SP nodes that are physically not located in an SP frame.

The changes implemented in PSSP 3.1 to support SP-attached servers are discussed in this chapter and are subdivided into the following five sections:

1. The system attachment of the SP-attached server to the SP is discussed in "Hardware Attachment" on page 125.
2. The changes in installation and configuration of an SP-attached server are discussed in "Installation and Configuration" on page 136.
3. The changes in the PSSP software to support the SP-attached server are discussed in "Software Changes" on page 142.
4. The changes in the different user interface panels and commands are discussed in "Changes in the User Interface" on page 151.
5. Different attachment scenarios to the SP are discussed in "Attachment Scenarios" on page 156.

---

### 4.1 Hardware Attachment

In this section, we describe the hardware architecture of the SP-attached server and its attachment to the SP system, including areas of potential concern of the hardware or the attachment components.

### 4.1.1 SP-Attached Server

The RS/6000 Enterprise Server Model S70 (7017) is a 64-bit symmetric multiprocessing (SMP) system that supports 32- and 64-bit applications concurrently.

Until now, all nodes in an SP environment resided within the slot location of an SP frame. However, the SP-attached server is physically too large to reside in an SP frame slot location, as it is packaged in two side-by-side rack units, as shown in Figure 80 on page 127.

The first unit is a 22w x 41d x 62h-inch (56w x 104d x 157h-cm) Central Electronics Complex (CEC). The CEC system rack contains:

- A minimum of one processor card, and a maximum of three processor cards, with a 4-, 8-, or 12-way PowerPC processor configuration. The system can contain up to a maximum of 12 processors, sharing common system memory.
- Each processor card has four 64-bit processors operating at 125Mhz or 262Mhz.
- A 4MB ECC L2 cache memory per 125Mhz processor, and an 8MB per 262 Mhz processor.
- System memory is controlled through a multiport controller which supports up to 20 memory slots. All the system memory is contained in the system rack, up to a maximum of 16GB.
- An operator panel that consists of the display unit, scroll up and down push-button, an Enter button, and two indicator LEDs. The power on/off button is also located on the operator panel. In addition, it contains a port that can be used via an RS-232 cable to communicate to the S70. The operator panel is used for selecting boot options and initiating system dumps as well as for service functions and diagnostic support of the entire system.
- Reliability from redundant fans, hot-swappable disk drives, power supplies and fans, and a built-in service processor.

The second unit is a standard I/O rack, similar in size to the CEC. Each I/O rack accommodates up to two I/O drawers, with a maximum of four drawers per system. Up to three more I/O racks can be added to a system. The base I/O drawer contains:

- Up to 14 PCI slots per drawer.
- Drawer 0 reserves slots 2 and 8 for support of system media.
- Service processor and hot-pluggable DASD.

- Drawers 1 through 3 are reserved for supported PCI adapters.
- One fully configured system of four I/O drawers and up to 56 PCI slots.
- Support for SCSI/SSA 6-packs, looped SSA and SIO.

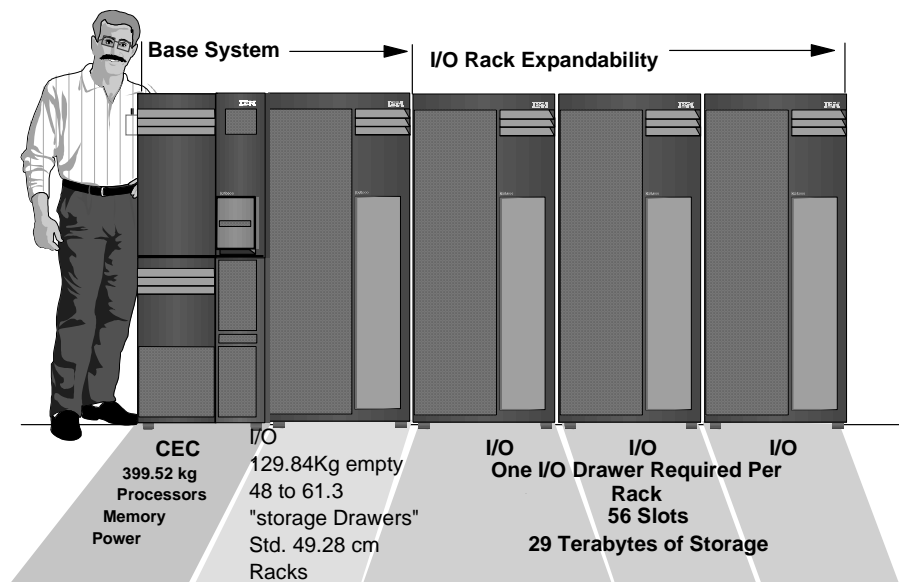


Figure 80. The S70 Components

Since the CEC and I/O racks are so large, the SP-attached server must be attached to the SP system externally.

#### 4.1.2 SP-Attached Server Attachment

This section describes the attachment of the SP-attached server to the SP, highlighting the potential areas of concern that must be met before installation. The physical attachment is subdivided and described in three connections:

- Connections between the CWS and the SP-attached server are described in "Control Workstation Connections" on page 130.
- Connections between the SP Frame and the SP-attached Server are described in "SP Frame Connections" on page 132.
- An optional connection between the SP Switch and the SP-attached server are described in "Switch Connection (Optional)" on page 132.

These connections are illustrated in Figure 81 on page 128.

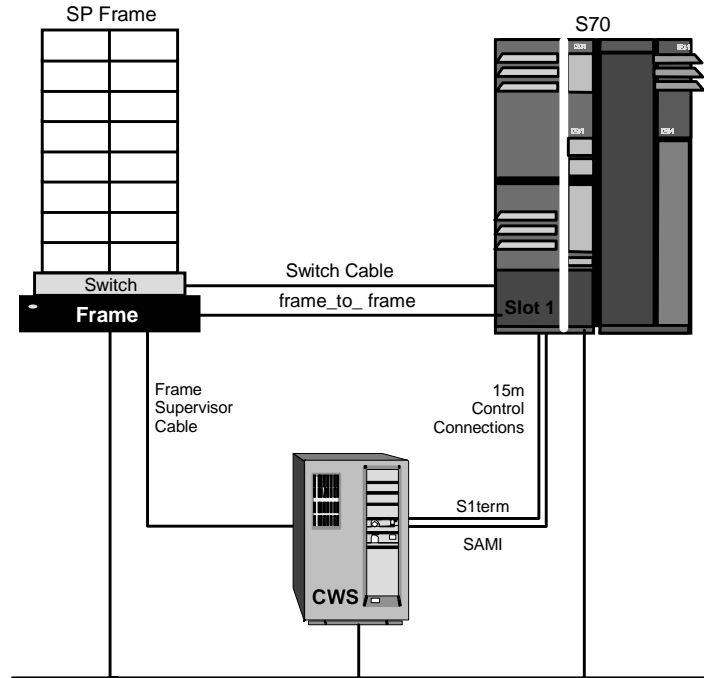


Figure 81. The S70 Attachment to the SP

The following diagram outlines the two RS-232 connections to the S70 machine.

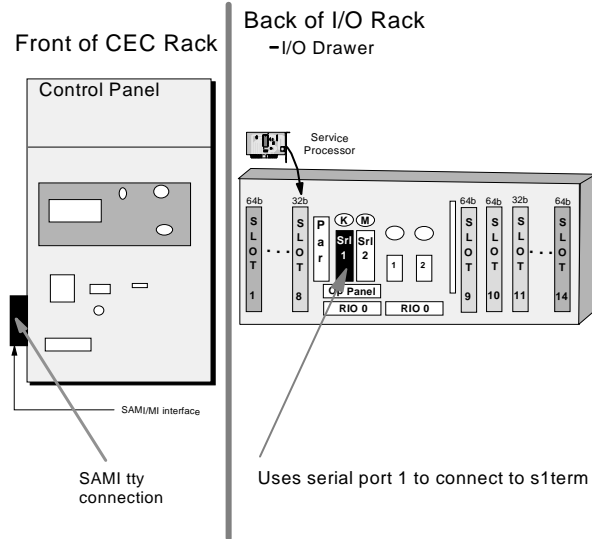


Figure 82. RS-232 Connections to the S70

It is important to note that the size of the S70 prohibits it from being physically mounted in the SP frame. Since the SP-attached server is mounted in its own rack, and is directly attached to the CWS using RS-232, the SP system must view the SP-attached server as a frame. The SP-attached server is also viewed as a node; because the PSSP code runs on the machine, it is managed by the CWS and you can run standard applications on the SP-attached server. Therefore, the SP system views the SP-attached server as an object with both frame and node characteristics.

However, as the SP-attached server does not have *full* SP frame characteristics, it cannot be considered as a standard SP expansion frame. Therefore, when assigning the server's frame number, you have to abide by the following rules:

- The SP-attached server cannot be the first frame in the SP system.
- The SP-attached server cannot be inserted between a switch configured frame and any non-switched expansion frame using that switch. It can, however, be inserted between two switch-configured frames. Different attachment configurations are described in 4.5, "Attachment Scenarios" on page 156.

Once the frame number has been assigned, the server's node numbers, which are based on the frame number, are automatically generated. The following system defaults are used:

- The SP-attached server is viewed as a single frame containing a single node.
- The SP-attached server occupies slot one position.
- Each SP-attached server installed in the SP system subtracts one node from the total node count allowed in the system. However, as the server has frame-like features, it reserves sixteen node numbers that are used in determining the node number of nodes placed after the attached server. The algorithm for calculating the node\_number is demonstrated in Figure 83 on page 130; for further information on the frame numbering issue, refer to Figure 99 on page 156:

$$\text{node\_number} = (\text{frame\_number} - 1) * 16 + \text{slot\_number}$$

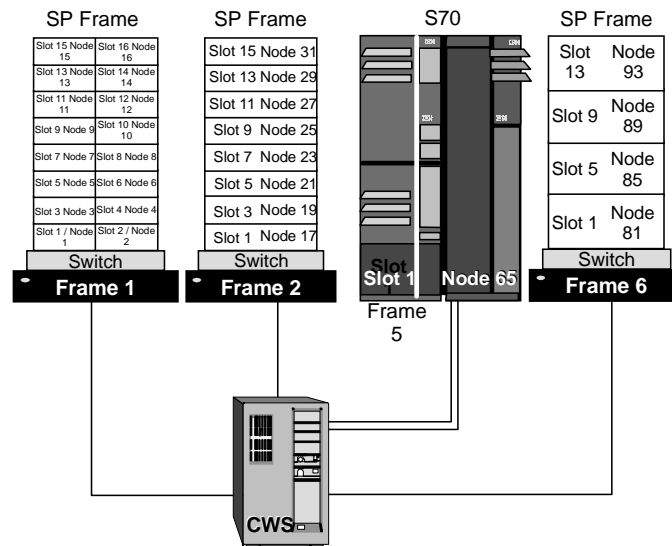


Figure 83. Node Numbering

#### 4.1.2.1 Control Workstation Connections

The SP-attached server does not have a frame or node supervisor card, which limits the full hardware, control and monitoring capabilities of the server from the SP CWS (unlike other SP nodes). However, it does have some basic capabilities, such as power on/off.



Three CWS connections to the SP-attached server are required for hardware control and software management:

- An Ethernet connection to the SP-LAN for system administration purposes.
- A custom-built RS-232 cable connected from the S70 operator panel to a serial port on the CWS. It is used to emulate operator input at the operator panel. An S70-specific protocol is used to monitor and control the S70 hardware. This protocol is known as Service and Manufacturing Interface (SAMI).
- A second custom-built RS-232 cable that must only use the S70 S1 serial port. This is used to support the s1term connectivity. This is a custom-built RS-232 cable, which is part of the order features, with a null modem and a gender-bender.

### **CWS Considerations**

In connecting the SP-attached server to the CWS, it is important to keep the following CWS areas of concern in mind:

- When connecting the SP-attached frame to the system, you need to make sure that the CWS has enough spare serial ports to support the additional connections. However, it is important to note that there is one restriction with the 16-port RS-232 connection. By design, it does not pass the required ClearToSend signal to the SAMI port of the SP-attached server, and therefore the *16-port RS-232 cannot be used* for the RS-232 connectivity to the SP-attached server. The 8-port and the 128-port varieties will support the required signal for connectivity to the SP-attached server.
- There are two RS-232 attachments for each S70/S7A SP-attachment. The first serial port on the S70/S7A *must* be used for S1TERM connectivity.
- Floor placement planning to account for the effective usable length of RS-232 cable.

The CWS-to-S70 connection cables are 15 meters in length, but only 11.5 meters is effective. So, the S70 must be placed at a distance where the RS-232 cable to the CWS is usable.

- In a HACWS environment, there will be no S70 control from the backup CWS. In the case where a failover occurs to the backup CWS, hardmon and s1term support of the S70 is not available until fail back to the primary CWS. The node will still be operational with switch communications and SP Ethernet support.

#### 4.1.2.2 SP Frame Connections

The SP-attached server connection to the SP frame is as follows:

- 10 meter frame-to-frame electrical ground cable.

The entire SP system must be at the same electrical potential. Therefore, the frame-to-frame ground cables provided with the S70 server must be used between the SP system and the S70 server, in addition to the S70 server electrical ground.

#### **Frame Considerations**

In connecting the SP-attached server to the SP Frame, it is important to have the following in mind:

- The SP system must be a *tall frame*, as the 49inch short "LowBoy" frames are not supported for the SP-attachment.
- The tall frame with the 8 port switch is not allowed
- The SP-attached server *cannot* be the first frame in the SP system. So, the first frame in the SP system must be an SP frame containing at least one node. This is necessary for the SDR\_config code, which needs to determine whether the frame is with or without a switch.
- Maximum of 8 SP-attached servers are supported in one SP system. This means that if a switch is installed, there must be 8 available switch connections in the SP system, one per SP-attached server

For complete power planning information, refer to *Site and Hardware Planning Information, Volume 1, SA38-0508*.

#### 4.1.2.3 Switch Connection (Optional)

This is an optional connection, if the SP-attached server is to be connected to a switched SP system:

- The TB3PCI adapter, known as the RS/6000 SP system attachment adapter, of the SP-attached server connects to the 16-port SP switch, via a 10 meter switch cable.

This TB3PCI adapter is used in those systems that are connected to the switch board using a PCI adapter, and it has the following characteristics:

- It is driven by a 99Mhz 603e PowerPC processor.
- It has a sustained bandwidth of 85MByte/sec.
- It has components familiar to the SP environment.
- Its device driver is derived from TB3MX.
- It is supported *only* in the S70 server family.

### **Switch Considerations**

In connecting the SP-attached server to the SP Switch, it is important to note the following:

- The High Performance switch (HiPS) cannot be used with an SP-attached server since this switch is not supported in PSSP 3.1.
- The S70/S7A servers will be the first, and currently the only, nodes attached to the switch using an RS/6000 SP Attachment adapter.
- Only *one* RS/6000 SP Attachment adapter is allowed per SP-attached server.
- The RS/6000 SP Attachment adapter that is placed in the SP-attached server requires:
  - One valid, unused switch port on the SP switch, corresponding to a legitimate node slot in your SP configuration.
  - The SP attachment adapter reserves 3 media slots in the I/O tower of the S70 server, and has the following placement restrictions:
    - Must be installed in slot 10 of the SP-attached server's I/O tower
    - Slot 9 must be left open to ensure that the adapter has sufficient bandwidth.
    - Slot 11 must be left open to provide clearance for the switch adapter's heat sinks.

These restrictions are illustrated in Figure 84., "S70 Switch Adapter Attachment Slot" on page 134.

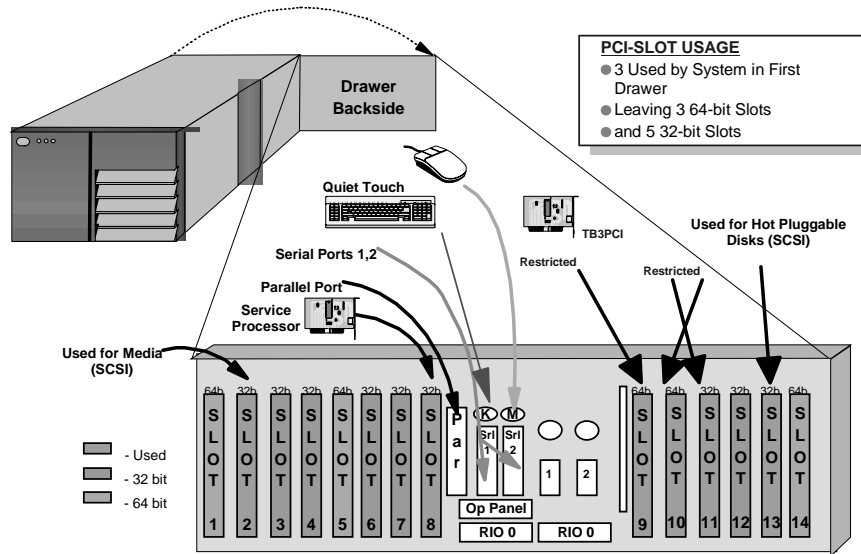


Figure 84. S70 Switch Adapter Attachment Slot

- Floor placement planning to account for the effective usable switch cable.

The SP switch-to S70 connection cable is 10 meters in length but only 6.5 meters is effective. So, the S70 switch adapter located in slot 10 must be within 6.5 meters of the SP switch, as illustrated in Figure 85 on page 135.

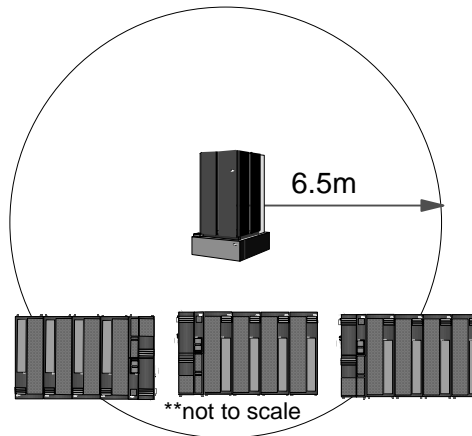


Figure 85. S70 Floor Placement

### **SP-Attached Server Considerations**

In connecting the SP-attached server to the SP system, it is important to have in mind the following potential concerns:

- Supported Adapters

All adapters currently supported in the SP environment are supported with the SP-Attached Servers (S70). However, not all currently supported SP-attached server adapters are supported in the SP switch-attached server environment. If the S70 possesses adapters that are not currently supported in the SP environment, so they *must* be removed from the SP-attached server.

The following is a list of supported adapters:

- F/C 2741 FDDI SK-NET LP SAS
- F/C 2742 FDDI SK-NET LP DAS
- F/C 2743 FDDI SK-NET UP SAS
- F/C 2751 S/390 ESCON Channel Adapter
- F/C 2920 Token Ring Auto Lanstream
- F/C 2943 EIA 232/RS-422 8-port Asynchronous Adapter
- F/C 2944 WAN RS-232 128-port
- F/C 2962 2-port Multiprotocol X.25 Adapter
- F/C 2963 ATM 155 TURBOWAYS UTP
- F/C 2968 Ethernet 10/100 MB
- F/C 2985 Ethernet 10 MB BNC
- F/C 2987 Ethernet 10 MB AUI
- F/C 2988 ATM 155 MMF

- F/C 6206 Ultra SCSI SE
  - F/C 6207 Ultra SCSI DE
  - F/C 6208 SCSI-2 F/W SE
  - F/C 6209 SCSI-2 F/W DE
  - F/C 6215 SSA RAID 5
- SP-attached server Ethernet required as en0:

For the S70 server, only the 10Mbps BNC or the 10Mbps AUI Ethernet adapters are supported for SP-LAN communication, in accordance with the existing SP-LAN configuration. Note that the BNC adapters provides the BNC cables, but the AUI ethernet adapter does *not* provide the twisted pair cables.

The SP-LAN adapter must be configured as the en0 adapter of the SP-attached server (that is, the lowest numbered Ethernet bus slot in the first I/O tower).
  - Minimum code requirements:

The CWS and SP-attached server must be running AIX 4.3.2 and PSSP 3.1 at the minimum. Hence an existing S70 may require an AIX upgrade before installation of PSSP 3.1, to achieve SP-attachment.

**Each SP-attached server S70 must have a PSSP 3.1 licence, separately chargeable against each S70's serial number.**

---

## 4.2 Installation and Configuration

The SP-attached server is treated as similarly as possible to a frame with a node. However, there are some important distinctions that have to be addressed during SP-attached server configuration, namely the lack of frame and node supervisor cards and support for two TTYs instead of one, as described in 4.1.2, "SP-Attached Server Attachment" on page 127.

Information that is unique to the SP-attached server is entered in the configuration of this server. Once the administrator configures the necessary information about the SP-attached server processor in the SDR, then the installation should proceed the same as any standard SP node in the SP administrative network.

### ***Configuration Considerations***

- Add two TTYs on the CWS.
- Define the Ethernet adapter on the SP-attached server.
- In a switched system, configure the SP-attached server to the SP Switch.

- Frame definition of SP-attached server:

The rules for assigning the frame number of the SP-attached server are detailed in section 4.1.2, “SP-Attached Server Attachment” on page 127.

The SP-attached server must be defined to PSSP, using the `spframe` command, using the new options that are available for SP-attached server for this command:

```
/usr/lpp/ssp/bin/spframe -p {hardware protocol}
-n {starting_switch_port}
[-r {yes|no}] [-s {sltty}]
start_frame frame_count starting_tty_port
```

Alternatively, you can use the `smitty nonsp_frame_dialog` menu, as shown in Figure 86.

Non-SP Frame Information

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

	[Entry Fields]	
* Start Frame	[ ]	#
* Frame Count	[ ]	#
* Starting Frame tty port	[/dev/tty0]	
* Starting Switch Port Number	[ ]	#
sl tty port	[ ]	
* Frame Hardware Protocol	[SAMI]	
Re-initialize the System Data Repository	no	+

F1=Help	F2=Refresh	F3=Cancel	F4=List
Esc+5=Reset	Esc+6=Command	Esc+7=Edit	Esc+8=Image
Esc+9=Shell	Esc+0=Exit	Enter=Do	

Figure 86. Non-SP Frame Information

This menu will request frame number, tty ports and switch port numbers. This will establish hardmon communications with the SP-attached server and create the frame object in the SDR.

- Hardware Ethernet address collection:

The MAC address of the SP-Attached server is retrieved by `sphrdwrdd`, in just the same way as a normal SP node, and placed in the SDR.

Now that the SP-attached server is configured as a SP-attached server frame in the SDR, it is ready for standard configuration and installation as a normal node. Full instructions are defined in *PSSP Installation and Migration Guide*, GA22-7347.

- **Boot/Install consideration:**

The default setup for boot/install servers is that the CWS is the boot/install server for a single frame system. In a multiple frame system, the CWS installs the first node in each frame and defines this node as the boot/install server for the remaining nodes in its frame.

If, however, the multiple frame system contains an SP-attached server, the CWS remains as the default boot/install server for the first node in each frame. The first node in each SP frame becomes the boot/install server, with the exception of the SP-attached server, which is treated as a node instead of a frame.

- **Installing the Node:**

The configuration and installation of the SP nodes and SP-attached servers is identical. All of the installation operations will be performed over the Ethernet, with one of the TTY lines providing the *s1term* capabilities, and the other TTY line providing the hardware control and monitoring functions.

- **System Partitioning consideration:**

If the system has multiple partitions defined and you wish to add an SP-attached server, you do not need to bring the system down to one partition, as the SP-attached server appears as a standard SP node to the system partition.

Each SP-attached server has appropriate frame, slot values and switch port numbers. These values are accommodated for existing attributes in the relevant Frame, Node and Syspar\_map SDR classes.

When the SP-attached server frame/node is defined to the system with the `spframe` command, the switch port number to which the node is connected is identified. This number is also necessary in a switchless system, to support system partitioning.

If it is necessary to change the switch port number of the SP-attached server, then the node has to be deleted and redefined with a new switch port number. Deleting this node should be done by deleting the frame to ensure that no inconsistent data is left in the SDR:

- If more than one partition exists, repartition to a single partition.



- Invoke `spdelfram` to delete the SP-Attached server frame and node definitions.
- Recable the server to a new switch port.
- Invoke `spframe` to redefine the SP-attached server frame and node to specify the new switch port number.
- If the system was previously partitioned, repartition back to the system partitioning configuration.

- Considerations when integrating an existing SP-attached server:

Perform the following steps to add an existing SP-attached Server and preserve its current software environment.

1. Physical attachment.

When integrating an existing SP-attached server node to your system, it is recommended (though not mandatory) that the frame be added to the end of your system, to prevent having to reconfiguring the SDR. Different attachment scenarios are described in “Attachment Scenarios” on page 156.

2. Software levels.

If your SP-attached server is not at AIX 4.3.2, upgrade to that level. Ensure that the PSSP `code_version` is set to PSSP-3.1.

3. Customize node.

To perform a preservation install of an SP-attached server with PSSP software, the node must be set to *customize* instead of *install* in the SDR. For example:

```
spbootins -r customize -l 33
```

4. Mirroring.

If the root volume group of the SP-attached server has been mirrored and the mirroring is to be preserved, the information about the existing mirrors must be recorded in the SDR, otherwise the root volume group will be unmirrored during customization.

For example, if the root volume group of the S70 Advanced Server has two copies on two physical disks in locations 30-68-00-0,0 and 30-68-00-2,0 with quorum turned off, enter the following to preserve the mirroring:

```
spchvgobj -r rootvg -c 2 -q false -h 30-68-00-0,0:30-68-00-2,0 -l 33
```

To verify the information, enter:

```
splstdata -b -l 33
```

5. Set up Name Resolution of the SP-attached server.

For PSSP customization, the following must be resolvable on the SP-attached server:

- The control workstation host name.
- The name of the boot/install server's interface that is attached to the SP-attached server's en0 interface.

6. Set up routing to the control workstation host name.

If a default route exists on the SP-attached server, it must be deleted. If it is not removed, customization will fail when it tries to set up the default route defined in the SDR. In order for customization to occur, a static route to the control workstation's hostname must be defined. For example, the control workstation's hostname is its Token Ring address, such as 9.114.73.76 and the gateway is 9.114.73.256:

```
route add -host 9.114.73.76 9.114.73.256
```

7. FTP the SDR\_dest\_info file.

During customization, certain information will be read from the SDR. In order to get to the SDR, the /etc/SDR\_dest\_info file must be FTPed from the control workstation to the /etc/SDR\_dest\_info file of the SP-attached server, ensuring the mode and ownership of the file is correct.

8. Verify perfagent.

Ensure that perfagent.tools 2.2.32.x is installed on the SP-attached server.

9. Mount the pssplpp directory.

Mount the /spdata/sys1/install/pssplpp directory from the boot/install server on the SP-attached server. For example, issue:

```
mount k3n01:/spdata/sys1/install/pssplpp /mnt
```

10. Install ssp.basic.

Install spp.basic and its prerequisites onto the SP-attached server. For example:

```
installp /aXgd/mnt/PSSP-3.1 ssp.basic 2>&1 | tee /tmp/install.log
```

11. Unmount the pssplpp directory.

Unmount the /spdata/sys1/install/pssplpp directory on the boot/install server from the SP-attached server. For example:

```
umount /mnt
```

#### 12. Run pssp\_script.

Run the pssp\_script by issuing

```
/usr/lpp/ssp/install/bin/pssp_script
```

#### 13. Reboot.

Perform a reboot of the SP-attached server.

### 4.2.1 Pre-Installation Checklist

Using the SP configurator, the following hardware and software components for the SP-attached server should be ordered.

#### 1. Feature 9122 Node Attachment.

The feature provides the following:

- 15 meters RS-232 cable between S70 and CWS (S1TERM).
- 15 meters RS-232 cable between S70 and CWS (SAMI).
- This feature includes the frame-to-frame electrical ground cable.

#### 2. Feature 9123 Frame Attachment.

This feature keeps track of how many frames are in your SP system, to avoid exceeding the limit.

#### 3. Feature 5700/1/2 for SP-Attached Server PSSP.

PSSP 3.1 is a separately charged software license for each SP-attached server.

AIX 4.3.2 is included with the SP-attached server and preloaded at the factory, and therefore does not need to be ordered separately.

This feature must be ordered for a non-switched system as well.

#### 4. 9222 Node Attachment Ethernet BNC Boot Feature.

Includes BNC cable for SP Ethernet Communications.

#### 5. 9223 Node Attachment Ethernet Twister pair Boot Feature.

This feature does not provide twisted pair cable.

#### 6. The following features are optional and are only required if the SP-attached server should be attached to the switch. In a switchless system, this feature is not necessary.

- Feature 8396 RS/6000 SP System Attachment Adapter.
- Feature 9310, 10 meter SP switch cable.

---

## 4.3 Software Changes

This section describes the PSSP software enhancements to support the SP-attached server. Of special interest is the fact that the SP-attached server does not use the SP node or frame supervisor cards. Hence the software modifications and interface to the SP-attached server must simulate the architecture of the SP Frame Supervisor Subsystem, such that the boundaries between an SP node and an SP-attached server node are minimal.

### 4.3.1 SDR Changes

The SDR contains system information describing the SP hardware and operating characteristics. Several class definitions have changed to accommodate this new hardware, such as Frame, Node and Syspar\_map classes. A new class definition has been added, the NodeControl class.

The classes that have been modified or new classes created are briefly described:

- Frame Class

Currently, the Frame Class is used to contain information about each SP frame in the system. This information includes physical characteristics (number of slots, whether it contains a switch, and so forth), tty port, hostname and internal attributes used by the switch subsystem.

SP-attached server nodes do not have physical frame hardware and do not contain switch boards. However, they do have hardware control characteristics, such as tty connections and associated Monitor and Control Node (MACN). Therefore, an SDR Frame Object is associated with each SP-attached server node to contain these hardware control characteristics.

Two new attributes have been added to the Frame class:  
*hardware\_protocol* and *s1\_tty*.

The *hardware\_protocol* attribute distinguishes the hardware communication method between the existing SP frames and the new frame objects associated with SP-attached server nodes. For these new nodes, the hardware communication method is SAMI (Service and Manufacturing Interface), which is the protocol used to communicate across the serial connection to the SP-attached server service processor.

The attribute *s1\_tty* is used only for the SP-attached server nodes and contains the tty port for the S1 serial port connection established by the `s1term` command.

A typical example of a frame class with the new attributes and associated values is illustrated in Figure 87.

frame_number	tty	frame_type	MAC	b_MACN	slots	f_in_config	snn_index	switch_config	hardware_protocol	s1_tty
1	/dev/tty0	switch	spcw	**	16	1	0	0	sp	**
2	/dev/tty2	**	spcw	**	1	**	**	**	SAMI	/dev/tty1

Figure 87. Example of a Frame Class with an SP-Attached Server

- **Node Class**

The SDR Node class contains node-specific information used throughout PSSP. Similarly, there will be an SDR Node object associated with the SP-attached server.

SP frame nodes are assigned a node\_number, based on the algorithm described in section 4.1.2, “SP-Attached Server Attachment” on page 127.

Likewise, the same algorithm is used to compute the node number of a SP-attached server frame node, where the SP-attached server occupies the first and only slot of its frame. This means that for every SP-attached server frame node, 16 node numbers will be reserved of which only the first one will ever be used.

The node number is the key value used to access a node object.

Some entries of the Node Class Example are outlined in Figure 88 on page 144.

Node Class	Nodes is an SP	attached S70 Node
Node Number	1-16	17
Slot Number	1-16	1(always)
Switch_node_number	0-15	1
Switch_chip_port	0-15	any port used from 0-15
Switch_chip	4-7	any chip used from 4-7
Switch_number	1	1
Boot_device	en0	en0
Description	112_MHZ_SMP_High 66_MHZ_PWR2_Thin 66_MHZ_PWR2_Wide	7017-S70
Platform	rs6k	chrp
hardware_control_type	161 high, 97 thin, 81 wide, ...,etc.	10 (S70/S7A)

Figure 88. Entries of the Node Class for SP Nodes and SP-Attached Server

The platform attribute has a value of Common Hardware Reference Platform (chrp) for the SP-attached server.

The hardware\_control\_type key value is used to access the NodeControl class. A value of 10 suggests an SP-attached server.

- **Syspar\_map Class**

The Syspar\_map class contains one entry for each switch port in potential switch port, assuming each frame would contain a switch.

As the SP-attached server has node characteristics, it has an entry in the Syspar\_map class for that node, with no new attributes.

The *used* attribute of the Syspar\_map will be set to 1 for the SP-attached server node to indicate that there is a node available to partition. Since this node will be attached to the switch, the *switch\_node\_number* will be set appropriately based on the switch port in an existing SP frame that the SP-attached server node is connected to.

In a switchless system, the switch\_node\_number will be assigned by the administrator using the *spframe* command.

An example of the syspar\_map class is shown in Figure 89 on page 145.

syspar_name	syspar_addr	node_number	switch_node_number	used	node_type
k48s	9.114.11.48	1	0	1	standard
k48s	9.114.11.48	17	1	1	standard
k48s	9.114.11.48	3	2	1	standard
k48s	9.114.11.48	16	15	1	standard

Figure 89. Example of the Syspar\_map Class with SP-Attached Server

The SDR\_config command has been modified to accommodate these new SDR attribute values, and now to handle the assignment of switch\_port\_numbers for SP-attached server nodes.

- NodeControl Class

In order to support different levels of hardware control for different types of nodes, a new SDR class has been defined to store this information.

The NodeControl class is a global SDR class, that is not partition-sensitive. It contains one entry for each type of node that can be supported on an SP system. Each entry contains a list of capabilities that are available for that type of node. This is static information that is loaded during installation and is not be changed by any PSSP code. This static information is required by the SDR\_config script to properly configure the node.

An example of the NodeControl class is illustrated in Figure 90 on page 145.

NodeControl Class

Type	Capabilities	Slots_used	Platform_type	Processor_type
65	Power,reset,ty,KeySwitch,LED,NetworkBoot	1	rs6k	UP
161	Power,reset,ty,KeySwitch,LCD,NetworkBoot	4	rs6k	MP
33	Power,reset,ty,KeySwitch,LED,NetworkBoot	1	rs6k	UP
10	Power,ty,LCD,NetworkBoot	1	chrp	MP
177	Power,reset,ty,LCD,NetworkBoot	1	chrp	MP
115	Power,reset,ty,KeySwitch,LED,NetworkBoot	2	rs6k	UP

Figure 90. Example of the NodeControl Class with the SP-Attached Server

The key link between the Node class and the NodeControl class is the node type, which is a new attribute stored in the SDR Node object. The SP-attached server has a node type value of 10, with hardware

capabilities of power on/off, tty, LCD, and network boot as outlined in Figure 91.

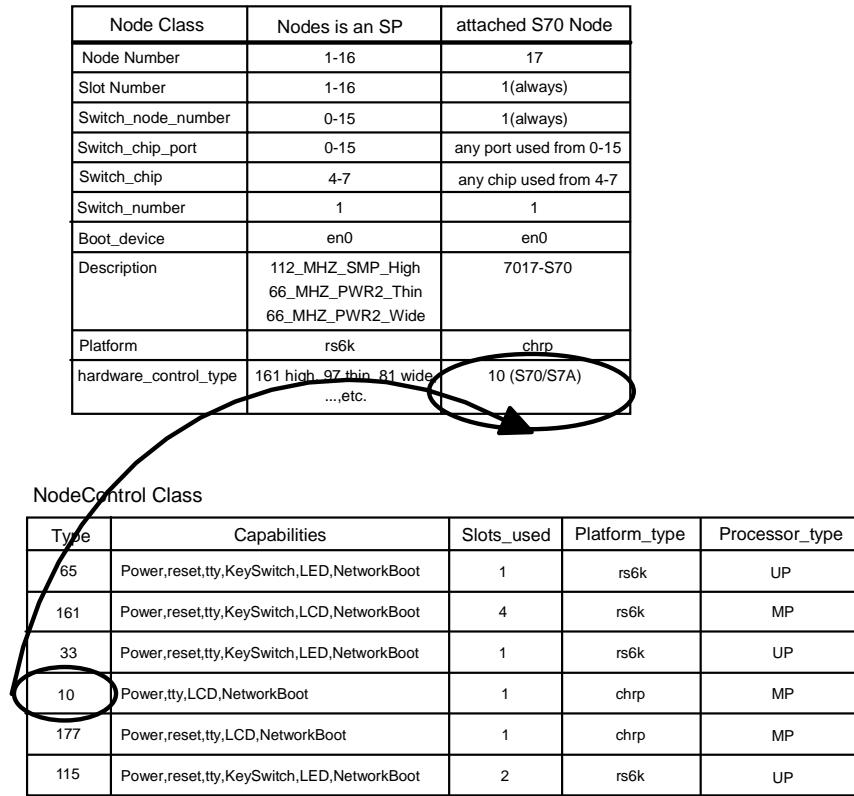


Figure 91. The Relationship Between Node and NodeControl Class

Perspectives routines and hardmon commands access this class to determine the hardware capabilities for a particular node, before attempting to execute a command for a given node.

### 4.3.2 Hardmon

Hardmon is a daemon that is started by the System Resource Controller (SRC) subsystem that runs on the CWS. It is used to control and monitor the SP hardware (Frame, Switch and Nodes) by opening a tty that communicates using an internal protocol to the SP Frame Supervisor card, via a serial RS-232 connection between the CWS and SP Frame.



The new SP-attached server does not have a frame or node supervisor card that can communicate with the hardmon daemon. Therefore a new mechanism to control and monitor SP-attached servers is provided in PSSP3.1.

Hardmon provides support for SP-attached servers in the following way:

- It discovers the existence of SP-attached servers.
- It controls and monitors the state of SP-attached servers, such as power on/off.

#### ***Discover the SP-Attached Server***

For hardmon to discover the hardware, it must first identify the hardware and its capabilities. Today, for each frame configured in the SDR's frame class, hardmon opens a tty defined by the `tty` field. A 2-way communication to the frame supervisor via the RS-232 interface occurs, where hardmon sends hardware control commands and receives state data in the form of packets.

With PSSP 3.1, two new fields have been added to the SDR's frame class: `hardware_protocol` and `s1_tty`. They enable hardmon to determine the new hardware that is externally attached to the SP, and also what software protocol must be used to communicate to this hardware.

Currently, the only two supported values for the `hardware_protocol` field are SP and SAMI. However these values are extensible for new hardware protocol drivers that will emerge as more externally connected hardware is supported.

Upon initialization, hardmon reads its entries in the SDR Frame class, and also examines the value of the `hardware_protocol` field to determine the type of hardware and its capabilities. If the value read is SP, this indicates that SP nodes are connected to hardmon, through SP's Supervisor subsystem. A value of SAMI is specific to the S70/S7A hardware, since it is the SAMI software protocol that allows the communication, both sending messages and receiving packet data, to the S70/S7A's Service Processor.

Once hardmon recognizes the existence of one or more S70/S7As in the configuration, it starts a new process - the S70 daemon. One S70 daemon is started for each frame that has an SDR Frame class `hardware_protocol` value of SAMI. Now hardmon can send commands and process packets or serial data as it would to normal SP frames. This is illustrated in Figure 92 on page 148.

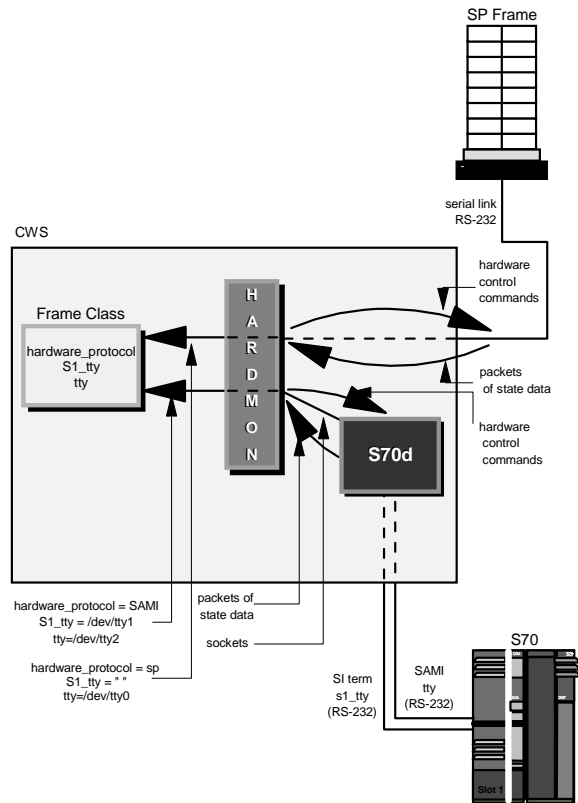


Figure 92. Hardmon Flow of Control

It is important to note that only hardmon starts the S70 daemon, and no other invocation external to hardmon is possible. In addition, the parent hardmon daemon starts a separate S70 daemon for each S70 frame configured in the SDR Frame class.

The S70 daemon starts with the following flags:

```
/usr/lpp/ssp/install/bin/S70d -d 0 2 1 8 /dev/tty2 /dev/tty1
```

where "-d" indicates the debug flag, 0 is the debug option, 2 is the frame number, 1 is the slot number (which is always 1), 8 is the file descriptor of the S70d's side of the socket that is used to communicate with hardmon, /dev/tty2 is the tty that is used to open SAMI/MI operator panel port, and /dev/tty1 serial tty.

## ***S70 Daemon***

The S70 daemon interfaces to the S70 hardware, and emulates the frame and node supervisor by accepting commands from hardmon and responding with hardware state information in the same way as the frame supervisor would. Its basic functions are:

- It polls the S70 for hardware changes in hardware status and returns the status to hardmon in the form of frame packet data.
- It communicates with the S70 hardware through the SAMI/MI interface.  
It accepts hardware control commands from hardmon to change the power state of the S70 and translates them into SAMI protocol, the language that the Manufacturing Interface (MI) understands. It then sends the command to the hardware.
- It opens the tty defined by the tty field in the SDR Frame class, through which the S70 daemon communicates to the S70 serial connection.
- It supports an interface to the S70 S1 serial port to allow console connections via s1term.
- It establishes and maintains data handshaking in accordance with the S70 Manufacturing Interface (MI) requirements.

## ***Dataflow***

Hardmon requests are sent to the S70 daemon, where the command is handled by one of two interface components of the S70 daemon, the Frame Supervisor Interface, or the Node Supervisor Interface.

The frame supervisor interface is responsible for keeping current that state data in the frames' packet and formats the frame packet for return to hardmon. It will accept hardware control commands from hardmon that are intended for itself and "pass-on" to the node supervisor interface commands intended to control the S70/S7A node.

The node supervisor interface polls state data from the S70/S7A hardware, for keeping current the state data in the Nodes' packet. The node supervisor interface will translate the commands received from the frame supervisor interface into S70/S7A software protocol and sends the command through to the S70/S7A service processor.

If the hardmon command is intended for the frame, the frame supervisor entity of the S70d handles it. If intended for the node, the node supervisor entity converts it to SAMI protocol and sends it out the SAMI/MI interface file descriptor, as illustrated by Figure 93 on page 150.

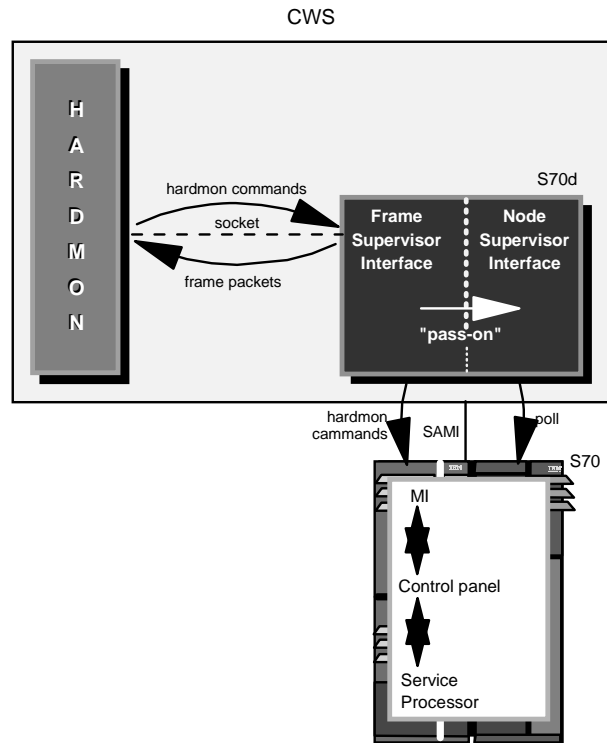


Figure 93. S70 Daemon Internal Flow

The S70 daemon uses SAMI protocol, which takes the form of 4-byte command words, to talk to the S70's Manufacturing Interface. This interface communicates with the S70's operator panel, which in turn communicates with the S70's Service Processor. It is the Service Processor that contains the instruction that acts upon the request. Data returned to the S70 daemon follows the reverse flow.

### **Monitoring of SP-attached Server**

For hardmon to monitor the hardware, it must first identify the hardware and its capabilities.

The hardware control type is determined from the SDR Node class, as a hardware\_control\_type attribute. This attribute is the key into the NodeControl class. The NodeControl class will indicate the hardware capabilities for monitoring. This relationship is illustrated in Figure 91 on page 146.

### ***Hardmon Resource Monitor Daemon***

The Hardmon Resource Monitor Daemon (hmrmd) supports the Event Management resource variables to monitor nodes. With the new SP-attached servers, new resource variables are required to support their unique information.

There are four new hardmon variables that will be integrated into the Hardmon Resource Monitor for the SP-attached servers. They are SRChasMessage, SPCNhasMessage, src, and spcn. Historical states such as nodePower, serialLinkOpen and type are also supported by the SP-attached servers. The mechanics involved with the definition of these variables are no different than with previous variables and can be viewed via Perspectives and in conjunction with Event Manager.

In order to recognize these new resource variables, the Event Manager must be stopped and restarted on the CWS, as are all the nodes in the affected system partition.

---

## **4.4 Changes in the User Interface**

This section highlights the changes in the different user interface panels and commands that have been made to represent the SP-attached server to the user.

### **4.4.1 Perspectives**

As SP must now support nodes with different levels of hardware capabilities, an interface was architected to allow applications such as Perspectives to determine what capabilities exist for any given node and respond accordingly. This interface will be included with a new SDR table, the NodeControl class.

The Perspectives interface needs to reflect the new node definitions, those that are physically not located on an SP frame, and those nodes that do not have full hardware control and monitoring capabilities

There is a typical object representing the SDR Frame object for the SP-attached server node in the Frame/Switch panel. This object has a unique pixmap placement to differentiate it from a high and low frame, and this pixmap is positioned according to its frame number in the Perspectives panel.

An example of the Perspective representation of the SP-attached server is shown in Figure 94 on page 152.

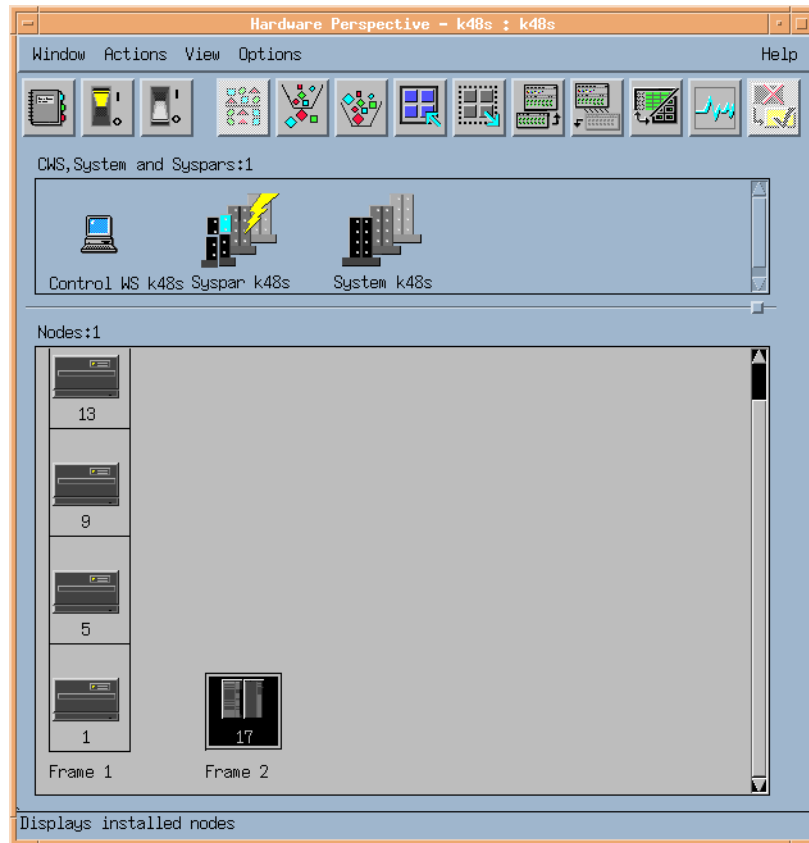


Figure 94. Example of Perspectives with SP-Attached Server

The monitored resource variables are handled the same as for standard SP nodes. Operations, status, frame and node information are handled the same as for standard SP nodes.

Only the Hardware Perspective (sphardware) GUI is affected by the new SP-attached server nodes. The remaining panels, Partitioning Aid Perspective (spsyspar), Performance Monitoring Perspective (spperfmon), Event Perspective (spevent), and VSD Perspective (spvsd) are all similar to the sphardware Perspective node panel since they are based off the same class. Therefore the pixmaps placement will be similar to that of the sphardware Perspective node panel.

### ***Event Manager***

With the new SP-attached server nodes, new resource variables are required to support their unique information.

These new resource variables will be integrated into the Hardmon Resource Monitor for the SP-attached server:

- IBM.PSSP.SP\_HW.Node.SRChasMessage
- IBM.PSSP.SP\_HW.Node.SPCNhasMessage
- IBM.PSSP.SP\_HW.Node.src
- IBM.PSSP.SP\_HW.Node.spcn

In order to recognize these new resource variables the Event Manager must be stopped and restarted on the CWS and all the nodes in the affected system partition.

#### **4.4.1.1 System Management**

Various system management commands display new SDR attributes for SP-attached servers are:

- spmon

Figure 96 on page 155 outlines the `spmon -d -G` output in an SP system that consists of an SP Frame and an SP-attached server.

```

1. Checking server process
   Process 11454 has accumulated 9 minutes and 27 seconds.
   Check ok

2. Opening connection to server
   Connection opened
   Check ok

3. Querying frame(s)
   2 frame(s)
   Check ok

4. Checking frames

      Controller  Slot 17  Switch  Switch  Power supplies
Frame  Responds  Switch  Power  Clocking  A  B  C  D
-----
   1      yes      no      N/A      N/A      on  N/A N/A N/A
   2      yes      no      N/A      N/A      N/A N/A N/A N/A

5. Checking nodes

----- Frame 1 -----
Frame  Node  Node          Host/Switch  Key  Env  Front Panel  LCD/LED is
Slot  Number  Type  Power  Responds  Switch  Fail  LCD/LED  Flashing
-----
   1      1  high    on  yes no    normal  no  LCDs are blank  no
   5      5  high    on  yes no    normal  no  LCDs are blank  no
   9      9  high    on  yes no    normal  no  LCDs are blank  no
  13     13  high    on  yes no    normal  no  LCDs are blank  no

----- Frame 2 -----
Frame  Node  Node          Host/Switch  Key  Env  Front Panel  LCD/LED is
Slot  Number  Type  Power  Responds  Switch  Fail  LCD/LED  Flashing
-----
   1      17  extrn  on   no  no    normal  no  no  no

LCD2 is blank

```

Figure 95. The Output of the `splmon` Command

- `splstdata`

Figure 96 on page 155 is the output of `splstdata -n`. It shows two frames. Figure 97 on page 155 shows the output from `splstdata -f`, where the S70 is shown as a second frame. Figure 97 on page 155 shows the hardware description of each node in the SP system.



- The SP frame has frame number 1 with 4 high nodes of node numbers 1,5,9 and 13, each occupying 4 slots.
- The SP-attached server has frame number 2, with 1 node of node\_number 17 occupying 1 slot

```

List Node Configuration Information

node# frame# slot# slots  initial_hostname  reliable_hostname  dcehostname
      default_route  processor_type  processors_installed  description
-----
  1     1     1     4  c60n01.ppd.pok.i  c60n01.ppd.pok.i  ""
      9.114.88.94      MP                4 112_MHz_SMP_High
  5     1     5     4  c60n05.ppd.pok.i  c60n05.ppd.pok.i  ""
      9.114.88.94      MP                4 75_MHz_SMP_High
  9     1     9     4  c60n09.ppd.pok.i  c60n09.ppd.pok.i  ""
      9.114.88.94      MP                4 75_MHz_SMP_High
 13     1    13     4  c60n13.ppd.pok.i  c60n13.ppd.pok.i  ""
      9.114.88.94      MP                4 112_MHz_SMP_High
 17     2     1     1  c60tpln02.ppd.po  c60tpln02.ppd.po  ""
      9.114.88.1       MP                1 ""

```

Figure 96. `splstdata -n` Output

Figure 97 is the output of `splstdata -f`, which shows 2 frames:

```

List Frame Database Information

frame#          tty          sl_tty          frame_type  hardware_protocol
-----
  1          /dev/tty0          ""          switch          SP
  2          /dev/tty1          /dev/tty2          ""          SAMI

```

Figure 97. `splstdata -f` Output

Figure 98 is the output of `spgetdesc -u -a`, which shows the hardware description obtained from the Node class.

```

spgetdesc: Node 1 (c188n01.ibm.com) is a Power3_SMP_Wide.
spgetdesc: Node 5 (c188n05.ibm.com) is a 332_MHz_SMP_Thin.
spgetdesc: Node 9 (c188n09.ibm.com) is a 332_MHz_SMP_Thin.
spgetdesc: Node 13 (c188n13.ibm.com) is a Power3_SMP_Wide.
spgetdesc: Node 17 (c187-S70.ibm.com) is a 7017-S70.

```

Figure 98. `spgetdesc -u -a` Output

## 4.5 Attachment Scenarios

The following sections describe the different attachment scenarios of the SP-attached server to the SP system, but they do not show all the cable attachments between the SP frame and the SP-attach server.

### **Scenario 1: SP-attached server to a one-frame SP system**

This scenario shows a single frame system, with 14 thin nodes located in slots 1 through 14. The system has two unused node slots in position 15 and 16. These two empty node slots have corresponding switch ports which provide valid connections for the RS/6000 SP Attachment adapter.

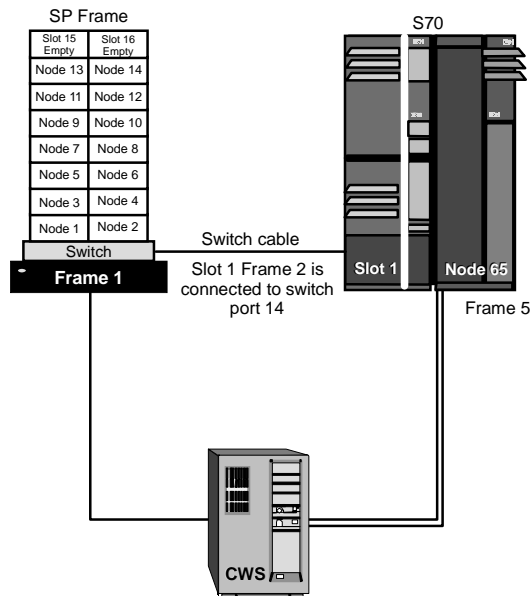


Figure 99. Scenario 1: SP-Attached Server and One SP Frame

### **Scenario 2: SP-attached server to a two-frame SP system**

This scenario shows a two-frame system, with 4 high nodes in each frame. This configuration will use 8 switch ports and leave 8 valid switch ports available for future scalability. Therefore, it is important that the frame number assigned to the S70 must allow for extra non-switched frames (in this example, frames 3 and 4), as the S70 frame must be attached to the end of the configuration. On this basis, the S70 frame number must be at the very least 5, to allow for the 2 possible non-switch frames.

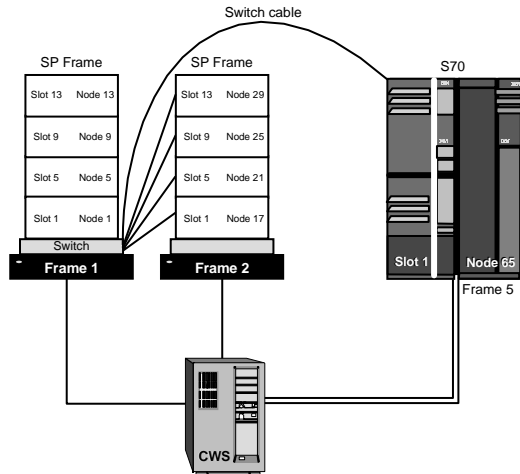


Figure 100. Scenario 2: SP-Attached Server to Two SP Frames

Note that the switch cable from frame 1 connects to the S70; for example, in this case, slot 1 frame 5 connects to switch port 3 of switch chip 5.

### **Scenario 3: One SP frame and multiple SP-attached servers**

This scenario illustrates three important considerations:

1. The minimum requirement of one node in a frame, to be able to attach one or more SP-attached servers to an SP system, as the SP-attached server cannot be the first frame in an SP environment.
2. It cannot interfere with the frame numbering of the expansion frames, and therefore the SP-attached server is always at the end of the chain.
3. A switch port number must be allocated to each SP-attached server, even though the SP system is switchless.

In this example, the first frame has a single thin node only, which is mandatory for any number of SP-attached servers.

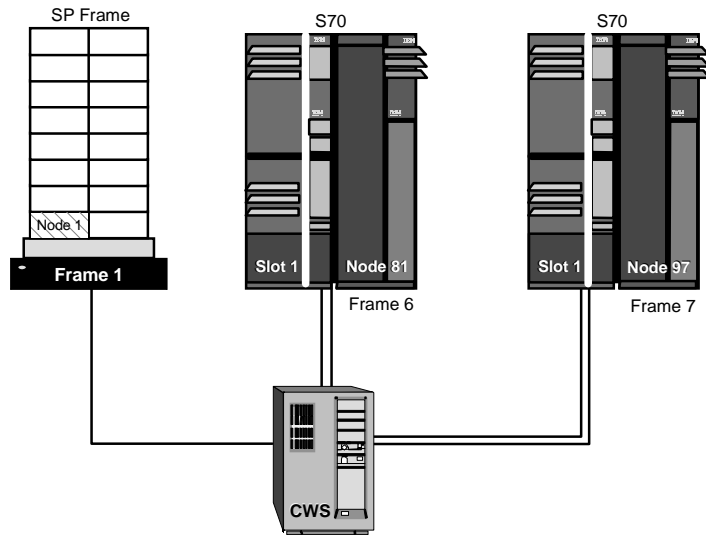


Figure 101. Scenario 3: SP Frame and Multiple SP-Attached Servers

**Scenario 4: Non-contiguous SP-attached server configuration**

Frame 1 and 3 of the SP system are switch-configured. Frame 2 is a non-switched expansion frame attached to frame 1. In this configuration, the SP-attached server could be given frame number 4, but that would forbid any future attachment of nonswitched expansion frames to frame 1's switch. If, however, you assigned the SP-attached server frame number 15, your system could still be scaled using other switch-configured frames and nonswitched expansion frames.

Frame 3 is another switch-configured frame, and the SP-attached server has previously been assigned frame number 10, for future scalability purposes.

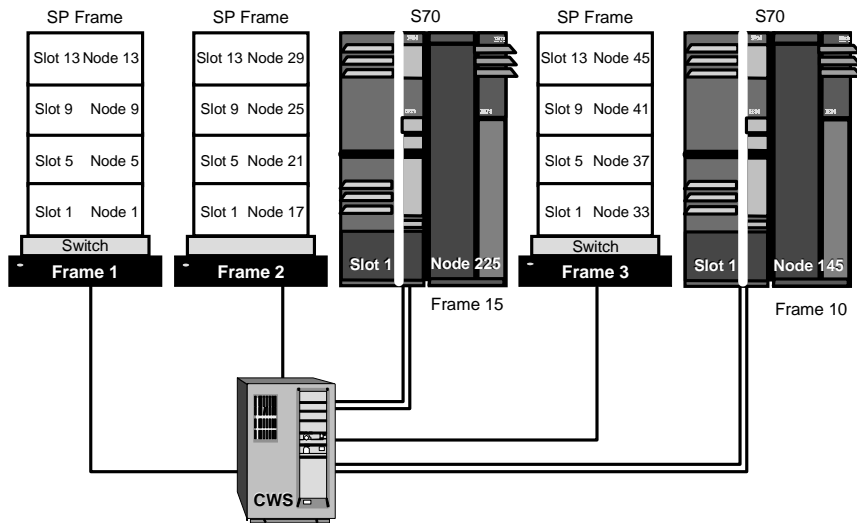


Figure 102. Scenario 4: Non-Contiguous SP-Attached Server

For more information see: *IBM RS/6000 SP: Planning Volume 2, Control Workstation and Software Environment*, GA22-7281.



---

## Chapter 5. Switch Support Enhancements

Various improvements have been made to the switch support in PSSP 3.1. These enhancements were inspired by customers' requirements gathered via surveys and interviews by the PSSP development laboratory.

The objective of these changes is to help automate switch management and improve its serviceability.

---

### 5.1 Automatic Node Unfence

Prior to PSSP 3.1, when a node was shut down or rebooted, it was fenced off the switch automatically by the primary node.

However, this caused an extra workload since the system administrator was required to manually unfence the node after rebooting.

On the other hand, other networks (such as Ethernet or token ring) can be used immediately after reboot. Thus, it seemed that the manual intervention step needed here for the switch network should be eliminated to make the switch network behave more like other LANs.

With PSSP 3.1, nodes that are automatically fenced by the primary node will be automatically unfenced by the primary node itself.

This makes the switch network behave more like other LANs, that is, it can be used after rebooting is completed. There is no need for manual intervention.

This also helps automate the switch management task. Now the system administrator needs to be concerned only with the unfence of the nodes that were explicitly fenced.

#### 5.1.1 Implementation Overview

Several changes were made to implement this function:

##### 5.1.1.1 Autojoin is Now Default

Prior to PSSP 3.1, the autojoin attribute in the switch\_respond class was normally turned off for all nodes. The only exception was that a node that was explicitly fenced with the `Efence -autojoin` command had this attribute turned on.

With PSSP 3.1, the autojoin attribute is now turned on for all nodes. Exceptions are the node fenced with the `Efence` command and the node that the fault service daemon considers to have a persistent problem.

#### 5.1.1.2 Changes to the `switch_scan` Function

The switch scan, conducted every two minutes by the primary node, has been modified to unfence any nodes that have the autojoin attribute set when they are "ready" to come up on the switch.

The meaning of "ready" in this context is that the switch adapter microcode and the fault service daemon (or its dependent node equivalent) are running.

Thus, all nodes that are rebooted or powered off and then on will automatically come up on the switch after the primary node found that they are ready.

However, if a node has an intermittent problem with the switch adapter, it may continually be unfenced and refenced. This causes unnecessary work to be done by the primary node.

To avoid this problem, two mechanisms are put into place:

- If the fault service daemon on the failing node reaches an error threshold or detects an unrecoverable error, it puts its TBIC<sup>1</sup> into reset and sets the autojoin attribute to off. Once this occurs, the node will not unfence until `rc.switch` is run to recover the node, and `Eunfence` is run.

The node failing to `Eunfence` signals the administrator that the node has a problem and needs to be diagnosed and recovered.

- If the primary fault service daemon fails to unfence a particular node three consecutive times after its link initializes, its autojoin attribute is set off.

No further attempt to auto-unfence will be made until the administrator runs `Eunfence`.

### 5.1.2 Example Scenarios

Figure 103 on page 163 shows that when a node is fenced with autojoin option, it will be automatically unfenced within two minutes.

<sup>1</sup> *Trail Blazer Interface Chip*, a chip in the switch adapter card that manages the link from the switch adapter to the switch board.



```

sp3en0{ / } SDRGetObjects switch_responds node_number==12
node_number switch_responds autojoin isolated adapter_config_status
12 1 1 0 css_ready
sp3en0{ / } E fence -autojoin 12
All nodes successfully fenced.
sp3en0{ / } date
Tue Sep 15 19:37:12 EDT 1998
sp3en0{ / } SDRGetObjects switch_responds node_number==12
node_number switch_responds autojoin isolated adapter_config_status
12 0 1 1 css_ready
sp3en0{ / } SDRGetObjects switch_responds node_number==12
node_number switch_responds autojoin isolated adapter_config_status
12 1 1 0 css_ready
sp3en0{ / } date
Tue Sep 15 19:38:37 EDT 1998

```

Figure 103. A Node with the Autojoin Bit On Automatically Joins the Switch Network

Figure 104 on page 163 shows that once a node is rebooted, it automatically joins the switch network.

```

sp3en0{ / } SDRGetObjects switch_responds node_number==15
node_number switch_responds autojoin isolated adapter_config_status
15 1 1 0 css_ready
sp3en0{ / } date
Tue Sep 15 19:50:04 EDT 1998
sp3en0{ / } cshutdow -Fr -N 15
Progress recorded in /var/adm/SPlogs/cs/cshut.0915195011.35264.
Progress recorded in /var/adm/SPlogs/cs/cstart.0915195234.35378.
sp3en0{ / } date
Tue Sep 15 19:59:04 EDT 1998
sp3en0{ / } SDRGetObjects switch_responds node_number==15
node_number switch_responds autojoin isolated adapter_config_status
15 1 1 0 css_ready
sp3en0{ / }

```

Figure 104. A Rebooted Node Automatically Joins the Switch Network

### 5.1.3 Coexistence Consideration

In order to enable the automatic node unfence function for a system partition, the primary node for that system partition must be at the PSSP 3.1 level.

For all nodes, it is required that whenever a node does not want an unfence attempted by the primary node, it must put its TBIC in the reset state.

When the TBIC is in a reset state, the primary fault service daemon knows that the node is not ready to be sent any packet.

The TBIC is taken out of the reset state after the fault service daemon has started, loaded and started the switch adapter microcode.

The TBIC is set to reset in the following cases:

- Whenever the fault service daemon exit.
- After it has been determined that the TBIC has a permanent error condition such as a hot interrupt, or exceeds an error threshold
- When the switch adapter diagnostic code finishes
- When the switch adapter microcode is not running

Suppose that the switch adapter microcode is not running and the node's autojoin bit is on but the TBIC was not set to reset. The primary node will try to unfence it by sending the service packet. Since the node is not "ready", the packets are not received. The effect of this is that those packets start to accumulate in the switch chip and block the switch network. This can cause severe performance degradation of the switch network.

Prior to PSSP 3.1, TBIC was not always left in the reset state when it was supposed to be. Thus, in order to allow earlier PSSP levels to coexist with PSSP 3.1 primary node, a certain level of ssp.css fileset is required for earlier PSSP releases.

*Table 14. Required ssp.css Level for Coexistence*

PSSP level	ssp.css level
PSSP 2.2	2.2.0.13
PSSP 2.3	2.3.0.8

PSSP 2.4 and PSSP 3.1 always put TBIC into the reset state and thus need no PTFs.

---

## 5.2 Startup of Switch-Dependent Applications at Boot Time

From RAS and usability studies conducted with customers, the ability of a node to automatically join the switch without administrator intervention is only part of the solution of automating switch management. The other areas required are to automate the startup of the switch network (see "Switch Admin Daemon" on page 167), and to synchronize the starting of switch-dependent applications and subsystems automatically when the switch interface comes up.

Prior to PSSP 3.1, configuring applications to use the switch network had to be done by writing a separate script to start the switch and checking that it was available before starting the applications.

In PSSP 3.1, with the help of this function and the switch admin daemon (which is described in the next section), setting up the application to use the switch network is just two simple steps:

- Change the rc.switch entry in /etc/inittab from *once* to *wait*.
- Put the application startup script after the rc.switch entry.

### 5.2.1 Implementation Overview

The following figure shows the additional code that was added to the rc.switch script in PSSP 3.1.

```
# declare integers
# 72*(5 seconds) = 6 minutes
integer waitlimit=72
integer n=0
.....
# if the adapter is TB3 we will wait 5 minutes for the switch interface
# to come up.
grep fsd /etc/inittab | cut -d: -f 3 | read CSS0_WAIT
.....
if [[ "$ADAPTER" = TB3 ]] && [[ "$CSS0_WAIT" = wait ]]
then
  echo "Waiting for switch interface to come up."
  echo "Waiting for switch interface to come up." >> $LOG/rc.switch.log
  SP_READY=0
  while (( $n < $waitlimit ))
  do
    $CSS/estat -n css0 >> $LOG/rc.switch.log
    if (( $? == 1 ))
    then
      SP_READY=1
      break
    fi
    sleep 5
    (( n = $n + 1 ))
  done

  if (( $SP_READY != 1 ))
  then
    echo "ERROR: Timed-out waiting for switch interface to come up." >> $LOG/rc
    .switch.log
  else
    echo "Switch interface is up." >> $LOG/rc.switch.log
  fi
fi
```

Figure 105. Changes in the /usr/lpp/ssp/css/rc.switch Script in PSSP 3.1

Prior to PSSP 3.1, the rc.switch script exited after starting the Worm daemon.

In PSSP 3.1, the script was changed so that after starting the Worm daemon, if "wait" was specified in the inittab file for the rc.switch entry, it waits (up to a maximum of six minutes) for the switch interface to come up before exiting.

Thus, if we change the rc.switch entry in /etc/inittab to "wait" and place any startup scripts for switch-dependent applications after it, we can be sure that the switch interface has already come up before the applications are started.

For example, suppose that the primary node is available when a node comes up. After starting Worm, the node has both switch adapter microcode and the fault service daemon running and thus is "ready" to join the switch. The switch scan that is conducted by the primary node every two minutes will unfence the node to let it join the switch network.

The rc.switch script will then exit and inittab processing continues. The applications that start after this can make use of the switch interface.

#### Important Note

It should be noted that there is a potential risk, albeit low, that the switch interface has not been made available before your applications are started. For example, the switch is powered off or it has not been initialized by an Estart.

In addition, in PSSP 3.1, the rc.switch script is now inserted immediately following the rc.sp entry in the /etc/inittab file. This was done to try to have the switch interface up as early as possible during the system startup cycle so that other applications that depend on the switch network can start sooner.

```
rtcpip:2:wait:/etc/rc.tcpip > /dev/console 2>&1 # Start TCP/IP daemons
rcnfs:2:wait:/etc/rc.nfs > /dev/console 2>&1 # Start NFS Daemons
sp:2:wait:/etc/rc.sp > /dev/console 2>&1
fsd:2:once:/usr/lpp/ssp/css/rc.switch
cron:2:respawn:/usr/sbin/cron
nimclient:2:once:/usr/sbin/nimclient -S running
piobe:2:wait:/usr/lib/lpd/pio/etc/pioint >/dev/null 2>&1 # pb cleanup
qdaemon:2:wait:/usr/bin/startsrc -sqdaemon
```

Figure 106. A Portion of /etc/inittab in PSSP 3.1

## 5.2.2 Coexistence Consideration

For those who wrote their own scripts to provide the startup of the switch-dependent applications, it is recommended that they review the scripts and use this new function instead, wherever possible.

---

## 5.3 Switch Admin Daemon

The objective here is to automate the switch startup process by having a daemon monitor certain node and switch adapter events in all partitions and respond with an automatic Estart whenever appropriate.

Prior to PSSP 3.1, some setup, either in the form of writing a script to automate Estart or having a person manually do the Estart was required before the switch network could be used after the power-on of the system.

In PSSP 3.1, there is no need to do this since now the switch network is normally automatically started by the switch admin daemon!

### 5.3.1 Implementation Overview

The new switch admin daemon, *cssadm*, which runs on the CWS makes use of Event Management to receive notification of events on nodes by registering for:

- Node-down-on-switch-adapter event<sup>2</sup> on all nodes.
- Node-up-on-host\_respond event<sup>3</sup> on all nodes.

Figure 107 on page 168 shows the action the *cssdam* daemon takes in response to a node-down-on-switch-adapter event.

<sup>2</sup> X==0 && X@P==1 on IBM.PSSP.Membership.LANAdapter.state

<sup>3</sup> X==1 && X@P==0 on IBM.PSSP.Response.Host.state

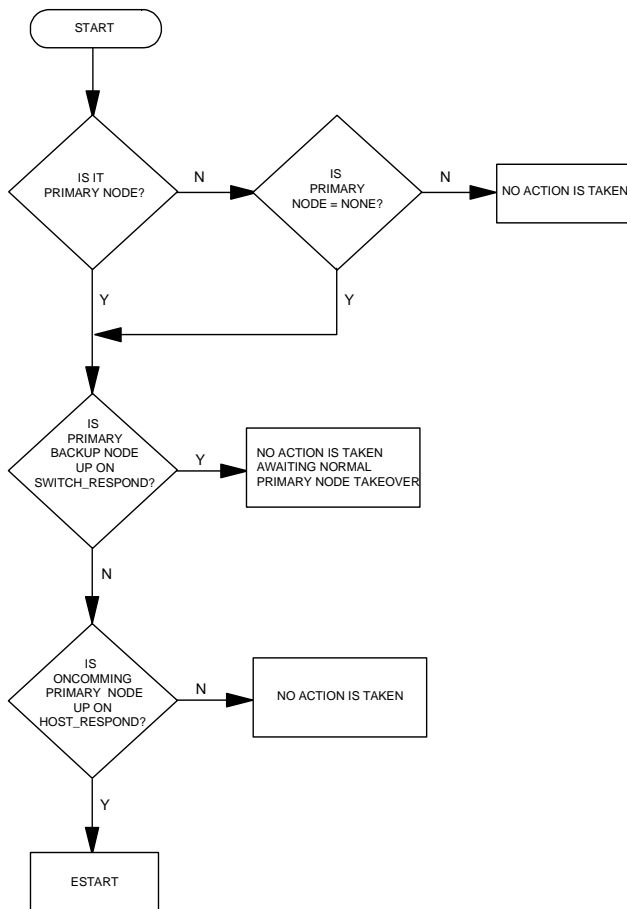


Figure 107. Cases When a Node Went Down on the css0 Interface

Notice that when the primary node is down on the switch but the primary backup node is active, the cssadm daemon takes no action, because the primary backup node will take over the primary node responsibility.

In case both the primary and the primary backup node are down on the switch but the oncoming primary node is not yet up on host\_respond, it takes no action but waits until another significant event occurs.

Figure 108 on page 169 shows the action the cssdam daemon takes in response to a node-up-on-host\_respond event.

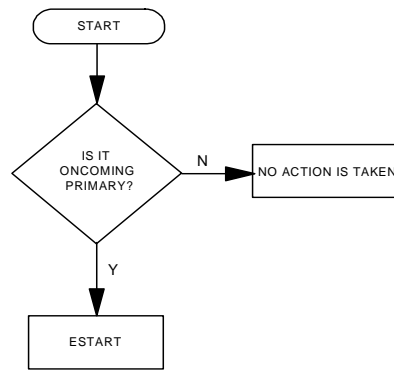


Figure 108. Cases When a Node Came Up on *host\_respond*

Notice that *cssadm* will issue an *Estart* only when it finds that the oncoming primary node came up on *host\_respond*.

In all cases, if *Estart* fails, it logs the errors in the *cssadm.stderr* log file located in the */var/adm/SPlogs/css* directory, but takes no additional recovery actions.

The *cssadm* daemon runs under SRC control with the subsystem name of *swtadmd*. It is added to */etc/inittab* by the *install\_cw* script during the installation, so that it automatically starts up when the CWS boots.

The configuration file for the *cssadm* daemon is *cssadm.cfg* in the */spdata/sys1/ha/css* directory. By default, this file contains only one non-comment line, Node 1.

"One" here means that the *cssadm* daemon will perform the node recovery, that is, it tries to *Estart* whenever it detects significant node events in the system (for example, the primary node goes down, the oncoming primary backup comes up).

To disable node recovery, change this line from "Node 1" to "Node 0", stop and restart the cssadm daemon by using the `stopsrc -s swtadmd` and `startsrc -s swtadmd` command.

The cssadm daemon generated three output files in the `/var/adm/SPlogs/css` directory:

- `cssadm.stdout` contains the stdout of the commands executed by the daemon.
- `cssadm.stderr` contains the stderr of the commands executed by the daemon.
- `cssadm.debug` contains entries for each event received, how it was handled, and what the results were.

Figure 109 on page 170 shows an example of the `cssadm.debug` file on the CWS.

```
(i) open_EM_sessions: ha_em_start_session for sp3en0 succeeded - continuing.
-----
Received Event:
-----
event_type      = node down
node number = 13
time = Mon Sep 14 18:38:48 1998

complete = 2

(i) cssadm: Processing event:
-----
Processing Event:
-----
event_type      = node down
node number = 13
time = Mon Sep 14 18:38:48 1998

complete = 2
(i) cssadm: Primary node is down on switch responds in partition sp3en0.
Checking primary backup.
(i) cssadm: Primary backup node sp3n14.msc.itso.ibm.com is up. No action taken, awaiting
normal primary node takeover.
```

Figure 109. Example of the `cssadm.debug` File on the CWS

### 5.3.2 Example Scenarios

In Figure 110 on page 171, we simulate the event that the oncoming primary backup node (node 13 here) is up on `host_respond` by stopping and restarting the hats daemon on node 13.



```

sp3en0{ / } Eprimary
14 - primary
13 - oncoming primary
5 - primary backup
14 - oncoming primary backup
sp3en0{ / }
sp3en0{ / } dsh -w n13 stopsrc -s hats
n13: 0513-044 The stop of the hats Subsystem was completed successfully.
sp3en0{ / }
sp3en0{ / } dsh -w n13 startsrc -s hats
n13: 0513-059 The hats Subsystem has been started. Subsystem PID is 19394.
sp3en0{ / } tail -21 /var/adm/SPlogs/css/cssadm.debug
-----
Received Event:
-----
event_type = node up
node number = 13
time = Tue Sep 15 13:38:17 1998

complete = 2

(i) cssadm: Processing event:
-----
Processing Event:
-----
event_type = node up
node number = 13
time = Tue Sep 15 13:38:17 1998

complete = 2
(i) cssadm: Oncoming primary node has come up on Host Responds in partition sp3
en0.
Estart will be run
(i) cssadm: Estart successful in partition sp3en0.
sp3en0{ / }
sp3en0{ / } Eprimary
13 - primary
13 - oncoming primary
14 - primary backup
14 - oncoming primary backup

```

Figure 110. Example of the cssadm Daemon Response

We can see from the cssadm.debug log file that Estart was issued in response to the oncoming-primary-up-on-host\_respond and it completed successfully. Eprimary also shows that now node 13 is the primary node.

In Figure 111 on page 172, we simulate the event that the primary node (node 5) and the primary backup node (node 8) are both down on switch\_respond by killing the Worm daemon on both nodes at the same time.

```

sp3en0{ / } Eprimary
5      - primary
13     - oncoming primary
8      - primary backup
14     - oncoming primary backup
sp3en0{ / } dsh -w n05,n08 'ps -ef|grep orm|grep -v grep'
n05:   root 12930  1  0  Sep 13   -  0:14 /usr/lpp/ssp/css/fault_serv
ice_Worm_RTG_SP -r 4 -b 1 -s 5 -p 1 -a TB3 -t 22
n08:   root 3884  1  0  Sep 12   -  0:01 /usr/lpp/ssp/css/fault_serv
ice_Worm_RTG_SP -r 7 -b 1 -s 6 -p 3 -a TB3 -t 22
sp3en0{ / } dsh -w n05 kill 12930; dsh -w n08 kill 3884
sp3en0{ / } tail -23 /var/adm/SPlogs/css/cssadm.debug
-----
Received Event:
-----
event_type      = node down
node number     = 5
time = Tue Sep 15 14:21:03 1998

complete       = 2

(i) cssadm: Processing event:
-----
Processing Event:
-----
event_type      = node down
node number     = 5
time = Tue Sep 15 14:21:03 1998

complete       = 2
(i) cssadm: Primary node is down on switch responds in partition sp3en0.
Checking primary backup.
(i) cssadm: Primary backup is not up on switch responds. Checking
if oncoming primary is up on host responds.
(i) cssadm: Oncoming primary up on host responds. Going to Estart.

sp3en0{ / } Eprimary
13     - primary
13     - oncoming primary
14     - primary backup
14     - oncoming primary backup

```

Figure 111. Another Example of the cssadm Daemon Response

Though the cssadm daemon detected that both the primary node and the primary backup node are down, it found that the oncoming primary node is up, so it issued an Estart to get the switch network back to normal.

Prior to PSSP 3-1, the administrator had to realize that the primary and primary backup node were both down and issue Estart manually.

**5.3.3 Coexistence Consideration**

The switch admin daemon works with any combination of nodes, since it does not require any modifications to the code in a node.

It is required that the event manager daemon on pre-PSSP 3.1 nodes be recycled in order to pick up the new resource variables being monitored.

---

## 5.4 Centralized Error Logging

PSSP 3.1 introduces a centralized switch error logging as a step to improve the serviceability of the switch by making the problem determination and resolution process easier and faster.

Prior to PSSP 3.1, when there was a problem with the switch, we had to log on to the switch primary node and other possible problem nodes to see what had happened and try to figure out how they happened.

We had to coordinate all events that had happened on the nodes because we had to know exactly how things had happened so that we could see the relationships and possibly the effect they had on one another. This enabled us to find out the probable "culprit" that caused the problems.

However, this process is quite time-consuming and thus hinders fast problem determination and resolution.

In PSSP 3.1, there is a new log file, *summlog*, in the */spdata/sys1/ha/css* directory on the CWS that provides a summary of all significant switch-related events that happen in the entire system.

Since this log file consolidates all switch-related entries in the AIX error log for the entire system, it provides a single point from which to monitor system-wide switch activity.

Since it is ordered by timestamp, it shows the sequence of events that happened and enables us to figure out the relationship between them. This helps decrease the switch problem determination and resolution time.

Additionally, some switch events now do not only create a summary record on the CWS, but also trigger a snapshot, *css.snap*, on the node logging the error. This helps capture the data necessary for further problem determination at the most appropriate point in time, the time when the error happened, and thus helps shorten problem resolution time.

### 5.4.1 Implementation Overview

A new daemon, *css.summlog*, was implemented on the CWS to watch for a switch-related entry added to the AIX error log on any nodes. Once such an

event occurs, a summary log record is generated in a new log file, `summlog`, in the `/spdata/sys1/ha/css` directory, on the CWS.

#### 5.4.1.1 New Resource Variable

A new resource variable, `IBM.PSSP.CSSlog.errlog`, which is specific to CSS log consolidation functionality is defined.

The intent is to support the generation of events which reflects that a new switch-related entry was made to the local AIX error log. It provides the index of the entry within the error log, the label and the time of the entry.

This resource variable is made part of the new `IBM.PSSP.CSSlog` resource class since its resource monitor is `IBM.PSSP.CSSLogMon`. (It would have been part of `IBM.PSSP.CSS` if it had used `harmlid` as the resource monitor.)

Table 15 on page 174 shows the detail of each field in this resource variable.

Table 15. `IBM.PSSP.CSSlog.errlog` is a Structured Byte String with These Fields

Name	Type	Description
<code>lastupdt</code>	long	time of last update in seconds
<code>entrynum</code>	long	entry number in log
<code>version</code>	cstring	version number (for future use)
<code>symptom</code>	cstring	symptom string

#### 5.4.1.2 Changed in Error Notification Objects

Sixty-eight ODM error notification objects for switch error log entry are modified to use a new command, `css.logevent`, as the error notification method (instead of `errlog_rm` as in the previous releases).

Table 16 shows that all AIX error log entries that occur on a node will generate a summary log record on the CWS.

Table 16. AIX Error Log Entries

Error Label	Description
<code>HPS_FAULT6_ER</code>	Switch Fault Service Daemon Terminated
<code>HPS_FAULT9_ER</code>	Switch Adapter - Bus error
<code>SP_CLCK_MISS_RE</code>	Switch (non-master) lost clock
<code>SP_CSS_IF_FAIL_ER</code>	Switch adapter i/f system call failed
<code>SP_MCLCK_MISS_RE</code>	Switch (master oscillator) lost clock

Error Label	Description
SP_PROCESS_KILLD_RE	Process killed due to link outage
SP_SW_ACK_FAILED_RE	Switch daemon ACK of svc command failed
SP_SW_BCKUP_TOVR_RE	Switch primary backup node takeover
SP_SW_CBCST_FAIL_RE	Switch daemon command broadcast failed
SP_SW_CRC_SVCPKT_RE	Switch service logic incorrect CRC
SP_SW_DNODE_FAIL_RE	Switch daemon dependent node svc failure
SP_SW_ECLOCK_RE	Eclock command issued by user
SP_SW_EDCTHRSHLD_RE	Switch rcvr EDC errors exceed threshold
SP_SW_EDC_ERROR_RE	Receiver EDC-class error
SP_SW_ESTRT_FAIL_RE	Estart failed
SP_SW_FENCE_FAIL_RE	Fence of node failed
SP_SW_FIFOVRFLW_RE	Switch receiver FIFO overflow error
SP_SW_GET_SVCREQ_ER	Switch daemon could not get svc request
SP_SW_INIT_FAIL_ER	Switch daemon initialization failed
SP_SW_INVALID_RTE_RE	Switch sender invalid route error
SP_SW_IP_RESET_ER	Switch daemon could not reset IP
SP_SW_LNK_ENABLE_RE	Switch svc logic invalid link enable
SP_SW_LOGFAILURE_RE	Error writing switch log files
SP_SW_LST_BUP_CT_RE	Primary backup node not responding
SP_SW_MISWIRE_ER	Switch - cable mis-wired
SP_SW_NCLL_UNINT_RE	Switch central queue NCLL uninitialized
SP_SW_NODEMISW_RE	Switch node miswired
SP_SW_OFFLINE_RE	Node fence request received
SP_SW_PE_INBFIFO_RE	Switch svc logic bad parity - inFIFO
SP_SW_PE_ON_DATA_RE	Switch sender parity error on data
SP_SW_PE_ON_NCLL_RE	Switch central queue parity error - NCLL
SP_SW_PE_ON_NMLL_RE	Switch central queue parity error - NMLL

Error Label	Description
SP_SW_PE_RTE_TBL_RE	Switch svc logic bad parity - route tbl
SP_SW_PRI_TAKOVR_RE	Switch primary node takeover
SP_SW_RCVLNKSYNC_RE	Switch receiver link sync error
SP_SW_RECV_STATE_RE	Switch receiver state machine error
SP_SW_REOP_WIN_ER	Switch daemon reopen windows failed
SP_SW_ROUTE_VIOL_RE	Recvr Route Violation Error
SP_SW_RSGN_BKUP_RE	Resigning as switch primary backup
SP_SW_RSGN_PRIM_RE	Resigning switch primaryship
SP_SW_RTE_GEN_RE	Switch daemon failed to generate routes
SP_SW_SCAN_FAIL_ER	Switch scan failed
SP_SW_SDR_FAIL_RE	Switch daemon SDR communications failed
SP_SW_SEND_TOD_RE	Switch svc logic send TOD error
SP_SW_SIGTERM_ER	Switch daemon received SIGTERM
SP_SW_SNDLNKSYNC_RE	Switch sender link sync error
SP_SW_SNDLOSTEOP_RE	Sender Lost EOP Error
SP_SW_SNDTKNTHRS_RE	Switch snd token errors exceed threshold
SP_SW_SND_STATE_RE	Switch sender state machine error
SP_SW_STIDATARET_RE	Recvr STI Data Re-Time Request
SP_SW_STITOKN_RT_RE	Sender STI Token Re-Time Request
SP_SW_SVC_PKTLEN_RE	Switch svc logic saw bad packet length
SP_SW_SVC_Q_FULL_RE	Switch service send queue full
SP_SW_SVC_STATE_RE	Switch svc logic state machine error
SP_SW_UBCST_FAIL_RE	Switch daemon DBupdate broadcast failed
SP_SW_UNINI_LINK_RE	Links not initialized during Estart
SP_SW_UNINI_NODE_RE	Nodes not initialized during Estart
TB3_BAD_PACKET_RE	Bad packet received
TB3_CONFIG1_ER	Failed to update ODM during CSS config

Error Label	Description
TB3_HARDWARE_ER	Switch adapter hardware/microcode error
TB3_LINK_RE	Switch adapter link outage
TB3_MICROCODE_ER	Switch adapter microcode error
TB3_PIO_ER	I/O error
TB3_SLIH_ER	Switch adapter interrupt handler error
TB3_SVC_QUE_FULL_ER	Switch adapter svc interface overrun
TB3_THRESHOLD_ER	Switch adapter error threshold exceeded
TB3_TRANSIENT_RE	Switch adapter transient error
TBS_HARDWARE_ER	Switch board hardware error

#### 5.4.1.3 New Error Notification Method

The new error notification method, `css.logevnt`, does the following:

1. Invokes `errlog_rm` if `ssp.pman` is installed.
 

`errlog_rm` is a resource monitor that generates an event indicating that the AIX error log has been updated.
2. Optionally takes a full or soft `css.snap`.
 

Example of errors that cause a full `css.snap` to be taken:

  - Switch adapter link outage
  - Switch adapter error threshold exceeded
  - Switch adapter service interface overrun

Example of errors that cause a soft `css.snap` to be taken:

  - Switch sender parity error on data
  - Switch central queue parity error
  - Switch adapter device driver I/O error
3. Generates log change event to trigger summary record creation on the CWS.
4. Makes sure that its stdout/stderr file, `logevnt.out` in the directory `/var/adm/SPlogs/css`, is not larger than 1MB. When it is more than 1MB, it is moved to `logevnt.out.old` and a new `logevnt.out` is created.

#### 5.4.1.4 New Log File on CWS

A new consolidated summary log file `summlog`, in the directory `/spdata/sys1/ha/css`, is created on the CWS containing summary records for each switch-related event occurring in the AIX error log on all nodes.

The summary record format contains:

- timestamp - in the form of MMDDhhmmYYYY
- nodename - short reliable hostname
- snap? - Yes or No, indicates whether a snap was taken
- partition - system partition name or global
- index - the sequence number field in the AIX error log
- label - the label field in the AIX error log

When the `summlog` file size is greater than 3MB, it is renamed to `summlog.old` and a new `summlog` file is created.

Figure 112 on page 178 shows an example of `/var/adm/SPlogs/css/summlog` file, which is a symbolic link to `/spdata/sys1/ha/css/summlog`, on the CWS.

```
082611441998 sp3en0 N global 101 SP_SW_ECLOCK_RE
082715261998 sp3n05 N sp3en0 27 SP_SW_SDR_FAIL_RE
082715271998 sp3n05 N sp3en0 28 SP_SW_UNINI_NODE_RE
082715271998 sp3n05 N sp3en0 29 SP_SW_UNINI_NODE_RE
082715271998 sp3n05 N sp3en0 30 SP_SW_UNINI_LINK_RE
082715271998 sp3n05 N sp3en0 32 SP_SW_UNINI_LINK_RE
082715271998 sp3n05 N sp3en0 31 SP_SW_UNINI_LINK_RE
082715271998 sp3n05 N sp3en0 33 SP_SW_SDR_FAIL_RE
082716301998 sp3n09 N sp3en0 26 SP_SW_RSGN_BKUP_RE
082716301998 sp3n12 Y sp3en0 21 TB3_LINK_RE
082716301998 sp3n15 Y sp3en0 20 TB3_LINK_RE
090511081998 sp3n07 N sp3en0 32 SP_SW_SNDLNKSYNC_RE
```

Figure 112. Example of the `summlog` File on the CWS

#### 5.4.1.5 New Daemon

A new daemon `css.summlog`, which runs on the CWS, connects to the event manager daemon in each partition and generates a summary log record in the `summlog` file for every event it receives.

The `css.summlog` daemon runs under SRC control with the subsystem name of `swtlog`. It is added to `/etc/inittab` by the `install_cw` script during the installation so that it automatically starts up when the CWS boots.



Figure 113 on page 179 shows an example of the `/var/adm/SPlogs/css/summlog.out` file on the CWS which contains the events received and the actions responded to by the `css.summlog` daemon.

```
css.summlog: ha_em_start_session for sp3en0 succeeded - continuing.
css.summlog sp3en0: EM error for IBM.PSSP.CSSlog.errlog on sp3en0: gen: 2 spec: 3
css.summlog sp3en0: EM error for IBM.PSSP.CSSlog.errlog on sp3n01: gen: 2 spec: 3
css.summlog sp3en0: EM error for IBM.PSSP.CSSlog.errlog on sp3n13: gen: 2 spec: 3
css.summlog sp3en0: EM error for IBM.PSSP.CSSlog.errlog on sp3n05: gen: 2 spec: 3
css.summlog sp3en0: EM error for IBM.PSSP.CSSlog.errlog on sp3n06: gen: 2 spec: 3
css.summlog: ha_em_start_session for sp3en0: 8 - 2521-630 Attempt to connect to an Event Manager daemon
with a UNIX domain socket
failed: connect(): Connection refused.
We will retry until we have success.
css.summlog: ha_em_start_session for sp3en0 succeeded - continuing.
css.summlog sp3en0: EM error for IBM.PSSP.CSSlog.errlog on sp3en0: gen: 2 spec: 3
css.summlog sp3en0: EM error for IBM.PSSP.CSSlog.errlog on sp3n08: gen: 2 spec: 5
```

Figure 113. Example of the `summlog.out` File on the CWS

Other changes made in order to support this function are:

- The `css.snap` command was modified to accept a new flag, `-s`, to indicate that this is a soft dump and thus, no `tb3dump` is needed.

`css.snap` now also extracts the switch adapter information from ODM.

- The `haemloadlist` was modified to include the new resource variables.

#### 5.4.2 Example Scenario

In the following example, we killed the Worm daemon on the primary node. After a while we got seven new records in the `summlog` file on the CWS.

Three of them came from node 13 (primary node), indicating:

1. `SP_SW_SNDLNKSYNC_RE` - Switch sender link sync error
2. `SP_SW_SIGTERM_ER` - Switch daemon received SIGTERM
3. `HPS_FAULT6_ER` - Switch Fault Service Daemon Terminated

Four records came from node 14 (primary backup node), indicating:

1. `SP_SW_UNINI_NODE_RE` - Nodes not initialized during Estart
2. `SP_SW_UNINI_LINK_RE` - Links not initialized during Estart
3. `SP_SW_SDR_FAIL_RE` - Switch daemon SDR communications failed
4. `SP_SW_PRI_TAKOVR_RE` - Switch primary node takeover

```

sp3en0{ / } tail /spdata/sys1/ha/css/sumlog
091019481998 sp3n15 N sp3en0 167 SP_SW_UNINI_LINK_RE
091109501998 sp3n13 N sp3en0 70 SP_SW_RSGN_BKUP_RE
091109521998 sp3n15 N sp3en0 169 SP_SW_RSGN_PRIM_RE
091110221998 sp3n15 N sp3en0 170 SP_SW_SDR_FAIL_RE
091110221998 sp3n15 N sp3en0 171 SP_SW_SDR_FAIL_RE
091110271998 sp3n13 N sp3en0 71 SP_SW_RSGN_BKUP_RE
091110301998 sp3n15 N sp3en0 172 SP_SW_RSGN_PRIM_RE
091111121998 sp3n15 N sp3en0 173 SP_SW_SDR_FAIL_RE
091111121998 sp3n15 N sp3en0 174 SP_SW_SDR_FAIL_RE
091114531998 sp3n15 N sp3en0 175 SP_SW_SNDLNKSYNC_RE
sp3en0{ / }
sp3en0{ / } Eprimary
13      - primary
13      - oncoming primary
14      - primary backup
14      - oncoming primary backup
sp3en0{ / }
sp3en0{ / } dsh -w n13 'ps -ef|grep orm|grep -v grep'
n13: root  8384 1 0 Sep 12 - 0:55 /usr/lpp/ssp/css/fault_servi
ce_Worm_RTG_SP -r 12 -b 1 -s 4 -p 1 -a TB3 -t 22
sp3en0{ / } dsh -w n13 kill 8384
....
....
sp3en0{ / } tail /spdata/sys1/ha/css/sumlog
091111121998 sp3n15 N sp3en0 173 SP_SW_SDR_FAIL_RE
091111121998 sp3n15 N sp3en0 174 SP_SW_SDR_FAIL_RE
091114531998 sp3n15 N sp3en0 175 SP_SW_SNDLNKSYNC_RE
091416431998 sp3n13 N sp3en0 140 SP_SW_SNDLNKSYNC_RE
091418381998 sp3n13 N sp3en0 142 HPS_FAULT6_ER
091418381998 sp3n13 N sp3en0 141 SP_SW_SIGTERM_ER
091418421998 sp3n14 N sp3en0 166 SP_SW_UNINI_NODE_RE
091418421998 sp3n14 N sp3en0 167 SP_SW_UNINI_LINK_RE
091418421998 sp3n14 N sp3en0 168 SP_SW_SDR_FAIL_RE
091418421998 sp3n14 N sp3en0 169 SP_SW_PRI_TAKOVR_RE
sp3en0{ / }
sp3en0{ / } Eprimary
14      - primary
13      - oncoming primary
1       - primary backup
14      - oncoming primary backup

```

Figure 114. The sumlog File When a Primary Node Takeover Occurs

In the summlog file, we found that someone had killed the Worm daemon on the primary node (node 13), and in less than two minutes the primary backup node (node 14) detected this situation and took over the primary node responsibility.

You also noticed that no snap was taken, since snap data is not needed in the resolution of these events.

Thus, with PSSP 3.1, you can easily find out what switch-related events are going on in the system just by monitoring the summlog file!

### **5.4.3 Coexistence Consideration**

This function works for a PSSP 3.1 node only. It does not work in earlier PSSP releases because the error notification objects in those releases still use `errlog_rm`, which does not trigger a summary record creation on the CWS.

Thus, if you have mixed levels of PSSP nodes, only PSSP 3.1 nodes will have summary records in the consolidated log file on the CWS.



---

## Chapter 6. RS/6000 Cluster Technology

SP High Availability Infrastructure (HAI) has been replaced by RS/6000 Cluster Technology (RSCT). This term refers to the overall package of technologies, which currently includes:

1. Topology Services (TS)
2. Group Services (GS)
3. Event Management (EM)

New terminology for RSCT consists of the following terms:

### Cluster

The term *cluster* is used to refer to a collection of RS/6000 machines on which RSCT components are executing. These machines may exclusively be nodes on an RS/6000 SP, standalone RS/6000 workstations, or a combination of both sorts of machines. This is a significant change in our concept of a cluster.

Note that a cluster does not have to be exclusive. A machine may be contained in multiple clusters. An example would be an SP node that has HACMP installed. Within the PSSP, the node is part of the SP cluster but it is also part of HACMP cluster. Each cluster is independent of the other and the subsystems within each are independent.

### Domain

The term *domain* is subtly more specific than cluster. A domain describes the boundary of a set of machines within which the executing RSCT components provide their services. In general, the boundaries of a cluster match the boundaries of a domain. The differentiation is that a cluster does not become a domain until Group Services has established its domain via one of the Group Services daemons on a node within the cluster. At this point, the clients of Group Services are allowed to form their groups and begin offering their services.

As with the description of a cluster, a domain may overlap with another domain; in other words, it is possible for a machine to join multiple domains. The expected domains are:

- *SP domain*

The boundaries of SP domain are the control workstation (CWS) and the nodes within an SP partition. Thus each SP partition is a separate domain.

- *HACMP domain*

The boundaries of HACMP domain differ based on the release:

- For HACMP/ES V4.2.2 or lower, the domain was a proper subset of an SP domain.
- For HACMP/ES V4.3, the domain is any allowable cluster.

### Realm

A *realm* is used to refer to the previously mentioned domains, but in general terms, rather than when discussing a specific domain. Thus, the term SP realm refers to any domain established on any generic SP to support the PSSP. The term HA realm refers to any domain established on any generic cluster to support HACMP/ES.

### Dual Daemons

The term *dual daemons* represents the fact that there are separate sets of RSCT components supporting the separate domains. Thus, on hardware that is part of both the SP realm and the HA realm, each node will execute:

- A Topology Services (TS) daemon for each realm.
- A Group Services (GS) daemon for each realm.
- An Event Management (EM) daemon for each realm.

Figure 115 on page 184 illustrates the RSCT infrastructure.

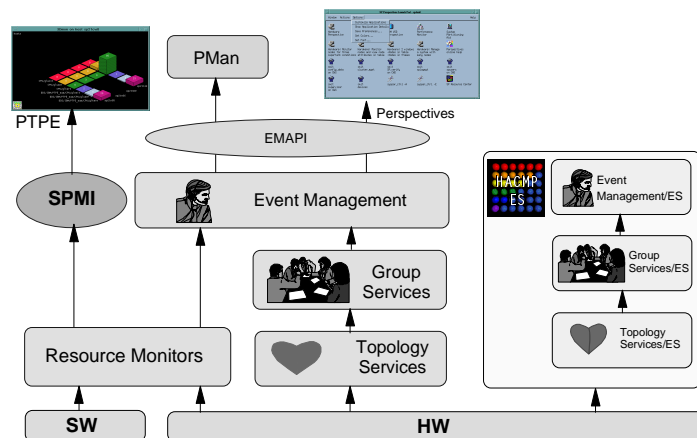


Figure 115. RSCT Infrastructure

For more information on High Availability components, refer to *RS/6000 SP High Availability Infrastructure*, SG24-4838.

## 6.1 RSCT Packaging

In previous releases of PSSP, the RSCT components were packaged and shipped as part of the ssp.\* filesets. The ssp.ha fileset includes all HAI files. With PSSP 3.1, a new package is created. It is called RSCT 1.1. The ssp.ha fileset is removed from the PSSP package. There are two install images in this package, as shown in Table 17.

Table 17. RSCT Install Images

Fileset	Description
rsct.basic.rte	RSCT basic function (all realms)
rsct.basic.sp	RSCT basic function (SP realm)
rsct.basic.hacmp	RSCT basic function (HACMP realm)
rsct.clients.rte	RSCT client function (all realms)
rsct.clients.sp	RSCT client function (SP realm)
rsct.clients.hacmp	RSCT client function (HACMP realm)

### 6.1.1 Coexistence of rsct and ssp.ha/ssp.topsvcs on a Node

The ssp.ha and ssp.topsvcs filesets are removed from the PSSP package. The ssp.topsvcs fileset was added in PSSP-2.3 to support HACMP/ES V4.2.2. The rsct.\* filesets supercede the ssp.ha and ssp.topsvcs filesets and the following parts of the ssp.clients fileset:

```
/usr/lib/nls/msg/en_US/ha_em.cat  
  
/usr/lpp/ssp/lib/libha_em.a  
  
/usr/lib/libha_em.a /usr/lpp/ssp/lib/libha_em.a  
  
/usr/lpp/ssp/lib/libha_em_r.a  
  
/usr/lib/libha_em_r.a /usr/lpp/ssp/lib/libha_em_r.a  
  
/usr/lpp/ssp/lib/ha_em_partinfo
```

The ssp.ha\_topsvcs.compat fileset is added for HACMP/ES interoperability. The rsct.\* filesets interoperate with different level of PSSP code.

## 6.1.2 Directory Structure of RSCT

The rsct filesets install in the following directories instead of in the /usr/lpp/ssp/bin directory:

```
/usr/sbin/rsct/bin
```

```
/usr/sbin/rsct/include
```

```
/usr/sbin/rsct/lib
```

```
/usr/sbin/rsct/install
```

The /var/ha/\*, /spdata/sys1/ha, and /etc/ha/cfg directories are still used. In these directories, parts are differentiated by domain name (partition name).

Some parts continue to be installed in /usr/include, /usr/lib, and /usr/lib/nls/msg/\*. These parts are symbolic links and message catalogs.

## 6.1.3 Prerequisites

The rsct.basic fileset is a prerequisite for the ssp.basic fileset. The perfagent.tools fileset is a prerequisite for rsct.basic. Prior to AIX 4.3.2, SPML technology comes from the PTX Performance Agent (PAIDE), which includes the perfagent.server fileset. SPML libraries are now moved from the perfagent.server fileset to the perfagent.tools fileset in AIX 4.3.2. However, the perfagent.server file set must be installed on the CWS to support any node at PSSP 2.2, 2.3, 2.4. If all nodes are at PSSP 3.1, then the perfagent.server is not required. Note that the perfagent.server is still a prerequisite for Performance Toolbox Parallel Extension (PTPE), which is an optional feature of PSSP 3.1.

---

## 6.2 Topology Services

The Topology Services subsystem provides the foundation for the PSSP 3.1 RSCT Infrastructure. This subsystem is distributed across all nodes in an environment and maintains availability information about the nodes and adapters. Another primary responsibility of TS is to provide the network roadmap, or Network Connectivity Table (NCT) for use by GS's Reliable Messaging library. The operation of this process is illustrated in Figure 116 on page 187.



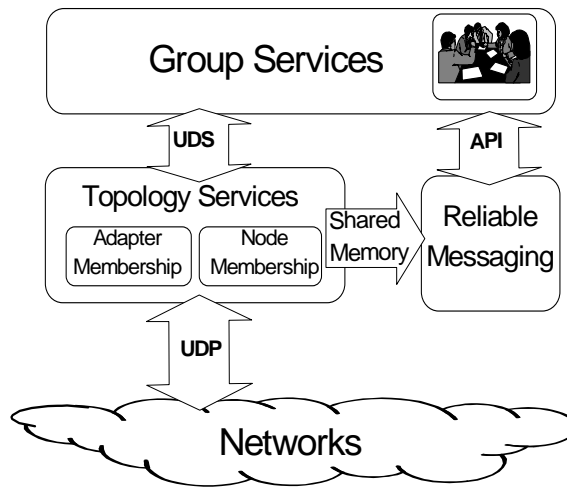


Figure 116. TS and GS Interfaces

The following terms are components used in TS terminology:

*Adapter Membership* describes the process of monitoring the availability of different network adapters defined in the environment. *Adapter Membership Group* is an association of adapters that establish their relationship for monitoring and routing purposes.

*Node Membership* is the process for maintaining node availability information based on Adapter Membership. If adapters on different nodes are in the same Adapter Membership Group (AMG), then the nodes are able to communicate. Nodes can also communicate indirectly by routing across different AMGs.

*Group Leader (GL)* is the adapter of the highest IP address within an AMG. The GL is responsible for maintaining the topology and connectivity information for the group and distributing it to members.

*Crown Prince (CP)* is the adapter with the second highest IP address in the group. CP takes over as GL when the GL fails. If GL comes back, it takes over.

A *Singleton Group* is an Adapter Membership Group which consists of only one member. All adapters initialize into a Singleton before they join a larger group.

TS configuration flow:

The Topology Services control script is contained in the executable file `/usr/sbin/rsct/bin/hatsctrl`. This script is normally invoked by the `syspar_ctrl` script, which provides an interface to all of the system partition-sensitive subsystems.

Note that, before you run this command, you should ensure that the `SP_NAME` environment variable is set to the appropriate system partition name and exported. If the `SP_NAME` variable is not set, the `/etc/SDR_dest_info` file is referenced.

- TS control script `hatsctrl` executes the hats startup script. It is a shell script that is used to obtain the configuration from the SDR and start up the TS daemon. This script is invoked by SRC. The first part of initialization is done by the startup script `hats`.
  - The startup program obtains the number of the node on which it is running using the `/usr/lpp/ssp/install/bin/node_number` command.
  - The startup program obtains the name of the system partition from the Syspar SDR class.
  - The startup program receives node and adapter configuration information from the SDR at the CWS and builds the Machines List file. The name of Machines List file is `machines.lst` which can be found in TS current working directory, `/var/ha/run/hats.<syspar_name>`.
  - The startup program performs file maintenance in the log directory and current working directory.
  - Finally, the startup script executes the TS daemon `hatsd`. Figure 117 on page 188 is an example of the output of the `lssrc -ls hats.<syspar_name>` command.

```
[root@sp4en0]# lssrc -ls hats.sp4en0
Subsystem      Group      PID      Status
hats.sp4en0    hats       20900    active
Network Name   Indx Defd Mbrs St Adapter ID      Group ID
SPether        [ 0]  11   10  S 192.168.4.130  192.168.4.130
SPether        [ 0]           0x45fbcfe6      0x45fd16f9
HB Interval = 1 secs. Sensitivity = 4 missed beats
  2 locally connected Clients with PIDs:
haemd( 19370) hagsd( 20124)
Configuration Instance = 904854369
Default: HB Interval = 1 secs. Sensitivity = 4 missed beats
CWS = 192.168.4.130
```

Figure 117. `lssrc -ls hats.sp4en0` Command

- The TS daemon continues the initialization with the following steps:
  - Read the current Machines List file.
  - Form the adapter rings.
  - Create Adapter Membership Groups.
  - Build a topology table and graph to indicate individual node connectivity, therefore availability.
  - Build the Network Connectivity Table in shared memory, specifying all valid routes between all node combinations.
  - Accept connections from local clients, primarily GS.
  - Monitor adapters via heartbeats and update the tables as necessary.

Figure 118 shows the process flow of TS.

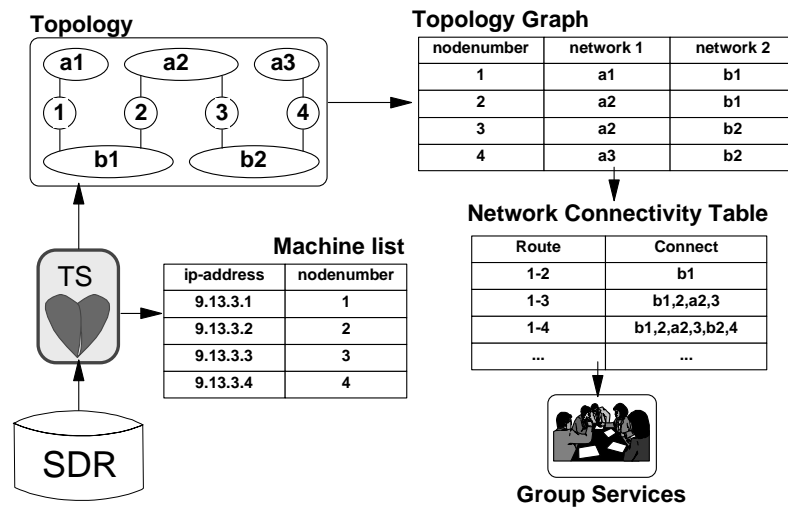


Figure 118. TS Process Flow

### 6.2.1 Heartbeat on Additional Networks

Topology Services daemon supports for HACMP/ES heartbeating on the following networks: Ethernet, token-ring, FDDI, ATM, tmSCSI, tmSSA, rs232, SLIP and the SP Switch. However, for PSSP, only Ethernet and SP Switch are supported.

Three new classes are added into the SDR for supporting heartbeat on other network types. These classes are Network, Subnet, and TS\_Tunable. The configuration of these classes will be provided in a later release. All these

new classes and attributes are described in Appendix A, “Changes to the SDR” on page 319.

In the HACMP domain, TS allows each network type to have different tuning parameters. The HACMPtopsvcs GODM class is updated to include the new tuning parameters. All networks have different capabilities and therefore can have different heartbeat rates. This provides the ability to reduce the network load and have a faster fail-over in HACMP clusters. Figure 119 is an example of the HACMPtopsvcs GODM class.

```
sp4n13:/etc/objrepos >odmget HACMPtopsvcs

HACMPtopsvcs:
  hbInterval = 1
  fibrillateCount = 5
  runFixedPri = 1
  fixedPriLevel = 38
  tsLogLength = 5000
  gsLogLength = 5000
  instanceNum = 18
```

Figure 119. HACMPtopsvcs GODM Class

### 6.2.2 Dynamic Update (Refresh)

TS now fully supports dynamic update ("refresh") for configuration changes, for both PSSP and HACMP/ES Dynamic Automatic Reconfiguration Event (DARE). The refresh operation consists of the hats daemon obtaining the new system configuration from the SDR in the SP realm or from the GODM in the HACMP realm. Secondly, the refresh operation applies the new configuration to the daemon's data structures. The new configuration may differ from the old by:

- Changing daemon parameters
- Adding new adapters or nodes
- Removing adapters or nodes
- Adding or removing whole networks

If new nodes or adapters are added/removed from the TS configuration, the TS has to be aware of the new configuration. This is done through refreshing TS. Run the following command from CWS to refresh TS:

```
# hatsctrl -r
```

The daemon handles IP address changes automatically.

### 6.2.3 New SDR Classes

New classes are added to SDR for future release support. For more information about new SDR classes, refer to Appendix A, “Changes to the SDR” on page 319.

### 6.2.4 Topology Services in HACMP Domain

According to the new RSCT infrastructure, there are two different domains and two different data repositories. HACMP/ES uses Global ODM and PSSP uses SDR. If an HACMP cluster consists of standalone RS/6000s and SP nodes, it may have network interfaces not known to PSSP. Topology Services relies on the fact that all nodes have the same view of the configuration. In other words, the machine list file should be same in an SP domain. In the case where SP and HACMP domains coincide in a node, dual daemons work for different domains.

The Topology Services running in the HACMP domain uses the topology information in the GODM at start time. At TS start time, the System Resource Controller (SRC) subsystem calls the `/usr/sbin/rsct/bin/topsvcs` script, which creates the `machines.lst` file and starts the TS daemon. The `machines.lst` file is located in the `/var/ha/run/topsvcs.<cluster_name>` directory. Its name is `machines.<cluster_id>.lst`.

Figure 120 on page 192 is an example of machines list file in an HACMP domain. The `<cluster_name>` is ES43 and `<cluster_id>` is 1.

```

sp4n13:/var/ha/run/topsvcs.ES43 >pg machines.1.lst
*InstanceNumber=18
*configId=248620498
*!TS_realm=HACMP
*!TS_EnableIPAT
*!TS_PinText
TS_Frequency=1
TS_Sensitivity=5
TS_FixedPriority=38
TS_LogLength=5000
Network Name ether1_0
Network Type ether
*
*Node Type Address
  3 en1 128.100.10.130
  4 en1 128.100.10.150
*!Service Address=128.100.10.15
*!Service Address=128.100.10.15
Network Name ether1_1
Network Type ether
*
*Node Type Address
  3 en2 128.100.20.13
  4 en2 128.100.20.15
Network Name spether_0
Network Type ether
*
*Node Type Address
  3 en0 192.168.4.13
  4 en0 192.168.4.15
Network Name HPS_2_0
Network Type hps
*!TS_Sensitivity=4
*
*Node Type Address
  3 css0 140.4.4.130
  4 css0 140.4.4.150
*!Service Address=140.4.4.15
*!Service Address=140.4.4.15
Network Name HPS_1_0
Network Type hps
*!TS_Sensitivity=4
*
*Node Type Address
  3 css0 192.168.14.13
  4 css0 192.168.14.15

```

Figure 120. Machines List File in HACMP Domain

## 6.2.5 Node/Adapter Numbering

Node numbering in PSSP is more or less contiguous. But some configurations may leave gaps in Topology Services such as wide nodes (2 slots) and high nodes (4 slots). Different node numbering utilities are used to prevent duplication of node numbers in an SP domain or an HACMP domain. An HACMP/ES cluster can consist also of nodes in different SP frames. This includes clusters made up of SP nodes from different SP systems.

When TS operates in PSSP domain, it uses the `hats_node_number` utility to determine its own node number. Figure 121 on page 193 is an example of the `hats_node_number` utility. It looks for domain name.

```
[root@sp4en0]</>export HA_DOMAIN_NAME=sp4en0
[root@sp4en0]</>hats_node_number
0
[root@sp4en0]</>hats_node_number -d sp4en0
0
```

Figure 121. `hats_node_number` Utility

The corresponding utility in HACMP domain is `clhandle` command. For more information about this command, refer to 8.5.1, “`clhandle`” on page 242.

### 6.2.6 Large Systems Improvements

RSCT 1.1 supports PSSP partitions up to 512 nodes. There are some improvements to increase performance and reliability in large configurations, such as those over 128 nodes.

In large configurations, SP internal network traffic is heavy and SP Ethernet cannot handle it. In PSSP 3.1, node connectivity messages of Topology services are reduced. Group members of Topology Services stop now to send these messages after their groups have been stabilized. This reduces CPU utilization by the daemons and decreases the traffic on the SP internal network.

For more information on this subject, refer to "Topology Services Tuning for RPQ systems" in PSSP Release Notes, which is called `rsct.basic.README` and is located in the directory `/usr/sbin/rsct/README`.

---

## 6.3 Group Services

The Group Services subsystem is a partition-sensitive, fault-tolerant and highly available service that provides a general purpose facility for coordinating and monitoring changes to the state of a subsystem running on a set of nodes.

A Group Services daemon (`hagsd`) runs on each RS/6000 SP node and on the CWS. If there is more than one partition, then one daemon is executed on the CWS for each partition. GS daemons exchange their own reliable-protocol messages over UDP/IP. The communication paths selected

are provided by TS through the NCT. Figure 122 on page 194 illustrates the GS structure.

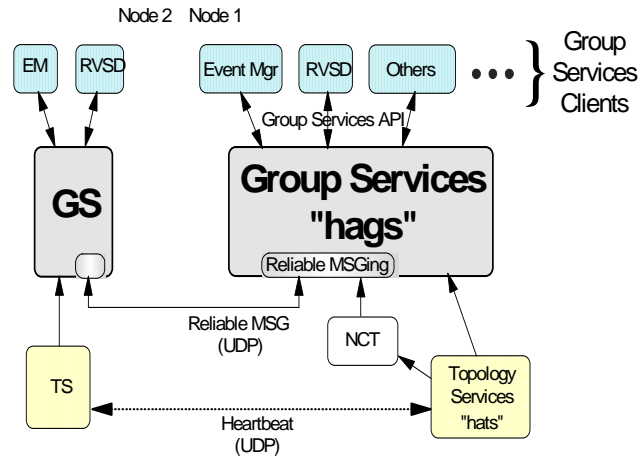


Figure 122. Group Services Structure

GS is based on forming groups. If an application wants to take advantage of the coordination and monitoring functions of GS, it must first form a group. Any process in a GS domain can create a new group, and any process in the domain may ask to become a member of a group. Such group members are called *providers*. Another process may only wish to be informed of the group's activities. If the subscribe request of this process is accepted, they become a *subscriber* to the group. A subscriber is only notified of the group's activities. Providers and subscribers are Group Services clients.

Groups are defined by the following three attributes:

**Name**

A token uniquely identifies each group in the system.

**Membership List**

A membership list is a list of one or more providers. In a group, each provider is identified by its identifier, which consists of an Instance ID and the node number on which the provider is running.

**Group State Value**



A Group State Value is a byte field whose length is between 1 and 256 bytes. It is not interpreted by GS.

TS provides necessary information that consists of Adapter Membership and Node Membership to form a group by GS. GS clients can subscribe to these groups to monitor hardware status.

Any process of an application or subsystem that uses the Group Services Application Programming Interface (GSAPI) can create groups. Examples from IBM are EM, RVSD, GPFS, and HACMP/ES. Use of GS is optional for other applications. GS itself does not perform recovery actions, but provides the synchronization and commit protocols.

Figure 123 on page 196 illustrates how Group Services clients can get services from a GS subsystem and how a GS subsystem works internally.

A GS subsystem consists of several modules:

- **TS Client Module.** GS receives services of TS through this module.
- **Reliable Messaging Module.** This module, using the NCT created by TS in shared memory, provides reliable, sequenced delivery of messages between GS daemons.
- **Name Server Module.** This module controls the GS namespace.
- **Client Control Module.** This module handles the connection with GS clients. It also accepts requests from GS clients and manages them.
- **Meta Group Control Module.** Meta groups are the collection of the Group Services daemons that supports the user groups. This module manages the behavior of GS daemons.

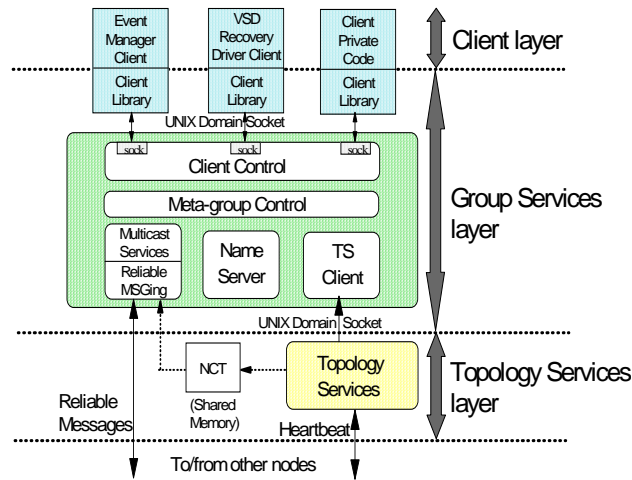


Figure 123. GS Functional Flow

### 6.3.1 New GS Protocols

GS has three new protocols to provide better support for its client subsystems to perform migration and recovery. These new protocols are used by the new release of HACMP/ES, and are available to any user of the GSAPI.

#### 6.3.1.1 ha\_gs\_change\_attributes()

The group attributes define various control and behavior conditions for a group, and remain established as long as there are providers in the group. Previously, if the group's providers wanted to change the group attributes, all providers would have to leave the group, and rejoin the group using new attributes. In PSSP 3.1, the providers can dynamically change most of the attribute settings using the `ha_gs_change_attributes()` asynchronous interface without having to leave the group. This provides sufficient function for HACMP/ES to migrate from 1-phase to N-phase joins.

The `ha_gs_change_attributes` subroutine is used by a provider of a GS group to propose a change to the group's attributes. The following attributes can be changed via an `ha_gs_change_attributes()` subroutine call:

- The `gs_client_version` attribute contains a user-defined version code.
- The `gs_batch_control` attribute controls the batching of multiple group joins and failure leaves.
- The `gs_num_phases` attribute specifies whether join protocols and failure leave protocols are to be n-phase or one-phase protocols.

- The `gs_source_reflection_num_phases` attribute contains the number of phases for source-reflection protocols, which are run in the target-group when the source-group changes its state value.
- The `gs_group_default_vote` attribute contains the default vote to use for the providers in this group.
- The `gs_merge_control` attribute specifies how the merging of groups should be handled.
- The `gs_time_limit` contains the voting time limit in seconds.
- The `gs_source_reflection_time_limit` contains the time limit in seconds for each voting phase of a source reflection protocol, which is run in the target-group when the source group changes its state value.

For details of these attributes, see *RSCT Group Services Programming Guide and Reference*, SA22-7355.

For the group to successfully use the `ha_gs_change_attributes()` subroutine to dynamically change the group's attributes, all providers in the group must be at the PSSP 3.1 level for a specific domain (SP or HACMP). The semantics of the `ha_gs_change_attributes()` subroutine includes synchronous and asynchronous returns. Figure 124 on page 197 illustrates the sample protocol execution for this protocol. Note that not all return codes are covered. For more information, refer to *RSCT Group Services Programming Guide and Reference*, SA22-7355.

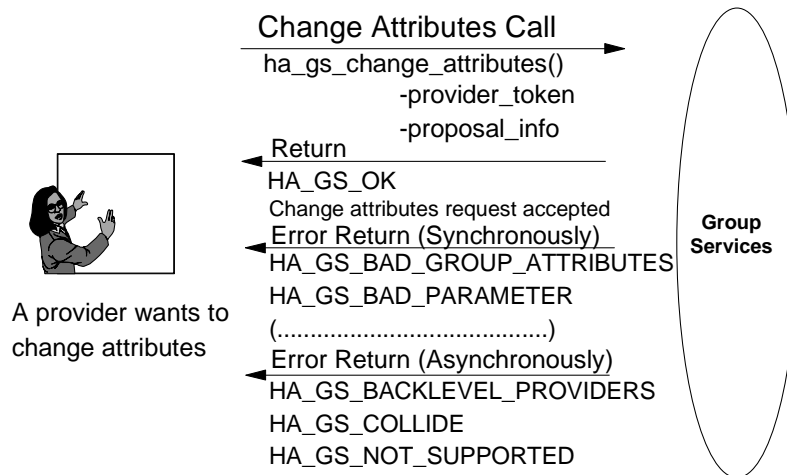


Figure 124. `ha_gs_change_attribute` Sample Protocol Execution

### 6.3.1.2 ha\_gs\_goodbye()

GS providers can leave a group in a number of ways:

- They may leave voluntarily.
- They may be expelled at the request of another provider.
- They may leave involuntarily when the provider process, or the node on which it is running, fails.

The `ha_gs_goodbye()` subroutine enables a provider to immediately leave its group as if it had failed, while informing the group that it "failed" voluntarily. A provider says goodbye when it wants to immediately leave a group. It should be used instead of the voluntary "leave interface" (`ha_gs_leave()`) when the provider does not need to specify a specific leave code or if it cannot risk the possibility where the voluntary leave may not execute immediately.

The `ha_gs_goodbye()` subroutine is called by a provider of a group to immediately leave the group. This is a synchronous interface. If this call returns with an `HA_GS_OK` return code, then the calling provider is no longer in the group. Figure 125 on page 198 illustrates sample protocol execution for this protocol.

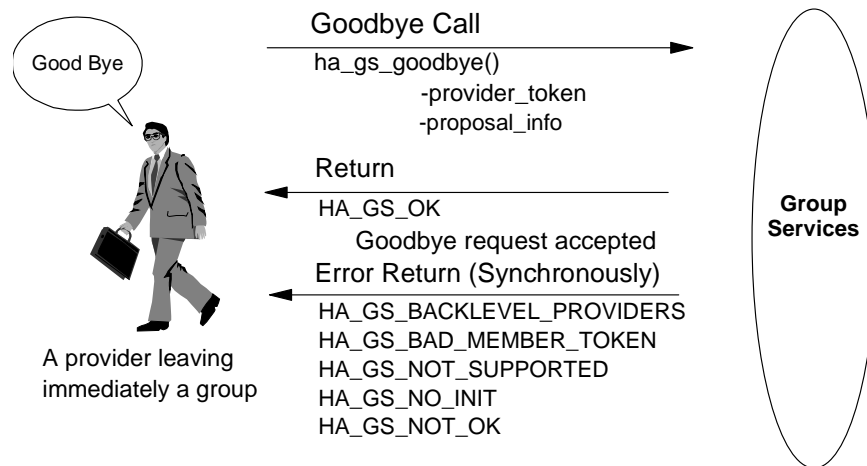


Figure 125. `ha_gs_goodbye` Sample Protocol Execution

### 6.3.1.3 Deactivate on Failure

In this release, Group Services will optionally execute a deactivate script when a provider's process fails to allow recovery and clean-up actions on a node. The deactivate script will be executed in the case of a local provider's process failure as well as in the case of a provider being forced out of the

group via an expel protocol. When a provider is failing, its group is forced into a failure leave protocol. This support is called "deactivate-on-failure" and is activated by setting the appropriate values in the group's attributes. The deactivate script is executed by GS during the first phase of the failure leave protocol that is executed for the provider's failure.

The following example describes the execution of a deactivate script.

1. Group **foo** has three providers, **P1**, **P2**, and **P3** on nodes **1**, **2**, and **3**. They all specify a deactivate script `/foo/deactivate` and specify that failure-leave protocols should be n-phase.
2. Group **foo** provider on the node 3, **P3**, fails because its process has failed (for example, killed by a signal, core dumps due to a programming error, encounters an unexpected situation).
3. GS starts the n-phase failure-leave protocol for group **foo**, notifying all remaining providers, **P1** and **P2**, that **P3** failed.
4. During the first phase of failure-leave protocol, providers **P1** and **P2** perform whatever actions are appropriate.
5. On node 3, GS executes the deactivate script `/foo/deactivate` and waits for it to complete. Assuming the deactivate script succeeds with an exit code of 0, the GS votes APPROVE on behalf of **P3**.
6. If there are subsequent voting phases for the failure-leave protocol, then for each phase, GS continues to vote APPROVE on behalf of **P3**.

### 6.3.2 RSCT Enhancements to Support HACMP/ES in GS

In order to provide standalone RS/6000 support, HACMP/ES V4.3 and the RSCT components run on HACMP domain have been changed and do not use SDR information. The RSCT components run on HACMP domain use only Global Object Data Manager (GODM).

TS maintains separate heartbeat rings on each defined set of SP switch alias addresses. Group Services provides indications of each separate alias address when notifying any GS clients subscribing to the `cssRawMembership` adapter group for adapter up/down changes. HACMP/ES uses this information to determine which `css` alias address is affected by the up/down change.

---

## 6.4 Event Management

The Event Management (EM) subsystem provides information by monitoring various hardware and software resources in the system. For example,

monitored resources are CPUs, disks, filesystems, processes, and database systems.

The Event Management subsystem consists of three types of components:

### **Event Manager Daemon**

The Event Manager daemon (haemd) is the central program that gathers resource variable instances as they are reported by the resource monitors. It monitors the resource variables and expressions for which EM clients have registered. A condition in which an EM client is interested is called an *expression*. The Event Manager subsystem generates an event and notifies the appropriate EM client that has registered for this event when an expression evaluates to TRUE.

### **Event Management Clients**

An Event Management Client is an application program or subsystem. It uses the Event Management Application Programming Interface (EMAPI) to register interest in particular events, to receive information when the events occur, and to query resource variable values and definitions.

### **Resource Monitors**

The Resource Monitors are programs that monitor the state of specific system resources and transform this state into several resource variables. The Resource Monitors periodically provide these variables to the Event Manager daemon through the Resource Monitor Application Programming Interface (RMAPI).

Figure 126 on page 201 illustrates EM's functional design.

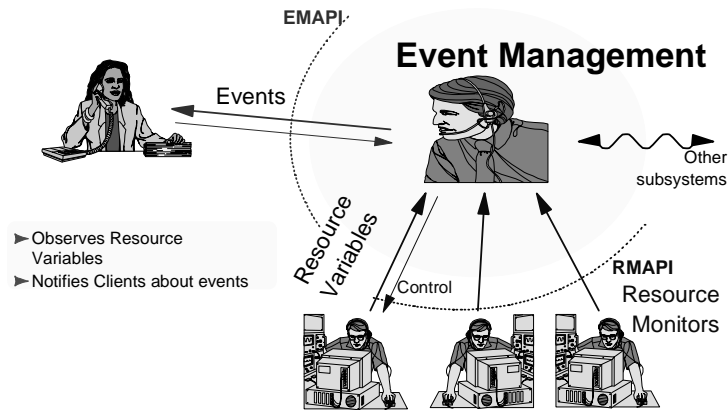


Figure 126. EM Functional Design

### 6.4.1 Shared Memory Segment

Use of Performance Toolbox (PTX) Technology is removed from Event Management. The mechanism to transfer the values of resource variable instances from a resource monitor to the EM daemon has changed in PSSP 3.1. Instead of using the SPMI library functions to manage shared memory, the daemon now uses System V Inter Process Communication (IPC) shared memory directly.

The following enhancements are done in PSSP 3.1:

- SPMI calls are removed from the EM daemon.
- System V IPC is used as a shared memory mechanism between the Event Management daemon and the Resource Monitor Application Programming Interface (RMAPI).
- The RMAPI continues to use SPMI to supply data to PTPE. However, SPMI is only initialized if PTPE requests monitoring of resource variables.
- The internal resource monitor for AIX variables (aixos) is removed from the EM daemon. A resource monitor of the type server is created to supply AIX resource variables to the EM daemon using the RMAPI. For details of a server resource monitor refer to 6.4.2, "Resource Monitors" on page 202.

New Shared Memory architecture is illustrated in Figure 127 on page 202.

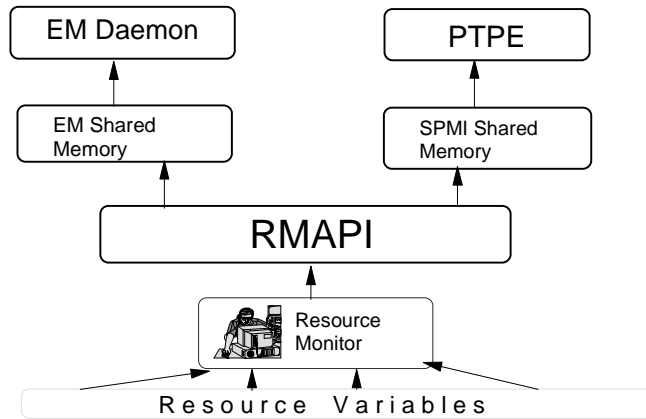


Figure 127. New Shared Memory Architecture

When the Event Manager daemon establishes a connection to a resource monitor that provides Counter and/or Quantity value type resource variables, EM Shared Memory is created by the Event Manager daemon. SPMI shared memory is created by the RMAPI only when the resource monitor is requested to monitor Counter and/or Quantity value type resource variables by PTPE.

### 6.4.2 Resource Monitors

Resource Monitors are those software entities that provide the actual resource variables to a higher level of manager software. Examples of manager software are Performance Toolbox and the Event Management. The resource variables are passed to the manager software through the RMAPI. Resource variables of type state are sent as a message directly to the manager software.

There are two types of resource monitors:

#### Server type

The server type of Resource Monitor expects the manager program to connect to the resource monitor. In this case, after the Resource Monitor program has started, the manager program sends control commands to the Resource Monitor. These commands control the flow of resource variables. Server type resource monitors are usually daemon-based.

#### Client type



Client type resource monitors are command-based. This type of resource monitor connects to the manager program to establish communication. They can provide resource variables only of state type to the Event Management subsystem.

Additionally, the Event Management daemon itself performs some resource monitoring function. This function is considered to be a resource monitor with a connection type of internal.

RSCT supplies the following external resource monitors:

- **IBM.PSSP.harmlid**

This monitor supplies resource variables for the CSS, VSD and LoadLeveler subsystems. This data is also transferred through SPMI to the Performance Monitor subsystem. This is a daemon (harmlid) with a connection type of server.

- **IBM.PSSP.harmpd**

This monitor provides 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 (harmpd) with a connection type of server.

- **IBM.PSSP.hmrmd**

This monitor provides resource variables obtained from the PSSP hardware monitoring subsystem (hardmon). This is a daemon with a connection type of server.

- **IBM.PSSP.pmanrmd**

This monitor supplies the resource variables of the PSSP Problem Management subsystem. This is a command-based resource monitor with a connection type of client (pmand).

- **aixos**

This monitor provides resource variables that represent AIX operating system resources. This is a daemon (harmad) with a connection type of server.

- **IBM.PSSP.CSSLogMon**

This monitor supplies a resource variable that represents the state of CSS error log entries. This is a command-based resource monitor with a connection type of client.

- **IBM.PSSP.SDR**

This monitor provides a resource variable that represents the modification state of SDR classes. This is a command-based resource monitor with a connection type of client.

Figure 128 illustrates SP Resource Monitors of all types.

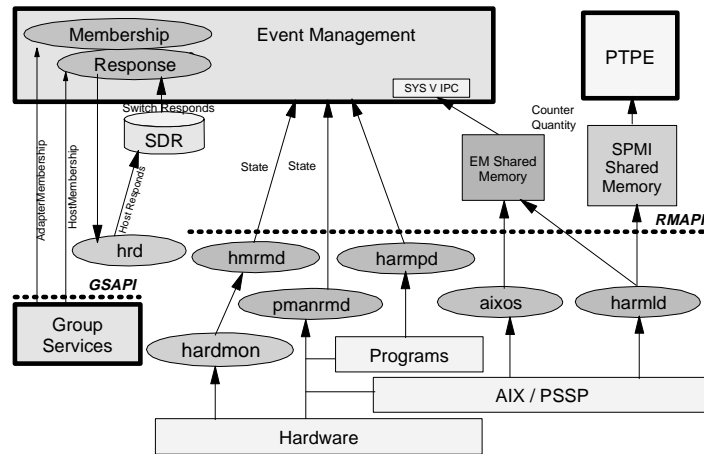


Figure 128. SP Resource Monitors

There are also several internal types of resource monitors. These resource monitors are incorporated in the Event Manager daemon.

- **Membership**

This monitor 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**

This monitor provides resource variables that represent the information in the host\_responds and switch\_responds SDR classes.

Figure 129 on page 205 is an example of the following command:

```
# lssrc -ls haem.sp4en0
```

The resource monitor information in the output of this command specifies the type of resource monitors (C for Client, S for Server, I for Internal). Aixos resource monitor is the type of server (S) as shown in the example.

Also the configuration database version specifies the EMCDB version string and an indication as to whether the version string is taken from the SDR or from the peer group state (SDR or peer). For more information for this command, refer to the *PSSP Administration Guide*, SA22-7348.

```

Subsystem      Group      PID      Status
haem.sp4en0    haem      19370    active

Trace flags set:  None

Configuration Data Base version: 905344401,461095936,0(SDR)

Daemon started on 09/12/98 at 10:05:48.546477056
  running 0 days, 5 hours, 11 minutes and 53 seconds
Daemon connected to group services: TRUE
Daemon has joined peer group:      TRUE
Daemon communications enabled:     TRUE
Peer count:                          7

Peer group state:
  905344401,461095936,0

Logical Connection Information
Type  LCID  FD  Node/PID  Start Time
local  0    11  23478    Sat Sep 12 14:16:22 1998
local  1    14  14708    Sat Sep 12 14:16:24 1998
local  2    15  21930    Sat Sep 12 14:16:27 1998
local  3    16  17030    Sat Sep 12 14:16:27 1998
local  4    17  24012    Sat Sep 12 14:16:41 1998
local  5    18  24012    Sat Sep 12 14:16:42 1998
local  7    24  19196    Sat Sep 12 14:52:44 1998
local  8    25  19196    Sat Sep 12 14:53:16 1998

Resource Monitor Information
Resource Monitor Name  Inst  Type  FD  SHMID  PID  Locked
IBM.PSSP.CSSLogMon    0     C    -1  -1     -2  No 00/00
IBM.PSSP.SDR          0     C    -1  -1     -2  No 00/00
IBM.PSSP.harmlld      0     S    22   6    31242  No 01/01
IBM.PSSP.harmpd       0     S    20  -1    29974  No 01/01
IBM.PSSP.hmrmd        0     S    23  -1    28582  No 03/03
IBM.PSSP.pmanrmd      0     C    13  -1     -2  No 00/00
Membership            0     I    -1  -1     -2  No 00/00
Response              0     I    -1  -1     -2  No 00/00
aixos                 0     S    12  393221 -2  No 00/01

```

Figure 129. *Issrc -ls haem.sp4en0* Command

The purpose of the aixos resource monitor is to provide the resource variables that represent AIX Operating System resources like CPU (idle, kern, user, wait), disks, file systems, LAN, memory, paging space, and processes (runque, swpque). Aioxos resource monitor collects values for selected AIX statistics and feeds that data to the Event Management daemon through the RMAPI.

In previous releases of PSSP, the aixos resource monitor was an internal resource monitor of the Event Management daemon. In PSSP 3.1, the aixos

resource monitor is controlled by SRC and is a daemon with a connection type of server (harmad).

For migration and coexistence purposes, the aixos resource monitor is still defined as an internal resource monitor in EMCDB and SDR. The RMAPI and Event Management daemon for PSSP 3.1 converted the type internal to a server for the aixos resource monitor. The aixos resource monitor is controlled by SRC. Figure 130 on page 206 is an example of the haemaixos subsystem on CWS.

```
#lssrc -ls haemaixos.sp4en0
Subsystem      Group          PID    Status
haemaixos.sp4en0 haem          18598  active

Trace Level:      None
Domain Type:      SP
Domain Name:      sp4en0
RMAPI Initialized: TRUE
Data Initialized: TRUE
Data Init. Attempts: 1
Data Init. Delay: 5
Inst. Interval:   600
Inst. Count:      32
SRC FD:           3
Server FD:        7
Class Count:      7
Variable Count:   41
```

Figure 130. *lssrc -ls haemaixos.sp4en0* Command

A new functionality in the aixos resource monitor for the PSSP 3.1 is to pick up automatically new definitions of AIX-related resources. These can be new Volume Groups, Logical Volumes, disks, LAN adapters, and paging space. Every ten minutes it looks for new resources. As you can see in Figure 130 on page 206, the instance interval variable has a value of 600 seconds (10 minutes). If it discovers new resources, they are registered with RMAPI. Figure 131 on page 207 describes ODM data for the haemaixos SRC subsystem.

```
[root@sp4en0]#>odmget -q subsysname=haemaixos.sp4en0 SRCsubsys

SRCsubsys:
  subsysname = "haemaixos.sp4en0"
  synonym = ""
  cmdargs = "-t SP -n sp4en0"
  path = "/usr/sbin/rsct/bin/haemRM/harmad"
  uid = 0
  auditid = 0
  stdin = "/dev/null"
  stdout = "/dev/null"
  stderr = "/dev/null"
  action = 1
  multi = 0
  contact = 3
  svrkey = 0
  svrmttype = 0
  priority = 20
  signorm = 0
  sigforce = 0
  display = 1
  waittime = 20
  grpname = "haem"
```

Figure 131. haemaixos SRC Subsystem

### 6.4.3 Event Management Enhancements to Support HACMP/ES

The configuration mechanism for the Event Management Configuration Database in recent releases is heavily dependent on the SDR.

Prior to PSSP 3.1, HACMP/ES uses the same Event Management daemon in PSSP. In other words, the HACMP domain and the SP domain shared Event Management. HACMP/ES V4.3 can run also on a standalone RS/6000. Therefore, HACMP/ES V4.3 needs its own Event Management daemon. The emsvcs subsystem is designed for this purpose to work only in an HACMP domain. The Event Manager daemon is started either by the program haemd\_SP or haemd\_HACMP. These programs pass the following arguments:

- Domain type (SP or HACMP)
- Node number
- Domain name

Additionally, the haemd\_SP program passes the following arguments:

- Control Workstation name
- CDB Version
- IP address of system partition (only on CWS)
- Port number for remote connections (only on CWS)

In the SP environment, the `haemctrl` command is used to configure the Event Management subsystem for operation. The `haemctrl` command is modified to configure the haem SRC subsystem with the `haemd_SP` command to be started by SRC rather than `haemd`. After `haemd_SP` obtains the necessary information to be passed to the daemon, it starts the `haemd` daemon.

The Event Management control script is contained in the executable file `/usr/sbin/rsct/haemctrl`. This script is invoked by the `syspar_ctrl` script. The configuration of the Event Management in an SP domain consists of the following steps:

- Add an Event Management Daemon communications port number into the `/etc/services` file.
- Add the Event Management startup program to the SRC using the `mkssys` command. In PSSP 3.1, the startup program for the `haemd` daemon is the `haemd_SP` program. This program is specified by the `mkssys` command as to be started.
- Add the aixos resource monitor daemon (`harmad`) to the SRC using the `mkssys` command.
- Add an entry to the `/etc/inittab` file so that the EM daemon and aixos resource monitor will be started during boot.
- Create the Event Management AIX group `haermm`.
- Create Event Management `/var/ha/soc` and `/var/ha/lck` directories.
- If it is the CWS, the `haemloadcfg` program is executed to load the default configuration data into the SDR.
- Compile the data in the SDR and create the binary EM CDB by executing the `haemcfg` command.

Figure 132 on page 209 illustrates the ODM data for the haem SRC subsystem.

```
[root@sp4en0]#>odmget -q subsysname=haem.sp4en0 SRCsubsys

SRCsubsys:
  subsysname = "haem.sp4en0"
  synonym = ""
  cmdargs = "192.168.4.130"
  path = "/usr/sbin/rsct/bin/haemd_SP"
  uid = 0
  auditid = 0
  stdin = "/dev/null"
  stdout = "/dev/null"
  stderr = "/dev/null"
  action = 1
  multi = 0
  contact = 3
  svrkey = 0
  svrmttype = 0
  priority = 20
  signorm = 0
  sigforce = 0
  display = 1
  waittime = 120
  grpname = "haem"
```

Figure 132. haem SRC Subsystem

In the HACMP/ES environment, the `emsvcsctrl` command is used to configure Event Management. The `emsvcsctrl` command is executed by HACMP scripts during the configuration.

The configuration of the Event Management in an HACMP domain consists of the following steps:

- Add the `emsvcs` subsystem to the SRC by using the `mkssys` command. The startup program for this daemon is `haemd_HACMP` program.
- Add the aixos resource monitor for the HACMP domain `emaixos` to the SRC.
- Create the EM AIX group `haemrm`.
- Create the EM `/var/ha/soc` and `/var/ha/lck` directories.
- Copy the precompiled EMCDB for the HACMP domain from the `/usr/sbin/rsct/install/config` directory to the `/etc/ha/cfg` directory.

Figure 133 on page 210 illustrates the ODM data for `emsvcs` SRC subsystem.

```

sp4n13:/ >odmget -q subsysname=emsvcs SRCsubsys

SRCsubsys:
  subsysname = "emsvcs"
  synonym = ""
  cmdargs = ""
  path = "/usr/sbin/rsct/bin/haemd_HACMP"
  uid = 0
  auditid = 0
  stdin = "/dev/null"
  stdout = "/dev/null"
  stderr = "/dev/null"
  action = 1
  multi = 0
  contact = 3
  svrkey = 0
  svrmtpe = 0
  priority = 20
  signorm = 0
  sigforce = 0
  display = 1
  waittime = 120
  grpname = "emsvcs"

```

Figure 133. emsvcs SRC Subsystem

The cross-relationship of RSCT subsystems in the SP domain and the HACMP domain is described in Table 18.

Table 18. Dual Daemons in SP and HACMP Domain

SRC Subsystems	SP Domain	HACMP Domain
Topology Services	hats	topsvcs
Group Services	hags	grpsvcs
GS Switch	hagsglsm	grpglsm
Event Management	haem	emsvcs
Aixos Resource Monitor	haemaixos	emaixos

#### 6.4.4 Event Management Configuration Database

The Event Management Configuration Database (EMCDB) contains the definitions of all resource monitors and the resource variables. The EMCDB is a binary file that is created by the haemcfg utility from Event Management SDR classes. The SDR classes are loaded and modified using the haemloadcfg and loadsdr commands.

In an SP environment, the haemcfg command builds the EMCDB for a system partition. The haemcfg command follows this procedure while creating database:



- It compiles the Event Management objects stored in the SDR.
- It places the compiled information into a binary EMCDB file in a staging directory as `/spdata/sys1/ha/cfg/em.<syspar_name>.cdb`.
- It updates the `haem_cdb_version` attribute in the SDR Syspar class for the system partition with the current EMCDB version string. The EMCDB version string contains a timestamp and a sequence number.

You should stop and restart all of the system partition's Event Manager daemons to activate the new EMCDB. When the EM daemon restarts, it copies the EMCDB from the staging directory to the runtime directory. The name of the runtime EMCDB is `/etc/ha/cfg/em.<syspar_name>.cdb`.

HACMP/ES V4.3 requires only a fixed set of resource monitors that the customer cannot modify. Assuming no other resource monitors need to be added, then for HACMP/ES V4.3, an EMCDB is already created in the `rsct.basic.hacmp` fileset. The following resource monitors are defined in this EMCDB:

- IBM.PSSP.harmpd
- aixos
- Membership

During the configuration of HACMP/ES V4.3, the EMCDB is copied from `/usr/sbin/rsct/install/config/em.HACMP.cdb` to `/etc/ha/cfg/em.<domain_name>.cdb`, where `domain_name` is the cluster name of the HACMP cluster. When the EM daemon starts in the HACMP domain, it looks only for a local copy of the EMCDB and does not attempt to copy an EMCDB from the staging area.

#### 6.4.5 EMAPI and RMAPI Changes

The Event Management Application Programming Interface (EMAPI) is a shared library. An EM client uses EMAPI to obtain the services of the Event Management subsystem. This shared library is provided in two versions. One of them is for non-thread safe programs, and the other is 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/sbin/rsct/lib/libha_em.a` and `/usr/sbin/rsct/lib/libha_em_r.a`, respectively.

The Event Management Application Programming Interface (EMAPI) is re-versioned. In this new version, the start session interface has another argument added, which is the domain type. If a version 1 interface is used, then the domain type is assumed to be SP. The EMAPI is changed to support the HACMP domain.

The Event Management subsystem operates in a domain. A *domain* is a set of RS/6000 machines upon which the Event Management subsystem executes and provides its services. On the RS/6000 SP, a domain is a system partition. On an HACMP/ES cluster, a domain is the entire cluster.

Note that a machine may be in more than one RSCT domain. If an SP node is also a node in an HACMP/ES cluster, then the node is a member of both the SP domain and the HACMP domain. The control workstation is a member of each system partition and, therefore, a member of each RSCT domain. When a machine is a member of more than one domain, there is an executing copy of each RSCT component per domain.

Since a node may be in both an SP and an HACMP/ES cluster, a domain type must be specified. If the domain is an SP system partition, the domain name is the system partition name. If the domain is an HACMP/ES cluster, the domain name is the cluster name.

An EM client may establish multiple sessions. Multiple sessions can also be established with a single domain.

The `ha_em_start_session()` subroutine starts an EM client session with the Event Management subsystem. The session validates that the EM client is permitted to use Event Management services, and then provides a communication path to the specified Event Management subsystem.

The `em_domain_type` argument is added to this subroutine for supporting multiple domains. The `em_domain_type` indicates the type of the domain in which a session is to be established.

The `part_name` argument in previous releases of PSSP is replaced by the `em_domain_name` argument. The `em_domain_name` argument indicates the domain in which a session is to be established.

The Resource Monitor Application Programming Interface (RMAPI) is used by Resource Monitors to provide resource variables to the Event Management daemon. The RMAPI is designed also to provide data to Performance Monitoring subsystem.

Resource Monitor Application Programming Interface is restructured with PSSP 3.1.

RMAPI uses the private haem shared memory segment to pass Counter/Quantity values to the EM daemon. Prior to PSSP 3.1, resource variable data of type counter and quantity is placed in System Performance Measurement Interface (SPMI) shared memory. The SPMI interface is continues to be used by the RMAPI to support PTPE.

RMAPI supports multiple copies of a resource monitor to execute within the same node/domain.

RMAPI looks for two environment variables to determine if it is executing on an SP or HACMP domain:

- HA\_DOMAIN\_TYPE specifies the type of domain the RMAPI executing in. Valid parameters are SP and HACMP.
- HA\_DOMAIN\_NAME specifies the actual domain name to be used by the RMAPI. On an SP, this is a system partition name the RMAPI executing in. On an HACMP cluster, it is the HACMP cluster name.

These environment variables can be set by SRC. If the EM daemon starts the resource monitor, then the EM daemon sets the environment variables. If the resource monitor is not started by the EM daemon, then whatever mechanism is used to start the resource monitor has to set the environment variables (if it is necessary). The following rules apply to the setting of the domain environment variables:

- If the HA\_DOMAIN\_TYPE variable is not set, the RMAPI assumes it is running on an SP, and looks for the system partition name.
- If the HA\_DOMAIN\_NAME variable is set, the HA\_DOMAIN\_TYPE also has to be set. This is required for the RMAPI to perform a validation check on the domain name.
- The HA\_DOMAIN\_TYPE variable can be set without setting the domain name variable. In this case the RMAPI determines the name based on the value of the type variable.

#### **6.4.6 New Command - haemqvar**

Resource data is gathered and sent to the Event Management subsystem by resource monitors.

There are two ways to determine what resource data is being collected in the system:

- You can use the new command `haemqvar`.
- On an RS/6000 SP, you can use the SP Perspectives GUI.

Perspectives is not available in an RS/6000 HACMP cluster. In this case the `haemqvar` command queries the EM subsystem for information about resource variables. By default, the `haemqvar` command produces a listing of all defined resource variables in the current SP domain. The current SP domain is the current SP system partition name as defined by the `SP_NAME` environment variable. If `SP_NAME` variable is not set, the `/etc/SDR_dest_info` file is referenced. You can use the `-H` flag to query variables in an HACMP domain. Figure 134 on page 214 is an example of an `haemqvar` usage statement.

```

haemqvar [-S domain | -H domain] [ -c | -d | -i ] [ -f file ] [ -h ]
[ class var rsrcID [ ... ] ]
-S      Get definitions for the specified SP domain
-H      Get definitions for the specified HACMP domain
-c      Query current resource variable values
-d      Query definitions, but output short form
-i      Query instances of resource variable values
-f      File containing lines of class var rsrcID
-h      Only display this usage statement
class  Name of resource variable class or quoted null string
var    Name of resource variable or quoted null string
rsrcID Resource ID or "*"

```

Figure 134. `haemqvar` Command

The following information is reported for each resource variable definition:

- Variable name
- Value type
- Data type
- SBS Format (if data type is Structured Byte String)
- Initial value
- Class
- Locator
- Variable description
- Resource ID and its description
- Default expression (if defined) and its description

Since the output of information produced can be quite large, the output of this command should be redirected to a file:

```
# haemqvar > vardefs.out
```

This command can also take arguments requesting information about a particular resource variable. The following example in Figure 135 on page 215 illustrates information about the IBM.PSSP.Response.Host.state resource variable.

```
[root@sp4en0]~/usr/sbin/rsct/bin>haemqvar "" "IBM.PSSP.Response.Host.state" ""
Variable Name:  IBM.PSSP.Response.Host.state
Value Type:    State
Data Type:     long
Initial Value:  0
Class:         IBM.PSSP.Response
Locator:
Variable Description:
    Indicates if the node has connectivity over the en0 adapter.

    IBM.PSSP.Response.Host.state indicates if a node has connectivity
    over the en0 LAN adapter, as determined by the High Availability Topology
    Services subsystem and reported by the High Availability Group Services
    subsystem. A value of 1 indicates connectivity; 0 indicates no connectivity.

    This variable is supplied by the "Response" resource monitor.

    The resource variable's resource ID specifies the number of the node.
    To register an event that indicates node 5 has lost connectivity over its
    en0 adapter, the variable, resource ID and expression would be:

        Resource Variable: IBM.PSSP.Response.Host.state
        Resource ID:      NodeNum=5
        Expression:       X == 0

    Resource ID wildcarding:

    The resource ID element may be wildcarded.

    Related Resource Variables:

        IBM.PSSP.Response.Switch.state
        IBM.PSSP.Membership.LANAdapter.state
        IBM.PSSP.Membership.Node.state

Resource ID:    NodeNum=int
                The number of the node.
```

Figure 135. haemqvar Output

For more information, refer to the *PSSP Command and Technical Reference Volume 1*, SA22-7351.

#### **6.4.7 Event Registration Acknowledgment**

EM clients need event registration acknowledgment to verify that the event registration is successfully completed. A new acknowledgment response is added to the `ha_em_send_command()` subroutine.

The `HA_EM_SCMD_RACK` subcommand returns a registration acknowledgment once the Event Management subsystem has validated the registration request. The specification of the `HA_EM_SCMD_RACK` subcommand on the registration request causes successful registrations to be reported through the registration error responses too. For more information, refer to the *RSCT Event Management Programming Guide and Reference*, SA22-7354.

#### **6.4.8 New SDR Classes and Attributes**

Perspectives is the PSSP application that most frequently uses the Event Management EMAPI interface. The program behind the Graphical User Interface (GUI) registers for events and receives notification about occurrence of registered events. The Perspectives environment also enables the user to monitor resource variables and query them.

Several terms in the SDR are changed to create consistency between Event Management and the Perspectives GUI.

The Event Management subsystem requires information that defines the resource variables and describes how to obtain them. This information is stored in the SDR.

For more information about new SDR classes, refer to Appendix A, “Changes to the SDR” on page 319.

#### **6.4.9 What is New in EM Security?**

Prior to PSSP 3.1, Resource Monitors have to execute as root. This restriction is due to a requirement that Performance Toolbox (PTX) Dynamic Data Suppliers must be root. One of the Dynamic Data Suppliers is RMAPI. In PSSP 3.1, RMAPI now only uses the SPMI when transferring data to PTPE, thus this restriction is removed. If the Resource Monitor provides data to PTPE, it has to be executed as root.

The calling processes of RMAPI subroutines must have a real or effective group id of haermm, or must have the haermm AIX group id in its supplemental group list. If the process is not in the haermm group list, its effective user ID must be root.

If the calling process is instance 0 of the resource monitor, and the monitor is configured to supply Counter or Quantity variables to the Performance Monitor, it must have an effective user ID of root.





---

## Chapter 7. Recoverable/Virtual Shared Disk 3.1

This chapter first introduces IBM Virtual Shared Disk (VSD) and IBM Recoverable Virtual Shared Disk (RVSD) concepts, then discusses changes made to R/VSD in PSSP 3.1. Finally, migration and coexistence are reviewed.

---

### 7.1 R/VSD Concepts

We briefly review the concepts of VSD and RVSD here in order to provide some background information for readers not familiar with these concepts.

#### 7.1.1 VSD Overview

Virtual Shared Disk (VSD), is the software that enables nodes in the RS/6000 SP to share disks with the other nodes in the same system partition.

A *Virtual Shared Disk* is a logical volume that can be accessed not only from the node it belongs to, but also from any other node in the system partition.

A *VSD server* is a node that owns a number of VSDs. It reads and/or writes data to VSDs as requested by client nodes, and transfers data back, usually via SP Switch.

A *VSD client* node is a node that requests access to VSDs. It should be noted that a node can be both a VSD server node and a client node at the same time.

#### 7.1.2 RVSD Overview

If a VSD server node fails, access to data on all VSDs that it owns is lost. In order to avoid this situation, we implement Recoverable Virtual Shared Disk (RVSD) and twin-tailed or loop cabling between nodes.

The RVSD concept is to allow not only one node (the VSD server primary node) to have access to a set of VSDs, but also a second node (the VSD server secondary node), in case one of the following fails:

- VSD server primary node
- Switch adapter
- Disk adapter
- Disk or network cable

RVSD provides protection against node failure by subscribing to Group Services. When a node fails, RVSD is informed by Group Services.

If the failed node is the VSD server primary node, RVSD will have the VSD server secondary node take over the disk subsystems from the primary node and become the server for those VSDs while the primary node is unavailable.

Twin-tailed or loop cabling of the disk subsystem between nodes is needed in order to provide an alternate path to the disk subsystem from the VSD server secondary node.

Thus, with RVSD, the disk subsystem becomes highly available since you can have continuous access to the VSDs even when the VSD server primary node is down.

The amount of time required to failover to a secondary server depends on the number of volume groups that must be varied online, the number of virtual shared disks that make up the volume group, and whether the volume groups need to be re-imported due to configuration changes that have occurred on the primary server.

---

## 7.2 R/VSD 3.1 Enhancements

The most significant enhancement made in R/VSD 3.1 is the ability to add/delete VSD nodes and to add/delete VSD/HSD devices without the need to stop and restart the VSD subsystem.

This eliminates the need to stop and restart applications running on the VSD subsystem, such as Oracle Parallel Server and GPFS, and thus allows these to be continuously available to end users.

### 7.2.1 Packaging Changes

Changes have been made to the naming convention of VSD, VSD perspective and RVSD filesets.

#### 7.2.1.1 VSD Packaging Changes

VSD filesets were renamed to make them easier to recognize. Instead of `ssp.csd.xxx`, they are now `vsd.xxx`.

Table 19. Changes to VSD Fileset Names

Before PSSP 3.1	PSSP 3.1	Description
ssp.csd.cmi	vsd.cmi	VSD SMIT panels
ssp.csd.vsd	vsd.vsdd	VSD device driver
ssp.csd.hsd	vsd.hsd	VSD hash shared disk
ssp.csd.sysctl	vsd.sysctl	VSD sysctl commands

### 7.2.1.2 VSD Perspective Packaging Changes

VSD perspective filesets were renamed to make them consistent with the convention used for VSD filesets. Instead of ssp.csd.xxx, they are now ssp.vsdgui.

Table 20. Changes to VSD Perspective Fileset Names

Before PSSP 3.1	PSSP 3.1	Description
ssp.csd.gui	ssp.vsdgui	VSD perspective
ssp.csd.loc.ma_RP.gui	ssp.vsdgui.loc.ma_RP	VSD perspective locale information
ssp.csd.msg.ma_RP.gui	ssp.vsdgui.msg.ma_RP	VSD perspective messages

### 7.2.1.3 RVSD Packaging Changes

RVSD filesets were renamed to reflect the fact that they are now a part of the VSD installp package, not a separate LPP.

Table 21. Changes to RVSD Fileset Names

Before PSSP 3.1	PSSP 3.1	Description
rscd.docs	---removed---	(included in ssp.docs now)
rscd.rvsd	vsd.rvsd.rvsdd	RVSD daemon
rscd.hahc	vsd.rvsd.hc	RVSD connection manager
rscd.vsd	vsd.rvsd.scripts	RVSD recovery scripts

## 7.2.2 Dynamic Node and Device Changes

The objective here is to allow the addition/deletion of VSD nodes and the addition/deletion of VSD/HSD devices on a running system.

Prior to PSSP 3.1, these changes required the VSD subsystem to be brought down, that is, stop all VSDs and unconfigure from all VSD nodes, then reconfigure and start up again. This was quite disruptive to applications using VSDs/HSDs.

The option `refresh` has been added to the `ha.vsd` command. The command `ha.vsd refresh` propagates changes that have been made to the SDR to the VSD device driver on all VSD nodes. This refresh ensures that all nodes have the same, and latest, configuration level.

**Note**

- The cache option and VSD size can also be changed dynamically. However, you still need to stop and restart the VSD subsystem for other parameters (such as the number of pbuf and buddy buffers or buddy buffer size)
- Since this function is in RVSD, you have to install the RVSD fileset `vsd.rvsd.rvsdd` if you want to use it.
- The refresh option reconciles the SDR with the VSD device driver. This means that if a VSD has been unconfigured but not undefined in the SDR, a refresh will reconfigure the VSD.

### 7.2.3 Separate File System for VSD Configuration and Log Files

To alleviate the space utilization problem in the `/var` file system and to make problem determination easier, a separate 8MB file system, `/var/adm/csd`, is now created in `rootvg` to contain various VSD configuration files read from the SDR when VSD is installed.

The log files created by VSD, HSD and RVSD also reside in this directory.

In case there is not enough space in `rootvg` to create a separate file system, the directory `/var/adm/csd` is created in the `/var` file system, as in previous releases.

---

## 7.3 Migration and Coexistence Considerations

Each of the currently supported levels of R/VSD (1.2, 2.1, 2.1.1 and 3.1) can *interoperate* with each of the supported levels of PSSP (2.2, 2.3, 2.4 and 3.1).

Interoperate means that nodes in the same partition, but at different PSSP levels, can access each other's VSD at the lowest functional level.

Table 22. Which R/VSD Level is Supported in Which PSSP Level

	PSSP 2.2	PSSP 2.3	PSSP 2.4	PSSP 3.1
R/VSD 1.2	Y	Y	Y	Y
R/VSD 2.1	N	Y	Y	Y
R/VSD 2.1.1	N	N	Y	Y
R/VSD 3.1	N	N	N	Y

You will be able to use the new R/VSD 3.1 functions only when the CWS and all nodes that will be using R/VSD are migrated to AIX 4.3.2, PSSP 3.1 and R/VSD 3.1.

In the following section, we discuss the new command that allows us to choose which R/VSD functional level we would like to run.

### 7.3.1 The `rvsdrestrict` Command

Prior to PSSP 3.1, if we have mixed levels of R/VSD in a system partition, R/VSD will be set to operate with the functionality of the lowest PSSP level in the system partition, regardless of whether that lowest level PSSP node participated in R/VSD or not.

With PSSP 3.1, we have the capability of setting the functional level of R/VSD that we want.

For example, consider these three nodes in a system partition:

- Node 1 : PSSP 2.2, no R/VSD
- Node 2 : PSSP 2.3, R/VSD 2.1
- Node 3 : PSSP 2.4, R/VSD 2.1

Prior to PSSP 3.1, node 1 has the lowest level of PSSP in this partition, so R/VSD in this partition would operate at the functional level supported by PSSP 2.2 (that is, R/VSD 1.2).

In this case, since node 1 does not have R/VSD, the group should operate at R/VSD 2.1 level, the lowest R/VSD level in this partition. However, R/VSD assumes that we can install R/VSD on node 1 at any time and activate it. Thus, to be safe, R/VSD always operates at the lowest PSSP level found in the system partition.

In PSSP 3.1, this limitation has been changed. With the `rvsdrestrict` command, you now have the flexibility to tell R/VSD which functional level it should operate. When R/VSD is initialized, only the nodes that have "more or the same" functionality will start; the nodes that are backleveled will not.

For example, in the preceding case, suppose that we have migrated the CWS to R/VSD 3.1:

- If we issue the `rvsdrestrict -s RVSD2.1` command, node 2 and node 3 and the CWS will start and operate at R/VSD 2.1 level.
- If we issue the `rvsdrestrict -s RVSD3.1` command, node 2 and node 3 will not start since they are backleveled.

Note: The quorum concept in R/VSD 3.1 has not been changed. A quorum for the proceeding example is 2 (51% of all nodes having R/VSD and CWS).

To determine the functional level of R/VSD, issue the following command:

```
/usr/lpp/ssp/csd/bin/rvsdrestrict -l
```

The `rvsdrestrict` command makes a change to the *level* attribute in the new `RVSD_Restrict_Level` class in SDR to reflect the current level of R/VSD in that system partition; see Figure 136.

```
sp3en0{ / } /usr/lpp/csd/bin/rvsdrestrict -l
rvsdrestrict level is not set.
sp3en0{ / } SDRGetObjects RVSD_Restrict_Level
domain      level
""          ""

sp3en0{ / } /usr/lpp/csd/bin/rvsdrestrict -s RVSD1.2
rvsdrestrict level is RVSD1.2
sp3en0{ / } SDRGetObjects RVSD_Restrict_Level
domain      level
""          1020000

sp3en0{ / } /usr/lpp/csd/bin/rvsdrestrict -s RVSD3.1
rvsdrestrict level is RVSD3.1
sp3en0{ / } SDRGetObjects RVSD_Restrict_Level
domain      level
""          3010000

sp3en0{ / } /usr/lpp/csd/bin/rvsdrestrict -l
rvsdrestrict level is RVSD3.1
```

Figure 136. The `rvsdrestrict` Command Usage

After you have migrated the CWS, use this command to set the functional level to the lowest R/VSD node in that system partition (otherwise, R/VSD on those older nodes will not start).

Once all nodes are migrated, the `rvsdrestrict` command can then be issued again with 3.1 as the functional level so that all nodes can now operate with R/VSD 3.1 functionalities.

For more information on how to migrate R/VSD, see the publication titled *PSSP:Managing Shared Disks, Version 3 Release 1, SA22-7349*.

### 7.3.2 PTFs for Coexistence

For VSD which provides basic functions (for example, read or write), we do not need any PTFs for coexistence.

However, for RVSD which provides advanced functions (for example, fence or unfence VSD, change VSD primary server, dynamically add/delete VSD nodes/devices and so on.), we need the following PTFs for coexistence with RVSD 3.1:

*Table 23. PTFs for Coexistence with RVSD 3.1*

RVSD Level	PTFs Number
RVSD 1.2	IX80283 (PTF set 20)
RVSD 2.1	IX79109 (PTF set 10)
RVSD 2.1.1	IX79110 (PTF set 2) and IX80414 (PTF set 4)

---

## 7.4 Recommendation

Since all advanced functions added to R/VSD are in RVSD, it is highly recommended that you install and run RVSD subsystem even if you do not care about the VSD primary server takeover function.

You will not be able to use the advanced functions (for example, dynamically add/delete VSD nodes/devices) if you do not install RVSD or RVSD subsystem is not active!





---

## Chapter 8. HACMP/ES 4.3.0

This is the second release of HACMP/ES, version 4.3.0. The previous release of HACMP/ES had two modification levels 4.2.1 and 4.2.2.

For detailed information on planning, installing and administering HACMP/ES refer to *The Enhanced Scalability Installation and Administration Guide*, SC23-4284. There is also a redbook available for HACMP/ES version 4.3, *HACMP Enhanced Scalability Handbook*, SG24-5328 (available on Dec, 1998).

---

### 8.1 Overview

HACMP/ES is an LPP which provides high availability of resources to its using community. Resources are normally shared IP addresses for high availability of access and shared volume groups for high availability of data. These shared resources exist within an HACMP cluster. An HACMP cluster consists of a number of nodes which provide access to these resources in the event of a failure of hardware or software within the cluster. The cluster design strives to eliminate the *Single Point Of Failure*, SPOF, anywhere within this environment. The successful integration of a cluster also relies on the ability to minimize any external influence from services it requires such as power, environment and network infrastructure.

With this release HACMP/ES is able to run in a number of different scenarios. It is no longer dependent on PSSP. It can be configured as:

- A clusters of RS/6000s outside an SP system.
- A mixed cluster of RS/6000 and SP Nodes.
- A cluster of SP nodes in different system partitions or different SPs.

HACMP/ES uses the services provided by RS/6000 Cluster Technology, *RSCT*. For more information regarding these services see Chapter 6, "RS/6000 Cluster Technology" on page 183

#### 8.1.1 Terminology

A number of new terms are introduced to describe the new functionality.

##### 8.1.1.1 Domain

For a full explanation of this term, refer to Chapter 6, "RS/6000 Cluster Technology" on page 183.

### 8.1.1.2 Realm

For a full explanation of this term, refer to Chapter 6, “RS/6000 Cluster Technology” on page 183.

## 8.1.2 Additional Support

There are new additions to supported hardware. The list is not complete but contains information on hardware relevant to using HACMP/ES in an SP environment:

- 7017 S7A SMP Processor Rack
- 2105 VSS Versatile Storage Server B09 & 100
- 2920 PCI T/R Adapter
- 2944 PCI 128 Port Async Adapter
- 2969 PCI Gigabit Ethernet Adapter <sup>1</sup>

and networks:

- ATM Classic IP and LAN Emulation

### 8.1.2.1 ATM

Support is now available for ATM adapters. ATM adapters can be used in two modes Classic IP mode or LAN emulation, LANE, mode. If you are unsure of these modes, in Classic IP mode the interface name is atX, in LANE the interface name appears like the ethernet interface enX, etX or the token ring interface trX, where X is the interface number. HACMP/ES supports both these modes of operation. Configuration for each mode is quite different. For more details refer to *The Enhanced Scalability Installation and Administration Guide*, SC23-4284.

<sup>1</sup> This adapter is not supported in PSSP 3.1

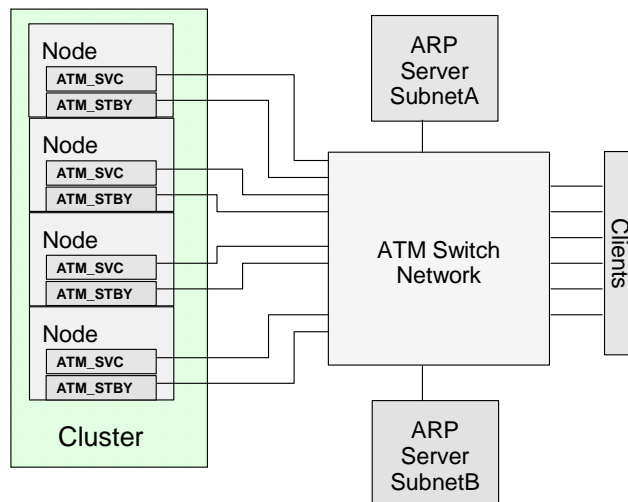


Figure 137. ATM Adapters Using Classic IP

Figure 137 shows ATM adapters in a Classic IP environment. A classic IP implementation relies on ARP servers for resolving an IP address to an ATM circuit connection. There are some rules if using ATM in this mode:

- If an ARP server is an RS/6000, it cannot be a part of the HACMP cluster.
- Adapters must be defined for a private network. ATM does not broadcast.
- Any ARP server must be able to refresh its cache after an HACMP adapter event.
- Only Switched Virtual Circuits (SVCs) can be used.
- MAC address takeover is not supported.

Figure 138 shows ATM adapters in a LAN Emulation (LANE) environment.

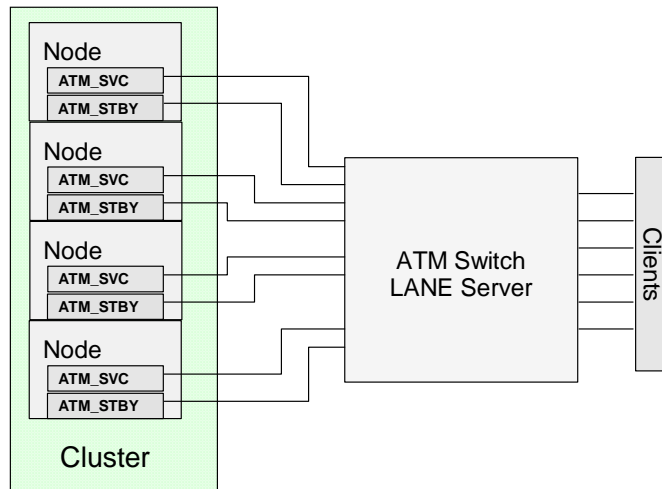


Figure 138. ATM Adapters Configured for LAN Emulation

- Adapters configured for LANE appear as ethernet or TR interfaces (en and tr) and are configured as such.
- Network is public.
- MAC address takeover is not supported.

#### 8.1.2.2 FDDI MAC Address Takeover

MAC address takeover of FDDI adapters is now available. The recommended changes for the address are:

1. Use 4, 5, 6 or 7 as the first digit of the new address.
2. Use the last six digits of the manufacturers default address as the last six digits of the new address.

These recommendations are shown in the following example:

40.00.00.b8.10.89

50.00.00.b8.10.89

60.00.00.b8.10.89

7f.ff.ff.b8.10.89

### 8.1.3 Software Enhancements

The following enhancements are provided by release 4.3 of HACMP/ES:

- Thirty two node support.
- Mixed RS/6000 and SP node clusters.
- Global network support allows subnets to be defined into a global network object.
- Topology Dynamic Reconfiguration Event, DARE, utilizing enhancements to RSCT.
- Security, new utility `cl_setup_kerberos` adds Kerberos principals for all defined interfaces to use `rcmd` and `godm` services.
- All information for the HACMP cluster is moved from the SDR to the Global ODM for PSSP independence.
- Concurrent Access support.
- Tunable heartbeat on individual networks utilizing enhancements to RSCT.

---

## 8.2 HACMP ES Release 2 LPP

This section describes aspects of HACMP ES Release 2 LPP.

### 8.2.1 Packaging

The HACMP/ES product is packaged for installation on any RS/6000 including SP nodes. The installation media contains the RSCT components necessary for running HACMP/ES.

### 8.2.2 Dependencies

- HACMP/ES 4.3.0 requires AIX 4.3.2
- The RSCT package for HACMP/ES consists of the following:
  - `rsct.basic 1.1.0.0`, the two subcomponents required are:
    - `rsct.basic.rte 1.1.0.0`, RSCT basic function, all domains.
    - `rsct.basic.hacmp 1.1.0.0`, RSCT basic function, HACMP domains.
  - `rsct.clients 1.1.0.0`, the two subcomponents required are:
    - `rsct.clients.rte 1.1 0.0`, RSCT client function, all domains.
    - `rsct.clients.hacmp 1.1.0.0`, RSCT client function, HACMP domains.

### 8.2.3 Changes and Restrictions

The following restrictions for earlier releases have been changed.

- Maximum number of nodes per cluster is increased from sixteen to thirty two. This applies to SP nodes, a mixed SP nodes and RS/6000 cluster and RS/6000 only clusters.
- Cluster nodes can now span SP partitions.

The following restrictions are enforced in this release of HACMP/ES:

- Maximum number of nodes in any cluster using concurrent resource groups is eight.
- The maximum number of nodes that can be viewed and managed using the `xhacmpm` command is eight
- The `cl_setup_kerberos` utility only works in an SP cluster.

---

## 8.3 Migration and Coexistence

There are two different migration paths to be considered with this release of HACMP/ES. The first, HACMP/6000 installations moving to the ES version. The second, installations currently running a previous version of HACMP/ES.

### 8.3.1 Migration from HACMP/6000

The HACMP/6000 and HACMP/ES products do not support coexistence in the same cluster. In order for an installation to migrate to HACMP/ES, there must be some system downtime.

Because of the very nature of the HACMP product, downtime is likely to be a limited resource. A precise migration plan should be produced, with fall back procedures at critical time points. If possible, the migration should be tested on a test or non-production cluster. This test should include recovery testing at the defined critical time points. The migration plan should be able to restore the current cluster should the migration fail or overrun these time points.

These are the HACMP steps required to convert a cluster:

- Take a snapshot of the HACMP/6000 cluster. For versions of HACMP/6000 prior to 4.1, which do not have the snapshot facility, a more detailed analysis of the migration plan should be made.
- Take HACMP/6000 down on all cluster nodes.
- Deinstall HACMP/6000 on all cluster nodes.

- Install HACMP/ES on all cluster nodes.
- Restore the cluster configuration from the snapshot.
- Convert the Global ODM by running the clconvert command.
- Synchronize the cluster configuration.
- Start HACMP/ES on the cluster nodes.

### 8.3.2 Migration from HACMP/ES 4.2.1 and 4.2.2

This version, 4.3.0, of HACMP/ES can coexist in an active cluster with nodes running 4.2.1 and 4.2.2. This makes possible a rolling upgrade where a production node can be failed over and its resources moved to another node in the cluster. The failed node is then upgraded before being reintegrated into the cluster.

Again, planning is paramount. If a node has been failed over for the upgrade, then a deliberate failure has occurred and most clusters are designed to eliminate only *Single Points Of Failure*. Before beginning the cluster upgrade, it may be wise to introduce another node into the cluster to take the place of the node that is currently being upgraded.

These are the steps for the HACMP upgrade:

1. Take a snapshot of the cluster configuration.
2. Stop HACMP/ES on the node to be migrated.
3. Install HACMP/ES 4.3.0 on the node.
4. Convert the Global ODM on the node using the clconvert command.
5. Start HACMP/ES 4.3.0 on the node.
6. Repeat steps 2 - 5 for each node in the cluster.

After all nodes have been migrated to HACMP/ES 4.3.0, HACMP informs Group Services that no more single phase joins are allowed for this cluster. This prevents any more pre-4.3.0 HACMP nodes from joining this cluster.

A new field is present in the HACMPcluster ODM class, *lowest version*. This field is set to 0 in the 4.3.0 release and does not exist in the previous releases. Using this allows an orderly switch to the new 4.3.0 functionality.

With all nodes running 4.3.0, a migration protocol is initiated. Each node will update the *lowest version* field in their HACMPcluster ODM class and run the clmigrated script. The clmigrated script refreshes topology services and group services. The migration protocol is setup as a default yes vote protocol.

If a node fails during the migration protocol, group services will vote for the node. If the failure is prior to the update of the HACMPcluster ODM class, that node will not be allowed to join the running cluster until manual intervention updates the ODM on the node. This requires a DARE from one of the nodes which remained up.

#### **8.3.2.1 Introduction of RS/6000**

If any RS/6000 is to be introduced into an HACMP/ES cluster running on SP nodes, all the nodes should be fully migrated to HACMP/ES 4.3.0 before this is attempted.

---

## **8.4 New Functionality**

This section describes the new functionality introduced in HACMP/ES 4.3.

### **8.4.1 Global Network Support**

Global network support is a new concept for HACMP clusters. In previous versions, a network existed at the subnet level. A global network contains more than one subnet. Heartbeats takes place across this global network. For the heartbeat to take place, TCP/IP routes must also exist.

Consider the example of a multi-frame SP that has a CWS with separate Ethernet connections to node1 in each frame. Each node 1 has a network which connects to every node in that frame. This is shown in Figure 139. Routes exist so that any node on the 10.1.3.0 network can contact any node on the 10.1.4.0 network via node1 in each frame and the CWS.



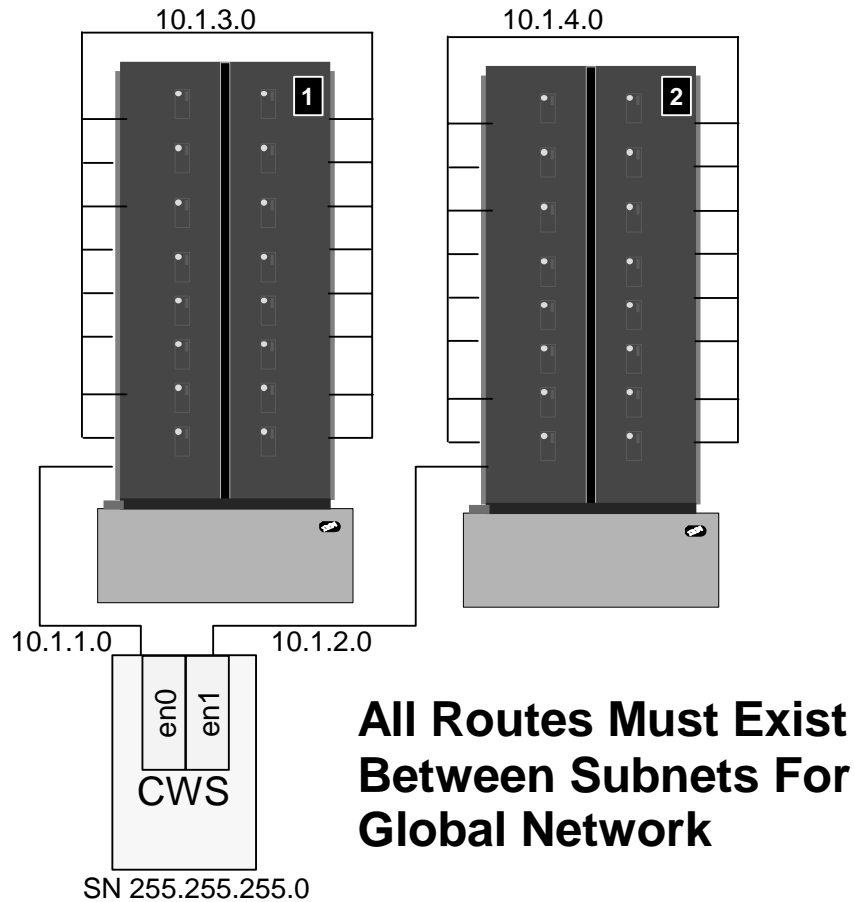


Figure 139. Example of Subnets on an SP Private Ethernet

In this release of HACMP, these two subnetworks can be defined as one global network by using the `claddnetwork` command or by using the SMIT panel shown in Figure 140.

The use of this new definition eliminates the situation which could previously have occurred in the system shown in Figure 139. A cluster partition could occur during a frame startup or switch fault where switch communications goes down for a time. The cluster would partition as the ethernetets were effectively 2 separate ethernet HACMP networks. By combining the ethernetets into one global network, when the switch comes back up the cluster is not partitioned.

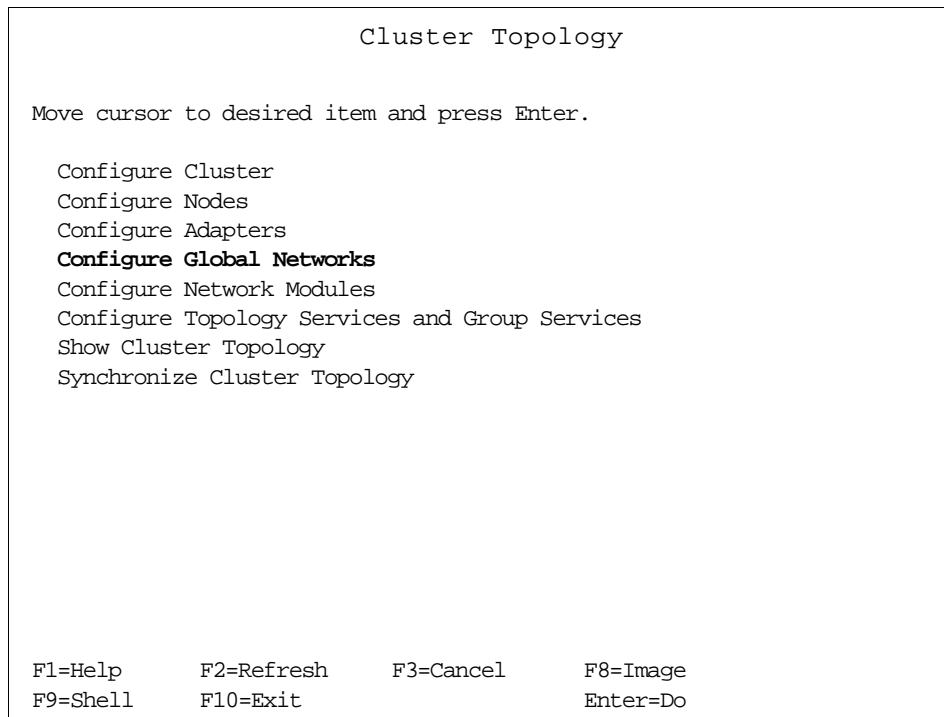


Figure 140. SMIT Menu for Global Network Addition

Using this new facility, networks can be brought together as a single global network. If the subnet on frame 1 went down, that would not cause a network down event. Both subnets would have to be down for the network down event to take place.

Consider the configuration in Figure 139 on page 235. The two subnets, 10.1.3.0 defined to HACMP as spether1 and 10.1.4.0 defined to HACMP as spether2, can be combined as a global network and called SPnet using the `claddnetwork` command as follows:

```
claddnetwork -u spether1:SPnet
```

```
claddnetwork -u spether2:SPnet
```

This updates the HACMP ODM class HACMPnetwork as shown in Figure 141 on page 237. To remove the HACMP network spether1 from inclusion in a global network, run the command:

```
claddnetwork -u spether1
```

```

HACMPnetwork:
    name = "spether1"
    attr = "public"
    network_id = 5
    globalname = "SPnet"

HACMPnetwork:
    name = "spether2"
    attr = "public"
    network_id = 4
    globalname = "SPnet"

```

Figure 141. HACMPnetwork ODM with Global Network Defined

## 8.4.2 Dynamic Reconfiguration Event (DARE)

The default configuration data for an HACMP cluster is stored in the system default ODM directory `/etc/objrepos`. The HACMP ODM in this directory is referred to as the Default Configuration Directory, *DCD*. In order for the dynamic reconfiguration of a running cluster to occur, the cluster manager, on initialization, creates a private copy of these HACMP ODM classes. This private copy is called the Active Configuration Directory, *ACD*. All the HACMP daemons, scripts and utilities running on a node reference this private copy. The default directory for this private area is `/usr/sbin/cluster/etc/objrepos/active`. When you configure a cluster, the DCD is modified, not the ACD.

Any changes made to an active cluster modify the DCD. Using SMIT, these changes are synchronized throughout the cluster. Synchronization updates the DCD on each cluster node from the modified DCD on the node initiating the change. The ACD on each node is replaced by the new DCD using a group services protocol to coordinate a cluster-wide transition. The reconfiguration event also refreshes the RSCT daemons and the cluster manager on each node.

For example in a two node cluster consisting of node1 and node2, a third node, node3 is added. The configuration is updated on node1. The changes in the DCD on node1 are copied to the DCDs on node2 and node3 and a DARE is triggered. The DCD on the existing cluster nodes, node1 and node2, is copied to a temporary location, the Staging Configuration Directory, *SCD*. Using the temporary location allows you to start making additional configuration changes, while the DARE is in progress. A second synchronization initiated while a DARE is in progress is not allowed. The

presence of the SCD acts as a lock. The local cluster manager on a node verifies the configuration in the SCD before moving it into the ACD.

### 8.4.3 Security

Both HACMP/6000 and HACMP/ES have support for an enhanced security mode that utilizes Kerberos instead of the traditional TCP/IP .rhosts file. In previous releases. If a cluster had enhanced security-enabled, it was a time-consuming repetitive task to define service principals for all HACMP-defined adapter labels, boot, service and standby to use rcmd and godm services. The new utility command `cl_setup_kerberos` is used to automate the previous repetitive manual actions. The utility is found in the `/usr/es/sbin/cluster/sbin` directory. If you run HACMP/ES on RS/6000 only and use Kerberos authentication, you may wish to create a version for your environment.

#### 8.4.3.1 Configuring Kerberos Automatically, SP Only.

Before running `cl_setup_kerberos`, be aware that it makes a number of assumptions:

1. Kerberos for the SP is configured and functional across the SP Ethernet.
2. HACMP/ES is installed on all cluster nodes.
3. There is a CWS and a functional SDR.

If the assumptions that `cl_setup_kerberos` makes are all correct, the following procedure will update Kerberos authentication to allow godm and rcmd services on all defined interfaces within the cluster:

1. Make sure all .rhost files are deleted.
2. Configure the cluster topology on one node.
3. Run `cl_setup_kerberos` on the configured node to create the new Kerberos service principals and configure all the HACMP adapter labels for Kerberos authentication.
4. Set the cluster security mode to enhanced and synchronize the topology.

The `cl_setup_kerberos` utility performs the following functions:

- Extracts all the HACMP adapter labels from the configured node.
- Prompts for a password for the new principals.
- Adds the new entries to the Kerberos database. For each defined adapter label using the rcmd and godm services.
- Extracts the new service principals from the database and updates the `krb-srvtab` on each cluster node.

- Updates the .klogin file and the realms file on each cluster node.

This automation helps to eliminate SPOFs possibly created by the manual method.

Take care, if node customizing is carried out on a cluster node you should run the utility again. PSSP will create a Kerberos krb-srvtab file that contains only service principals for those interfaces defined in the SDR.

#### 8.4.4 Global ODM

In the previous versions of HACMP/ES configuration data relied on the SDR and therefore the CWS. This release uses HACMP ODM classes stored in the RS/6000 ODM system directory /etc/objrepos. Removing SDR reliance is one of the features that make HACMP/ES able to run in a non SP environment.

The global ODM is an RPC based wrapper around the standard AIX ODM that provides remote ODM operations analogous to the local versions. In order to preserve the expected behavior on a single node and to provide failure resistance, HACMP/ES configuration uses a synchronization step to update all of the ODMs in a cluster. The ODM on each cluster node is updated in a serial sequence.

The global ODM daemon /usr/sbin/cluster/godmd is configured during the HACMP/ES installation as an inetd subserver. The global ODM will function using either standard or enhanced security.

Standard security relies on the use of the TCP/IP access control list file .rhosts being correctly configured in the root directory of a target node. Enhanced security makes use of Kerberos version 4. The Kerberos database, the node krb-srvtab and .klogin files must all be updated to use the godm service. There is a utility `cl_setup_kerberos` which automates the modifications to the Kerberos authentication system. For more information refer to 8.4.3 on page 238.

#### 8.4.5 Concurrent Access

Concurrent access is the ability for up to eight nodes to access shared volume groups and shared logical volumes concurrently. There are a number of restrictions with the use of concurrent shared VGs over non-concurrent VGs.

- Nodes using concurrent VGs only use raw logical volumes in these VGs.
- Concurrent VGs are supported on the following disk systems:

- Any SSA non RAID configurations using 7133 and 7131-405. 7131-405 has limited flexibility in clusters greater than two nodes. SSA will support up to eight concurrently attached nodes. This ability is dependent on the SSA adapter which is used.
- 9333 disk subsystems. Support for up to eight attached nodes.
- 7135-110 and 7135-210 RAIDiant arrays. These arrays support up to four attached nodes.
- A concurrent resource group consists only of application servers and concurrent volume groups.
- The application must manage the locking required for access to data stored in the shared concurrent logical volumes.
- Volume groups created or imported for use in a concurrent environment must have the relevant fields in the SMIT panels set to yes. These fields are:
  - In the SMIT create a volume group:  
Create VG Concurrent Capable? = yes
  - In the SMIT import a volume group:  
Make this VG Concurrent Capable? = yes

The 9333 and SSA subsystems can use disk fencing to prevent data integrity for problems that can occur in partitioned clusters.

These subsystems use fence registers, one per disk, capable of permitting or denying access to any of the attached nodes. This provides a means of preventing uncoordinated access by one or more nodes.

The 9333 and 7133 hardware support a fencing command to update the registers. This command provides a tie-breaking function within the controller for nodes independently attempting to update the same fence register. A compare and swap protocol of the fence command requires that each node provide the current and desired contents of the fence register. If competing nodes attempt to update a register at about the same time, the first succeeds, but the second fails because it has the wrong revised contents.

Part of the concurrent resource manager (CRM) package is a lock daemon. This lock daemon is provided with an API. Concurrent applications can be developed to use this daemon.

## 8.4.6 Supported Networks

Table 24 summarizes the network support available in HACMP/ES 4.3.0.

Table 24. Network Support in HACMP

IP Networks	Non IP Networks
Ethernet	RS232
Token Ring	tmSCSI
SP Switch	
FDDI	
ATM	

MAC Address takeover is supported for Ethernet, Token Ring and FDDI.

## 8.4.7 Tunable Heartbeat

Because of the enhancements to topology services, it is now possible to have different heartbeat parameters for different network types. The tuning parameters are held in the HACMP ODM class, HACMPnim, for each network type. This file is read by the topsvcs script during initialization and the parameters are extracted. If any of the parameters are invalid or the file is not readable, default global parameters are supplied from the HACMP ODM file HACMPtopsvcs.

The HACMPnim file contains a field *hbrate*. This field can be modified using SMIT. The fastpath is `smitty cm_config_networks.chg.select` For any supported network type, the options for the SMIT *Failure Detection Rate* field are:

- Fast
- Normal
- Slow

For both IP and Non-IP networks, these preset values set 1, 2 or 3 seconds respectively. The *Failure Cycle* field is a count of the number of missed heartbeats allowed before action is taken. The default and also the minimum value is 4. The failure cycle is used for heartbeat tuning on systems with busy or slow networks or systems that are resource starved. In these types of systems it is possible for the deadman switch to be invoked on a node because of missed heartbeats.

For an example of an HACMPnim stanza for an Ethernet network, see Figure 142.

```
HACMPnim:
  name = "ether"
  desc = "Ethernet Protocol"
  addrtype = 0
  path = "/usr/sbin/cluster/nims/nim_ether"
  para = ""
  grace = 30
  hbrate = 1000000
  cycle = 4
```

Figure 142. HACMPnim Class Example

---

## 8.5 New Commands

The following are new commands for HACMP/ES 4.3.0.

### 8.5.1 clhandle

The clhandle command is used to obtain the following information from the local ODM when used with the following flags:

- -a, for every node in the cluster display the handle and name.
- -n nodename, display the handle for this node.
- -h handle, display the nodename for this handle.
- -c, display colon separated fields.
- No argument, display this node's handle and node name.

This information is required for a user to define resource identifiers for user-defined events. If an error occurs, a non-zero exit code will be returned. For more information about user-defined events, refer to 3.4, "Event Perspectives" on page 95.

### 8.5.2 cldomain

The cldomain command returns the cluster name that defines the domain for the RSCT infrastructure. If an error occurs, a non-zero exit code will be returned.



### 8.5.3 **clmixver**

Determine if this node is running a different version from the version that did the last topology synchronization. It displays the internal HACMP/ES version number on stdout. The return codes are:

- 1, this node is executing a version greater than the clstrmgr that did the last topology synchronization.
- 0, this node is executing the same version as the clstrmgr that did the last topology synchronization.
- -1, an error occurred.

### 8.5.4 **claddnetwork**

Once the cluster adapter topology has been configured, individual subnets can be combined together to heartbeat in a global network using the `claddnetwork` command. In a global network, when the last adapter goes down or comes up in a subnet, the corresponding network down or up event does not run as long as another adapter is up within the global network. We recommend that you configure the SP Ethernet as a global network in those systems with boot/install servers, as shown in Figure 139 on page 235.



---

## Chapter 9. GPFS 1.2

This chapter first discusses the goals in developing GPFS; then introduces the concept of GPFS and several enhancements made by GPFS 1.2 in the areas of scalability, usability, system management and performance. Finally, migration, coexistence and compatibility is reviewed.

---

### 9.1 Why GPFS?

GPFS was developed in order to solve various problems faced with existing file systems, for example:

1. A single file server does not provide sufficient performance.
2. Access to data on other systems via NFS is not good enough in terms of performance.
3. Existing file systems do not provide sufficient performance.
4. Existing parallel file systems do not provide sufficient availability.

The combination of a very high level of performance, a very high level of availability and standard conformance was not provided by any single currently available file system. GPFS was developed to address all these requirements.

---

### 9.2 GPFS Overview

GPFS is implemented as a standard AIX Virtual File System, which means that most applications using standard AIX VFS calls (such as JFS calls) will run on GPFS without any modification.<sup>1</sup>

GPFS allows parallel applications simultaneous access to the same files, or different files, from any node in the configuration while maintaining a high level of data availability. It offers an extremely highly available file system by utilizing both hardware and software redundancy where appropriate (for example, disk takeover when a node fails, replicated log files, selectable data/metadata replication, and so on).

#### 9.2.1 Implementation Overview

GPFS is designed to provide a highly available file system that can satisfy requirements for a very high performance file system in both serial and parallel applications.

<sup>1</sup> There are certain limitations, for example GPFS does not support memory-mapped files yet.

GPFS achieves a very high level of performance by having not one, but multiple nodes acting in cooperation to provide server functions for a file system. This solves the problem of running out of server capacity which occurs with NFS, for example, since in GPFS we have not one, but multiple servers and we can add more servers or disks when we want to get better performance.

Having multiple servers will not help much unless we can make sure that all nodes are working together and the system workload is spread out across them evenly, in order to not allow any one of them become a bottleneck of the system.

The following are some examples of what GPFS does to spread the workload across nodes:

- GPFS stripes data across disks on multiple nodes
- GPFS allows more disks and nodes to be added later
- GPFS can re-stripe the file system
- GPFS does not assign the same node to be the stripe group manager, if possible

When you create a file system in GPFS, you can specify a *block size* of 16KB, 64KB or 256 KB. Block size is the largest amount of data that is accessed in a single I/O operation. Each block is divided into 32 *subblocks* which is the smallest unit of disk space that can be allocated. For example, using a block size of 256KB, GPFS can read as much as 256KB in a single I/O operation and small files occupy at least 8KB of disk space.

When you create a file system, you also specify a set of disks that will be used to store the data and metadata for this file system. GPFS calls this set of disks a *stripe group*.

When GPFS writes data to the file system, it stripes the data into many blocks, each with the size of block size, then writes each block to each disk in the stripe group of this file system. You can also specify the method GPFS should use for allocating each block to disk, for example roundrobin or random, when you create the file stem.

GPFS achieves the goal of being a highly available file system through its ability to recover from various hardware and software failures that can happen in the system. In most cases, the end users can continue on the operations with only a slight delay or performance degradation.

With the help of RVSD and twin-tailed disk connection, GPFS is able to tolerate various hardware failures such as node failure, switch adapter failure, disk adapter failure and disk cable failure.

There are many ways that GPFS can handle disk failures:

- Use RAID-5 disks
- Implement disk mirroring
- Data and/or metadata replication

When you create the GPFS file system, you can specify whether you would like to replicate the data and/or the metadata for the file system. You can select up to two replicas for data and metadata.

GPFS ensures that a copy of replica will always be available even when there is a hardware failure. (In certain cases, it may be able to survive multiple failures.)

The limitation of replication is that when it is enabled, the maximum file system size and the maximum file size that can be created in the file system are reduced significantly (for example, by a factor of 2 or 4).

Even when we do not use RAID-5 disk, do not implement disk mirroring and data/metadata replication and multiple failures occur, GPFS can still provide access to the file as long as all the required metadata to access the file and its data can still be accessed!

Moreover, the file system integrity in GPFS is always maintained since GPFS replicates the log file of each node to another one. Thus when a node fails, the node that has the other copy of the log file can use that to recover from the failure. This not only allows the file system integrity to be maintained, but also allows the file system operation to continue with no disruption.

### 9.2.2 GPFS Components

GPFS is installed on all nodes that will be part of the GPFS domain. There is some part of GPFS that is installed on the CWS for the maintenance of the SDR. (The CWS cannot be a part of GPFS domain since there is no switch connection from the CWS. Anyway, we can use any GPFS file system from the CWS by mounting it from any node in the GPFS domain for that file system in the same way as we use NFS.)

There is a daemon, *mmfsd*, running on every node that is a part of the GPFS domain. This daemon provides some services such as allocation of disk space to new file, initiation of disk I/O, management of security and disk quotas, file locking, and so on.

However, since these tasks must be carried out in many nodes at the same time, some coordination, synchronization and management need to be performed by certain GPFS daemons.

The following are the additional personalities that a GPFS daemon may assume:<sup>2</sup>

### **9.2.2.1 Configuration Manager**

*Configuration Manager* is a component of GPFS that is responsible for selecting the Stripe Group Manager for each file system. It also determines whether a quorum exists, which in turn determines whether the file system can continue to be used.

When a quorum is lost, GPFS unmounts the file system (thus not allowing it to be used), since there can be cases where problems with the network cause the GPFS domain to be divided into separate groups.

If GPFS does not force the group with no quorum to stop using the file system, multiple groups may try to write to the same metadata and/or file at the same time, compromising the integrity of the file system.

There is one Configuration Manager per system partition. It is the first node to join the group *MmfsGroup* in Group Services. In other words, it is the oldest node in this group.

If the Configuration Manager is down, Group Services will select the next oldest node in *MmfsGroup* to become the Configuration Manager.

### **9.2.2.2 Stripe Group Manager**

Each GPFS file system is comprised of a stripe group. A stripe group is simply a set of disks that belong to this file system.

There is one Stripe Group Manager per file system. The Configuration Manager selects a Stripe Group Manager for each file system. It tries not to overload any node in the system by selecting a different node to act as a Stripe Group Manager for each file system, if possible.

The Stripe Group Manager provides the following services to all nodes using that file system:

- Processes changes to the state or description of the file system
  - Adds/deletes/replaces disks
  - Changes disk availability

<sup>2</sup> A GPFS daemon may assume more than one of these personalities.

- Repairs the file system
- Restripes the file system
- Control disk region allocation

If needed, you can influence the Configuration Manager regarding the selecting of the Stripe Group Manager by creating a file called `cluster.preferences` in the `/var/mmfs/etc` directory and listing the switch hostname of the nodes that you want to act as a Stripe Group Manager, one per line.

The Configuration Manager will select any node in the list that is available at the time the choice is made. There is no relative priority or rank among the nodes.

When a Stripe Group Manager is down, the Configuration manager selects a node from a preference file or any node if you don't use the preference file.

In both cases, it tries to select a node that is not currently a Stripe Group Manager for any other file system.

### **9.2.2.3 Metadata Manager**

There is one Metadata Manager for each open file in the file system. It is responsible for the integrity of the metadata of that file.

Even though each VSD server node can read and/or write the data to the disks directly, the update of metadata of a file is restricted to the node containing the Metadata Manager for that file.

The Metadata Manager is selected to be the first node that had the file opened. It continues to provide metadata services for that file until one of the following events occurs:

- The file is closed everywhere.
- The node fails.
- The node resigns.

When the Metadata Manager is down, the next node that needs the metadata service will become the Metadata manager.

### **9.2.2.4 Token Manager Server**

Tokens are used to coordinate various activities occurring at the same time in the file system across nodes to make sure that the integrity of the file system

is not compromised. They are used in much the same way that locks are used to coordinate the activities on a single node.

There is a Token Manager on every GPFS node.

When a node wants to access data, its Token Manager contacts the Token Manager Server requesting a token. The Token Manager Server determines whether there is any locking conflict among the tokens that have already been granted and the currently requested one.

If there is no conflict, the Token Manager Server can allow the request to proceed by granting a token to that node, so that it can continue with what it wants to do. If there are conflicts, it sends a list called a *copy set* that lists the nodes that have conflict locks.

In order to reduce the workload at the Token Manager Server, it is the responsibility of the requesting Token Manager to negotiate with any node in the list to obtain the token.

There is one Token Manager Server per file system. It is located on the same node as the Stripe Group Manager. It is responsible for granting tokens to the requesting Token Managers.

For the purpose of availability and recoverability, two copies of the token are kept in the system: one in the Token Manager Server (the server copy), and one in the Token Manager (the client copy).

When a node is down, all the tokens it had can be recovered by obtaining the server copy from the Token Manager Server.

When the Token Manager Server is down, the new Token Manager Server can recover all the tokens that the old Token Manager Server had by obtaining the client copy from all Token Managers in the GPFS domain.

---

## 9.3 Hardware and Software Requirements

The following lists the hardware and software required by GPFS 1.2:

### 9.3.1 Hardware Requirements

- RS/6000 SP
- SP Switch
- Sufficient disk capacity to support file systems.



### 9.3.2 Software Requirements

- AIX 4.3.2
- PSSP 3.1 with the following options installed:
  - ssp.basic
  - ssp.css
  - ssp.sysctl
- RSCT 1.1
- VSD 3.1
- RVSD 3.1

#### Note

RVSD is a prerequisite for GPFS. You need RVSD even when you do not care about the high availability of your file system, or do not plan to use any external disk subsystem. This is because GPFS needs some commands in RVSD, for example, `fencevsd` and `unfencevsd`, that are necessary to ensure that the integrity of the GPFS file system will not be compromised.

---

## 9.4 GPFS 1.2 Enhancements

Various enhancements have been made to GPFS 1.2 in terms of scalability, usability, system management and performance.

### 9.4.1 Scalability Enhancements

GPFS 1.2 provides several enhancements that enable it to scale better with the increased workload and/or resources provided.

#### 9.4.1.1 Movement of Token Manager From Kernel

In GPFS 1.1, the Token Manager is a kernel extension that uses kernel heap storage to store its token.

The amount of kernel heap storage needed for the Token Manager to store the client copy depends on how much file system activities that node has.

However, the amount of kernel heap storage needed for Token Manager Server to store the server copy depends on the total activities of that file system. In case of a very active file system, this amount can become very

large and may conflict with other applications running in the system (or even with the system itself!).

To eliminate this potentially serious problem, in GPFS 1.2, the Token Manager function has been moved from the kernel extension to the GPFS daemon and now uses shared segments, instead of the kernel heap storage, to store tokens.

This not only provides additional kernel heap storage for other system functions but also results in quicker recovery in case of token manager server failures.

#### 9.4.1.2 Stripe Group Descriptor Limit Increased

When you create a GPFS file system with the `mmcrfs` command, you can provide a descriptor file with the option `-F`.

This descriptor file can contain:

- A disk descriptor which consists of the disk name, the primary VSD server name and secondary VSD server name for that disk, how the disk is used to store data and/or metadata, and its failure group<sup>3</sup>.
- A VSD descriptor which consists of VSD device name, how the disk is used to store data and/or metadata, and its failure group.

In GPFS 1.1, you can provide at most 512 descriptors in this file, which means that the GPFS file system size is limited to the amount of disk storage the 512 descriptors can provide.

In GPFS 1.2, this limit has been increased to 1024. Thus the maximum file system size that can be created with GPFS 1.2 is roughly two times that of GPFS 1.1.

#### Note

- The maximum file system size and file size in a GPFS file system depend on a number of parameters specified when we create the GPFS file system.

See Chapter 2: Planning for GPFS in *GPFS Installation and Administration Guide*, SA22-7278 for more detail.

- One Terabyte is the maximum supported file system size now.

<sup>3</sup> Failure Group is a group of disks that have a common point of failure, for example all internal disks in a node belong to the same failure group.

## 9.4.2 Usability and System Management Enhancements

GPFS 1.2 provides several enhancements aimed at improving ease of use and ease of management; for example, a reduction in the need to stop and restart the GPFS subsystem.

### 9.4.2.1 Extensible inodes

In GPFS 1.1, the maximum number of files that can be created in a GPFS file system (in other word, the number of inodes) is specified when you create the file system with the option `-N` of the `mmcrfs` command.

This number cannot be changed later. If you would like to change it, you have to create another GPFS file system with the new maximum number of files that you want, then copy the files from the old file system to the new one.

This greatly reduces the flexibility provided by the `mmaddisk` command that allows us to add more disks to the file system as needed, since even though we can add more disk to the file system later, we cannot add more files because the maximum number of files cannot be changed!

In GPFS 1.2, the number of inodes can now be changed by specifying the `-F` option in the `mmchfs` command. If you changed it and later found that it is not enough, you can use the same command again to increase it.

The inode file will expand on demand, from the initial minimum value specified in the file system creation time, up to the new maximum value specified in the `mmcrfs` command.

### 9.4.2.2 Dynamic GPFS Buffer

When you configure GPFS, you can specify the amount of memory that can be used by GPFS. This area is pinned in the memory. It is used to increase performance through read-ahead and write-behind operations, as well as for reuse of cached data.

With the `mmconfig` command, you can specify:

- `pagepool`

The area used to store user data. It can range from 4MB to 512MB per node, with a default of 20MB.

- `malloysize`

The area used to store metadata and GPFS control structures. It can range from 2MB to 128MB per node, with a default of 4MB.

The sum of these two areas cannot exceed 80% of real memory.

In GPFS 1.1, if you need to change either of these areas, you need to stop and restart GPFS.

In GPFS 1.2, the pagepool area can now be changed dynamically with the command:

```
mmchconfig pagepool=<size> -i
```

The -i option tells GPFS to make the change take effect immediately.

Note: You still need to stop and restart GPFS to change the malloc size.

#### **9.4.2.3 Multiple GPFS Configurations for Testing/Migration**

In GPFS 1.1, we can have only one GPFS domain in a system partition.

The consequence of this is that, if you need to test a new GPFS code, what you have to do is to set up a separate system partition for this purpose.

Though setting up a system partition is not a difficult thing to do, system partition places certain restrictions (such as all nodes that use the same switch chip must belong to the same system partition) such that you may not be able to comply. It also requires you to power off the nodes that will change the partition, which may not be the thing you can do.

In GPFS 1.2, we can have multiple GPFS domains coexist in the same system partition. Each domain is also now called a *GPFS configuration*.

This allows you to test a new level of GPFS code in a subset of nodes without the need to set up a separate system partition and without interfering with your production system.

For example, in an SP system now you can have ten nodes running GPFS 1.1 as your production system and six nodes running GPFS 1.2 for testing. Once you are satisfied with the testing, you can upgrade your ten production nodes to GPFS 1.2.

See more information in “GPFS Configuration For Migration” on page 256.

#### **9.4.2.4 Add/Delete VSD Nodes Dynamically**

In GPFS 1.1, although you can add more nodes and disks to the running GPFS subsystem, those nodes and disks must already be defined to the VSD subsystem.

If they are new, you need to stop and restart the VSD subsystem to make it know them. Since GPFS runs on the VSD subsystem, you need to stop and restart GPFS too.

With VSD 3.1, it is now possible to add/delete VSD nodes and VSD devices without stopping and restarting the VSD subsystem.

Since GPFS 1.2 makes use of VSD 3.1, with GPFS 1.2, you can always add more nodes and disks to the running GPFS subsystem without the need to stop and restart GPFS!

### 9.4.3 Performance Enhancements

Several performance enhancements have been added to GPFS 1.2.

#### 9.4.3.1 MPI-IO Support

Prior to Parallel Environment 2.4, MPI does not provide support for parallel file I/O in MPI calls, and this was considered a major drawback since it makes portable code involving parallel file I/O impractical.

Parallel Environment 2.4 now supports a subset of MPI-IO, as defined in the MPI-2 standard.

GPFS 1.2 is highly recommended as a parallel file system for MPI-IO programs.

#### 9.4.3.2 Pre-allocate Support

GPFS 1.2 now supports an API which provides the ability to preallocate space for a file.

This allows you to preallocate some amount of space for a file that has already been opened, prior to writing data to the file.

The preallocation of disk space for a file provides an efficient method for allocating storage without having to write any data. This can result in faster I/O compared to a file which gains disk space incrementally as it grows.

Existing data in the file is not modified. Reading any of the preallocated blocks returns zeroes.

#### Note

You need to compile any program that uses this function from the `libgpfs.a` library with the `-lgpfs` flag.

---

## 9.5 Migration Considerations

Due to the changes in the token manager function in GPFS 1.2, it is required that all nodes that use a given file system are not only at the same level of GPFS, but are also upgraded to that level at the same time.

Some new functions in GPFS 1.2, such as extensible inodes and preallocation, create data structures which are not recognized by GPFS 1.1.

In order to ease the migration, GPFS 1.2 does not allow you to exploit these new functions until you have explicitly authorized these changes via the `mmchfs -v` command.

### 9.5.1 GPFS Configuration For Migration

A *GPFS configuration* is a group of nodes that all run the same level of GPFS code and operate on the same file system.

With GPFS 1.2, you can define more than one GPFS configuration in the same SP complex. This allows you to create a separate configuration for testing without interfering with the production.

Each node may belong to only one GPFS configuration. Nodes may be moved from configuration A to configuration B by `mmdelnode` from configuration A and `mmaddnode` to configuration B.

A GPFS file system may only be accessed from one GPFS configuration. To show which GPFS configuration a given GPFS file system belongs to, use the `mmfsfs -C` command.

### 9.5.2 Migration Approach

If your system has at least five nodes, you can use multiple GPFS configurations and do a staged migration. Otherwise, it is recommended that you use a full migration.

This limitation is due to the fact that GPFS requires quorum in order to operate. For example, if you have 5 nodes, the quorum is 3, so you can move two of them to test configuration; the three remaining nodes can still run. But if you have 4 nodes, the quorum is also 3, so you can only move one node to test configuration. However, having a one-node GPFS configuration does not make much sense!

### 9.5.3 A Full Migration

In this method, you stop GPFS on all nodes, install GPFS 1.2 to all nodes in the system at the same time, reboot and then spend some time making sure that everything works fine.

Once you decide that you will permanently accept GPFS 1.2, issue the command: `mmchfs <file system> -V`.

### 9.5.4 A Staged Migration

In a staged migration, you first choose a subset of nodes that will be a *test configuration*. The CWS will also always be in a test configuration.

You then stop GPFS on all test configuration nodes, use `mmdelnode` to delete them from the current configuration, install GPFS 1.2, and reboot.

You then use `mmconfig` to create the test configuration and `mmcrfs` to create the file system you want and operate with the new code for some time.

Once you are satisfied with the new code, you can upgrade the rest of the nodes by using the same procedure. When you decide that you will permanently accept GPFS 1.2, issue the command: `mmchfs <file system> -V`.

---

## 9.6 Coexistence Considerations

The following sections highlight coexistence considerations.

### 9.6.1 Within a Partition

Due to the changes in the token manager function, it is not possible for different levels of GPFS to coexist in the same configuration.

However, it is possible to run multiple configurations at different levels of GPFS in the same system partition.

### 9.6.2 Multiple Partitions

A GPFS file system can only be accessed from a single partition since switch communication, needed for communication between GPFS nodes, does not exist between system partitions.

Note that you can use a GPFS file system from another partition by mounting it via NFS like other file systems.

---

## 9.7 Compatibility

All applications which run with GPFS 1.1 will continue to run with GPFS 1.2.

File systems created under GPFS 1.1 may continue to be used under GPFS 1.2.

However, once a GPFS 1.1 file system has been explicitly changed to GPFS 1.2 by issuing the `mmchfs -v` command, the disk image can no longer be read by a GPFS 1.1 file system.



---

## Chapter 10. LoadLeveler Version 2.1

The primary purpose of this chapter is to acquaint the reader with LoadLeveler and to show how it can be used in the SP environment. For more detailed information, refer to *IBM LoadLeveler for AIX: Using and Administering Version 2 Release 1*, SA22-7311.

For the reader who is experienced with LoadLeveler, we also review the changes that have been made since the previous version, LoadLeveler Version 1.3.

---

### 10.1 LoadLeveler Introductory Concepts

LoadLeveler was initially developed at the University of Wisconsin as part of the CONDOR project. It is a software program designed to automate workload management. In essence, it is a scheduler which also has facilities to build, submit and manage jobs. The jobs can be processed by any one of a number of machines, which together are referred to as the LoadLeveler cluster. Any standalone RS/6000 may be part of a cluster, although LoadLeveler is most often run in the RS/6000 SP environment. A sample LoadLeveler cluster is shown in Figure 143.

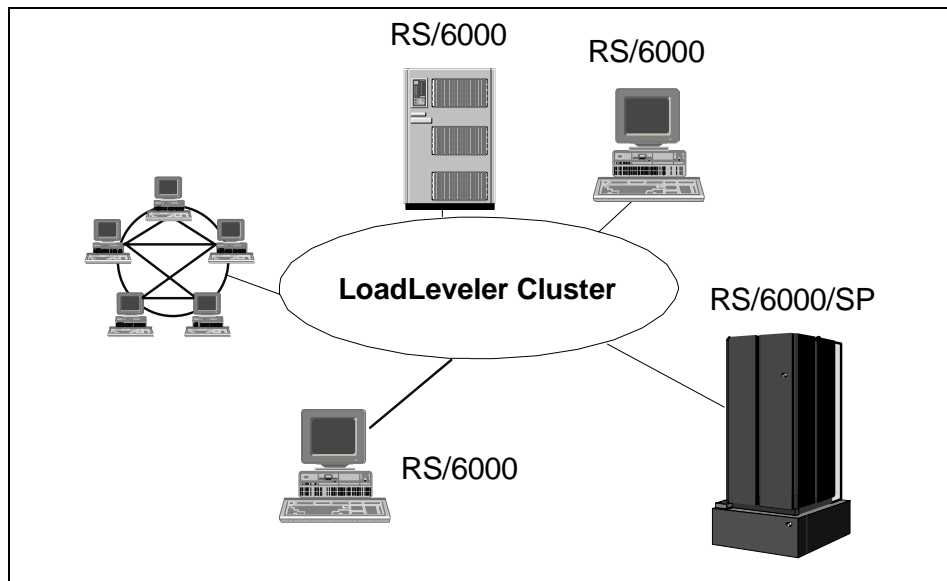


Figure 143. Example LoadLeveler Configuration

Important concepts in LoadLeveler are:

**Cluster.** A group of machines which are able to run LoadLeveler jobs. Each member of the cluster has the LoadLeveler software installed.

**Job.** A unit of execution processed by Loadleveler. A serial job runs on a single machine. A parallel job is run on several machines simultaneously and must be written using a parallel language Application Programming Interface (API). As LoadLeveler processes a job, the job moves in to various job states such as "Pending", "Running" and "Completed".

**Job Command File.** A formal description of a job written using LoadLeveler statements and variables. The command file is submitted to LoadLeveler for scheduling of the job.

**Job Step.** A job command file specifies one or more executable programs to be run. The executable and the conditions under which it is run are defined in a single job step. The job step consists of several LoadLeveler command statements.

By way of example, Figure 144 on page 261 schematically illustrates a series of job steps. In this figure, data is read from storage in job step one. Depending on the exit status of this operation, the job is either terminated or continues on to job step two. Again LoadLeveler examines the exit status of job step two and either proceeds on to job step three which, in this example, prints the data that the user requires or terminates.

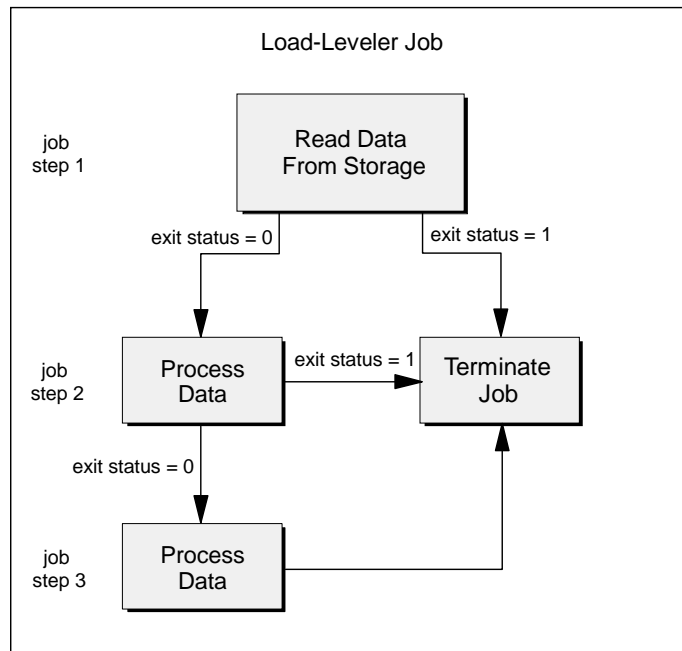


Figure 144. A LoadLeveler Job

### 10.1.1 LoadLeveler: A Breakdown of How It Works

There are three important functional machine types in LoadLeveler.

**Scheduling machine.** When a job is submitted to LoadLeveler, it gets placed in a queue which is managed by the scheduling machine. The latter then asks the central manager to find a machine which can process the job.

**Central manager machine.** This machine evaluates the resources required by the job that were specified in the job command file and selects a machine which is capable of running it. The central manager is also called the negotiator.

**Executing machines.** Machines which are assigned and run jobs.

Figure 145 shows how these machine types fit together and the order in which they communicate.

1. A job has been submitted to LoadLeveler

2. The scheduling machine contacts the central manager to inform it that a job has been submitted and to find out if there is a machine available that matches the job's requirements.
3. The central manager checks to determine if a machine exists that is capable of running the job. Once a machine is found, the central manager informs the scheduling machine which machine is available.
4. The scheduling machine contacts the executing machine and sends it the job information and executable program. The executing machine sends job status information to the scheduling machine, and notifies it when the job has completed.

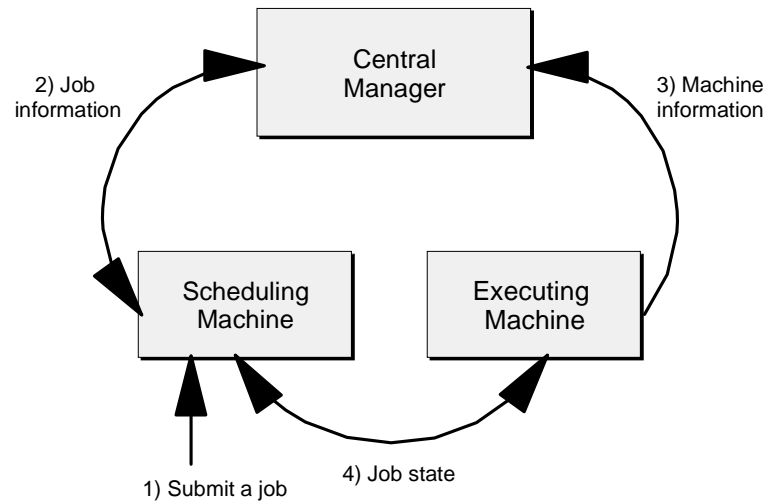


Figure 145. LoadLeveler Job Flow

In addition, there is another type of machine known as a submit-only machine. As its name indicates, this type of machine can only submit jobs, although it is also able to query and cancel them.

Jobs do not get dispatched to the executing machines on a first-come, first-served basis unless LoadLeveler is specifically configured to run that way, that is, with a first in first out (FIFO) queue. Instead, the negotiator calculates a priority value for each job called SYSPRIO that determines when the job will run. Jobs with a high SYSPRIO value will run before those with a low value.

The system administrator can specify several different parameters that are used to calculate SYSPRIO. Examples of these are: how many other jobs the user already has running; when the job was submitted; what priority the user

has assigned to it. The user assigns priorities to his own jobs by using the `user_priority` keyword in the job command file.

SYSPRIO is referred to as a job's *system priority*, whereas the priority that a user assigns his own jobs is called *user priority*. If two jobs have the same SYSPRIO calculated for them by LoadLeveler, then the job which runs first will be the job which has the higher user priority.

The priority of a job in the LoadLeveler queue is completely separate and must be distinguished from the AIX `nice` value, which is the priority of the process the executable program is given by AIX.

LoadLeveler also supports the concept of job classes. These are defined by the system administrator and are used to classify particular types of jobs. For example, we define two classes of job which run in the cluster called "night" jobs and "day" jobs. We might specify that executing machine A, which is very busy during the day because it supports a lot of interactive users, should only run jobs in the night class. However, machine B, which has a low workload in the day, could run both. LoadLeveler can be configured to also take job class in to account when it calculates SYSPRIO for a job.

As SYSPRIO is used for prioritizing jobs, LoadLeveler also has a way of prioritizing executing machines. It calculates a value called MACHPRIO for each machine in the cluster. The system administrator can specify several different parameters that are used to calculate MACHPRIO, such as load average, number of CPUs, the relative speed of the machine, free disk space and the amount of memory.

Machines may be classified by LoadLeveler into pools. Machines with similar resources, for example a fast CPU, might be grouped together in the same pool so that they could be allocated CPU-intensive jobs. A job can specify as one of its requirements that it run on a particular pool of machines. In this way, the right machines can be allocated the right jobs.

Pools, in Version 2.1 of LoadLeveler, are a replacement for "features", which were the equivalent concept in Version 1.3 of LoadLeveler.

### 10.1.2 LoadLeveler Daemons

LoadLeveler has a number of daemon processes to accomplish its tasks:

**LoadL\_master.** The master daemon runs on all the machines in the cluster and manages all the LoadLeveler daemons on the machine.

**LoadL\_schedd.** The schedd daemon runs on the scheduler machine and manages the jobs submitted to the job queue.

**LoadL\_startd.** The startd daemon runs on all executing machines in the cluster and accepts jobs to be run.

**LoadL\_starter.** The starter daemon is spawned by startd and is responsible for running a job on the executing machine.

**LoadL\_kbdd.** The keyboard daemon runs on all executing machines and monitors all keyboard and mouse activity.

**LoadL\_negotiator.** The negotiator daemon runs on the central manager machine. It collects job status and machine status from all machines in the cluster and makes decisions on where jobs should be run.

A summary of how these daemons work together to process a job in the LoadLeveler environment can be described as a series of steps. The job status is changed by negotiator as these steps are executed.

- A user submits a job command file to LoadLeveler either through the LoadLeveler GUI or the `llsubmit` command. Then schedd stores the job information on disk and places the job in the queue.
- When the job is ready to run, schedd sends job description information to the negotiator daemon. The negotiator continually receives machine state information from startd on the cluster machines and is able to make a decision on where the job should be run based on the job requirements. When it has done so, it sends schedd an authorization called a *permit* to run. Job status is Pending or Starting.
- Then schedd contacts startd on the executing node and requests that it start a job. Next startd spawns a starter process, and schedd then sends the starter process the job information and the name of the executable. Then schedd notifies the negotiator that the job has been started. Job status is changed to Running.
- The starter forks and executes the user's job and the starter parent waits for the child to complete.
- When the job has finished, the starter process notifies the startd daemon and startd notifies schedd. Then schedd forwards the information to the negotiator. Job status is changed to Completed.

### 10.1.3 Checkpointing

Checkpointing is a method of periodically saving the state of a job so that if the job does not complete, it can be started again from the saved state. Two different types of checkpointing may be specified:

**User initiated** - The user's application program decides when the checkpoint is taken. This type of checkpointing is permissible for serial and parallel jobs. User-initiated checkpointing is enabled in the job command file using the `checkpoint = user_initiated` keyword.

**System initiated** - This checkpoint is taken at intervals determined by the administrator and is only permissible for serial jobs. System-initiated checkpointing is enabled in the job command file using the `checkpoint = system_initiated` keyword. It is further configured by specifying time intervals, in the LoadLeveler configuration file, at which a running job is checkpointed by LoadLeveler.

Briefly, to enable user-initiated checkpointing, the user must call the checkpointing API that is supplied with LoadLeveler in his program and must link his program with the LoadLeveler checkpointing libraries. Precise details of this procedure are given in *IBM LoadLeveler for AIX. Using and Administering Version 2 Release 1*, SA22-7311.

When a checkpoint is initiated, the default behavior of LoadLeveler is to create a checkpoint file on the executing machine and store that same file on the scheduling machine. The checkpoint file contains the program's data segment, stack, heap, register contents, signal state and the states of the open files at the time of the checkpoint, and may be much larger than the executable. If an executing machine fails, then when LoadLeveler restarts, it will reschedule the job using its most recent checkpoint file. Since the checkpoint file is stored on the scheduling machine, the job may be restarted on any of the cluster machines.

There are a number of UNIX system calls which may not be used in a program which needs to use checkpoints. These are:

- Signals
- Shared libraries (dynamic loading)
- Memory-mapped files
- Threads
- Fork and exec system calls

- Interprocess communication (shared memory, semaphores, message queues and pipes)
- Set/get user or group IDs, process IDs
- Time and timer services

#### 10.1.4 Scheduling

LoadLeveler may be configured to run two different scheduling algorithms:

- Default algorithm. Intended for use primarily with serial jobs, it uses CPU time efficiently by scheduling jobs on those machines which have the highest value for MACHPRIO. Once it has allocated a machine to a job, it is then able to continue to monitor the machine workload. If the workload is too high, then the scheduler may choose to suspend the job and resume it at a later time, or perhaps delay the start of the job.
- Backfill algorithm. This is intended for use with parallel jobs. A drawback of the default algorithm is that when it has a parallel job to run, it reserves nodes until it has the number that the job requires. During this waiting period, the reserved nodes are not allowed run other jobs. Also it is possible that the default algorithm is not able to accumulate all the nodes that it needs to run, and the job may not be dispatched at all.

The backfill algorithm overcomes these disadvantages by requiring that each job, or at least job class, specify the maximum amount of time in which it will complete. This maximum time is known as the *job wall clock time*. Using the job wall clock times, the scheduler can determine the latest time that any job will run and can ensure that high priority jobs are never delayed. It also has a backfill capability which operates while it is waiting to start any large jobs that require many nodes. This means that it is able to schedule and run small or short jobs which arrive during the wait period.

- Job control API. This API is provided with LoadLeveler to allow you to enable an external scheduler.

#### 10.1.5 Parallel Jobs

LoadLeveler allows you to schedule parallel batch jobs that have been written using the following environments:

- IBM Parallel Environment Library (POE/MPI/LAPI) 2.4.0
- Parallel Virtual Machine (PVM) 3.3 (RS6K architecture)
- Parallel Virtual Machine (PVM) 3.311+ (SP2MPI architecture)



PVM is a public domain package which is distributed by Oak Ridge National Labs. Information on PVM can be obtained by sending a mail message to a list server:

```
echo "send index from pvm3" | mail netlib@orn1.gov
```

In the previous release, Version 1.3, LoadLeveler interacted with the PSSP Resource Manager, `ssp.jm`, to run parallel batch jobs. In particular, it relied on Resource Manager setting up and processing the Job Switch Resource Table (JSRT). The JSRT is used to map task ID to a node number for a user space job so that the parallel tasks know how to communicate with each other. However, LoadLeveler is now able to load and unload the JSRT itself using the switch table API, and no longer requires Resource Manager for this function. It is also able to provide access to the JSRT to other programs which need it, such as POE. The JSRT discussed in more detail in Chapter 11, "Parallel Environment 2.4" on page 295. The switch table API is documented in *PSSP Command and Technical Reference*, SA22-7351.

Another function of Resource Manager is to provide node and adapter configuration information that is required for running parallel interactive jobs. POE and LoadLeveler have now been modified so that POE can get node and adapter information from LoadLeveler instead. In addition, LoadLeveler has been enhanced so that it is now able to read node and adapter information directly from the SDR, which can then be used to build its administration file.

The new functions of LoadLeveler which provide equivalency to Resource Manager are described in more detail in "New Features in LoadLeveler Version 2.1" on page 289.

Scheduler machines which are intended for use with parallel jobs should be configured to use the backfill scheduler by using the `SCHEDULER_TYPE = BACKFILL` keyword in the local LoadLeveler configuration file.

---

## 10.2 LoadLeveler Jobs

The aim of this section is to assist the user in understanding three processes in the life cycle of a LoadLeveler job:

- Writing a description of the job in the job command file
- Submitting the command file to LoadLeveler for processing
- Managing the job after it has been submitted

## 10.2.1 Writing a Job Command File

It is not intended that this section be a complete reference guide for the LoadLeveler job command language. In particular, we do not describe any of the elements necessary for submitting parallel batch jobs. The novice reader can read this section and then refer to *IBM LoadLeveler for AIX: Using and Administering Version 2 Release 1*, SA22-7311 for detailed information.

A job command file is a plain text file which is either written using a text editor or by using the "Build a Job Window" in the LoadLeveler GUI. The file may contain:

**Comments.** Comment lines begin with a #.

**LoadLeveler keyword statements.** Keyword statements begin with a # @ character sequence and specify the name of the executable program to run, instructions to LoadLeveler on how to run the job, and the command which submits the job. Keywords are not case-sensitive. The back slash character (\) may be used as a line continuation character.

**Shell command statements.** A shell script can be used as the executable program in the job command file. The shell script can be written inside the job command file using shell command statements.

**Loadleveler variables.** LoadLeveler keeps track of information such as the name of the host that the job is running on, the job ID number, and the job step ID number. These can be referenced in the job command file through LoadLeveler variables.

Figure 146 shows a simple example of a job command file. This example illustrates several features of the command language. Firstly, the author has commented the file to make its purpose clear to the reader. Secondly, it does not specify the name of an executable program to be run. Instead, the job takes the form of a shell script which is defined inside the job command file itself. By default the shell script takes its name from the name of the job command file. The shell script prints out the following data:

- The name of the shell script
- The name of the host it is running on
- The current working directory
- The program environment
- Who is logged in

The third feature of this example are the output, error and queue keyword statements. Together, these three statements make up a single job step. There is only one job step in this example, although we will illustrate another command file in the next example which uses more than one job step. The output keyword specifies the name of the file to use for the standard output of the job step. Similarly, the error keyword indicates where the standard error should go. If a full pathname is not specified, the standard output and standard error will go to a file in the current working directory on the machine where the job is submitted.

We also note that these keywords make use of three LoadLeveler variables:

\$(Host) - the hostname from which the job was submitted

\$(jobid) - a sequential number assigned to the job by the submitting machine

\$(stepid) - a sequential number assigned to the job step

These variables are useful for distinguishing between output and error files from different machines and different jobs.

Lastly, we must mention the use of the queue keyword. This is the final action in any job step and places the job step in the LoadLeveler queue.

Figure 147 shows another example of a job command file with two job steps, that is, there are two queue keyword statements. LoadLeveler will run all job steps independently from one another unless the dependency keyword is used. In Figure 147, step2 will only run if step1 completes successfully since it has a dependency on step1. step1 is called the *sustaining job step*. This figure also illustrates an example of how the job command file may specify the environment in which the job is run, in this case by the use of the requirements keyword. step1 will only be run on RS/6000 architecture which is running AIX Version 4.3. The architecture of a machine and its operating system version can be specified as keywords in the LoadLeveler configuration file.

```
# Sample script to show basic function of LoadLeveler
# - since no "executable" is specified, this file "job1.cmd" will be
# used as the executable
#
# @ error   = job1.$(Host).$(jobid).$(stepid).err
# @ output  = job1.$(Host).$(jobid).$(stepid).out
# @ queue
#
# Kill some time
#
sleep 60

echo The name of this job is $0
echo

echo This job is running on `hostname`
echo

echo This job is running from `pwd`
echo

echo The environment is `env`
echo

echo These ids are logged onto the system:
who
```

*Figure 146. Job Command File Using Shell Command Statements*

```

# This job command file illustrates the use of multiple job steps. step2
# has a dependency of step1 and will only run if step1 completes with an
# exit status of 0
#
# @ step_name = step1
# @ executable = program1
# @ requirements = (Arch == "RS6000" ) && (OpSys == "AIX43")
# @ input = step1.in
# @ output = step1.out
# @ error = step1.err
# @ queue
#
# @ dependency = ( step1 == 0 )
#
# @ step_name = step2
# @ executable = program2
# @ input = step2.in
# @ output = step2.out
# @ error = step2.err
# @ queue

```

Figure 147. Job Command File Using Dependencies

## 10.2.2 Submitting a Job Command File

After building a job command file, you can submit it for processing either by using the LoadLeveler GUI or by using the `llsubmit` command with the job command filename as the argument. For example,

```
llsubmit myjob.cmd
```

LoadLeveler responds by issuing a message similar to:

```
submit: The job "sp4en0.22" has been submitted.
```

where `sp4en0` is the name of the machine to which the job was submitted and `22` is a job identifier. In fact, when a job is submitted, LoadLeveler assigns it a three part-identifier, as follows:

Machine name	The name of the machine that schedules the job
Job ID	An identifier given to the group of job steps that are specified in the job command file. This is the same as the <code>\$(jobid)</code> command file variable and has a value of <code>22</code> for

the job that was submitted to sp4en0 in the preceding example.

**Step ID** The step identifier is used when there is more than one job step in the job command file. Job steps in the same command file will have the same job ID but a different step ID. This is the same as the \$(stepid) command file variable. In the example above where job sp4en0.22 was submitted, LoadLeveler does not report a step ID to the user as there is only one step in the job command file. However, internally, LoadLeveler represents the job as sp4en0.22.0 where 0 is the first ( and only ) job step.

### 10.2.3 Managing a Job

There are several LoadLeveler commands that are available for managing a job once it has been submitted. For a full description of these commands, refer to *IBM LoadLeveler for AIX: Using and Administering Version 2 Release 1*, SA22-7311.

Some useful ones are:

**llq** Display the status of a job that has been submitted. For example,

```
llq sp4en0.22
```

where sp4en0 is the name of the machine to which the job was submitted and 22 is the job ID.

Useful flags for the llq command are:

-l. Generates a detailed description for a job

-s. Provides information on why a job is in the "Idle" state or "Deferred" state.

**llhold** Puts a temporary hold on a job in a queue. This command will only take effect if a machine has not yet been selected to run the job. For example,

```
llhold sp4en0.22 will hold the job and,
```

```
llhold -r sp4en0.22 will release it again.
```

**llcancel** Cancels a job. To cancel sp4en0.22, enter,

```
llcancel sp4en0.22
```

**llprio** Changes the user priority of a job in the LoadLeveler queue. To increase the priority of sp4en0.22 by a value of 10, enter:

```
llprio +10 sp4en0.22
```

The priority of a job can range from 0 to 100, with higher numbers corresponding to greater priority and the default being 50. The user may also alter the job priority in the job command file by using the user\_priority keyword.

**llstatus** Display the status of the LoadLeveler cluster.

llstatus with no command line options can also be used from any machine in the cluster to find the location of the central manager and scheduler machines. This is shown in Figure 148.

```
sp4n01:/home/loadl $ llstatus
Name                Schedd  InQ  Act  Startd  Run  LdAvg  Idle  Arch  OpSys
sp4en0.msc.itso.ibm.com  Avail  0   0  Idle    0  0.04   0  R6000  AIX43

sp4n01.msc.itso.ibm.com  Avail  0   0  Idle    0  0.00   1  R6000  AIX43

sp4n05.msc.itso.ibm.com  Avail  0   0  Idle    0  0.05  9999  R6000  AIX43

sp4n06.msc.itso.ibm.com  Avail  0   0  Idle    0  0.00  9999  R6000  AIX43

sp4n07.msc.itso.ibm.com  Avail  0   0  Idle    0  0.05  9999  R6000  AIX43

R6000/AIX43          5 machines    0  jobs    0  running
Total Machines      5 machines    0  jobs    0  running
```

Figure 148. Standard Listing of the llstatus Command

### 10.3 Installing and Configuring LoadLeveler

LoadLeveler installation and configuration can be complex. This section helps the reader by explaining the concepts of installation and configuration, but does not provide a definitive list of steps. For detailed information on installing and configuring LoadLeveler, refer to the following LoadLeveler publications:

*IBM LoadLeveler for AIX. Using and Administering Version 2 Release 1*

*IBM LoadLeveler for AIX Version 2 Release 1.0 Installation Memo*

Example configuration and administration files can be found in the release directory `/usr/lpp/LoadL/full/samples`.

### 10.3.1 Installation

Three steps need to be taken before installing LoadLeveler:

1. Decide which machine will act as the central manager and create the LoadLeveler user ID, LoadLeveler group ID and the LoadLeveler home directory on that machine. By default, `loadl` is the name used for LoadLeveler user ID and group ID.
2. Decide on LoadLeveler directory structure. There are several components to the directory structure which are listed as follows. An example configuration is shown in Table 25.
  - home directory. This is Loadleveler's home directory. It is located on the central manager machine and mounted on the nodes using a distributed file system such as NFS or AFS.
  - release directory. This defines the directory where all the LoadLeveler software resides. It may reside on the central master and be distributed to the other cluster machines.
  - local directory. Defines a directory tree which is private to each machine in the cluster.
  - log directory. Defines a local directory to store log files.
  - spool directory. Defines a local directory where LoadLeveler stores local job queue and checkpoint files.
  - execute directory. This defines the a local directory to store the executables of jobs submitted by other machines

Table 25. Example LoadLeveler Directory Tree

LoadLeveler Element	Location on Central Manager (hostname = cenman)	Location on machine in cluster (hostname = node1)
Home directory	<code>/home/loadl</code> (exported)	<code>/home/loadl</code> (mounted)
Release directory	<code>/usr/lpp/LoadL/full</code> (exported)	<code>/usr/lpp/LoadL/full</code> (mounted)
Local directory Tree	<code>/home/loadl/cenman</code>	<code>/home/loadl/node1</code>
Log directory	<code>/home/loadl/cenman/log</code>	<code>/home/loadl/node1/log</code>
Spool directory	<code>/home/loadl/cenman/spool</code>	<code>/home/loadl/node1/spool</code>
Execute directory	<code>/home/loadl/cenman/execute</code>	<code>/home/loadl/node1/execute</code>



- Decide on the location of the LoadLeveler configuration file and administration file. By default, this location is the home directory of the LoadLeveler user ID.

**Configuration file.** There is a single global configuration file that is stored on the central manager and is distributed to all the nodes. However, each machine also has a local configuration file which is referenced from the global configuration file. The local configuration file is located in the local directory tree on each machine in the cluster.

**Administration file.** There is only a single copy of the administration file that is stored on the central manager. This file must also be distributed to all nodes.

An example configuration follows in Table 26:

Table 26. Location of Configuration and Administration Files

File Type	File name	Location on Central Manager	Location on machine in cluster
Global Configuration File. One copy for all machines	LoadL.config	/home/loadl (exported)	/home/loadl (mounted)
Local Configuration File	LoadL_config.local	/home/loadl/cenman	/home/loadl/node1
Administration File. One copy for all machines	LoadL_admin	/home/loadl (exported)	/home/loadl (mounted)

Once these decisions have been made, LoadLeveler must be installed on the central manager machine, either using the `installp` command from the command line or by using the SMIT fastpath, `smit install_latest`. This procedure is documented in *IBM LoadLeveler for AIX Version 2 Release 1.0 Installation Memo* and is not repeated here. LoadLeveler will be installed in the `/usr/lpp/LoadL` directory.

Once the `installp` command has finished, five further steps are required before the LoadLeveler installation is complete:

- Specify the location of the local configuration file. We do this by modifying the global configuration file in the LoadLeveler samples directory. The path to this file is `/usr/lpp/LoadL/full/samples/LoadL.config`. Using the example in Table 25, you would write in `LoadL.config`:

```
LOCAL_CONFIG = $(tilde)/$(host)/LoadL_config.local
```

Also specify the locations of the LOG, SPOOL and EXECUTE directories in LoadL.config:

```
LOG           = $(tilde)/$(host)/log
SPOOL        = $(tilde)/$(host)/spool
EXECUTE      = $(tilde)/$(host)/execute
```

2. Change the ownership of all executable files in /usr/lpp/LoadL so that they are owned by the loadl user ID. This excludes LoadL\_master, the LoadLeveler master daemon which is owned by root. If you installed the entire LoadLeveler product, then the following commands issued as the root user will accomplish this:

```
cd /usr/lpp/LoadL; chown -R loadl.loadl full so
cd /usr/lpp/LoadL/full/bin; chown root.system LoadL_master
```

3. Make the LoadLeveler local directory and ensure that it is owned by the LoadLeveler user, loadl.

```
mkdir /home/loadl/cenman; chown loadl.loadl /home/loadl/cenman
```

4. Run `llinit`, the LoadLeveler installation script. This completes the installation of LoadLeveler. You must run `llinit` as the loadl user and also make sure that the environment variable `$HOME` is set to the home directory of loadl.

Part of the `llinit` process is to copy `LoadL.config` from the samples directory in to the LoadLeveler home directory. `llinit` will then read `LoadL.config` and create `LoadL_config.local` and the local directories as they were specified in step 1.

Here is how you should run `llinit`:

```
cd /usr/lpp/LoadL/full/bin
./llinit -local /home/loadl/cenman -release /usr/lpp/LoadL/full \
-cm cenman
```

This will install LoadLeveler with a local directory of `/home/loadl/cenman`, a release directory of `/usr/lpp/LoadL/full` on a central manager named `cenman`.

5. If the release directory has been exported to other nodes in the cluster with NFS or AFS, then you do not have to install the LoadLeveler images on them.

Instead, it is only necessary to mount the LoadLeveler release directory and the LoadLeveler home directory on the all machines in the cluster and then run the `llinit` script on each. Before running `llinit`, you must create symbolic links for the LoadLeveler shared libraries.

On each node in the cluster run the following steps:

1. Mount the LoadLeveler directories on the node:

```
mount cenman:/usr/lp/LoadL /usr/lpp/LoadL
mount cenman:/home/loadl /home/loadl
```

2. Make the symbolic links:

```
ln -s /usr/lpp/LoadL/full/lib/libllapi.a /usr/lib/lib/libllapi.a
ln -s /usr/lpp/LoadL/full/lib/libllmulti.a \
    /usr/lib/libllmulti.a
```

3. Make the local directory for the node in the LoadLeveler home directory:

```
mkdir /home/loadl/node1 ; chown loadl.loadl /home/loadl/node1
```

4. Run llnit for the node:

```
cd /usr/lpp/LoadL/full/bin
./llinit -local /home/loadl/node1 -release /usr/lpp/LoadL/full \
    -cm cenman
```

When these steps have been done for each node in the cluster, the installation of the LoadLeveler cluster will be complete.

### 10.3.2 LoadLeveler Administration File

The administration file is called `LoadL_admin`. It takes the form of a series of stanzas which define machines, users, classes, groups and adapters.

<b>Machine stanza</b>	Defines the machines in the LoadLeveler cluster
<b>User stanza</b>	Defines LoadLeveler users and their characteristics
<b>Class stanza</b>	Defines the characteristics of the LoadLeveler job classes
<b>Group stanza</b>	Defines the characteristics of a group of LoadLeveler users
<b>Adapter stanza</b>	Defines the network adapters available on machines in the LoadLeveler cluster

Figure 149 on page 279 shows an example of a LoadLeveler administration file. It has the following characteristics:

- Every stanza has a label associated with it. In this example, the labels are `mynode`, `myclass`, `myuser`, `mygroup` and `myadapter`.
- Every stanza has a type field that specifies it as machine, user, class, group or adapter.

- After the type field, the stanza consists of a series of keywords which are used to define the stanza. Some useful ones are shown in the example such as:
  - priority, used to set the default priority for a user, group or class
  - wall\_clock\_limit, used to set the maximum CPU time a job can use
  - maxjobs, the maximum number of jobs a user is allowed to run at once
  - maxqueued, the maximum number of jobs the user is allowed on the system queue
- LoadLeveler recognizes "default" as a special label. It can be used to specify default values for any keywords which are not included in the stanza.
- The pound sign # is used to insert comments in the file.

```

mynode:      type = machine
              central_manager = false # not central manager
              schedd_host = false    # not a public scheduler
              submit_only = false     # not a submit-only machine
              speed = 1               # machine speed
              cpu_speed_scale = false # scale cpu limits by speed
              adapter_stanzas = default # identifies the adapter stanza

myclass:     type = class             # class stanza
              priority = 0            # ClassSysprio
              max_processors = -1     # max processors for class (no
              # limit)
              wall_clock_limit = 30:00 # wall clock limit

myuser:      type = user              # user stanza
              priority = 0           # default UserSysprio
              default_class = No_Class # default class = No_Class (not
              # optional)
              default_group = No_Group # default group = No_Group (not
              # optional)
              maxjobs = -1           # maximum jobs user is allowed
              # to run simultaneously (no limit)
              maxqueued = -1         # maximum jobs user is allowed
              # on system queue (no limit). does not
              # limit jobs submitted.

mygroup:     type = group             # group stanza
              priority = 0           # default GroupSysprio
              maxjobs = -1           # maximum jobs group is allowed
              # to run simultaneously (no limit)
              maxqueued = -1         # default maximum jobs group is allowed
              # on system queue (no limit). does not
              # limit jobs submitted

myadapter:   type = adapter
              adapter_name = en0     # defines the adapter to use as en0

```

Figure 149. Example LoadLeveler Administration File

To get LoadLeveler to start, it is sufficient to use the sample administration file found in `/usr/lpp/LoadL/full/samples`. Make sure that all the default stanzas are uncommented and then put your machine names in the file. Figure 150 shows an example of a five-machine LoadLeveler cluster.

```
#####
# MACHINE STANZAS:
# These are the machine stanzas; the first machine is defined as
# the central manager. mach1:, mach2:, etc. are machine name labels -
# revise these placeholder labels with the names of the machines in the
# pool, and specify any schedd_host and submit_only keywords and values
# (true or false), if required.
#####
sp4en0: type = machine
        central_manager = false
sp4n01: type = machine
        central_manager = true
sp4n05: type = machine
        central_manager = false
sp4n06: type = machine
        central_manager = false
sp4n07: type = machine
        central_manager = false
```

Figure 150. Specifying Machine Names in the Administration File

### 10.3.3 LoadLeveler Configuration File

In this section we broadly describe the purpose of the LoadLeveler configuration file.

The global configuration file is called LoadL\_config. It contains configuration information which is common to all nodes in the LoadLeveler cluster.

The local configuration file is called LoadL\_config.local. It has the same format as the global configuration file, but the information in this file overrides the information contained in the global configuration file.

The reader must consult the manual *IBM LoadLeveler for AIX. Using and Administering Version 2 Release 1, SA22-7311* for detailed information on the format of the configuration files. See also the example configuration file in the LoadLeveler release directory /usr/lpp/LoadL/full/samples.

The configuration file defines the following:

- LL administrators. This is a list of the user IDs of those people who will have access to the LoadLeveler administrator only commands such as llctl, llfavorjb and llfavoruser.

- Cluster characteristics. These include what type of scheduling algorithm is used by LoadLeveler.
- Machine characteristics. These include machine architecture (whether the machine is an RS/6000), the level of operating system and whether the machine runs the scheduler, schedd.
- How many jobs a machine can run.
- What parameters are used to calculate SYSPRIO.
- What parameters are used to calculate MACHPRIO.
- Specify job control expressions which define how a job should be managed. There are five job control expressions in LoadLeveler:

**START** Defines the conditions for starting a job.

**SUSPEND** Defines the conditions for suspending a job, that is putting the process to sleep.

**CONTINUE** Defines the conditions for restarting suspended processes.

**VACATE** Defines the conditions for removing a suspended job from a machine and placing it back on the scheduler queue.

**KILL** Defines conditions for stopping a job and removing it from the scheduler queue.

Figure 151 is an example of how job control expressions are used to control a job. It shows an excerpt from a local LoadLeveler configuration file which instructs LoadLeveler to only run jobs at a certain time of day. The sample code will only start jobs after 5:00 PM and before 8:00 AM. If a job is submitted after 8:00 AM, it will be suspended. The `CONTINUE` expression will restart suspended jobs after 5:00 PM.

```
START: (tm_day >= 1700) || (tm_day <= 0800)

SUSPEND: (tm_day > 0800) && (tm_day < 1700)

CONTINUE: (tm_day >= 1700) || (tm_day <= 0800)
```

Figure 151. Running Jobs at a Specific Time of Day

- Whether job accounting should be enabled for a machine. Accounting can provide information on CPU usage, the resident set size (RSS), paging and I/O usage of a job.
- Specify alternate central manager machines which can take over the negotiator functions should the central manager machine fail.

- Where files and directories are located. The location of the local configuration file and administration file may be specified, as well as the location of the release and local directories.
- Log file management. This includes what type of messages are logged by the LoadLeveler daemons and how and when the log files are truncated.
- The TCP port numbers that are used by the LoadLeveler daemons.
- Whether job checkpointing is enabled.

---

## 10.4 Controlling LoadLeveler

When the LoadLeveler administration and local configuration files have been modified to suit the running environment, LoadLeveler can be started by running the `llctl` command on the central master machine:

```
llctl -g start
```

The `-g` flag specifies the `LoadL_master` daemon will try to start LoadLeveler on all the machines that are in the cluster, that is all the machine names that were specified in the administration file.

Conversely, LoadLeveler can be stopped using the command:

```
llctl -g stop
```

`llctl` commands must be issued as the LoadLeveler user, `loadl`.

If it is not desirable to stop or start everything at once, machines can be controlled one at a time by using the `-h` (host) flag. From the central master type:

```
llctl -h sp4n05 start
```

This command will start LoadLeveler on the machine `sp4n05`.

### 10.4.1 Using the LoadLeveler GUI

LoadLeveler provides a Motif-based GUI which can be used as an alternative to entering LoadLeveler commands from the shell prompt. The GUI is started using the command:

```
/usr/lpp/LoadL/bin/xloadl &
```



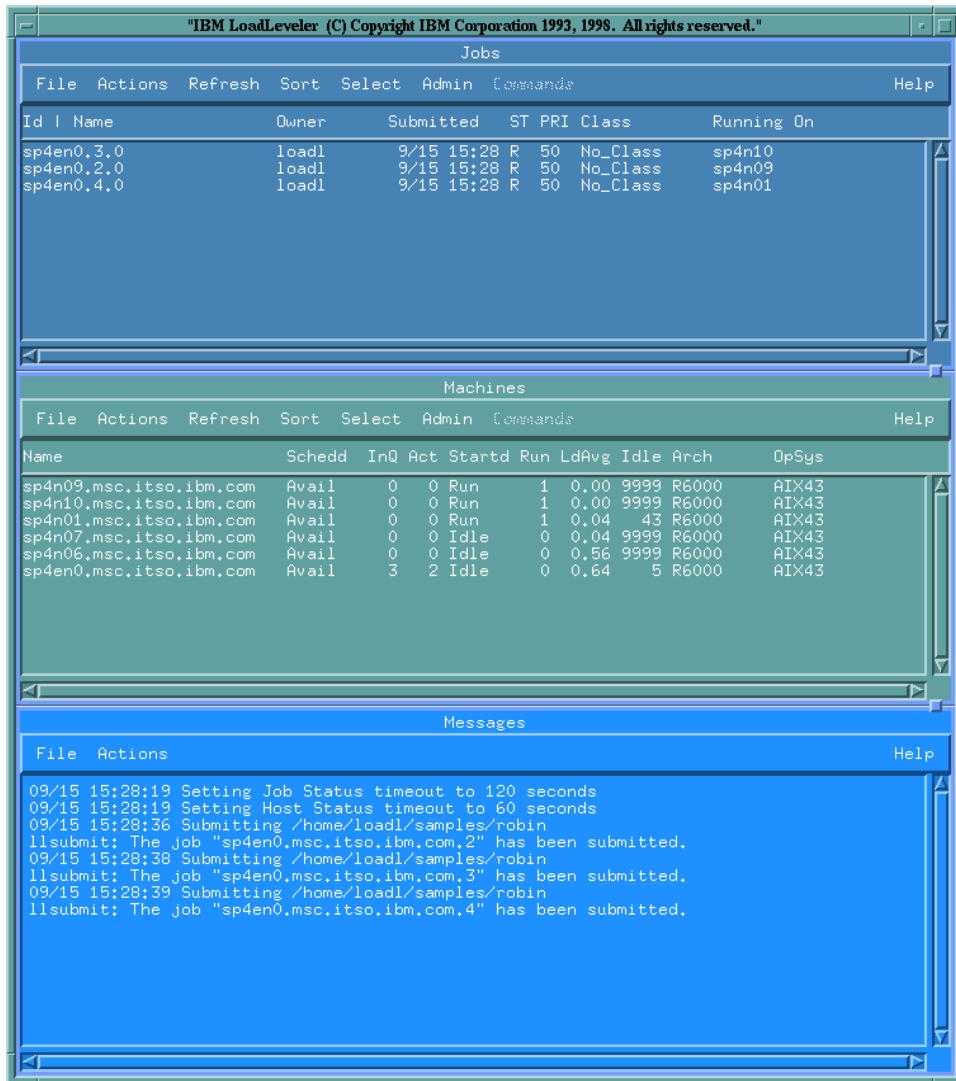


Figure 152. LoadLeveler GUI Main Window

Figure 152 shows the GUI main window which appears when LoadLeveler is started. There are three horizontal panes in the window which display information regarding:

- Jobs
- Machines
- LoadLeveler Messages

The messages pane shows that three jobs have been submitted. The name of the host that submitted the job is put in the job name so we see from the pane that all three jobs were submitted by host sp4en0. The number after the first period in the job name corresponds to the job number. The number after the second period corresponds to the job step.

The jobs pane shows more detailed information about the jobs:

- Job name
- Owner. The ID of the user who submitted the job.
- Submitted. The time that the job was submitted.
- Status. Job status such as "Starting", "Pending", "Running" or "Idle".
- Priority. Job priority.
- Running on. Name of the machine which the job is running on.

For example, you can see that job sp4en0.03.0 was submitted by loadl at 15:28 on the 15th of September. It has a status of R, which is "Running", and a priority of 50. It belongs to a class called No\_Class and it is running on host sp4n10.

Figure 153 shows another example of the jobs pane. In this example, three jobs have been submitted to LoadLeveler. These are job numbers 24, 25 and 26 and they are all in the "small" class. However, there are six entries listed in the jobs pane. This is because each job has two job steps (step 0 and step 1).

The ST (job state) column shows that only three steps are in the "R" or running state, whereas all the rest are in the "I" state which is Idle. Idle means that a step is being considered to run on a machine, but no machine has been selected yet.

For a machine to be able to run a job step in a particular class, in this example the "small" class, the following line must appear in the machine's local configuration file:

```
Class = { "small" }
```

This line means that this LoadLeveler machine will run one job step belonging to the class "small" at a time. If this line appears in the local configuration file of all the machines in the cluster and there are three machines in the cluster, then a total of three steps in the class "small" can be running in the cluster at any one time. Any remaining job steps in "small" must wait in the scheduler queue in the Idle state until the negotiator signals that a machine has finished running a job step and has become free to run another.

If you want a machine to be able to run more than one job step in a particular class at one time, then you must change the Class statement. For example, to run two job steps in "small" at once, you need to write in the local configuration file:

```
Class = { "small" "small" }
```

Jobs							
File	Actions	Refresh	Sort	Select	Admin	Commands	Help
Id	Name	Owner	Submitted	ST	PRI	Class	Running On
sp4en0.26.0		load1	9/25 14:04	I	50	small	
sp4en0.26.1		load1	9/25 14:04	I	50	small	
sp4en0.24.0		load1	9/25 14:04	R	50	small	sp4n09
sp4en0.24.1		load1	9/25 14:04	R	50	small	sp4n10
sp4en0.25.0		load1	9/25 14:04	R	50	small	sp4n01
sp4en0.25.1		load1	9/25 14:04	I	50	small	

Figure 153. LoadLeveler GUI Jobs Pane

Figure 154 shows the machines window which corresponds to the jobs pane in Figure 153. Features that should be noted from this pane are:

- There are six machines in the LoadLeveler cluster (sp4n01, sp4n06, sp4n07, sp4n09, sp4n10 and sp4en0).
- Schedd is running on all six machines.
- All machines are RS/6000 architecture and are running AIX Version 4.3.
- Machine sp4en0 has 6 job steps in its scheduler queue and three jobs are currently running. This is the machine that submitted the jobs.
- Machines sp4n01, sp4n09 and sp4n10 are running one job each.
- Machines sp4en0, sp4n06 and sp4n07 are not running any jobs. They are currently idle. This is because they have not been configured to run jobs in the "small" class.

Machines											
File	Actions	Refresh	Sort	Select	Admin	Commands					Help
Name	Schedd	InQ	Act	Startd	Run	LdAvg	Idle	Arch	OpSys		
sp4n09.msc.itso.ibm.com	Avail	0	0	Run	1	0,00	9497	R6000	AIX43		
sp4n07.msc.itso.ibm.com	Avail	0	0	Idle	0	0,00	9999	R6000	AIX43		
sp4n10.msc.itso.ibm.com	Avail	0	0	Run	1	0,00	9999	R6000	AIX43		
sp4n06.msc.itso.ibm.com	Avail	0	0	Idle	0	0,00	9999	R6000	AIX43		
sp4n01.msc.itso.ibm.com	Avail	0	0	Run	1	0,00	19	R6000	AIX43		
sp4en0.msc.itso.ibm.com	Avail	6	0	Idle	0	0,09	1	R6000	AIX43		

Figure 154. LoadLeveler GUI Machines Pane

Neither the jobs window nor the machines window are continuously updated. By default, LoadLeveler will automatically update the jobs window every 120 seconds and the machines window every 60 seconds. However, you can manually update both windows by dropping down the "Refresh" menu and selecting "Refresh All".

#### 10.4.2 Submitting a Job

You can use the GUI to submit a job by selecting the "File" menu on the jobs pane and choosing "Submit a Job". This will make the "Submit a Job" dialog box appear, Figure 155. When you have highlighted the file that you need, click on the "Submit" button. This will send the job to LoadLeveler scheduler and the GUI messages pane will be updated as shown in Figure 152. Unless you want to submit more jobs, click on the "Close" button to make the dialog disappear.

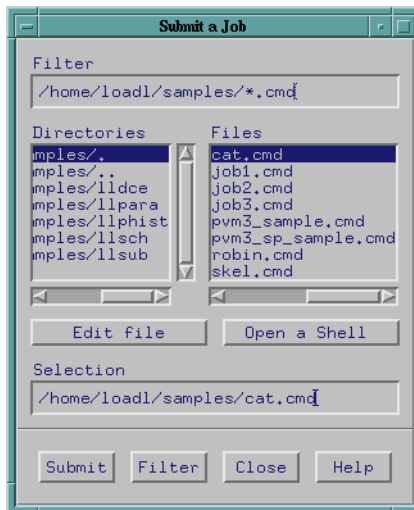


Figure 155. Submit a Job Dialog

### 10.4.3 Building a New Job

Writing a job command file in a text editor requires a knowledge of the LoadLeveler job command language syntax. Instead, you may use a dialog box in the GUI which will read your input and write the job command file for you. To do this drop down the "File" menu in the jobs pane and click on "Build a job", Figure 156. You need to choose the job type: serial, parallel or PVM. Figure 157 shows the "Build a Job" dialog for a serial job. The text boxes in the dialog box show the following details about the job:

- The executable program name is the `cat` command.
- The argument list is `"/etc/hosts"`. This job will run the command
 

```
cat /etc/hosts
```
- Standard output will go to a file called `cat.$(host).$(Process).out`, where `$(host)` is the name of the host which ends up running the command and `$(Process)` is the job ID.
- Standard error will go to a file called `cat.$(host).$(Process).err`
- The initial directory is `/usr/lpp/LoadL/full/samples`, located on the host which is submitting the job. This is where the standard output and standard error will go.
- The user `loadl@sp4en0` will be sent e-mail when the job is complete.

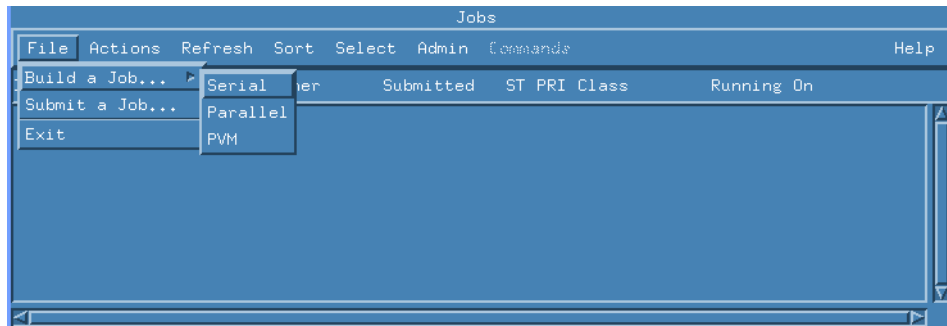


Figure 156. Choosing a Job Type for Building

You can click on the "Submit" button to submit the job immediately. Clicking on the "Save" button will save the job description as a job command file. You can then go and look at the file to review the syntax of the job command language statements that the GUI has placed in the file. You can submit the file at any time, either through the GUI or by using the `lsubmit` command.



Figure 157. Build a Job Dialog

## 10.5 New Features in LoadLeveler Version 2.1

The following is a list of new features in LoadLeveler Version 2 Release 1.

### 10.5.1 Enhanced Parallel Environment Support in LoadLeveler

There are three modifications that have been made to LoadLeveler to enhance the usability of LoadLeveler and POE, taking advantage of new functions available in the SP products:

- LoadLeveler will now support multiple user space processes on one SP node. Each SP switch adapter has four adapter windows available for use by user space processes. One user space process is allowed to run in each window. Switch adapter windows are discussed in detail in Chapter 11, “Parallel Environment 2.4” on page 295.
- LoadLeveler will now allow the use of multiple communications protocols within the same parallel task. Two protocols are currently supported, Message Passing Interface (MPI) and Low-level API (LAPI). MPI will run in both IP or user space mode, but LAPI can only run in user space. LoadLeveler can now run two MPI and two LAPI tasks concurrently on the same adapter. A job step can now specify protocol it uses by using the new network keyword in the job configuration file.
- LoadLeveler now allows multiple tasks of a parallel job to run on the same node. This improvement enables utilization of multiple processors in SMP nodes in an SP. Three new job command file keywords have been introduced support this: `node`, `task_per_node` and `total_tasks`.

### 10.5.2 Integration of Resource Manager Functions

The original purpose of the Resource Manager was to provide a workload management application that was responsible only for SP-specific tasks. The problem with this approach is that Resource Manager has only limited scheduling capabilities, and in order to get sophisticated workload management, the system administrator needed to use LoadLeveler as well.

A second drawback of using Resource Manager is that it relies on the SDR to store the state of nodes and adapters. If the SDR or the control workstation fails, then any LoadLeveler job whose nodes were allocated through the resource manager will be terminated and no new jobs will be started.

The functions that Resource Manager provided to LoadLeveler in its previous release were:

- Tracking of the usage of nodes and switch adapters by both batch and interactive jobs
- Processing of the switch table for parallel jobs that request to run in user space mode
- Controlling user logins

The following changes have been made to LoadLeveler to incorporate these Resource Manager functions:

- The available node and switch adapter information that Resource Manager used to get from the SDR is now defined in the LoadLeveler



administration file. LoadLeveler now provides a new command `llexSDR`, which extracts node and switch data from the SDR

- LoadLeveler now uses the switch table API that was made available in the Version 2.2 release of PSSP.
- LoadLeveler does not directly interact with the Login Control Facility. LoadLeveler logs in to nodes as root and switches to the user's ID.

Table 27 summarizes other Resource Manager functions and how their equivalent function can be accessed in LoadLeveler:

*Table 27. Resource Manager Functions Now in LoadLeveler*

Resource Manager Function	LoadLeveler function
Support for Node Pools	<b>pool_list</b> keyword in machine stanza +
Requesting dedicated use of nodes	<b>node_usage</b> keyword *
Requesting dedicated use of adapters	<b>network</b> keyword *
Displaying job information	<b>llq</b>
Displaying pool information	<b>llstatus -l</b>
Specifying batch, interactive or general use for nodes	<b>machine_mode</b> keyword +

+ specified in LoadLeveler administration file

\* specified in LoadLeveler job command file

It must be made quite clear that Resource Manager has been removed from the Version 3.1 Release of PSSP. The user now has three job management options:

- Use LoadLeveler as the job management application on the SP
- Run with no job management
- If third party parallel job management applications are already being used which interact with Resource Manager, then they will not work with PSSP 3.1. The applications will need to be modified to use the switch table API.

### 10.5.3 Changes and Enhancements to Checkpointing

There are a number of changes and enhancements to checkpointing which affect LoadLeveler:

- There are now new libraries for user and system initiated checkpointing which are supplied with LoadLeveler. All checkpointing programs must be statically linked with the libraries libchkrst.a and chkrst\_wrap.o.
- The following compilers are now supported for checkpointing:
  - FORTRAN: xlf 5.1.1 or later releases
  - C and C++: xIC 3.6.x, or Visual Age C, C++ (VAC++) 4.1

Compile scripts which assist with linking are provided in the bin subdirectory of the LoadLeveler release directory.

- A call to the checkpointing library used to take a core dump of the program which made the program exit. However, now the program's data segment (including program heap and stack areas, register values and program counter) is saved instead and it can continue to run without exiting. This has the additional benefit that less disk space is required for checkpoint file storage.
- When the checkpointing event occurs, the program's signal state is discovered by using signal system calls and then saved in the checkpoint image. When the job is restarted, the signal state can then be restored to what it was at the time of the checkpoint.
- System initiated checkpoints work by sending a checkpoint signal to a program. The checkpoint signal handler code has been rewritten so that it is signal-safe.
- It is now possible for the user to specify a checkpoint file in the job command file from which a job is restarted.
- The checkpointing library now supports restarting jobs on other machines in the cluster. This may need to happen in the event of node failure. This is referred to as *migration* and requires that the checkpoint file and the program object file be copied to the new machine on which the program is to be restarted.
- The user-initiated checkpoint library can now be used for parallel programs which use MPI for intertask communication and the library is now shipped with POE, as well as LoadLeveler.

Checkpointing for parallel programs currently works only for complete user- initiated parallel checkpointing. This means that each instance of the parallel program (that is, each node where the parallel program is running) must make a call to the checkpoint library and in each instance a local checkpoint is taken. The MPI messages which are being passed to other tasks are saved in the checkpoint image.

#### 10.5.4 New Scheduling Algorithm

A new algorithm called the *backfill scheduler* is included in LoadLeveler Version 2.1. It is designed for use with parallel jobs and is described in detail in 10.1.4, "Scheduling" on page 266.

#### 10.5.5 Migration from Version 1.3 of LoadLeveler

You should refer to *IBM LoadLeveler for AIX Version 2 Release 1.0 Installation Memo* for detailed instructions on how to migrate from Version 1.3. Some important changes which affect migration are mentioned here:

1. The LoadLeveler release directory has changed:
  - /usr/lpp/LoadL/nfs is now /usr/lpp/LoadL/full
  - /usr/lpp/LoadL/nfs\_so is now /usr/lpp/LoadL/so
2. The llq and llstatus commands have been changed to show information about parallel jobs and node pools respectively, in line with the removal of the Resource Manager from the PSSP software.
3. Use of the Adapter keyword in the job command file to select adapters for parallel jobs is now deprecated. The network keyword should be used instead. However, LoadLeveler will convert a single instance of the Adapter keyword present in the job command file to the equivalent network form. Instances after the first one are ignored and not converted.
4. The format of the LoadLeveler job queue files has changed. LoadLeveler provides a utility called llbconvert to convert the job\_queue.dir and job\_queue.pag files in the spool directory from 1.3 format to 2.0 format.

#### 10.5.6 Interactive Session Support (ISS)

The Interactive Session Support (ISS) function which was previously included with LoadLeveler is now only available as part of Interactive Network Dispatcher (IND). The ISS function in IND now allows various load measurements to be combined to form a weighted average of the overall system load. For more information, see *Interactive Network Dispatcher User's Guide*, GC31-8496.



---

## Chapter 11. Parallel Environment 2.4

IBM Parallel Environment for AIX provides the components required to develop, debug, analyze, tune and execute parallel applications, which are typically numeric- and compute-intensive problems requiring the high-performance available through parallel processing. This version supports AIX 4.3.2, and requires PSSP 3.1, when running on an SP system.

IBM Parallel Environment for AIX can be used on a single RS/6000 machine to develop parallel applications. These parallel applications will run, without modification, on an RS/6000 SP or cluster of RS/6000s. If such a parallel application does a significant amount of intertask communication, it is expected to run faster on an SP system with a switch than on an otherwise equivalent cluster of RS/6000 machines.

Parallel Environment supports the MPI-1.2 Standard API for message passing between tasks of a parallel application. Message passing by threaded application programs is also supported. There is some overhead for threads that will make single thread jobs, running on a uniprocessor slower with the thread library than with the non-threaded library. Programs written to exploit concurrent communication and calculation, and running on Symmetric Multiprocessing nodes (SMP), may run faster with the thread library. On an SMP node, this can allow an application with multiple tasks that can run in parallel to exploit multiple processes.

Parallel environment is being expanded in this release to support some portions of the MPI-2 Standard: a portion of MPI-IO and two new MPI\_Datatype accessor constructors. The new functions of MPI-IO will allow a file to be viewed as an object shared by several tasks and made up of a sequence of units described by an MPI\_Datatype. Refer to 11.7, "MPI I/O Subset" on page 312 for details on MPI-IO. The MPI-2 enhancements are supported only in the threaded version of the MPI library.

The MPL message passing library is supported on all node types, including SMP nodes, but does not support threaded applications.

Parallel Environment provides support for compiling and running application using the Low-Level API (LAPI) communication library. The LAPI and MPI libraries can both be used in the same program.

In the following sections we describe the enhancements provided for this new version of the IBM Parallel Environment for AIX.

---

## 11.1 Increased Tasks Limits Per Job

The new version of Parallel Operating Environment (POE), which is part of the IBM Parallel Environment for AIX software, provides an increased number of tasks that are supported in a single POE job to 2048 (1024 with the User Space MPI or LAPI libraries). Before Parallel Environment 2.4, it was 512 tasks.

For an MPI/IP job with N tasks, each task uses N+1 file descriptors for communication. In order to use more than 2000 file descriptors in AIX, there is a `ulimit` value (maintained in `/etc/security/limits`) that needs to be updated. The name of the field is `nfiles`, and the default value is 2000.

---

## 11.2 Multiple User Space Tasks Per Node

The SP communication subsystem (CSS) supports several different communication protocol implementations. The Internet Protocol (IP) is implemented as an interrupt-driven kernel mode device driver, while the Parallel Environment MPI is unconventionally implemented as a user mode process (termed User Space job).

The first implementation of a User Space protocol assumed an exclusive usage of the switch adapter (tb0) with one task per node at a time. It required the system to reboot when switching between switch IP jobs and user space parallel programs. Starting with the second switch generation adapter (tb2), and the current switch generation (tb3), this restriction is eliminated, allowing multiple protocols to work in the same adapter by having multiple send-recv buffer for different protocols.

Although the IP device driver supports multiple tasks via sockets and system-wide Memory Buffer (mbuf) implementation, the user space protocols were restricted to one per node until PSSP 2.3. Before PSSP 2.4, only one MPI user space process (parallel task) is permitted to run on one node. In the communication subsystem (CSS), each communication protocol instance is considered as one client to the kernel extension. The existing implementation associates each client with one Direct Memory Access (DMA) window, which is a system memory region mapped into I/O bus memory (accessible by the CPU on the adapter board) as a send and receive buffer. In PSSP 2.4, multiple windows are supported, but the window numbers are preallocated to MPI and LAPI by default. With PSSP 3.1, multiple windows are supported with no pre-allocation, but each window can (only) be used by one parallel task (not including the processes spawned by this task). So multiple processes can share one adapter to do message passing.

Figure 158 shows a diagram illustrating how the multiple user space tasks per adapter is implemented.

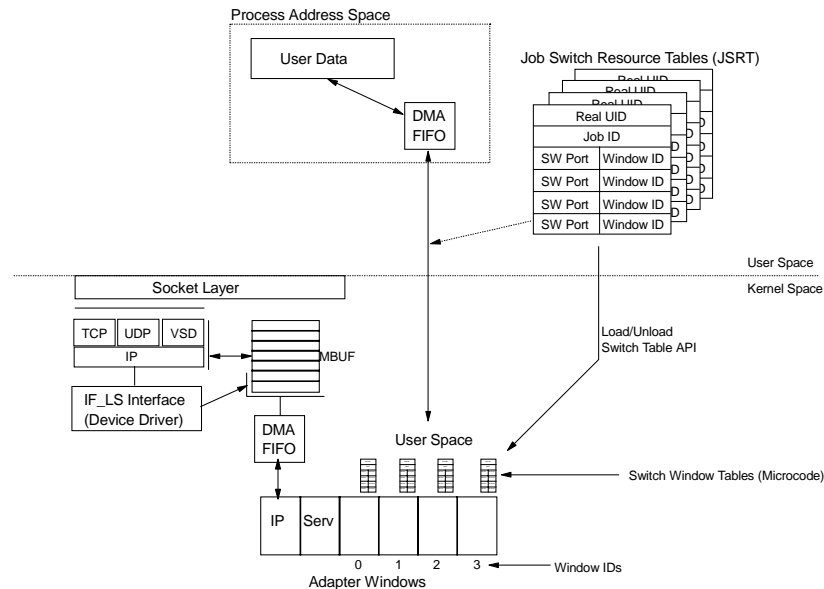


Figure 158. Multiple User Space Tasks Per Node

In this implementation, the adapter can handle several windows. One of them is allocated to IP traffic. The TCP/IP protocol implementation in AIX allows multiple tasks using IP protocol by having multiple queues. This implementation is very similar to other adapters, except for the fact that the switch adapter does not use mbuf memory for other than keeping the mbuf headers. The actual queues for send and receive through the switch are mapped to different memory segments (called switch pools) and they are not part of the mbuf structure.

IP protocol uses only one of the adapter windows for send and receive. The other adapter windows are used by the fault services daemon (switch daemon) and the user space protocols.

User space protocols use a different approach for providing multiple tasks. These protocols have direct access to the adapter windows. Four adapter windows are allocated to user space protocols.

To keep track of which task is using which window, a Switch table is maintained. This switch table maps the task, switch node number (switch port) and the window ID. In PSSP 2.2, switch tables are handled implicitly by

CSS. One major difference in PSSP 3.1 is that LoadLeveler and POE must handle the window numbers in the switch tables explicitly.

Before PSSP 2.4, since there is only one window available for user space processes on an adapter, there is no need to handle window numbers explicitly, the window number for a user space process on an adapter is known by default. In PSSP 2.4, there are two windows for user space processes on an adapter, but these windows are assigned to MPI and LAPI respectively by default, so no window numbers have to be handled explicitly. In PSSP 3.1, multiple windows are available for user space processes on an adapter, each user space process can use any (but only one per protocol) available window on the adapter. So the windows on an adapter have to be handled (allocated and deallocated) explicitly.

Let us say that we have an application or job that has six tasks and they use MPI/US protocol for communication, and we want to run this job in five nodes. In PSSP releases prior PSSP 3.1, we can not run all the tasks simultaneously or in parallel since previous releases of PSSP, multiple windows per node were not supported. Even though your five nodes could have been SMP nodes, and the number of processors (not nodes) could have been greater than the number of tasks.

With PSSP 3.1 you can run those tasks all in parallel (determine by the AIX scheduler, and if you have enough processors), and you do not allocate more than four tasks running in user space per node.

Assuming that someone (it may be you or the resource allocator you are using, such as LoadLeveler) has assigned the nodes and windows for your job. Let us say that you have the following distribution:

- Task 0 -> Node 1, Window 0
- Task 1 -> Node 7, Window 3
- Task 2 -> Node 8, Window 1
- Task 3 -> Node 9, Window 0
- Task 4 -> Node 10, Window 2
- Task 5 -> Node 10, Window 3

Which is organized into a list ordered by task ID. It does not matter "who" decided this distribution, at least not for this discussion.

Once this Switch Table<sup>1</sup> is built, it has to be loaded into the switch adapter. If you are using LoadLeveler, you do not need to worry about any of this because LoadLeveler takes care of all the details.

<sup>1</sup> The Switch Table API is documented in PSSP Command and Technical Reference, SA22-7351



But, if you are not using LoadLeveler, or any other scheduler who may load the table, you have to do it yourself. To do that, PSSP provides you with an API (called Switch Table API), which allows you to load and unload these switch tables. There is sample code provided with Parallel Environment 2.4, that shows you how to load and unload these tables. The sample code is located in /usr/lpp/ppe.poe/sample/swtbl (you will need a C compiler).

Once the table is loaded into the adapter, the adapter "knows" how to handle the communication between your tasks.

Figure 159 shows how the adapter handles these communication windows.

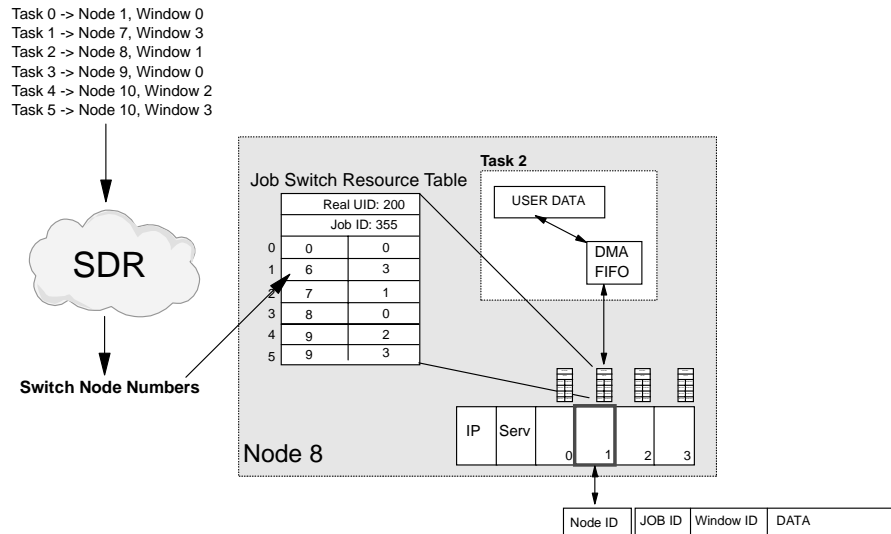


Figure 159. How the Adapter Handles These Communication Windows

Although the Job Switch Resource Table uses switch port numbers, the list of tasks presented in Figure 159 contains node numbers. These node numbers have to be converted to switch port numbers in order to load the switch table. The switch port use by a node is specified in the switch\_node\_number file located in the /spdata/sys1/st directory on the node. Another way to determine the switch port used by your nodes is by querying the SDR, using the SDRGetObjects command, as shown in Figure 160.

```
[root@sp4en0]# SDRGetObjects Syspar_map
syspar_name syspar_addr node_number switch_node_number used node_type
sp4en0 192.168.4.130 1 0 1 standard
sp4en0 192.168.4.130 2 1 0 standard
sp4en0 192.168.4.130 3 2 0 standard
sp4en0 192.168.4.130 4 3 0 standard
sp4en0 192.168.4.130 5 4 1 standard
sp4en0 192.168.4.130 6 5 1 standard
sp4en0 192.168.4.130 7 6 1 standard
sp4en0 192.168.4.130 8 7 1 standard
sp4en0 192.168.4.130 9 8 1 standard
sp4en0 192.168.4.130 10 9 1 standard
sp4en0 192.168.4.130 11 10 1 standard
sp4en0 192.168.4.130 12 11 0 standard
sp4en0 192.168.4.130 13 12 1 standard
sp4en0 192.168.4.130 14 13 0 standard
sp4en0 192.168.4.130 15 14 1 standard
sp4en0 192.168.4.130 16 15 0 standard
```

Figure 160. Getting the Switch Port Numbers from the SDR

In this case the switch node number and the node number differ by one (switch node number = node number - 1). However, with multiple frames, this is not true.

In the figure, *Task 2* runs on Node 8. It uses window number 1 in that node. The Job ID is 355 and the user ID is 200. When this task wants to send a message to any of the other tasks in the job, the task number is used to index the switch table in the adapter, and to obtain the switch node number, and the window ID. The microcode in the switch adapter will use the switch node number to send the packet to the target node. This packet contains the Job ID, Window ID and the data. At the target node, the adapter will receive the packet, and it will read the table associated to the window and it will compare the job ID within the packet with the job ID in the window. If they match, it will make the data available to the target task through the DMA FIFO connected to the address space for that task. The data in the DMA area is copied to the user buffer whenever MPI has a chance, such a when there is a MPI call, or the timer indicates that the DMA area has to checked.

So, in summary, before a job can use a user space protocol, its switch table has to be loaded into the adapter. There are basically three scenarios that can be presented here:

- POE jobs (both batch and interactive) run LoadLeveler, and it is LoadLeveler who loads the switch table for the corresponding job.

- POE uses a switch table loaded by the user to run jobs without LoadLeveler or other scheduler (see 11.8, “MUSPPA-lite” on page 315 for details about how to use the switch table API).
- POE jobs run under a third party scheduler which uses the switch table API without LoadLeveler.

POE interacts with LoadLeveler through the Job Management API which provides the window information to POE. For more information about LoadLeveler, refer to Chapter 10, “LoadLeveler Version 2.1” on page 259.

The state of each window is maintained in a file stored in /spdata/sys1/st/st\_datafile<id> on each node, where <id> is the window ID. If there are four available windows and only windows 1 and 2 are ever loaded, then there will only be st\_datafile1 and st\_datafile2. These files will never be removed from the system once they are created. The files contain the current status of the window, be it loaded or unloaded.

Everytime there is a request for loading a switch table, the inetd daemon will start a switchtbl process (with root permissions) which will do the operation. After the request has been satisfied, the switchtbl daemon will be terminated.

---

### 11.3 POE and Job Management

In previous versions of Parallel Environment, POE relied on the SP Resource Manager (RM), for keeping track of node and resource utilization (this is the ssp.jm component). Resource Manager was also in charge of loading the switch tables when new jobs were allocated to nodes.

The Resource Manager functionality is now integrated into LoadLeveler (version 2.1), which has been used for batch job submission in the past. Now it also processes the interactive jobs coming from POE. However, Resource Manager will still be used to serve nodes running previous levels of PSSP and PE. Although the compatibility is guaranteed, this job will not be able to use the new functionality, such as four adapter windows per node. Nodes running previous versions of PSSP are able to have only one user space task using both MPI and LAPI.

There five environment variables that, collectively, determine how nodes are allocated by the Partition Manager (the process that handle node allocations within POE). While these are the only ones you must set to allocate nodes, there are many other variables that you can set. See 11.6, “New Environment Variables” on page 310 for more information on the new environment

variables available in Parallel Environment 2.4. The environment variables for node allocation are:

<b>MP_HOSTFILE</b>	Specifies the name of the host list file. If not set, POE looks for a file called host.list in the current directory. Valid value is a file pathname.
<b>MP_RESD</b>	Specifies whether or not the Partition Manager should connect to a job management system (LoadLeveler or Resource Manager) to allocate nodes. Valid values are Yes or No.
<b>MP_EUILIB</b>	Specifies the communication protocol to use. Valid values are <i>ip</i> for TCP/IP communication, or <i>us</i> for user space protocol.
<b>MP_EUIDEVICE</b>	Specifies the adapter you want to use for IP communication. The switch IP device is css0. This variable is used only if the MP_RESD variable is set to Yes.
<b>MP_RMPOOL</b>	Specifies the pool identifier in LoadLeveler. This variable is only use if MP_RESD is set to Yes. Valid values are pool identifiers on LoadLeveler or Resource Manager (for PSSP levels prior PSSP 3.1).

### 11.3.1 Differences Between LoadLeveler and Resource Manager

LoadLeveler and Resource Manager differ in the:

**Pool specifications** - With RM, pools are specified with a pool number. With LoadLeveler, pools are alphanumeric strings.

**Host list file entries** - LoadLeveler ignores adapter and CPU specifications in the host list file. When using LoadLeveler, you can request how nodes are used with the MP\_CPU\_USE and/or MP\_ADAPTER\_USE environment variables, or their associated command line options.

#### Important

If you are using LoadLeveler, all the tasks must use the same pool, so the pool identifier specified in the host list file must be the same.

**Semantics of Usage** - Specifying dedicated adapter usage or unique CPU with LoadLeveler only prevents tasks of other parallel jobs from using the

resources. It does not prevent tasks from the *same* parallel job from using the resources.

---

## 11.4 User-Initiated Parallel Checkpoint/Restart

With the help of LoadLeveler, PE provides a mechanism for temporarily saving the state of a parallel program at a specific point (checkpointing), and then later restarting it from the saved state. When a program is checkpointed, the checkpointing function captures the state of the application as well as all data, and saves it in a file. If the program is restarted, the restart function retrieves the application information from the file it saved, and the program then starts running again from the place at which it was saved.

A user may initiate a checkpoint sequence from within a parallel MPI program by calling the `mp_chkpt()` function. All tasks in the parallel job must issue the call, which does not return until the checkpoint files have been created for all tasks. If the job subsequently fails and is restarted, the restart returns from the `mp_chkpt()` function with an indication that the parallel job has been restarted.

### 11.4.1 Limitations

Programs using the signal handling (non-threaded) MPI library may be linked as a checkpointable executable, which is run as a LoadLeveler batch job. LoadLeveler 2.1 or later is required. Restrictions on the program follow:

- For some processes, it is impossible to obtain or recreate the state of the process. For this reason, you should only checkpoint programs with states that are simple to checkpoint and recreate. A program that is long-running, computation-intensive, and does not fork any processes is an example of a job that is well-suited for checkpointing.
- In order to prevent unpredictable results from occurring, checkpointing jobs should not use the following system services:
  - Administrative (audit and swapqry, for example)
  - Dynamic loading
  - Forks
  - Internal timers
  - Messages
  - Semaphores
  - Set user ID or group ID
  - Shared memory
  - Threads

Another limitation of checkpointing jobs is file I/O. Because individual write calls are not traced, the file recovery scheme requires that all I/O operations, when repeated, must yield the same result. A job that opens all files as read-only can be checkpointed. A job that writes to a file and then reads the data back can also be checkpointed. An example of I/O that could cause unpredictable results is: reading an area of a file, writing to it, and then reading the same area of the file again.

You can only checkpoint POE and MPI applications that are submitted under LoadLeveler in batch mode; PE does not support checkpointing of interactive POE applications.

It is important to note that since the checkpointing library is part of LoadLeveler, and only POE batch jobs submitted with LoadLeveler are supported, LoadLeveler is required for checkpointing parallel programs.

**Important**

In the current release of LoadLeveler (2.1) and Parallel Environment (2.4), only user-initiated parallel checkpoints are supported. Only non-threaded programs may be checkpointed. See Parallel Environment documentation for additional limitations

### 11.4.2 How Checkpointing Works

Checkpointing occurs when an application calls the PE function `mp_chkpt()` in each task. Note that a program that calls `mp_chkpt()` must first be compiled with one of the POE checkpoint compile scripts (`mpcc_chkpt`, `mpCC_chkpt`, or `mpxlf_chkpt`). Before you submit the application, you first need to set the `MP_CHECKDIR` and `MP_CHECKFILE` POE environment variables to define the path name of the checkpoint file.

During checkpoint processing, each task executes the application, up to the point of the `mp_chkpt()` function call. At that point, the state and program data are written to the checkpoint file, which you defined with the `MP_CHECKDIR` and `MP_CHECKFILE` environment variables. The tasks then continue to execute the application. If the application is restarted, the `MP_CHECKDIR` and `MP_CHECKFILE` POE environment variables point to the checkpoint file that was previously saved. The application can be restarted on either the same or a different set of nodes, but the number of tasks must remain the same. The new nodes must also have identical system software level installed. When the restart function restarts a program, it retrieves the program state and data information from the checkpoint file.

Since large data files are often produced as a result of checkpointing a program, you need to consider the amount of available space in your filesystem. You should also consider the type of filesystem. Writing and reading checkpointing files may yield better performance on Journaled File Systems (JFS) or General Parallel File Systems (GPFS) than on Network File Systems (NFS), Distributed File Systems (DFS), or Andrew File Systems (AFS).

---

## 11.5 MPI Thread Compatibility

Parallel Environment 2.4 provides two MPI libraries: a non-threaded MPI library called `libmpi.a`, and a threaded MPI library called `libmpi_r.a` (both of them are located in the `/usr/lpp/ppe.poe/lib` directory).

The non-threaded MPI library is not affected by the new AIX thread structure, so it will not be discussed in this book. All applications compiled with previous versions of the non-threaded MPI library will be binary-compatible with the new AIX and PE versions.

However, for the threaded MPI library, there are some considerations to make in order to support applications compiled with previous versions of this library for running them in binary compatibility.

First, let us take a look to the thread structure for a MPI-POE job, as shown in Figure 161.

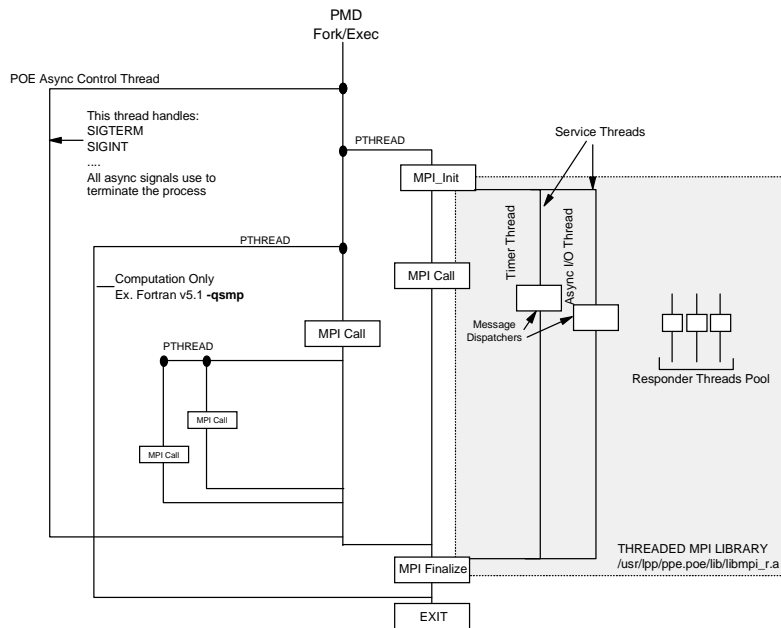


Figure 161. Thread Structure of a MPI-POE Task

In the figure, the Partition Manager Daemon (PMD), fork() and then exec() the POE job that has to be run in this node. Before calling the user's main program, POE will create a thread (using the pthread\_create() call) called POE Asynchronous Control thread. This thread will handle all the asynchronous signals used to terminate this process, for example SIGINT.

After the application is started, it can create its own threads. Before using any MPI call from any thread, the application has to call the MPI\_init() to initialize MPI. This call can be made from any thread. The MPI\_init() call will initialize the protocol, and it will create two service threads (Timer and Async I/O threads). At most one of these threads will run at the time. The Timer thread replaces the SIGALRM signal handle, and the Async I/O thread replaces the SIGIO signal handle both present in the non-threaded MPI library.

When MPI uses User Space protocol, there is only one service thread that handles both the time and the asynchronous I/O with the adapter window.

The library also creates Responder threads that handle MPI-IO and non-blocking communications within the library. These Responder threads are only created when needed, but they are not destroyed once created. The library will create these MPI Responder threads dynamically and based on



requests. When needing a thread for asynchronous execution, the library will check if there is one of these Responder threads available; if there is none, the library will create one.

When finished with MPI communication, the application has to call `MPI_Finalize()`, which will terminate the service threads and will deallocate any MPI resource associated to this application. This call has to be made from the same thread that called `MPI_init()`.

### 11.5.1 Responder Threads

In order to understand how the MPI implementation has been enhanced in Parallel Environment 2.4, especially the threaded library and the non-blocking communications between tasks, we need to understand how the MPI implementation works.

The MPI API provides an interface for message-oriented communication between tasks. These MPI calls made from the application are then converted to Message Passing Client Interface (MPCI) calls, which are stream-oriented. These calls will finally communicate with the device-dependent layer, which is packet-oriented, in order to deliver or receive data.

The MPI layered structure is shown in Figure 162.

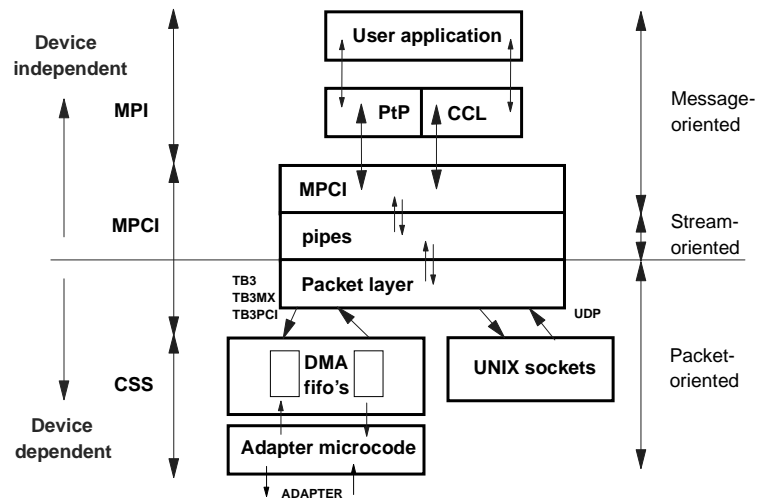


Figure 162. MPI Structure

MPCI supports non-blocking sends and receives. In normal use, a program will make a `MPI_Irecv` call and get back a request handle which represents the receive.

MPCI allows a handler (function pointer) to be attached to an `Irecv` or `Isend`. In normal use, like for `MPI_Irecv` or `MPI_Isend`, the pointer is `NULL` and so there is no special activity at the time MPCI recognizes a message has been sent or received. The next visible activity is when the user code calls `MPI_WAIT` with the handle returned by the `MPI_Isend/Irecv`. When there is a non-`NULL` pointer, the posted `Irecv` becomes a "posted responder". That means that when MPCI sees that the message has been matched, the responder is put on the responder queue and soon after, the function is called on a responder thread.

### 11.5.2 Threaded MPI Library Compatibility

The threaded MPI library in Parallel Environment 2.4 has changed, so in order to keep compatibility, and more important, binary compatibility for user executables, there are some issues that has to be considered.

Parallel Environment 2.3 requires PSSP 2.4, which requires AIX 4.2.1 or later. Applications compiled with Parallel Environment 2.3 are compatible with Parallel Environment 2.4. However, the thread structure in AIX has changed from AIX 4.3.1 or later.

The thread structure for AIX 4.3.0 and earlier is based on Draft 7 of the IEEE POSIX Thread Standard, while AIX 4.3.1 and later is based on IEEE POSIX 1003.1-1996 Thread Standard which is the final standard, and different from Draft 7.

This means that the AIX thread library (`libpthread.a`) has changed in AIX 4.3.1 and later, but AIX maintains binary compatibility across versions. AIX does that by having multiple shared objects in the library itself.

Executables built in AIX 4.3.0 or earlier, which links the thread library, reference the `shr.o` shared objects. Executables compiled in AIX 4.3.1 and later reference `shr_xpg5.o` shared objects. However, AIX 4.3.1 and later maintains both shared objects in the library, so "old" applications referencing `shr.o` objects will run without problems.

The thread library in AIX 4.3.1 and later contains both shared objects, as shown in Figure 163.

If an application, compiled in AIX 4.3.0 or earlier, runs in AIX 4.3.1 or later, it will reference the `shr.o` shared objects, which are present in the thread library.

The threaded MPI library in Parallel Environment 2.4 is compiled in AIX 4.3.1 which means it uses the shr\_xpg5.o shared objects. Applications compiled with previous versions of AIX and Parallel Environment will run, as long as mutexes (locks) and thread condition structures (signaling structures) are not shared.

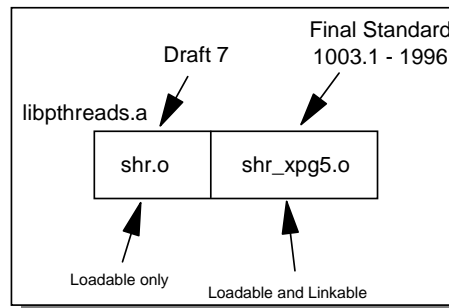


Figure 163. Structure of AIX Thread Library

The interesting case comes when part of the application has to be recompiled under AIX 4.3.1 and later. In that case, part of the application will use shr.o shared objects, and part will use shr\_xpg5.o shared objects. This is not guaranteed to run, mainly because it is likely that these recompiled parts of the application will share locks or signaling structures with "old" parts, which is not supported across these shared objects (shr.o and shr\_xpg5.o).

In order to recompile part of an "old" application, Parallel Environment provides a flag (-D7) in the mpcc\_r front end compiler to force the compilation to link the libpthreads\_compat.a which uses the shr.o shared objects.

### 11.5.3 AIX Thread Structure

AIX 4.3.1 and later has changed the default thread structure. In AIX versions previous to AIX 4.3.1, the mapping between user threads (pthreads) and kernel threads (kthreads) is 1:1 (also called System Contention Scope). This means that each user thread has a kernel thread allocated to it. The AIX Kernel Dispatcher (sometimes called Scheduler) allocates a kernel thread each time a user or process thread is created.

In AIX 4.3.1 and later, this thread structure has changed. The default is now M:N or Process Contention Scope, which means that the Kernel Dispatcher has a "pool" of kernel threads which are dynamically allocated to user or process threads when a user scheduling thread switches from one user thread to another, or when a user thread makes blocking system calls.

This new default in AIX brings some problems to multi-threaded parallel applications, because the kernel threads are allocated to process threads when they are available, which does not guarantee that your application will be able to execute all the threads you may want. However, in System Contention Scope, your threads have allocated a kernel thread each no matter what they are doing.

You can change this default to System Contention Scope by setting up an environment variable called `AIXTHREAD_SCOPE` to `S`. For applications compiled with Parallel Environment 2.4, this variable is not required because the MPI library changes each process scope thread which makes MPI calls into system scope. However it will not change the scope of a thread which does not call MPI. Applications executed with previous versions of Parallel Environment under AIX 4.3.1 or later need to set this variable to `S` (System Contention Scope) in order to change the default thread structure, since the previous versions of the MPI library does not change this.

---

## 11.6 New Environment Variables

POE behavior is controlled by environment variables. These variables must be set before the POE job gets executed. The POE command line can also be used to control POE by passing arguments instead of using environment variables.

The following list contains variables that are new in PE 2.4, and also variables that have been changed in order to support LoadLeveler as the job manager for interactive POE jobs.

**MP\_POLLING\_INTERVAL** - Sets the periodic packet transport check in microseconds. Default is 400,000 for User Space, and 180,000 for IP.

**MP\_CLOCK\_SOURCE** - Sets the time source for `MPI_Wtime`. Values are `AIX`, `SWITCH`. By default this variable is not set. Unless `AIX` is specifically set, `MPI_Wtime` will use the `SP` switch as clock source, if available. If `SWITCH` is set, MPI will report `MPI_WTIME_IS_GLOBAL` as `TRUE` and the job will fail if the switch clock is not available.

**MP\_MSG\_API** - Tells LoadLeveler if either MPI or LAPI will be used, so a window in the adapter has to be allocated. Default is `MPI`.

**MP\_CHECKFILE** - The base name for the checkpoint file.

**MP\_CHECKDIR** - The directory where the checkpoint file will reside.

**MP\_ADAPTER\_USE** - How the node's adapter should be used. If using LoadLeveler, the User Space communication subsystem library does not require dedicated use of the adapter on the node. The adapter use will be defaulted, as shown in Table 28, but shared usage may be specified. If using Resource Manager, this value is only used when POE is requesting non-specific nodes via the MP\_RMPOOL or -rmpool setting.

Table 28. Adapter/CPU Default Settings

	Adapter	CPU
If host list file contains non-specific pool requests	Dedicated	Unique
If host list file requests specific nodes	Shared	Multiple
If host list file is not used	Dedicated	Unique

If LoadLeveler is used, the meaning of this variable is as shown in Table 29.

Table 29. Adapter/CPU Use Under LoadLeveler

	If Node's CPU is "unique"	If Node's CPU is "Multiple"
Adapter is "Dedicated"	Intended for production runs of high performance applications. Only the tasks of that parallel job use the adapter and CPU.	The adapter specified with MP_EUIDEVICE is dedicated to the task. However, users still have access to the CPU through any other adapter.
Adapter is "Shared"	Only tasks of the same job have access to the node's CPU, but other job's tasks can shared the adapter.	Both the adapter and CPU can be used by a number of tasks from different jobs.

**MP\_CPU\_USE** - How the node's CPU should be used. If using LoadLeveler, the User Space communication subsystem library does not required unique CPU use on the node. CPU use will be defaulted, as in Table 28, but multiple use may be specified. If using Resource Manager, this value is only used when POE is requesting non-specific nodes via the MP\_RMPOOL or -rmpool setting.

**MP\_RESD** - This variable specifies whether or not the Partition Manager should connect to a job management system (LoadLeveler or Resource Manager) to allocate nodes. MP\_RESD only specifies whether or not to use a job management system. When the Resource Manager is used, the actual system to use is identified by the environment variable SP\_NAME.

When running POE from a workstation external to the LoadLeveler cluster, the *LoadL.so* fileset must be installed on the external node.

When running POE from a workstation that is external to the RS/6000 SP, and using the Resource Manager from PSSP 2.4, the *spp.clients* fileset must be installed on the external node.

**MP\_RMPOOL** - When using LoadLeveler, this variable specifies the pool name that should be used for non-specific node allocation. When Resource Manager is used, this variable specifies the pool number on the SP system pool that should be used for non-specific node allocation. This environment variable or command-line flag only applies to LoadLeveler or Resource Manager.

**MP\_NODES** - Specifies the number of physical nodes on which to run the parallel tasks. It may be used alone or in conjunction with **MP\_TASKS\_PER\_NODE** and/or **MP\_PROCS** variables. For more information about the use of these set of variables, refer to Table 7 on page 30 of the *IBM Parallel Environment for AIX: Operation and Use, Volume 1, SC28-1979*.

**MP\_TASKS\_PER\_NODE** - Specifies the number of tasks to be run on each of the physical nodes. It may be used alone or in conjunction with **MP\_TASKS\_PER\_NODE** and/or **MP\_PROCS** variables. For more information about the use of these set of variables, refer to Table 7 on page 30 of the *IBM Parallel Environment for AIX: Operation and Use, Volume 1, SC28-1979*.

**MP\_PMDSUFFIX** - When using LoadLeveler, this variable is used to determine a string to be appended to the normal partition manager daemon executable. The normal partition manager daemon executable specified is */etc/pmdv2*. By setting this variable, you can append a string to *pmdv2*. If **MP\_PMDSUFFIX** is set to *abc*, for example, the partition manager that gets run on each node of the parallel job is */etc/pmdv2abc*. When using Resource Manager, this variable is used to determine a string to be appended to the normal tcp service. The normal tcp service is called *pmv2*.

Using this variable with LoadLeveler or Resource Manager permits testing of alternate versions of the Partition Manager daemon. Typically, this environment variable is only used under the direction of the IBM Support Center in resolving PE-related problems.

---

## 11.7 MPI I/O Subset

The Message Passing Interface (MPI) standard provides an efficient way to communicate and synchronize multiple tasks, however, initial versions of this

standard do not provide support for parallel file I/O. Although applications may use a parallel file system to achieve this, the portability of these applications to other platforms and other file systems is not guaranteed at best.

Chapter 9 of the MPI-2 standard defines the set of MPI calls that allow parallel file I/O. This set of calls is called MPI-IO and a portion is being implemented as part of the threaded MPI library within Parallel Environment 2.4.

The MPI-IO implementation of the MPI-2 standard in Parallel Environment has been divided in two parts. Parallel Environment 2.4 includes most of the MPI-IO functions although not all. The decision as to which part of the subset needed to be implemented first was heavily based on customer requirements. For detailed information about the current implementation of the MPI-IO subset in Parallel Environment 2.4, refer to *IBM Parallel Environment for AIX: MPI Programming and Subroutine Reference*, GC23-3894. Future releases of the Parallel Environment for AIX will include the full implementation of the MPI-2 standard, including the MPI-IO functions.

The MPI-IO subset provides great flexibility to applications for defining the way how they will do their I/O. Tasks within an application can use MPI predefined and derived datatypes to partition the single file in multiple views. This allows to applications to partition the data and create their own access patterns based on these basic blocks or datatypes.

Figure 164 shows how multiple tasks can share a single file by using multiple views.

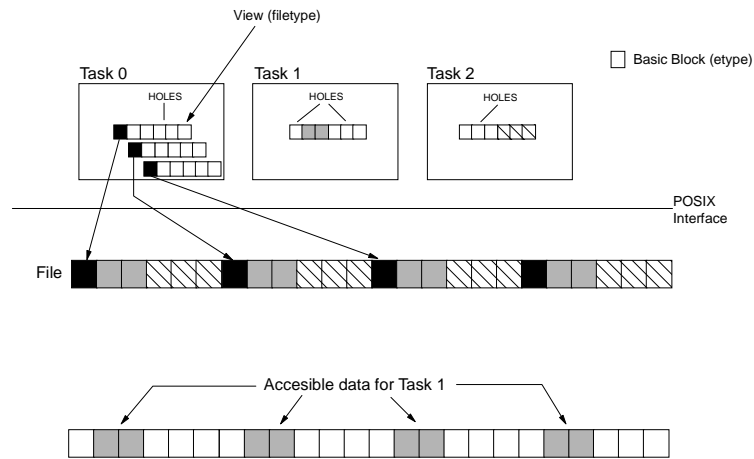


Figure 164. File Access Through MPI-IO

In Parallel Environment 2.4, MPI-IO requires GPFS for production use. The use of POSIX file systems that are not GPFS is severely restricted. A non-GPFS file system may be used only if all tasks are on a single node. This means that without GPFS, MPI-IO can be used only for certain program development. It would not be useful in production environments.

Figure 165 depicts the functional flow of the Parallel Environment 2.4 MPI-IO implementation. In this figure shows how applications, using MPI-IO, benefit from using the multinode/multidisk implementation of GPFS as the underlying file system.

With GPFS as the underlying file system, the multiple views from the different tasks can be handled by different VSD disk servers, which fully utilize the potential of the RS/6000 SP architecture.



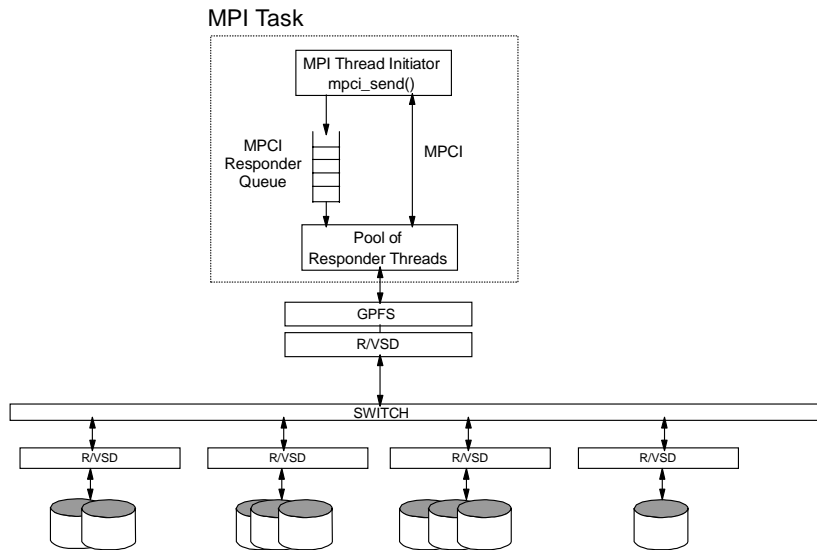


Figure 165. Functional Flow of MPI-IO Implementation

When an application makes an I/O request through the MPI-IO subset in one of the MPI threads, it is handled by sending a command to an MPI-IO posted responder. The MPCl responder queue is serviced on each MPI node by a pool of responder threads. This thread pool is created and managed by MPCl which creates responder threads as needed. Once created by MPCl, a responder thread remains alive and available for repeated use by MPCl.

MPI-IO active responders convert user I/O requests into system calls to the underlying file system. In the example, shown in Figure 165, the underlying file system is GPFS which provides the POSIX compliant interface needed by MPI-IO. GPFS in turn, takes the I/O request and gets or posts the data by using the Virtual Shared Disk (VSD) facility. Additionally, Recoverable Virtual Shared Disk (RVSD) can also be used if data availability is critical.

Communication and data transfer between user calls made on MPI threads and MPI-IO active responders are handled as MPCl send and receives. GPFS daemons communicate with the VSD servers responsible for disk access through IP.

## 11.8 MUSPPA-lite

This facility is part of POE and is shipped as a sample program to demonstrate how to use the Switch Table API. By using this, API programs

that do not want to use LoadLeveler or any other resource manager to load or unload the switch tables can use this API.

The sample code contains two C programs and a makefile file for compiling the programs. You will need IBM C for AIX Version 4.3 (5765-C64) or the IBM C and C++ Compilers Version 3.6 (5648-A81) compilers.

The sample can be found in the `/usr/lpp/ppe.poe/samples/swtbl` directory. Once compiled it provides two commands, one for loading and unloading the switch table, and another for starting POE applications.

#### **Important Note**

Users should exercise great care when running this sample code. A system administrator should carefully monitor the use of these programs, especially "swtbl\_api", which may be used to load and unload switch tables. It is suggested that these programs be used on a set of nodes that have been set aside for testing purposes.

A readme file can be found in the `/usr/lpp/ppe.poe/samples/swtbl` directory that explains how to use this facility.

---

## **11.9 Xprofiler Enhancements**

Xprofiler is a tool that helps you analyze your parallel application's performance. It uses data collected by the `-pg` compiler option to construct a graphical display of the functions within your application. Xprofiler provides quick access to the profiled data, which lets you identify the functions that are most CPU-intensive. The graphical user interface also lets you manipulate the display in order to focus on the application's critical areas.

Although Xprofiler uses the data generated for the AIX `gprof`, you do not need to be familiar with this command to use Xprofiler.

Xprofiler lets you profile both serial and parallel applications. The difference is that when you run a serial application, a single profile data is generated, while a parallel application produces multiple profile data files, one per each task in the parallel application.

The major enhancements in Xprofiler are:

- New options in the command line

- a Adds alternative paths to search for source code and library files, or changes the current path search order. When using this command line option, you can use the "at" symbol (@) to represent the default file path, in order to specify that other paths be searched before the default path.
  - c Loads the specified configuration file. If the -c option is used on the command line, the configuration file name with it will appear in the Configuration File (-c): text field in the *Load Files Dialog*. When both the -c and -disp\_max options are specified on the command line, the -disp\_max option is ignore, but the value that was specified with it will appear in the Initial Display (-disp\_max) field in the Load Files Dialog, the next time it is opened.
- Configuration file support
 

Xprofiler allows you to save the current configuration to a file for later use. It saves the names of the functions that are currently displayed. Later, in the same Xprofiler session or a different session, you can read in this configuration file using the Load Configuration option.
  - Screen dump capability
 

The File menu of the Xprofiler GUI includes an option called Screen Dump that lets you capture an image of the Xprofiler main window. You can send this image directly to the printer, or you can save it to a file in Postscript format. The default file name is Xprofiler.screenDump.ps.0.
  - Summary display mode
 

In summary mode, the size and shape of each function box is determined by the total CPU time of multiple gmon.out files used on that function alone, and the total time used by the function and its descendant functions. A function box that is wide and flat represents a function that uses relatively small amount of CPU on itself (it spends most of its time on its descendants). On the other hand, the function box for a function that spends most of the time executing only itself will be square-shaped.

Functions can also be represented in average mode. In average mode, the size and shape of each function box is determined by the average CPU time used on that function alone, among all loaded gmon.out files. The height of each function node represents the average CPU time, among all the input gmon.out files, used on the function itself. The width of each node represents the standard deviation of CPU time, among the gmon.out files, used on the function itself. The average mode representation is available only when more than one gmon.out file is entered.

- Undo option

Xprofiler allows you to undo operations that involve adding or removing nodes and arcs from the function call tree. When you undo an operation, you reverse the effect of any operation which adds or removes function boxes or call arcs to the function call tree.

Whenever you invoke the Undo option, the function call tree loses its zoom focus and zooms all the way out to reveal the entire function call tree in the main display.

- Other improvements
  - Case sensitive string search (Runtime option dialog)
  - New function node options
  - Minor GUI improvements

---

## 11.10 Message Queue Debugging

The message queue debugger is part of the pedb debugger interface. It is designed to help MPI application developers to debug internal message request information. This new feature allows you to view:

- A summary of the number of active messages for each task in the application. You can select criteria for the summary information based on message type and source, destination, and tag filters.
- Message queue information for a specific task.
- Detailed information concerning a specific message.

This facility only supports the threaded version of the MPI library, so developers writing applications using the non-threaded version of the MPI library cannot use this facility. Also, the version of the MPI library may not be supported by the version of the debugger. For more information refer to *IBM Parallel Environment for AIX: Operation and Use, Volume2*, SC28-1980.

---

## Appendix A. Changes to the SDR

The following are the changes we found. (Note that this may not be a complete list of all the changes.)

For more details, refer to *Administration Guide Version 3 Release 1*, SA22-7348, Appendix G.

### 1. Removed Classes:

EM\_Instance\_Vector:

EM\_Resource\_ID replaces EM\_Instance\_Vector . Number of attributes is unchanged. Names of attributes were started by iv (for Instance Vector) now replaced by ri (for resource id).

pmandConfig:

PMAN\_Subscription replaces pmandConfig. Old attributes "pmEventid" and "pmNodenum" are removed. Old attribute "pmActivate" is changed to "pmDeactivate". One new attribute is added, "pmInitEval".

### 2. New Classes:

EM\_Resource\_ID:

Used by RSCT (described in the Removed Classes part).

GS\_Config:

Used by RSCT to store the version of Group Services installed on each node.

Network:

Not currently used (created for future release support)

NodeControl:

Used by the PSSP to store information about the control it can have on the node based on architecture. Introduced to support external node diversity.

PMAN\_Subscription:

Used by PMAN (described in the Removed Classes part).

ProcessorExtensionNode:

Undocumented

RVSD\_Restrict\_Level:

Used by RVSD to manage RVSD operational level in case of coexistence.

Tec\_Agent\_Class:

Used by the T/EC Adapter to store events that will be forwarded to the TEC if triggered.

TS\_Tunable:

Not currently used (created for future release support)

Volume\_Group:

Used by the PSSP to manage the Alternate Volume Group function.

### 3. New attributes of existing classes:

Table 30. New Attributes in Adapter Class

<i>Attribute Name</i>	<i>Description</i>
enet_rate	Ethernet adapter rate (10, 100, auto).
duplex	Ethernet adapter duplex (full, half, auto).

Table 31. New Attributes in Frame Class

<i>Attribute Name</i>	<i>Description</i>
hardware_protocol	Type of hardware to be controlled: SP, SAMI.
s1_tty	The tty port used for serial (s1term) communications.

Table 32. New Attributes in SP Class

<i>Attribute Name</i>	<i>Description</i>
cw_dcehostname	Name by which DCE knows the control workstation.
sec_master	Not currently used.
cds_server	Not currently used.
cell_name	Not currently used.

Table 33. New Attributes in Node Class

<i>Attribute Name</i>	<i>Description</i>
hardware_control_type	Type of hardware control this node has.
RVSD_version	Represents the RVSD version in VRFM format.
selected_vg	The root volume group to install next.

<i>Attribute Name</i>	<i>Description</i>
dcehostname	Name by which DCE knows the host.

Table 34. New Attribute in EM\_Condition Class

<i>Attribute Name</i>	<i>Description</i>
type	Indicates if this condition is a default condition. default = default condition blank = a user-created condition

Table 35. New Attributes in Syspar Class

<i>Attribute Name</i>	<i>Description</i>
auth_install	Set of authentication methods to be installed on nodes in this partition.
auth_root_rcmd	Set of root authorization files defined for this part.
auth_methods	Set of active authentication methods for this partition.

#### 4. RSCT Class Changes:

Table 36. EM\_Resource\_ID New Attributes

<b>Old Attribute Name</b>	<b>New Attribute Name</b>
ivResource_name	riResource_name
ivElement_name	riElement_name
ivElement_description	riElement_description

A new SDR attribute is added to EM\_Resource\_Monitor Class.

Table 37. EM\_Resource\_Monitor Class

<b>SDR Class Name</b>	<b>New Attribute Name</b>
EM_Resource_Monitor	rmNum_Instances

rmNum\_Instances specifies the number of instances of a resource monitor that can execute simultaneously. If not specified the number is 1. The maximum value of this attribute is 8. It is a attribute of type integer.

A new SDR attribute is added to EM\_Resource\_Class.

Table 38. EM\_Resource\_Class

SDR Class Name	New Attribute Name
EM_Resource_Class	rcInstance_limit

rcInstance\_limit is the maximum number of resource variable instances for this class that the Event Management subsystem will accept from a resource monitor.

Two new attributes are added to EM\_Resource\_Variable SDR Class.

Table 39. EM\_Resource\_Variable

Old Attribute Name	New Attribute Name
rvPredicate	rvExpression
rvIndex_vector	rvIndex_element

Note that rvExpression and rvIndex\_element are added to the definition of the EM\_Resource\_variable class. rvPredicate and rvIndex\_vector are still present in this class, but are no longer used after migration. For more information about new SDR classes, refer to Appendix A, "Changes to the SDR" on page 319.



---

## Appendix B. New Commands and Changes to Old Commands

In this appendix we draw your attention to significant new commands in PSSP 3.1 and also to changes in the way in which some of the existing commands work. This appendix is not a definitive list of all new and changed commands. For further information, refer to the *PSSP: Command and Technical Reference*, SA22-7351.

---

### B.1 New Commands in PSSP 3.1

Table 40. New Commands in PSSP 3.1

Command name	Purpose of new command
<b>New Event Management Command</b>	
haemqvar	Queries the event management subsystem and writes to standard output all the names and values of resource variables in the current SP domain.
<b>New Volume Group Related Commands</b>	
spbootlist	Sets the bootlist on a node based on the disks defined in the volume group object.
spchvgobj	Changes the attributes of a node's root volume group object in the SDR.
spmkgobj	Creates a new root volume group object in the SDR.
sprmvobj	Removes an existing volume group object from the SDR.
<b>New Mirroring Commands</b>	
spmirrorvg	Mirrors a node's root volume group.
spunmirrorvg	Unmirrors a node's root volume group.
<b>New Security Commands</b>	
chauthpar	Runs on the CWS. Makes the specified authentication methods active for the designated system partition.
lsauthpar	Lists the active authentication methods for a specified SP system partition.
spauthconfig	Called from rc.sp when a node boots. This command checks the SDR and sets the correct authentication method for the node. It also calls the upauthfiles script, which updates the node's authorization files.

Command name	Purpose of new command
spsetauth	Run only on the CWS. Sets the authentication methods to be installed on the control workstation and in the partitions. Creates the authorization files for the selected authentication methods.

## B.2 Changes to Old Commands in PSSP 3.1

Table 41. Changes to Existing Commands in PSSP 3.1

Existing Command Name	Changes
spbootins	Addition of -c flag. This specifies the name of the root volume group object for the target node. Removal of the -h flag. This was used to specify the hdisk numbers used for installation of the nodes.
spframe	Addition of -p flag. This specifies the hardware protocol that is used to communicate within the frame. If the frame is an SP-attached server, the protocol will be SAMI. Addition of -s flag. This specifies the serial port on the CWS that is used to communicate with an SP-attached server.
splstdata	Addition of -v flag. This displays node volume group information.
spmon	Removal of -g flag. Equivalent functionality to the spmon GUI is now provided in Perspectives.

---

## Appendix C. Special Notices

This publication is intended to help IBM Customers, Business Partners, IBM System Engineers, and other RS/6000 SP specialists who are involved in Parallel System Support Programs (PSSP) Version 3, Release 1 projects, including the education of RS/60000 SP professionals responsible for installing, configuring, and administering PSSP Version 3, Release 1. The information in this publication is not intended as the specification of any programming interfaces that are provided by Parallel System Support Programs. See the PUBLICATIONS section of the IBM Programming Announcement for PSSP Version 3, Release 1 for more information about what publications are considered to be product documentation

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 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, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

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

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM ("vendor") products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this

information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

Any performance data contained in this document was obtained in a controlled environment based on the use of specific data and is presented only to illustrate techniques and procedures to assist IBM personnel to better understand IBM products. The results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data in their specific environment. No performance data may be abstracted or reproduced and given to non-IBM personnel without prior written approval by Business Practices.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

The following document contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples contain the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

You can reproduce a page in this document as a transparency, if that page has the copyright notice on it. The copyright notice must appear on each page being reproduced.

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

IBM ®	AIX
BookManager	Global Network
ESCON	HACMP/6000
LoadLeveler	OS/390
[POWERparallel	RS/6000
S/390	SP
System/390	TURBOWAYS
VM/ESA	

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

Java and HotJava are trademarks of Sun Microsystems, Incorporated.

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

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

Pentium, MMX, ProShare, LANDesk, and ActionMedia are trademarks or registered trademarks of Intel Corporation in the U.S. and other countries.

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

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



---

## Appendix D. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

---

### D.1 International Technical Support Organization Publications

For information on ordering these ITSO publications, see “How to Get ITSO Redbooks” on page 331.

- Inside the RS/6000 SP, SG24-5145
- RS/6000 SP PSSP 2.4 Technical Presentation, SG24-5173
- RS/6000 SP PSSP 2.3 Technical Presentation, SG24-2080
- RS/6000 SP PSSP 2.2 Technical Presentation, SG24-4868
- RS/6000 SP High Availability Infrastructure, SG24-4838
- GPFS: A Parallel File System, SG24-5165

---

### D.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

CD-ROM Title	Subscription Number	Collection Kit Number
System/390 Redbooks Collection	SBOF-7201	SK2T-2177
Networking and Systems Management Redbooks Collection	SBOF-7370	SK2T-6022
Transaction Processing and Data Management Redbook	SBOF-7240	SK2T-8038
Lotus Redbooks Collection	SBOF-6899	SK2T-8039
Tivoli Redbooks Collection	SBOF-6898	SK2T-8044
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RS/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RS/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
RS/6000 Redbooks Collection (PDF Format)	SBOF-8700	SK2T-8043
Application Development Redbooks Collection	SBOF-7290	SK2T-8037

---

### D.3 Other Publications

These publications are also relevant as further information sources:

- *IBM Parallel System Support Programs for AIX: Installation and Migration Guide*, GA22-7347 (Nov, 1998)
- *IBM Parallel System Support Programs for AIX: Administration Guide*, SA22-7348 (Nov, 1998)
- *IBM Parallel System Support Programs for AIX: Diagnosis Guide*, GA22-7350 (Nov, 1998)
- *IBM Parallel System Support Programs for AIX: Messages Reference*, GA22-7352 (Nov, 1998)
- *IBM Parallel System Support Programs for AIX: Command and Technical Reference, Volume 1 and Volume 2*, SA22-7351 (Nov, 1998)
- *IBM Parallel System Support Programs for AIX: Managing Shared Disks*, SA22-7349 (Nov, 1998)
- *IBM RS/6000 SP Planning Volume 1, Hardware and Physical Environment*, GA22-7280
- *IBM RS/6000 SP Planning Volume 2, Control Workstation and Software Environment*, GA22-7281
- *RS/6000 Cluster Technology: Event Management Programming Guide and Reference*, SA22-7354 (Nov, 1998)
- *RS/6000 Cluster Technology: Group Services Programming Guide and Reference*, SA22-7355 (Nov, 1998)



---

## How to Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at <http://www.redbooks.ibm.com/>.

---

## How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Redbooks Web Site on the World Wide Web**

<http://w3.itso.ibm.com/>

- **PUBORDER** – to order hardcopies in the United States

- **Tools Disks**

To get LIST3820s of redbooks, type one of the following commands:

```
TOOLCAT REDPRINT
TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
```

To get BookManager BOOKs of redbooks, type the following command:

```
TOOLCAT REDBOOKS
```

To get lists of redbooks, type the following command:

```
TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
```

To register for information on workshops, residencies, and redbooks, type the following command:

```
TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1998
```

- **REDBOOKS Category on INEWS**

- **Online** – send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL

### Redpieces

For information so current it is still in the process of being written, look at "Redpieces" on the Redbooks Web Site (<http://www.redbooks.ibm.com/redpieces.html>). Redpieces are redbooks in progress; not all redbooks become redpieces, and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

---

## How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Online Orders** – send orders to:

	<b>IBMMAIL</b>	<b>Internet</b>
In United States	usib6fpl at ibmmail	usib6fpl@ibmmail.com
In Canada	caibmbkz at ibmmail	lmannix@vnet.ibm.com
Outside North America	dkibmbsh at ibmmail	bookshop@dk.ibm.com

- **Telephone Orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	(long distance charges apply)
(+45) 4810-1320 - Danish	(+45) 4810-1020 - German
(+45) 4810-1420 - Dutch	(+45) 4810-1620 - Italian
(+45) 4810-1540 - English	(+45) 4810-1270 - Norwegian
(+45) 4810-1670 - Finnish	(+45) 4810-1120 - Spanish
(+45) 4810-1220 - French	(+45) 4810-1170 - Swedish

- **Mail Orders** – send orders to:

IBM Publications Publications Customer Support P.O. Box 29570 Raleigh, NC 27626-0570 USA	IBM Publications 144-4th Avenue, S.W. Calgary, Alberta T2P 3N5 Canada	IBM Direct Services Sortemosevej 21 DK-3450 Allerød Denmark
--	--	--

- **Fax** – send orders to:

United States (toll free)	1-800-445-9269
Canada	1-800-267-4455
Outside North America	(+45) 48 14 2207 (long distance charge)

- **1-800-IBM-4FAX (United States) or (+1) 408 256 5422 (Outside USA)** – ask for:

Index # 4421 Abstracts of new redbooks  
Index # 4422 IBM redbooks  
Index # 4420 Redbooks for last six months

- **On the World Wide Web**

Redbooks Web Site	<a href="http://www.redbooks.ibm.com">http://www.redbooks.ibm.com</a>
IBM Direct Publications Catalog	<a href="http://www.elink.ibm.link.ibm.com/pbl/pbl">http://www.elink.ibm.link.ibm.com/pbl/pbl</a>

### Redpieces

For information so current it is still in the process of being written, look at "Redpieces" on the Redbooks Web Site (<http://www.redbooks.ibm.com/redpieces.html>). Redpieces are redbooks in progress; not all redbooks become redpieces, and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

---

## IBM Redbook Order Form

Please send me the following:

Title	Order Number	Quantity
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____

---

First name \_\_\_\_\_ Last name \_\_\_\_\_

Company \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ Postal code \_\_\_\_\_ Country \_\_\_\_\_

Telephone number \_\_\_\_\_ Telefax number \_\_\_\_\_ VAT number \_\_\_\_\_

Invoice to customer number \_\_\_\_\_

Credit card number \_\_\_\_\_

Credit card expiration date \_\_\_\_\_ Card issued to \_\_\_\_\_ Signature \_\_\_\_\_

**We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.**



---

## List of Abbreviations

<b>AIX</b>	Advanced Interactive Executive	<b>GS</b>	Group Services
<b>AMG</b>	Adapter Membership Group	<b>GSAPI</b>	Group Services Application Programming Interface
<b>ANS</b>	Abstract Notation Syntax	<b>GVG</b>	Global Volume Group
<b>API</b>	Application Programming Interface	<b>HACMP</b>	High Availability Cluster Multiprocessing
<b>BIS</b>	boot/install server	<b>HACMP/ES</b>	High Availability Cluster Multiprocessing Enhanced Scalability
<b>BSD</b>	Berkeley Software Distribution	<b>hb</b>	heart beat
<b>BUMP</b>	Bring-Up Microprocessor	<b>HIPS</b>	High Performance Switch
<b>CP</b>	Crown Prince	<b>hrd</b>	host respond daemon
<b>CPU</b>	central processing unit	<b>HSD</b>	Hashed Shared Disk
<b>CSS</b>	communication subsystem	<b>IBM</b>	International Business Machines Corporation
<b>CWS</b>	Control Workstation	<b>IP</b>	Internet Protocol
<b>DB</b>	database	<b>ISB</b>	Intermediate Switch Board
<b>EM</b>	Event Management	<b>ISC</b>	Intermediate Switch Chip
<b>EMAPI</b>	Event Management Application Programming Interface	<b>ITSO</b>	International Technical Support Organization
<b>EMCDB</b>	Event Management Configuration Database	<b>JFS</b>	Journaled File System
<b>EMD</b>	Event Manager Daemon	<b>LAN</b>	Local Area Network
<b>EPROM</b>	Erasable Programmable Read-Only Memory	<b>LCD</b>	liquid crystal display
<b>FIFO</b>	first-in first-out	<b>LED</b>	light emitter diode
<b>FS</b>	file system	<b>LP</b>	logical partition
<b>GB</b>	gigabytes	<b>LRU</b>	last recently used
<b>GL</b>	Group Leader	<b>LSC</b>	Link Switch Chip
<b>GPFS</b>	General Purposes File System	<b>LV</b>	logical volume
		<b>LVM</b>	Logical Volume Manager
		<b>MB</b>	megabytes

<b>MIB</b>	Management Information Base	<b>RSI</b>	Remote Statistics Interface
<b>MPI</b>	Message Passing Interface	<b>R/VSD</b>	Recoverable/Virtual Shared Disk
<b>MPL</b>	Message Passing Library	<b>RVSD</b>	Recoverable Virtual Shared Disk
<b>MPP</b>	Massive Parallel Processors	<b>SBS</b>	structure byte string
<b>NFS</b>	Network File System	<b>SCSI</b>	Small Computer System Interface
<b>NIM</b>	Network Installation Management	<b>SDR</b>	System Data Repository
<b>NSB</b>	Node Switch Board	<b>SMIT</b>	System Management Interface Tool
<b>NSC</b>	Node Switch Chip	<b>SSA</b>	Serial Storage Architecture
<b>OID</b>	object ID	<b>VG</b>	volume group
<b>ODM</b>	Object Data Management	<b>VSD</b>	Virtual Shared Disk
<b>PIADE</b>	Performance Aide for AIX		
<b>PE</b>	Parallel Environment		
<b>PID</b>	process ID		
<b>PP</b>	physical partition		
<b>PSSP</b>	Parallel System Support Programs		
<b>PTC</b>	prepare to commit		
<b>PTPE</b>	Performance Toolbox Parallel Extensions		
<b>PTX</b>	Performance Toolbox for AIX		
<b>PV</b>	physical volume		
<b>RAM</b>	random access memory		
<b>RCP</b>	Remote Copy Protocol		
<b>RM</b>	Resource Monitor		
<b>RMAPI</b>	Resource Monitor Application Programming Interface		
<b>RPQ</b>	Request For Product Quotation		

---

## Index

### Symbols

.klogin 239  
.rhosts 238  
/etc/objrepos 237  
/etc/sysctl.acl 114  
/etc/sysctl.pman.acl 96  
/etc/sysctl.vsd.acl 114  
/var/adm/csd directory 222

### Numerics

7133 240  
7135-110 240  
7135-210 240  
8-Port Asynchronous PCI Adapter 3  
9333 240

### A

abbreviations 335, 336  
ACD 237  
acronyms 335, 336  
Action  
    Configure VSDs 119  
    Create VSDs 117  
    Filter Related Objects 122  
    Set Monitoring 122  
    vsd nodes 116  
Action page 111  
AIX 4.3.2 1  
AIX error log 104  
AIXLINK 4  
Aixos Resource Monitor 202  
Allocation 28  
alternate rootvg 1  
ATM 4, 228  
autojoin attribute 161

### B

block size 246  
bootlist 52  
bootp\_response 28  
bosinst.data 69  
Bubble Help 72

### C

CAT5 4  
Central Electronics Complex (CEC) 126  
Central Manager  
    see LoadLeveler  
Checkpointing  
    see LoadLeveler  
cl\_setup\_kerberos 238  
claddnetwork 235, 243  
clconvert 233  
cldomain 242  
clhandle 242  
clmixver 243  
Cluster 183  
cluster manager 237  
Coexistence 62  
Concurrent Capable 240  
Conditions panes 100  
Configure VSDs 119  
connwhere 45  
Controlling Hardware  
    see Hardware Perspective  
Create VSDs 117  
CRM 231  
    Concurrent Access 239  
css.logevnt 174  
css.snap 173, 179  
css.summlog daemon 173  
cssadm daemon 167  
cssadm.cfg file 169  
cssadm.debug file 170  
cssadm.stderr file 170  
cssadm.stdout file 170  
CSU 4  
customization 29  
CWS 20

### D

DARE 234, 237  
DB25 3  
DB78 3  
DCD 237  
Direct Memory Access 296  
DMA 296  
Domain 183, 227  
DSU 4  
Dual Daemons 184

## E

- Efence -autojoin command 161
- EIA-232 3
- element name 98
- element value 98
- EMAPI 200
- EMIF 3
- Enhanced Security 238
- Enterprise Server 125
- Error ID 108
- Error label 108
- ESCON Adapter 3
- Event activating 103
- Event Definitions 98
- Event expression 103
- Event Management 183, 199
- Event Management Application Programming Interface 200
- Event Management Client 200
- Event Manager 153
- Event Manager Daemon 200
- Event Perspectives 95
- Event rearming 103
- expression 98, 200

## F

- FDDI MAC Address 230
- fence registers 240
- fencevsd command 251
- fencing 240
- field name 98
- field value 98
- Filter 72
- Filter Related Objects 122
- firstboot.cust 29
- Fixed resource ID elements 103
- frame to frame 132

## G

- Global network 234
- Global ODM 239
- GODM
  - Global ODM 231
- GPFS cluster.preferences file 249
- GPFS configuration 254, 256
- GPFS Configuration Manager 248
- GPFS copy set 250
- GPFS descriptor file 252

- GPFS disk descriptor 252
- GPFS Failure Group 252
- GPFS file system integrity 247
- GPFS mallocsize 253
- GPFS Metadata Manager 249
- GPFS pagepool 253
- GPFS preallocation of disk space 255
- GPFS quorum 248, 256
- GPFS replication 247
- GPFS Stripe Group Manager 248
- GPFS Token 249
- GPFS Token Manager 250
- GPFS Token Manager Server 250
- GPFS VSD descriptor 252
- Group Services 183, 233

## H

- ha.vsd refresh command 222
- HA\_DOMAIN\_NAME 213
- HA\_DOMAIN\_TYPE 213
- HACMP 183
- HACMP Dependencies 231
- HACMP domain 184
- HACMP Packaging 231
- HACMP Restrictions 232
- HACMP/6000 232
- HACMP/6000 Migration 232
- HACMP/ES 184, 227
- HACMP/ES migration 233
- HACMPcluster 234
- HACMPcluster ODM 233
- haemd 200
- haemloadlist 179
- HAI 183
- Hardware Perspective 79
  - adding a pane 81
  - control operations 84
  - current pane 83
  - filter by monitored state 93
  - icon view 80
  - LED values 95
  - monitoring hardware 88
  - notebook 86
  - objects 80
  - panes 80
  - selecting objects 84
  - sphardware command 80
  - state 93



- table view 83, 91
- hasInactiveIBMVSds 114
- Heartbeats 234
- High Availability Infrastructure 183
- High Performance switch (HiPS) 133
- HiPS 7
- HiPS support 2
- hmcmds 100

## I

- I/O rack 126
- IBM 113
- IBM HSDs 113
- IBM VSDs 113
- IBM.PSSP.CSSlog resource class 174
- IBM.PSSP.CSSlog.errlog resource variable 174
- IBM.PSSP.CSSLogMon resource monitor 174
- Icon state 101
- IEEE POSIX 1003.1-1996 308
- inetd subserver 239
- install\_agent 8
- instance vector 98
- Inter Process Communication 201
- IPC 201
- ISS 293

## J

- Job
  - see LoadLeveler

## K

- kernel extension 251
- kernel heap storage 251
- krb-srvtab 238

## L

- LANE 4
- LAPI 295
- Launch 74
- Launch Pad
  - Adding applications 76
  - Application details 78
  - customize mode 76
  - Location of profiles 75
  - Predefined icons 78
- libgpfs.a 255
- libgpfs.a library 255

- lsubmit 271
- LoadLeveler 259
  - administration file 277
  - administration manual 259
  - backfill algorithm 266
  - central manager 261
  - checkpointing 265
  - checkpointing enhancements 291
  - cluster 259
  - configuration file 280
  - configuring 273
  - daemons 263
  - GUI 282
  - installation 273
  - job command file 268
  - job step 260
  - migration 293
  - parallel jobs 266
  - POE 289
  - resource manager 290
  - scheduler 261
  - scheduling 266
  - starting 282
  - submitting a job 286
  - SYSPRIO 262
- logevnt.out file 177
- LPP 2

## M

- MAC address takeover 230
- Manual Conditioning 31
- mbuf 296
- migration 53
- migration protocol 233
- Mirroring rootvg 41
- mmaddisk command 253
- mmaddnode command 256
- mmchfs command 253, 256, 257
- mmconfig command 253, 257
- mmcrfs command 252, 253, 257
- mmdelnode command 256, 257
- mmfsd daemon 247
- MmfsGroup group 248
- mmlsfs command 256
- Monitor VSDs 121
- Monitoring Hardware
  - see Hardware Perspective
- MPCI 307

MPI 1, 295  
MPI-1.2 295  
MPI-IO 255  
MPL 295  
Multiple rootvg 41, 63  
Multiprotocol PCI Adapter 4  
MUSPPA 296

## N

NCT 186  
Network Connectivity Table 186  
NIM 17  
NLS 1  
Node Object 43  
Notebook actions 103  
Notebooks 72  
Notification 101

## P

Packaging 23, 34  
PAIDE 186  
Panels 71  
perfagent 56  
perfagent.tools 21  
Performance Toolbox 201  
Performance Toolbox Parallel Edition 186  
Perspectives 31, 71  
POE 289, 296  
posted responder 308  
POWER3 2  
predicate 98  
Preferences 72  
Profiles 75  
    Saving 75  
    System profiles 75  
    User profiles 75  
PSSP 183  
PSSP 3.1 1  
PSSP domain 183  
PTPE 186, 201  
PTX 201  
PTX Performance Agent 186  
Pull 27  
Push 27  
PVC 4

## R

R/VSD 115  
RAIDiant 240  
rc.switch script 165  
Realm 184, 228  
Rearm expression 103  
rearm expression 98  
rearm predicate 98  
register events 96  
Registration 101  
Reliable Messaging 186  
Resource 290  
Resource Center 1  
resource group 240  
Resource ID 101  
resource identifier 98  
Resource Monitor Application Programming Interface 200  
Resource Monitors 200  
Resource Variable 108  
Resource variable 103  
Resource Variable Description 108  
resource variable instance 200  
responder thread 307, 308  
RMAPI 200  
rootvg 1  
route 234  
RS/6000 Cluster Technology 183  
RS-422 3  
RSCT 183  
RSCT integration 98  
RVSD 219  
RVSD fileset 221  
RVSD\_Restrict\_Level class 224  
rvsdInRecovery 114  
rvsdrestrict command 224

## S

S70 125  
S7A 125  
SCD 238  
Scheduling  
    see LoadLeveler  
script.cust 29  
SCSI-2 5  
SDLC 4  
SDR 18  
Service and Manufacturing Interface (SAMI) 131

- setup\_server 27
- Shared Memory Segment 201
- snapshot 232, 233
- SP-attached servers 125
- spbootins 48
- spbootlist 52
- spchvgobj 46, 47
- spcn 153
- SPCNhasMessage 153
- spevent 71
- sphardware 71
- splstdata 51, 154
- SPMI 186, 201
- spmirrorvg 49
- spmkgobj 44
- spmon 153
- SPMON Equivalence 73
- spmon functionality 73
- spmon -g 71
- spmon GUI 73
- SPOF 227
- spsyspar 71
- spunmirrorvg 50
- spvsd 71
- spvsd command 114
- src 153
- SRChasMessage 153
- SSA disks 65
- st\_datafile file 301
- STP 4
- stripe group 246
- structured byte string 108
- summlog file 178
- summlog.out file 179
- SVC 4
- switch scan 162
- switchtbl 301
- swtadmd subsystem 169
- swtlog subsystem 178
- sysctl svcrestart 114
- SYSPRIO
  - see LoadLeveler
- System V 201

## T

- T/EC adapter 1
- Table Attributes 120
- tb0 296

- tb2 296
- TB3MX 132
- TBIC 162
- Tec\_Agent\_Class 8
- tecad\_pssp 8
- TME 10 8
- Topology DARE 231
- Topology Services 183
- Tunable heartbeat 231

## U

- unfencevsd command 251
- User Space 1
- UTP 4

## V

- Virtual Shared Disk 219
- Volume\_Group 42
- VSD 219
  - Actions menu 114
  - Filter 113
  - management 113
  - Perspective 113
  - Using 114
- VSD client node 219
- VSD Enhancements 113
- VSD fileset 220
- VSD Perspective 113
  - Control 120
  - Table Attributes 120
  - Table view 120
- VSD perspective fileset 221
- VSD Perspective Filter 120
- VSD server node 219
- VSD server primary node 219
- VSD server secondary node 219

## W

- WAN 4
- wrappers 27

## X

- X.25 4



---

# ITSO Redbook Evaluation

PSSP 3.1 Announcement  
SG24-5332-00

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please complete this questionnaire and return it using one of the following methods:**

- Use the online evaluation form found at <http://www.redbooks.ibm.com>
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to [redbook@us.ibm.com](mailto:redbook@us.ibm.com)

Which of the following best describes you?

**Customer**    **Business Partner**    **Solution Developer**    **IBM employee**  
 **None of the above**

**Please rate your overall satisfaction** with this book using the scale:  
**(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)**

Overall Satisfaction \_\_\_\_\_

**Please answer the following questions:**

Was this redbook published in time for your needs?      Yes\_\_\_ No\_\_\_

If no, please explain:

---

---

---

---

What other redbooks would you like to see published?

---

---

---

**Comments/Suggestions:      (THANK YOU FOR YOUR FEEDBACK!)**

---

---

---

---

