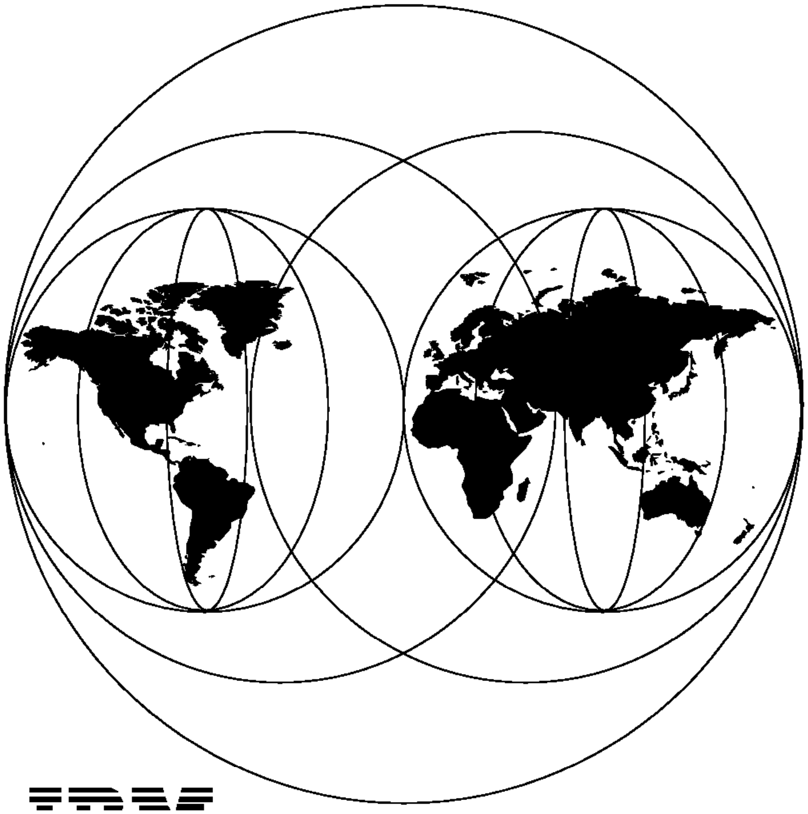
SG24-4868-00

RS/6000 SP PSSP 2.2 Technical Presentation

November 1996



IBM

International Technical Support Organization

Poughkeepsie Center

International Technical Support Organization

SG24-4868-00

RS/6000 SP PSSP 2.2 Technical Presentation

November 1996

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix A, “Special Notices” on page 309.

First Edition (November 1996)

This edition applies to PSSP Version 2, Release 2 for use with the AIX Version 4 Operating System.

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.

Contents

Tables . . . vii

Preface . . . ix
How This Redbook Is Organized . . . ix
The Team That Wrote This Redbook . . . x
Comments Welcome . . . xii

Chapter 1. PowerPC 604 High Nodes . . . 1
1.1 Introduction to SMP Concepts . . . 2
1.1.1 Multiprocessing Concepts . . . 2
1.1.2 SMP Architecture . . . 10
1.1.3 SMP Programming Model . . . 13
1.1.4 IBM SMP Design . . . 17
1.2 PowerPC 604 High Node . . . 22
1.2.1 SP PowerPC 604 High Node . . . 25
1.2.2 Supported Configurations . . . 31
1.3 PSSP Version 2 Release 2 and High Nodes . . . 39
1.3.1 High Nodes Installation Process . . . 41
1.3.2 High Node Boot Process . . . 43
1.3.3 System Monitoring and Control . . . 46
1.3.4 PSSP Resources and the High Nodes . . . 49
1.3.5 Conclusion . . . 52
1.4 System Management for SMP Nodes . . . 54
1.4.1 Overview . . . 56
1.4.2 System Management Infrastructure . . . 58
1.4.3 Administrative and Operations Interfaces . . . 65
1.4.4 System Administration and Operations . . . 66

Chapter 2. New Installation Methodology . . . 73
2.1 Agenda . . . 74
2.2 Design Objectives . . . 75
2.3 SP Installation, V2.2 . . . 77
2.4 Network Install Manager . . . 80
2.5 NIM Overview . . . 81
2.6 Setting Up Install Environment . . . 86
2.7 Wrappers . . . 87
2.7.1 delnimmast . . . 91
2.7.2 delnimclient . . . 92
2.7.3 mknimmast . . . 93
2.7.4 create_krb_files . . . 95
2.7.5 mknimint . . . 96
2.7.6 mknimres . . . 97
2.7.7 mknimclient . . . 98
2.7.8 mkconfig . . . 99
2.7.9 mkinstall . . . 100
2.7.10 allnimres . . . 101
2.7.11 unallnimres . . . 102
2.8 CWS Requirements . . . 103
2.9 Boot/Install Process . . . 104
2.10 pssp_script . . . 106
2.11 Recovering a Boot-Install Server . . . 107

2.12 Debugging NIM 108
2.13 The Frame Supervisor 109
2.13.1 The Old Situation 110
2.13.2 New with PSSP V2.2 111
2.13.3 The Supervisor Interface 112
2.13.4 SMIT and Supervisor 113
2.13.5 Supervisor Action Needed 115
2.13.6 The Supervisor Interface 116
2.13.7 The Supervisor Interface 118

Chapter 3. Software Coexistence 119
3.1 Table of Contents for Software Coexistence 121
3.2 Versions That Support Coexistence 122
3.2.1 Benefits 123
3.2.2 Coexistence versus Partitioning 124
3.3 Planning and Preparation for Coexistence 126
3.4 PSSP 2.2 Compared to PSSP 2.1 130
3.5 How Does PSSP Coexistence Work? 134
3.6 Unsupported Environment 135

Chapter 4. Migration Considerations 137
4.1 Overview 138
4.2 Reasons for Migration 139
4.3 Planning and Preparation for Migration 140
4.4 CWS Migration Steps 141
4.5 Migrate the CWS from AIX 3.2.5 to AIX 4.1.4 142
4.5.1 Migration from PSSP 1.2 to PSPP 2.2 143
4.5.2 Migration from PSSP 1.2 to PSPP 2.1 161
4.6 Migrating the CWS from PSSP 2.1 to PSSP 2.2 167
4.6.1 AIX Is at Level AIX 4.1.4 168
4.6.2 Create CWS System Backup 174
4.6.3 Restore System Backup on CWS 175
4.7 Node Migration 177
4.7.1 Preparation for Node Migration 179
4.7.2 Migrate 3.2.5 Node to AIX 4.1.4 181
4.7.3 Migrate 4.1.4 Node to PSSP 2.2 184
4.7.4 Preservation Installation for PSSP 2.2 186

Chapter 5. AIX 4.2 Support 189
5.1 Table of Contents 190
5.2 Why Use AIX 4.2? 191
5.3 Compatible Programs 193
5.4 Nodes with AIX 4.2 195
5.5 New Install with System Backup 198
5.6 Migration to AIX 4.2 200
5.6.1 Preparation for Node Migration 200
5.6.2 Node Migration for AIX 4.2 202
5.7 Year End Support 203

Chapter 6. Perspectives 205
6.1 Introduction to Perspectives 207
6.1.1 What Is Perspectives? 208
6.1.2 Design Objectives 209
6.1.3 Start Perspectives 211
6.2 Perspectives Launch Pad 212

6.2.1 Customizing the Launch Pad 213
6.2.2 Saving Preferences 215
6.2.3 Individual Perspectives User Interface 216
6.3 Hardware Perspective 218
6.3.1 Architectural Design 219
6.3.2 Available Information 224
6.3.3 Partition Driven 225
6.3.4 Node Grouping 226
6.3.5 Node Group Creation 228
6.3.6 Use of Node Groups 232
6.3.7 Node Grouping and Partitioning 234
6.3.8 Filtering 235
6.3.9 Object Label Modification 236
6.3.10 New Interfaced Functions 237
6.3.11 SP Health Definition 239
6.3.12 Monitoring in Action 241
6.3.13 Monitoring on the Desktop 242
6.3.14 Customizing Monitored Conditions 243
6.4 SPmon and Hardware Perspectives 248
6.4.1 Global Control 249
6.4.2 All Node Summary Display 250
6.4.3 Environmental Variables 251
6.4.4 Node Front Panel 252
6.4.5 3DigitDisplay 254

Chapter 7. System Partitioning Aid 255
7.1 System Partitioning Aid Overview 257
7.2 Introduction: System Partitioning 259
7.2.1 Partitioning Rationale 260
7.2.2 Different Expectations in System Partitioning 262
7.2.3 Partitioning Implementation 264
7.2.4 PSSP 2.1 Partitioning Limitations 266
7.2.5 PSSP 2.2 Solutions for System Management Flexibility 268
7.3 System Partitioning Aid Functions 270
7.3.1 System Partitioning Aid 271
7.3.2 System Partitioning Aid Infrastructure 274
7.3.3 Graphical User Interface Use 276
7.3.4 Functional Flow 278
7.3.5 System Partitioning on Switchless Systems 280
7.3.6 System Partitioning Aid Output Directories 282
7.3.7 System Partitioning Aid Output Files 284
7.3.8 System Partitioning Aid Added Value 286
7.4 Switch Partitioning Additional Information 288
7.4.1 Switch Elements Terminology 288
7.4.2 Switch Connectivity Characteristics 290
7.4.3 Switch Partitioning Limitations 294
7.4.4 Second Stage Switch 296
7.4.5 Example of Rejected Configuration 299
7.4.6 Example of Accepted Configuration 300
7.4.7 Switch Performance Metrics 304
7.4.8 Example of Metrics File 306

Appendix A. Special Notices 309

Appendix B. Related Publications 311

B.1 International Technical Support Organization Publications 311
B.2 Redbooks on CD-ROMs 311
B.3 Other Publications 311

How To Get ITSO Redbooks 313
How IBM Employees Can Get ITSO Redbooks 313
How Customers Can Get ITSO Redbooks 314
IBM Redbook Order Form 315

List of Abbreviations 317

Index 319

Tables

1. Directories for Software Coexistence 129
2. Hard Disk Layout 147
3. CWS File Systems 147
4. Hard Disk Layout 162
5. CWS File Systems 162
6. PSSP 2.x and AIX 4.x Directories 169
7. AIX 4.2 and PSSP 2.2 Directories 197
8. System Partitioning Aid Command Line Access 272
9. System Partitioning Aid GUI Access 273

Preface

This redbook provides detailed coverage of new functions and components that were made available with the recent release of PSSP 2.2 and the new 604 PowerPC High Nodes. This book is organized in the form of a technical presentation. It includes mid-size foils and related description and notes for each foil in the document.

This redbook was written for IBM customers, Business Partners, and IBM technical and marketing professionals. It will provide them with a detailed presentation of the different new functions that make PSSP 2.2 and the 604 PowerPC High Nodes major enhancements to the RS/6000 SP product line.

The book focuses on the following topics:

- PowerPC 604 High Nodes
- New installation methodology
- Software Coexistence
- Migration Considerations
- Perspectives, the new system GUI interface
- System Partitioning Aid

A good knowledge of AIX Version 4 and RS/6000 SP is assumed.

How This Redbook Is Organized

This redbook contains 324 pages. It is organized as follows:

- Chapter 1, “PowerPC 604 High Nodes”

  This provides a detailed presentation of the new PowerPC 604 High Nodes. It describes the internal hardware of the system and the level of software that supports it. It contains the system management changes associated with this new hardware and its benefits in commercial environment. Details of the possible supported configurations are also illustrated.

- Chapter 2, “New Installation Methodology”

  This focuses on the new PSSP Version 2 Release 2 installation methodology consisting mainly of wrappers. It contains detailed flow charts that describe the new changes to Network Installation Management (NIM) and how to execute each of the modules.

- Chapter 3, “Software Coexistence”

  This focuses on the new PSSP Version 2 Release 2 coexistence with other versions of PSSP, namely PSSP 2.1 and PSSP 1.2. It also provides the supported software configuration necessary to implement coexistence in a mixed environment.

- Chapter 4, “Migration Considerations”

  This provides an overview of the migration considerations supported by PSSP Version 2 Release 2 software. It contains specific migration examples and how they are implemented.

- Chapter 5, “AIX 4.2 Support”

  This chapter describes the AIX 4.2 support in PSSP 2.2. This support will be available by year-end. It discusses application support, directories organization, installation, and migration topics.

- Chapter 6, “Perspectives”

  This chapter describes the new consolidated graphical user interface for system management called Perspectives. It describes the concepts and architectural design and operational strategy, which is focused mainly on system control and monitoring with this release of PSSP 2.2.

- Chapter 7, “System Partitioning Aid”

  This chapter describes the new system partitioning tool introduced with PSSP Version 2 Release 2. It contains detailed examples of how this tool could be used to generate valid system partition configuration. It also describes how this tool is used in a switchless environment.

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.

Endy Chiakpo is a Project Leader at the International Technical Support Organization, Pougkeepsie Center. He writes extensively and teaches IBM classes worldwide on all areas of RS/6000 SP. He holds a B.S. degree in Physics and a Master of Science degree in Electrical Engineering from Syracuse University New York. Before joining the ITSO, Endy worked in the IBM Poughkeepsie Lab in New York, USA.

Peter Kes is a Project Leader at the International Technical Support Organization, Poughkeepsie Center. He writes extensively and teaches IBM classes worldwide on all areas of RS/6000 SP. Before joining the ITSO in 1996, Peter worked in AIX Systems in The Netherlands, as a System Engineer.

Adrian Demeter is an RS/6000 SP specialist at the IT Availability Services department of IBM Czech, in Prague (Czech Republic). He is responsible for RS/6000 SP and SMP support in the country, and the first RS/6000 SP and SMP installations in the country were done by him. He has prepared and taught other SP and non-SP education courses for customers in the Czech environment. He has worked at IBM for 3 years. He has 4 years of experience in UNIX and system engineering. He holds a degree in Electrical Engineering and Technical Cybernetics from Czech Technical University (CVUT) Prague.

Robert Gambs is a Technical Marketing Support Specialist at the AIX Systems Center in Westlake, Texas (USA). He is responsible for education, planning, installation, customer support, and various other aspects of RS/6000 SP, general RS/6000, and AIX technology. He co-authored the RS/6000 SP Implementation Workshop, available from IBM Education and Teaching. Robert holds a bachelor degree in Computer Science with emphasis in Physics from Texas A&M at Commerce, formerly known as East Texas State University. He has been with IBM for over four years.

Franz Gerharter-Lueckl is an AIX specialist at IBM Austria. He holds an engineering degree for a five years study at the TGM Technical School for communications engineering and electronics. Before joining IBM in 1989, Franz worked as project leader and programmer for a software house. At IBM, he was on assignment at the T.J. Watson Research Center for 20 months. During that time, he was also responsible for the initial releases of LoadLeveler for Sun Solaris (5765-227) and SGI Irix 5 (5765-228). His areas of expertise include RS/6000 SP, benchmarks, networking, porting software to HP, SUN, SGI, and AIX ESA. He is a co-author of two redbooks: *PSSP Version 2 Technical Presentation*, SG24-4542; and *Scientific and Technical Computing Overview*, SG24-4541. Franz teaches “TCP/IP in a Multivendor Environment” classes for IBM Education.

Yoram Gnat is a senior RS/6000 market specialist in Israel. He has 15 years of experience in UNIX operating systems. He has worked at IBM for 11 years, following AIX from Version 2, through Version 3, and up to Version 4. He holds a Ph.D. degree in Physics from Tel Aviv University, participated in experimental research projects in high energy physics and published several research studies. His areas of expertise include AIX, scientific/engineering programming and optimizations, parallel programming, and TCP/IP communications. He now also teaches the C programming language and the AIX Operating System.

Yann Guerin is a team leader at the EMEA Parallel Solutions Support Center (PSSC), in Montpellier (France). He is responsible for RS/6000 SP education courses and for the coordination of the system administrators team on the other RS/6000 SP activities of the PSSC (benchmarking, briefing, demonstrations, customized solutions). He has 12 years of experience in UNIX and project management.

Hisashi Shirai is an advisory IT specialist at IBM Japan Systems Engineering Co., Ltd. in Makuhari (Japan). He is mainly responsible for high availability (HA) of RS/6000 systems, including RS/6000 SP. He has 4 years of experience in AIX, and has worked as a team leader of an HA technical support group for two years. He is also a co-author of the redbook *Implementing High Availability on RS/6000 SP*, SG24-4742.

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

Marcelo Barrios
International Technical Support Organization, Poughkeepsie Center

IBM PPS Lab Poughkeepsie:
Mike Browne
Deepak Advani
Joseph Banas
Dr. Gili Mendel
Dr. Aruna Ramanan
Michael Schmidt
Ken Briskey
Stephen Tovcimak
James Gilman
Dennis Jurgensen
Tim Race
Robert Gensler Jr.
Peter Badovinatz
Paul Bildzok
Skip Russell
Richard Ferri
Mark Gurevich

Steven Cangemi
William Wajda
Margaret Moran
Bernard King-Smith
Patrick Meehan
John Simpson

IBM US Dallas System Center:
Mark Venator

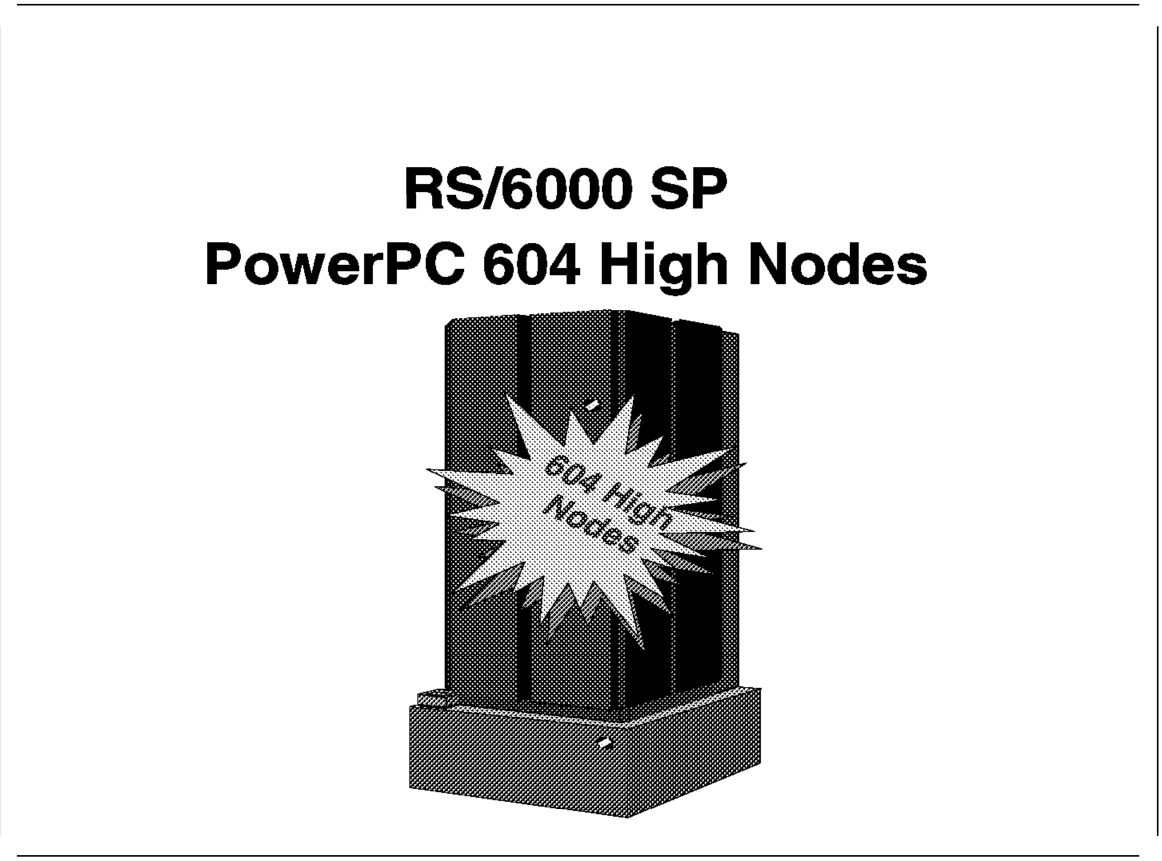
Comments Welcome

We want our redbooks to be as helpful as possible. Should you have any comments about this or other redbooks, please send us a note at the following address:

redbook@vnet.ibm.com

Your comments are important to us!

Chapter 1. PowerPC 604 High Nodes



RS/6000 SP
PowerPC 604 High Nodes

With the announcement of High Nodes, new concepts and terms have to be introduced to RS/6000 SP administrators and users. The IBM RS/6000 SP now benefits from both Symmetric Multiprocessors (SMP) and Massively Parallel Processors (MPP) architectures. To exploit these architectures, we have to understand how they work, and what their benefits and limitations are.

The first part of this chapter introduces some concepts of multiprocessing in general. It then focuses on SMP architecture, SMP programming model, and IBM implementation of the architecture.

The second part describes the PowerPC 604 High Node hardware, its relation to the RS/6000 model R40, the integration of High Nodes in an SP frame, and the configuration implications of the High Nodes on the RS/6000 SP system.

Special aspects of PSSP Version 2 Release 2 connected to the installation, management, and administration of the High Nodes are discussed in the third part of this chapter.

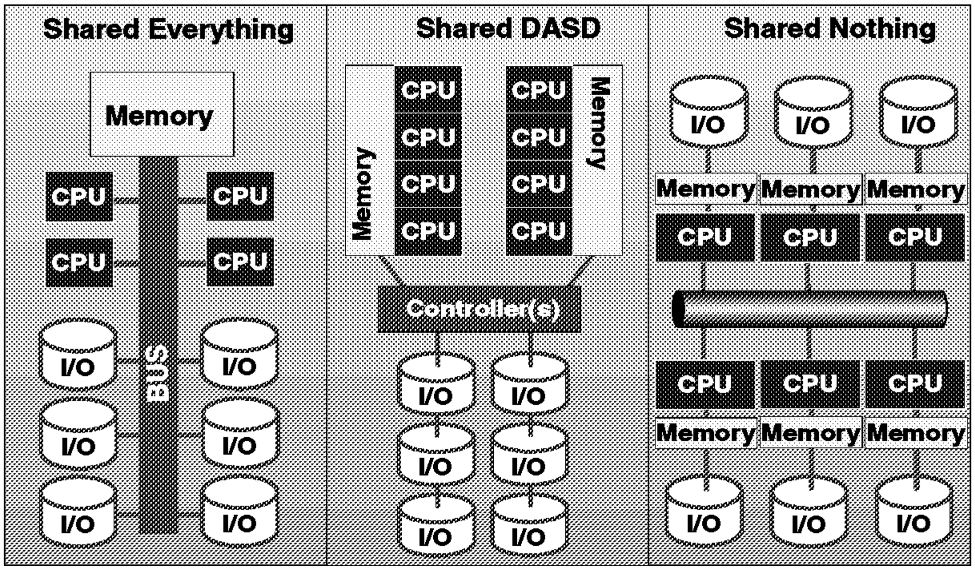
Notes on proper positioning and usage of the new High Nodes conclude the chapter.

1.1 Introduction to SMP Concepts

Following is a brief description of the SMP architecture concepts. For an in-depth discussion of SMPs and IBM implementation of this architecture, see *IBM RISC System/6000 SMP Servers Architecture and Implementation*, SG24-2583.

1.1.1 Multiprocessing Concepts

"Shared" Architectures



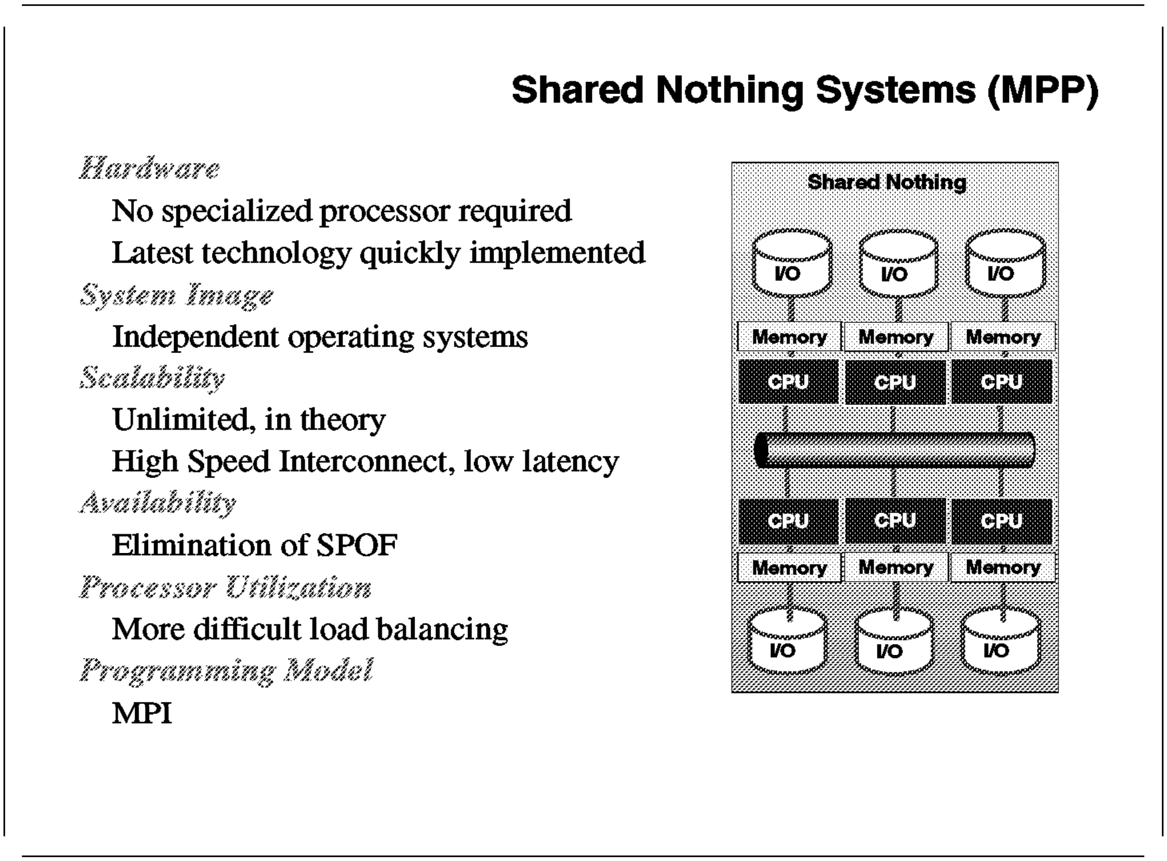
The standard uniprocessor design has some built-in bottlenecks. The main one is the fact that the instructions are forced to run in a strict sequence. Thus, only one task can be performed at a time. Multitasking is implemented by allowing different tasks to use the CPU in some predefined sequence. Introducing a new task to the CPU is performed by copying the registers′ state of a running task to some safe place (which is known as *context switching*), and ensuring that the memory content of this task will remain intact, sometimes by copying it to disk (which is known as *swapping*). Only then can we bring the next task to memory, restore its registers′ content, and start execution. This is a heavy load on the CPU, especially in multiuser, multiapplication environments.

Adding more processors to the system seems to be a good way to increase overall system performance. Each processor can perform different tasks at the same time, or perform different independent parts of the same task. However, since the processes running on the different CPUs may be related, the CPUs probably need to share resources like disk files or even data stored in memory.

They usually have to communicate with each other to synchronize the order in which the subtasks are executed. Sharing resources may introduce new bottlenecks, because different processes and CPUs are competing to get the resources.

The type of resources shared and the way they are shared define the different multiprocessor architectures. The next three sections briefly discuss the three most common architectures.

1.1.1.1 Shared Nothing Systems (MPP)



Shared Nothing Systems (MPP)

Hardware

No specialized processor required

Latest technology quickly implemented

System Image

Independent operating systems

Scalability

Unlimited, in theory

High Speed Interconnect, low latency

Availability

Elimination of SPOF

Processor Utilization

More difficult load balancing

Programming Model

MPI

The *Shared Nothing* multiprocessor systems, which are sometimes called Massively Parallel Systems (MPP), avoid the bottlenecks connected to resource sharing by simply not sharing resources. Each processor in such a system has its own cache, memory, I/O channels, disks, and adapters. It also runs its own copy of the operating system.

Very fast hardware and software interconnection fabrics are required to synchronize processes that run on different processors (*nodes*) but compose a single parallel application.

A set of administrative, management, and monitoring tools must be provided, so that a single system, rather than a set of computers, is seen by the system administrator.

The *advantages* of a shared nothing system are:

- No specialized processor is required, therefore the latest technologies can be quickly implemented.
- Very high scalability. Since the nodes share nothing, there are no contentions for accessing them.
- Very high availability. If one node fails, the rest of the system can continue to run. The failed application has to be restarted, but the overall system will keep running.

The *disadvantages* of such a system are:

- Load balancing is more difficult.
- To take advantage of the parallel architecture, specific programming interfaces such as MPI/MPL, PVME, and VSD need to be used. It is a programming model that requires specific skills.

The RS/6000 SP system is one of the most successful examples of a shared nothing architecture.

1.1.1.2 Shared DASD Systems

Shared DASD Systems

Hardware

No special hardware

System Image

Distributed, diverse

Scalability

Limited by technology

Availability

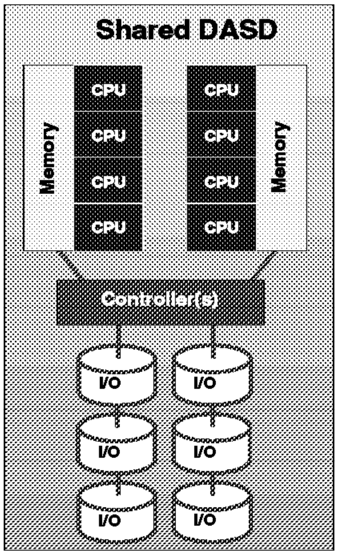
High, achieved by software

Processor Utilization

Complex load balancing

Programming Model

Fully customized



In a shared disk system, sometimes called shared disk cluster, each node (a node can be a uniprocessor, a symmetric multiprocessor, or a node in an MPP system) has its own cache, memory, and a copy of the operating system, but the disks are shared. Processors are interconnected through a LAN or a switch. Communication between members may be done through message passing.

The IBM RS/6000 HACMP clusters and DEC VAX-clusters are examples of this architecture.

The *advantages* of a shared disk system are:

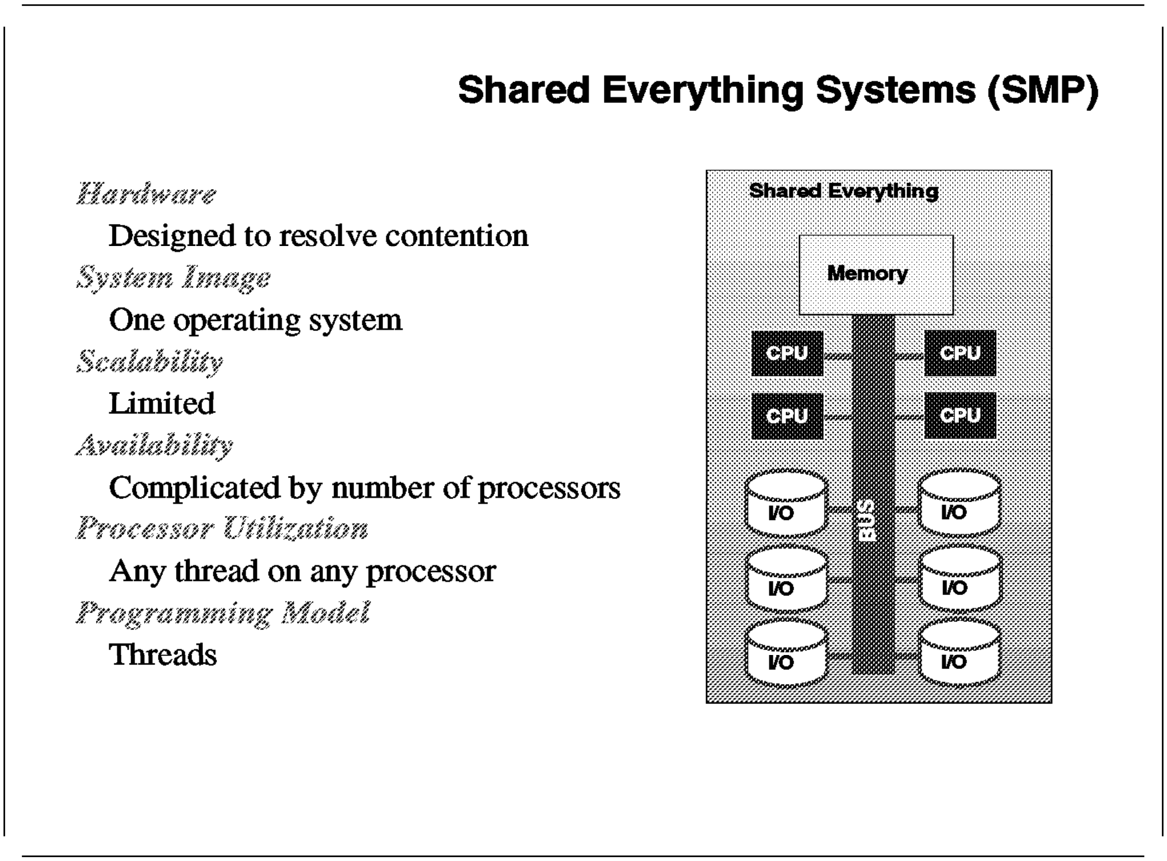
- No special computer hardware is required.
- Availability is high, since data on disk is available by other processors in case of processor failure.
- Part of the programming model that is familiar from the uniprocessor environment is preserved here. Data on disk is addressable by all nodes, and it is coherent.

The *disadvantages* of this architecture are:

- Load balancing is complex.
- Scalability is limited by the technology used to access the data shared on disks.

- The programming model is mixed. Data is shared on disks but not in memory. Communication between nodes must be done through specific interfaces.

1.1.1.3 Shared Everything Systems



Shared Everything Systems (SMP)

Hardware
Designed to resolve contention
System Image
One operating system
Scalability
Limited
Availability
Complicated by number of processors
Processor Utilization
Any thread on any processor
Programming Model
Threads

In this type of MP, the processors are tightly coupled inside the same box, with a high speed bus or switch between the processors, the memory, and the I/O subsystem. Although each processor has its own cache, the main memory, the disks, and the I/O devices are shared among the processors. A single copy of the operating systems runs across all processors, giving the user a *Single System Image (SSI).*

The *advantages* of shared everything architecture are:

- Easy system administration. A single image is presented to administrator and user; it is not different from the uniprocessor image.
- Programming model is familiar. The uniprocessor programming model can still be maintained.
- Load balancing is performed by the operating system; thus, in a multiuser/multitasking environment you gain throughput (not performance) even without adapting your applications to the new programming model.

The *disadvantages* of the architecture are:

- Scalability is limited due to the sharing of resources.
- In order to increase performance by taking advantage of the system processors, the use of a *thread* programming model is required. This programming model is not the same as the one used in uniprocessors; it requires specific programming skills.

Symmetric and Asymmetric Shared Memory Systems

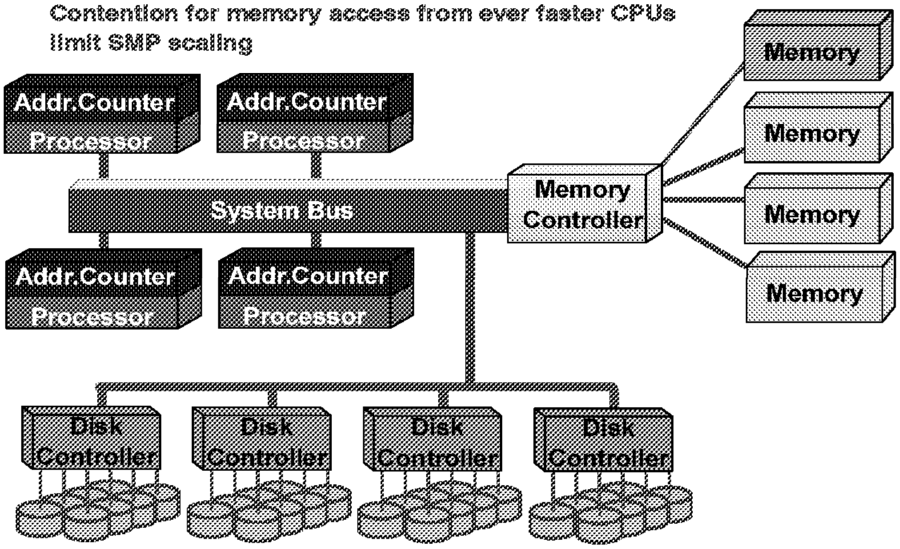
In an *asymmetric* shared memory multiprocessor, not all of the processors are equal. One of them is a *master processor*, while the others are slave processors. The master processor is a general purpose processor able to perform I/O operations as well as computations. Slave processors can perform computations only. All the I/O requests are routed from slave processors to the master processor. This can have severe performance implications if the master processor cannot service the slave processors requests efficiently. Failure of the master processor is catastrophic to the whole system.

In a *symmetric* shared memory system *(SMP)*, all processors are functionally equal. They can perform both I/O and computational operations. The operating system manages a pool of identical processors, and each of them is able to control any of the shared resources. In case of a processor failure, the system can be reconfigured and, after reboot, it will continue to run.

1.1.2 SMP Architecture

Symmetric Multiprocessors

Contention for memory access from ever faster CPUs limit SMP scaling



Sharing resources is the main technical issue in designing an SMP system. Contention for memory and disk access from increasingly faster processors must be efficiently resolved if we are to benefit from the multiple CPUs in the system. Specific hardware and software techniques must be provided to allow symmetric multiprocessing.

About 20 to 50 clock cycles are required to load data from main memory. If no other means are provided, the CPU has to wait through these cycles every time new instructions or data are loaded from memory.

To improve the hardware performance of a CPU (UP or MP), different levels of memory are used. The very fast L1 cache memory can be accessed in very few cycles (one cycle in PowerPC), but it is small, usually around 32 to 64 Kb. The L2 cache, external to the processor, has a higher capacity of about 1 to 4 Mb and can be accessed in 7 to 10 cycles.

In a symmetric multiprocessor, each processor has its own L1 cache and usually its own L2 cache. The main memory is shared between the processors.

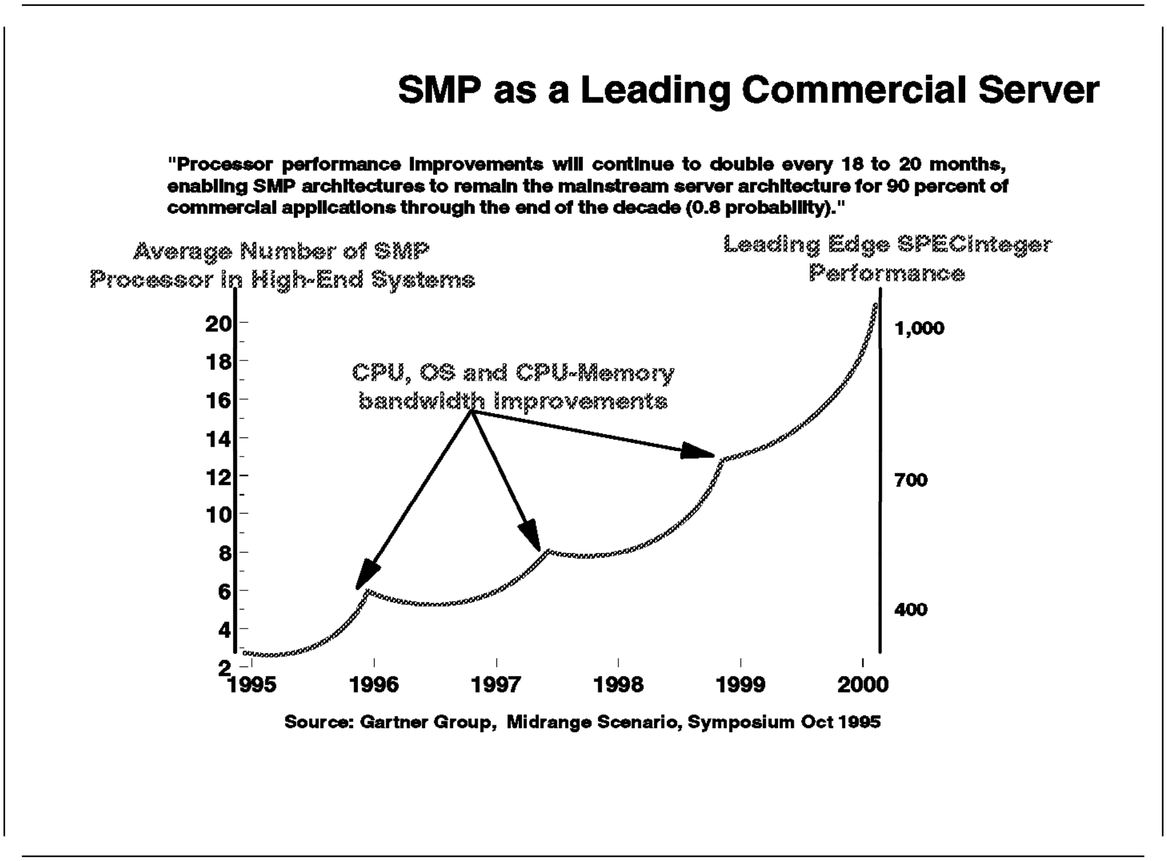
In an SMP system, parts of a single task (*threads*) may run simultaneously on different processors. These threads share the same memory space and access and update the same memory locations. Therefore, the memory image seen by all processors, both main memory and cache memory, must be the same.

Different algorithms and techniques are used to ensure *cache coherency*. All of them involve data traffic between the different processors′ cache memories, and between the cache memories and the main memory.

In SMP systems, this data traffic and the data transfers between memory and I/O devices are handled through a single data path: bus or switch. Contention for this data path is the main limiting factor for SMP scaling. At some point, adding a CPU to the system will result in saturating that data path, resulting in a performance decrease.

Even bearing all that in mind, upgrading from UP to SMP, with the possibility of adding more processors to the system when needed, is still a very cost-effective method to increase system throughput.

1.1.2.1 SMP as a Commercial Server



SMP as a Leading Commercial Server

"Processor performance improvements will continue to double every 18 to 20 months, enabling SMP architectures to remain the mainstream server architecture for 90 percent of commercial applications through the end of the decade (0.8 probability)."

Source: Gartner Group, Midrange Scenario, Symposium Oct 1995

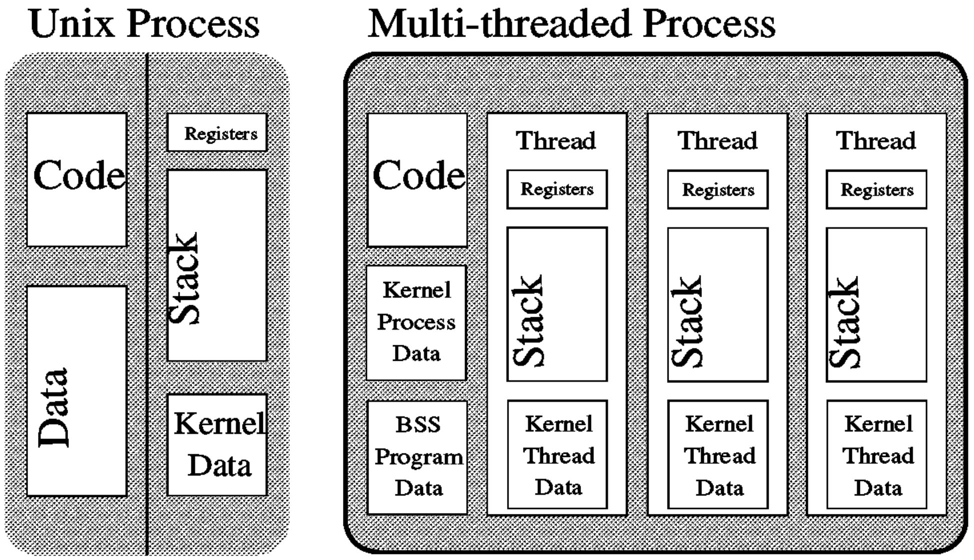
Improvements in processor performance and in bus/switch bandwidths will increase the maximal number of CPUs in an SMP system. The accompanying Gartner Group prediction shows that the SMP platform will have significant success in the four-to-eight processor configurations through 1998. However, this market prediction includes thousands of small and medium departmental servers in the 90% market saturation.

For large scale enterprise-wide applications, a single SMP system may not be the best solution. Therefore, distributed memory parallel machines are achieving commercial success with large corporations.

The introduction of SMP nodes in a distributed memory system, like the IBM RS/6000 SP, allows users to take advantage of the three most successful multiprocessor architectures in a single system.

1.1.3 SMP Programming Model

Processes versus Threads



In the UNIX environment, the term *process* denotes the combination of a program (a set of instructions and data required to perform a specific task), together with the current state of its execution, such as the program counter, registers, condition codes, and the content of the address space. In other words, a process is a program in execution.

The term *thread*, the most important term in the SMP programming model, denotes an independent flow of control that operates within the same address space as other independent flows of control within a process.

In older versions of UNIX, a process could have only one flow of control (*one thread*) within its address space. In recent versions of the operating systems with AIX among them, one process can have multiple threads, with each thread executing a different code concurrently while sharing data and files and synchronizing with other threads. In an SMP system, since memory and I/O devices are shared between processors, each thread within a process may execute on a different processor.

In such a multi-threaded environment, a ″regular″ process is seen as a single-threaded process.

In AIX V4, you will find three types of threads, namely:

User Threads

A user thread is an entity used by application programmers to handle multiple flows of control within a process. The Application Programming Interface for handling user threads is provided by the threads library. The user threads API is part of a portable programming model. A user thread only exists within a process. A user thread in one process cannot reference a user thread in another process.

Kernel Threads

A kernel thread is a kernel entity handled by the system scheduler. A kernel thread runs within a process, but it can be referenced by other kernel threads in the system. The application programmer has no direct control over this type of thread, unless it is writing kernel extensions. In AIX, each user thread is mapped to one kernel thread.

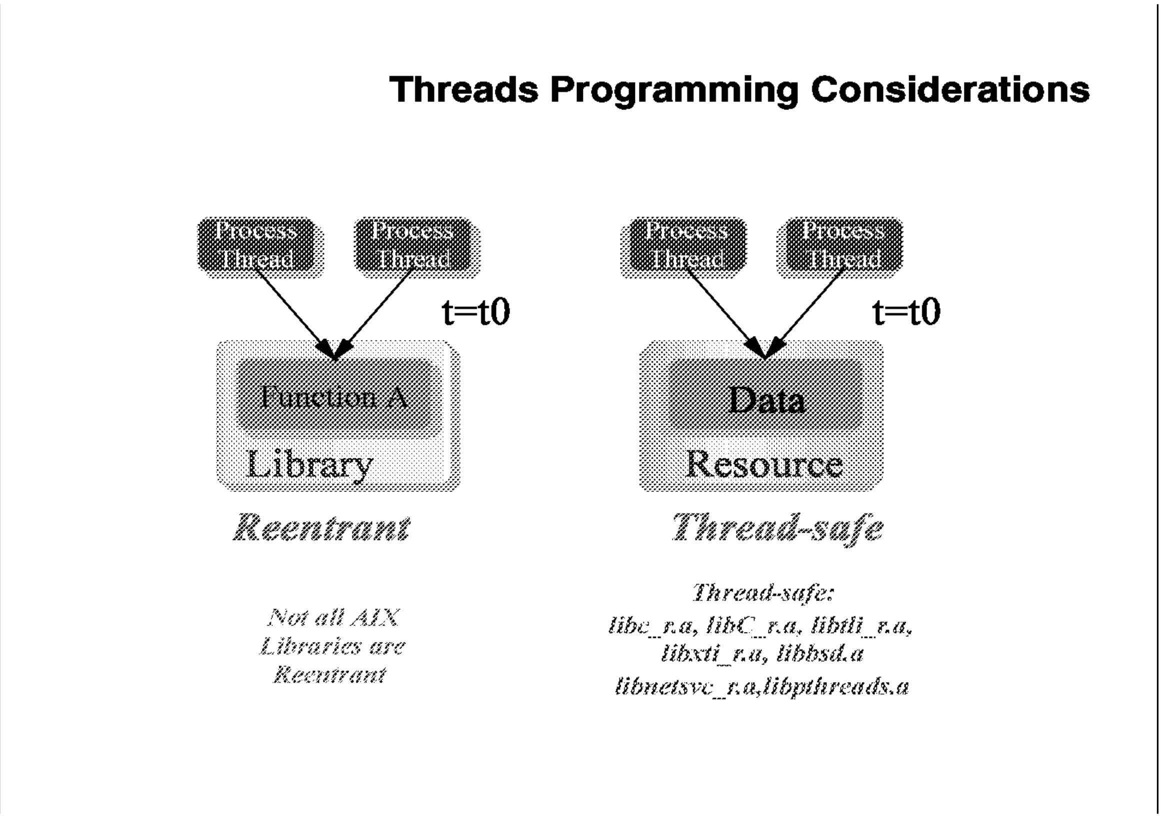
Kernel-Only Threads

A kernel-only thread is a kernel thread that executes only in a kernel mode environment. Kernel-only threads are controlled by kernel programmers through kernel services.

As stated before, a multi-threaded process has several independent flows of control. All threads within a process run in the same address space. Each thread holds the state of a single flow of execution within the process. The state of a thread consists of a minimum of the hardware state and a stack. Each thread keeps the data of the kernel thread it is mapped to.

Since threads run in the same address space, data communication can easily be achieved through shared variables within the address space.

1.1.3.1 Threads Programming Considerations


Threads Programming Considerations

Reentrant

Thread-safe

Not all AIX Libraries are Reentrant

Thread-safe: libc_r.a, libC_r.a, libtli_r.a, libxti_r.a, libbsd.a libnetsvc_r.a,libpthreads.a

The threads library includes facilities for threads creation, termination, synchronization, communication error recovery, and management. A user thread can make system calls just like a simple process. From a programmer′s point of view, the programming of threads is very similar to the programming of processes.

Since all threads belonging to same process share the same address space, locking and synchronizing mechanisms are used to ensure the coherency of data in all levels of memory. In a single-threaded process, there is only one flow of control. Therefore, the codes executed by such processes do not need to be reentrant or *thread-safe*.

In a multi-threaded process, two threads can call the same function at the same time. This means that to execute properly, this function must be reentrant.

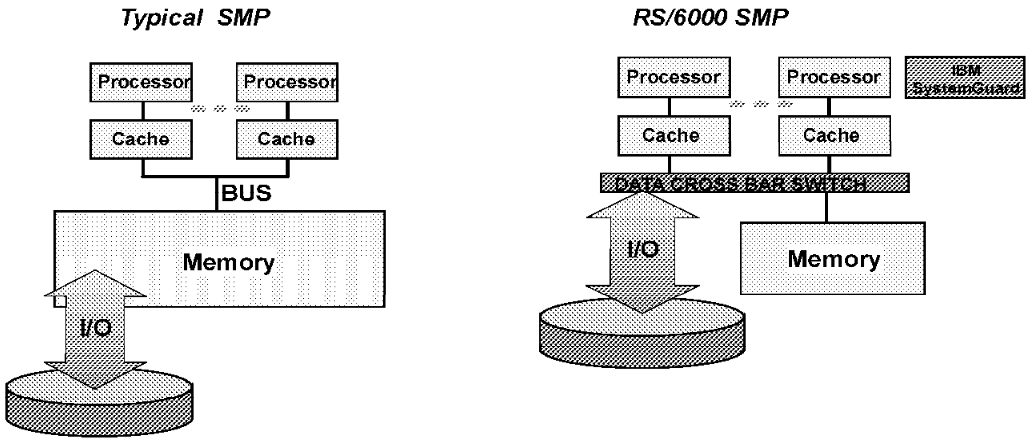
Two threads may also access the same resource at the same time. To avoid data corruption, the shared resources must be protected by locks. A *thread-safe* function does not use unprotected resources.

A multi-threaded program must use only functions that are both reentrant and thread-safe. A library is called thread-safe if all functions in the library are both reentrant and thread-safe.

Development and API libraries that are not thread-safe cannot be supported on SMP systems, since programmers may include calls to such libraries in their multi-threaded applications.

1.1.4 IBM SMP Design

IBM SMP Design Advantage



RS/6000 SMP : Designed for

**** Reliability, Availability, Serviceability***
**** System Capacity and Growth***
**** Investment Protection***

Two main aspects must be considered when designing an SMP system.

The first one is the performance issue that results from the way the processors perform memory communication, not just directly with memory, but among each other.

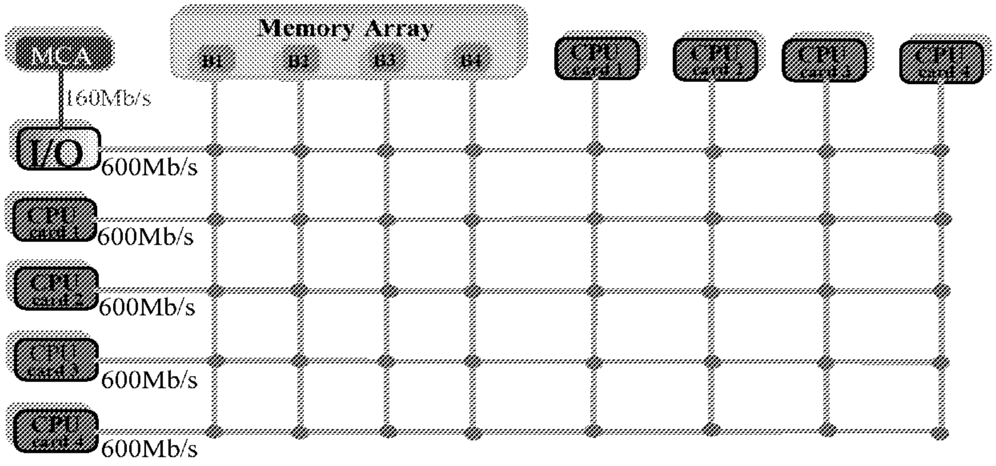
The scalability of an SMP system (that is, the ability to get much more performance when adding a new processor), depends very much on the way the processors communicate with each other and with the memory. As was already pointed out previously, these interprocessor communications are the main bottleneck of the shared everything architecture.

The second aspect is a combination of reliability, availability, and serviceability. In a shared everything system, the failure of one component affects all the other components. The failure of a processor in such a system will cause system halt. The system must be reconfigured (by removing the failing processor from the list of available resources), rebooted, and then the applications should be restarted. Automatic monitoring of system resources, notification of service personnel when required, and restarting the system when needed, are extremely important in a commercial SMP environment.

Two components in IBM SMP systems, the *cross bar switch* and the *SystemGuard processor*, distinguish the IBM design from most other available systems. These two components will be briefly discussed in next two sections.

1.1.4.1 Cross Bar Switch

Cross Bar Switch



- Each circle in the cross bar array represents a switch that can be turned on and off

The IBM SMP design is a result of extensive research into the performance characteristics of commercial applications. Typically, such applications include manipulating vast amounts of data and sharing data between many users and programs. Two effects should be noticed when running such applications in an SMP system:

- Due to the low probability of finding the appropriate data already in the cache, there will be a high level of data traffic generated between system memory and CPU caches.
- The default behavior of the scheduler in an SMP system is to execute the next runnable thread on the first processor that becomes free. This causes a dynamic increase in the level of data traffic between CPU caches. The physical implementation of cache coherency becomes a key to global system performance.

Therefore, the memory subsystem requires high speed large caches, a high bandwidth between processors and memory, a high bandwidth between the processors themselves (for cache-to-cache transfers), and a high bandwidth between the processors and the I/O subsystem.

Traditionally in SMP architecture, a single bus is used to interconnect the CPUs, the global memory, and the I/O subsystem. This is the most stressed point of the architecture, and tends to become saturated as the number of processors in the system increases. This situation occurs because of the increased cache-to-cache transfers caused by cache coherency protocols (*snooping*), and data transfers between the caches themselves and between the caches and memory.

The IBM SMP is designed in a different way. There is still a bus for *snooping* activity and addressing. But a new component has been added for data transfers. That component is a switch called a *Data Cross bar Switch (DCB)*. The switch allows point-to-point connections between a processor and another processor, between a processor and memory, and between memory and the I/O subsystem. It also allows several simultaneous transfers of data.

The idea of a cross bar is to provide multiple buses that can be used simultaneously in order to reduce contention and to provide several memory banks that can be used in parallel.

Each circle in the cross bar represents a switch that can be turned on or off. Normally, all switches are off until a processor or I/O device needs to access the memory or another processor.

For example, if processor card 1 needs to transfer data to memory bank 1, the switches on the data path will be turned on. At the same time, processor card 2 can transfer data to processor card 3. When the data transfer is complete, the corresponding switches are again turned off.

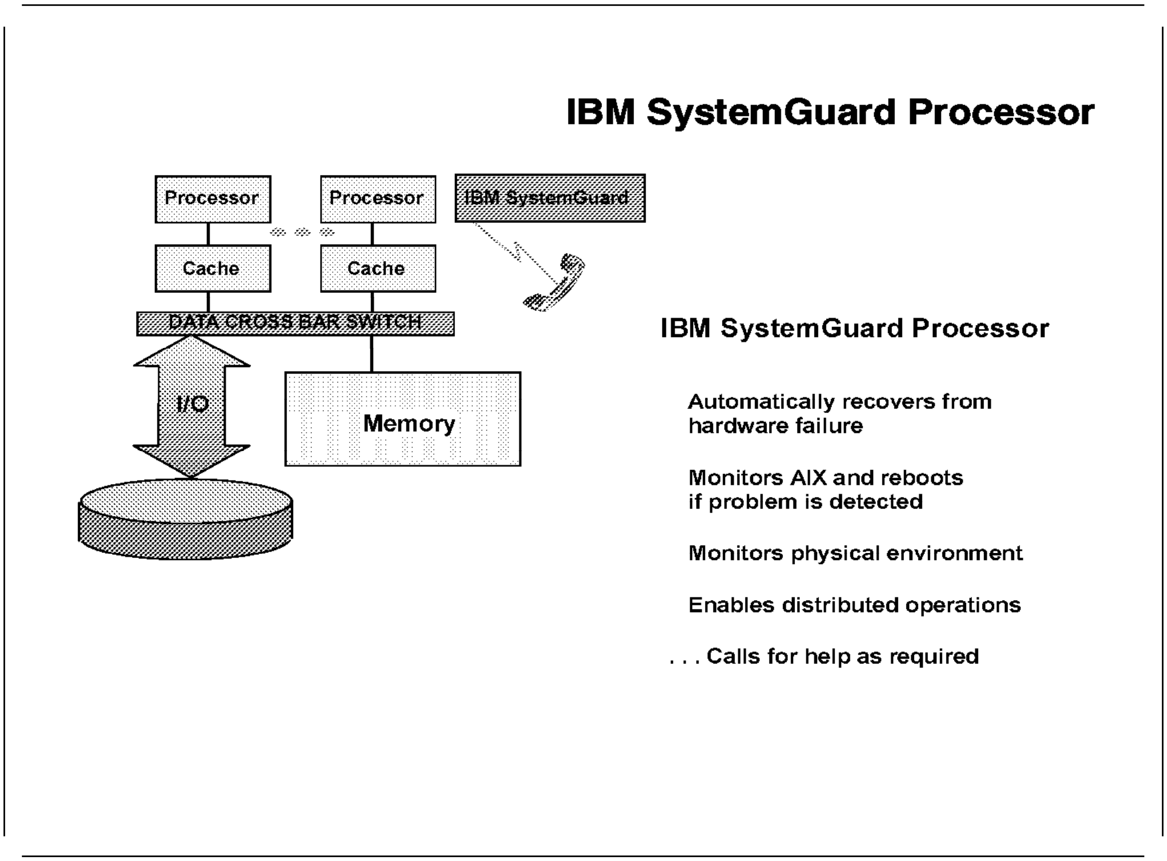
If the same switches are to be used simultaneously by several components, for example if two processor cards need to access the same memory bank, then the cross bar hardware arbitrates the requests in a way similar to that of a standard bus.

In a system equipped with four processor cards, and at least four memory banks, we can simultaneously have two memory-CPU transfers and one CPU-CPU transfer. With a rate of 600 Mb/s for each transfer, this leads to a *cross bar peak rate* of 1800 Mb/s.

Summarizing, the switch gives the IBM SMP the following advantages:

- It removes work from the bus.
- It can transfer data among several units simultaneously.
- Connections are point-to-point, allowing greater speed.

1.1.4.2 IBM SystemGuard Processor



IBM SystemGuard Processor

IBM SystemGuard Processor

Automatically recovers from hardware failure

Monitors AIX and reboots if problem is detected

Monitors physical environment

Enables distributed operations

. . . Calls for help as required

The IBM SMP servers were designed to operate in a commercial environment where RAS is extremely important. To handle the different RAS requirements, IBM RS/6000 SMP models are equipped with a service processor, called the *SystemGuard Processor*.

The SystemGuard Processor continually monitors the hardware as well as the operating system. If, for example, a CPU were to fail, the SystemGuard would detect it, reboot the system, and run without the failing CPU. The same would happen in case of a detected memory error that could not be corrected. The system will reboot and run without the failing memory component.

The SystemGuard Processor allows diagnostics and maintenance to be performed either locally or remotely. Having its own power boundary allows it to diagnose the problem and maintain the system even if the system power is off.

The main features of the SystemGuard are:

- Initialization process flow management
- Local as well as remote (remote not supported on High Nodes) control of the system (power-on/off, diagnostics, and reconfiguration maintenance)
- Run-Time surveillance
- Dial-out to a support center in case of system boot failure (not supported on High Nodes)

Note: In the RS/6000 SP environment, the Control Workstation is the single point of control for the system. Therefore, the *remote* features of the SystemGuard Processor are not supported on the RS/6000 SP.

The SystemGuard Processor introduces new hardware and firmware components that you need to know about in order to understand the installation and booting of an SMP system, and to understand the integration of High Nodes in the RS/6000 SP system.

The new hardware components are:

- A microprocessor called BUMP (Bring-UP Micro Processor) with its EPROM and RAM
- A Flash EPROM
- A Backup EPROM

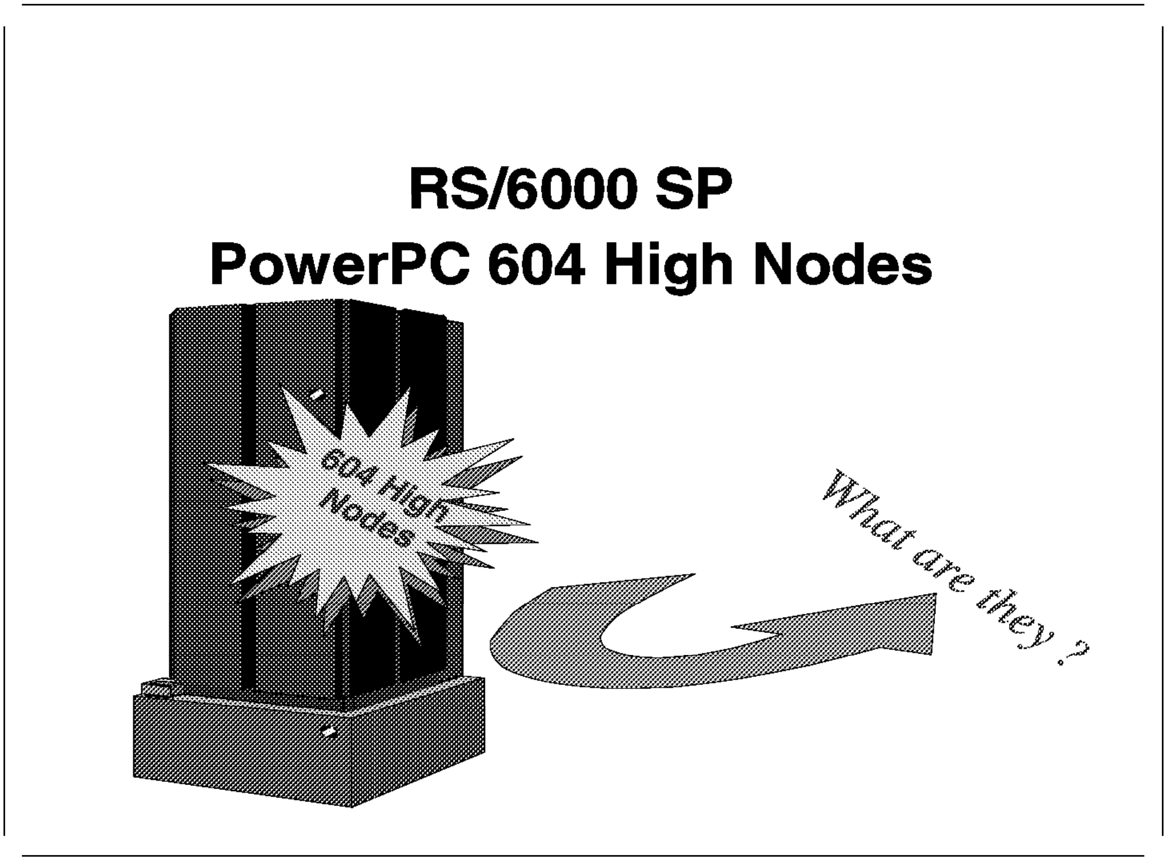
Part of the SystemGuard firmware is stored in the BUMP EPROM, and part in the Flash EPROM (allowing update of the BUMP firmware). The BUMP has access to existing system components, the power supply, all the boards′ EPROMS, the PowerPC COPs (Common On-chip Processor (COP) which is part of the PowerPC and used to test the chip), and so on.

The BUMP interfaces with the Operator Panel, the NVRAM, and the S1 and S2 serial ports. These interfaces are important to remember, since they are used in the High Node integration.

The SystemGuard controls the system when:

- The system power is off. In this state, the SystemGuard allows the system administrator or service personnel to run a specific test, set system parameters, reconfigure the system, or power-on the system. The interaction between the BUMP and the administrator is through a terminal connected to the S1 (or S2) serial port.
- The system is booting. SystemGuard controls the hardware Power ON tests and the loading of AIX.
- AIX is running. The SystemGuard monitors the system through a heartbeat protocol.

1.2 PowerPC 604 High Node



RS/6000 SP

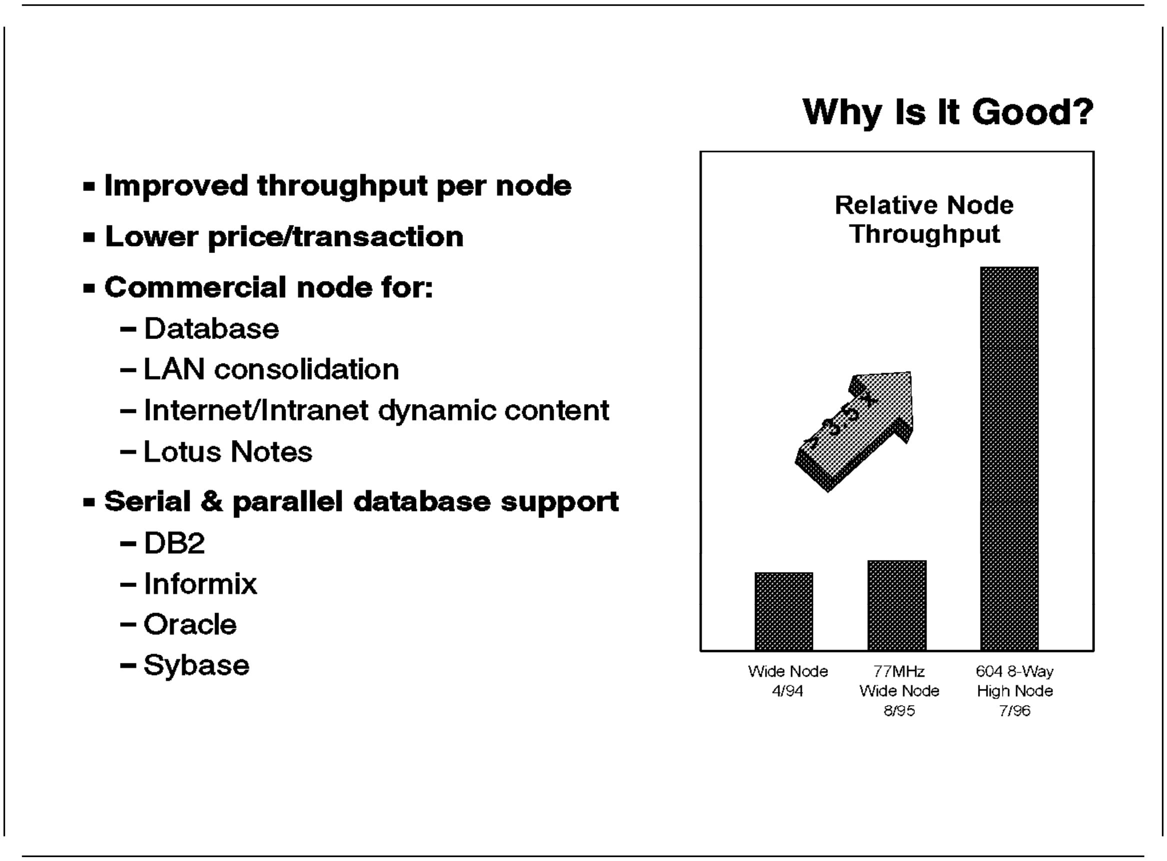
PowerPC 604 High Nodes

The RS/6000 SP High Node is another piece of evidence of IBM′s commitment to incorporate the top of RS/6000 technology into the Scalable Parallel systems.

With the announcement of the G40, J40, and R40 SMP systems, our customers expect to have the same commercial computing power in the RS/6000 SP.

In many cases of server consolidation, databases, and other commercial applications, the RS/6000 SP faced the competition of SMP clusters with conventional LANs as the interconnect layer. However, with the introduction of the High Nodes, the RS/6000 SP is now the fastest and most tightly coupled SMP cluster.

The flexibility of the configuration of the RS/6000 SP, the possibility of mixing, and the freedom to choose the node types most suitable for the task required make the RS/6000 SP one of best choices in parallel computing, in both the scientific/engineering and the commercial environments.



Why Is It Good?

- **Improved throughput per node**
- **Lower price/transaction**
- **Commercial node for:**
  - Database
  - LAN consolidation
  - Internet/Intranet dynamic content
  - Lotus Notes
- **Serial & parallel database support**
  - DB2
  - Informix
  - Oracle
  - Sybase

The new High Node delivers a large boost in commercial capability to the RS/6000 SP system. Compared to the 77 MHz wide node, the 8-way 604 High Node delivers over 3.5 times the transaction throughput for approximately 2.5 times the price (this relation may vary by geography).

With this transaction capability, the High Node is optimal for databases. It will be supported by the major database vendors DB2, Oracle, Informix, and Sybase, both as a serial database server and as a parallel database server.

IBM DB2 is already supporting the High Node with both version 2 for a serial transaction database (such as in SAP applications), and DB2 Parallel Edition for decision support applications. DB2 PE will scale to the maximum number of High Nodes, so it is particularly well-suited to an environment with a large database, a large number of interactive users, or both. Since each node is now able to handle three to four times more data than before, and since DB2 PE pricing is based on the number of nodes, the price/performance of DB2 PE will be even more attractive than before.

The other database vendors should be consulted to get availability dates.

In the LAN server consolidation applications, the High Node gives the best solution in cases where the application size exceeds the capacity of the wide node.

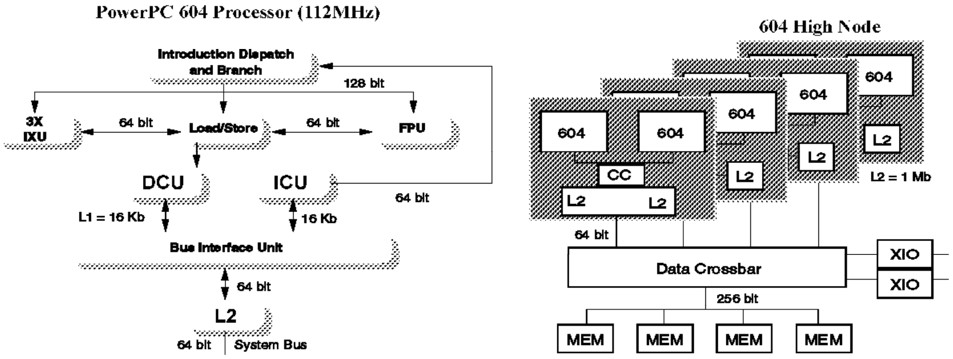
The High Node is also the best choice for Lotus Notes, if a large number of users are to be served by a single node.

Its outstanding transactions performance makes the High Node very well-suited for Internet/Intranet applications with dynamic content. (Dynamic content applications are applications which are constantly updated and searched upon request.)

Although the High Node is not intended for numeric intensive floating point calculations, it can be used in MPP configurations in cases where fast file server or data server is required.

1.2.1 SP PowerPC 604 High Node

High Node Architecture

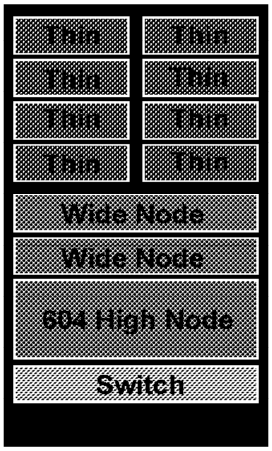


The High Node uses the PowerPC 604 processors at 112 Mhz. 32 Mb of instruction/data Level 1 cache is part of each processor. Three integer units, a branch unit, and a floating point unit allow the 604 PowerPC processor to execute up to five operations per cycle. A 64 bit wide data path connects the processor to the Level 2 cache controller.

Up to four dual CPU cards can be configured in the High Node. 2 Mb of second level cache (1 Mb per processor) are standard on the CPU cards. A 64 bit wide path connects the caches to the cross bar switch. A 256 bit wide path connects the cross bar to the main memory.

SP PowerPC 604 High Node

- **2-, 4-, 6-, or 8-Way 604 112 MHz**
- **64 Mb to 2 Gb memory**
- **1 Mb L2 cache**
- **2.2 Gb to 6.6 Gb internal disk**
- **16 Micro Channel slots**
  - 14 available
  - Supports most SP adapters
- **Supports:**
  - SP & High Performance Switches
  - TCP/IP (only)
- **Mix and match node types**
  - Max 16 High Nodes per system
- **New 604 High Node models:**
  - 206, 306, and 406
- **AIX 4.1.4 & PSSP 2.2**



The High Node is based on the RS/6000 model R40 and has the same basic configuration features. Up to four dual processor cards (one processor card in the base configuration) can be configured, resulting in a 2-, 4-, 6-, or 8-way SMP system. The name High Node reflects the fact that two drawers (four slots) in the RS/6000 SP frame are occupied when such a node is installed.

Each processor card has a 2 Mb L2 cache (1 Mb per processor).

The four memory slots in the node allow the expansion of the base 64 Mb memory to up to 2 Gb.

Of the 16 existing Micro Channel slots, one is used by an SCSI-2 Fast/Wide Single Ended controller and one by a required Ethernet adapter. This means that up to 14 slots are available in a non-switched RS/6000 SP. In an RS/6000 SP configuration with a switch, one additional Micro Channel slot will be used for the switch adapter.

Both the High Performance Switch adapter and the new SP Switch adapter are supported in the High Node.

The Network Terminal Accelerator cards (features 2402 and 2403) and the Ethernet/FDX 10 Mbps adapters (features 2992 and 2993) are not supported on the High Nodes.

Like the R40 model, the High Node supports only 2.2 Gb disks. In the R40, there are one disk bay and two media bays for a CD drive and optional tape drive. Since internal media drives are not supported in RS/6000 SP nodes, the three bays may be used for disks, giving a total internal capacity of 6.6 Gb.

A redundant power supply for each High Node may be optionally added.

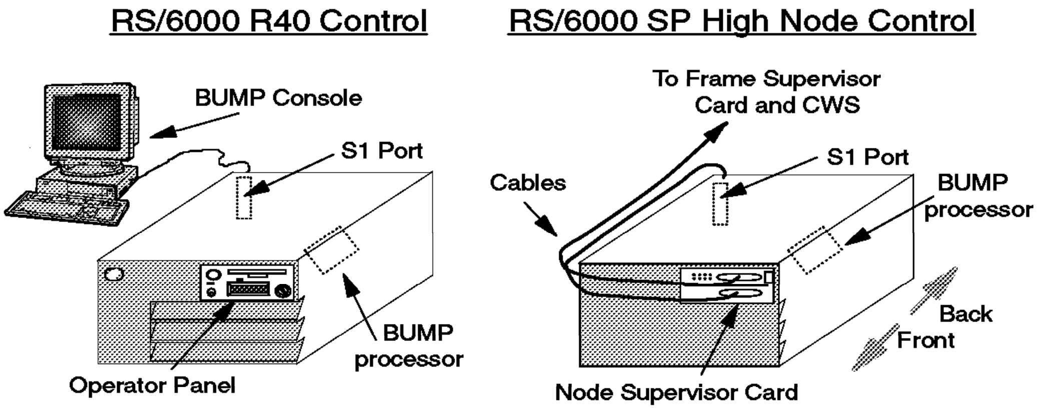
There are some limitations when configuring an RS/6000 SP system with High Nodes, namely:

- A maximum of 16 High Nodes can be configured in an RS/6000 SP system.
- High Nodes are not supported on systems using the 8-port version of the switch.
- High Nodes are not supported in short frames (low-cost frames).

As with the R40, AIX Version 4.1.4 with APAR #IX57164 is required on the 604-based High Node. PSSP Version 2 Release 2 must be installed on both the Control Workstation and the High Nodes in the RS/6000 SP system.

1.2.1.1 What Is a High Node?

How Was It Done?


RS/6000 R40 Control

RS/6000 SP High Node Control

SystemGuard tightly coupled to SMP processing hardware
➠ Interface between SP node supervisor card and SMP BUMP processor

The size of the RS/6000 model R40 allows it to be put inside an SP frame. It will occupy two drawers. The problem is that putting a computer system inside an SP frame is not enough to make it an RS/6000 SP node.

In each node, there is a supervisor card which is wired to a frame supervisor by an RS232 connection. The supervisor card monitors and activates the node hardware according to instructions received from the frame supervisor. The node supervisor supplies node data to the frame supervisor using the same connection path. The frame supervisor card itself is connected to the Control Workstation and can both transfer data and receive operational instructions (for example, power on node number 5).

As was already pointed out, the SystemGuard (or BUMP) is a very important part of IBM SMP systems. It basically plays a role similar to that of a node supervisor. It can receive commands from the operator panel and from the S1 serial port, and it can perform diagnostics, maintenance, power on/off operations, and other hardware monitoring and surveillance tasks.

If the R40 is to become an RS/6000 SP node, the operator panel is no longer meaningful, since the node is closed in the SP frame. What is required is a *special* node supervisor card that will connect to the BUMP processor through the front panel connection and through the S1 port.

This node supervisor module is the only real difference between an R40 and a High Node (if you do not consider removal of the plastics as a change). The whole operator panel (including diskette drive and operator panel display) was removed and the High Node supervisor card module was inserted instead, connected to the BUMP through the same connector as the original operator panel.

The S1 port, which is the other communication channel to the BUMP, is also connected to the node supervisor. Like any other node supervisor, it is connected to the frame supervisor with RS232 wiring.

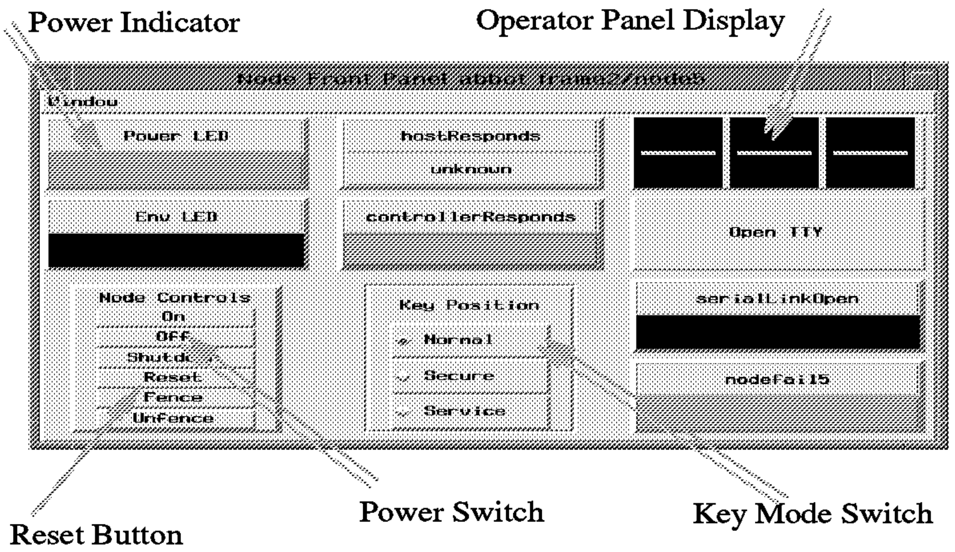
Trying to make the changes as simple as possible, the other communication ports of the R40 (the S2 and S3 ports) were not removed. These ports are not supported since they may not exist in future versions of High Nodes, but they are there and active.

The SystemGuard is performing most of its functions for configuration, initialization, monitoring, and problem determination, but it is now under the surveillance of the node supervisor.

Some of the functions like call home, battery backup, and cluster power control are no longer supported. Since this is now a node in a complex system, all of these functions are provided from the RS/6000 SP single point of control, which means from the Control Workstation.

1.2.1.2 Operator Panel

What Happened to the Operator Panel?

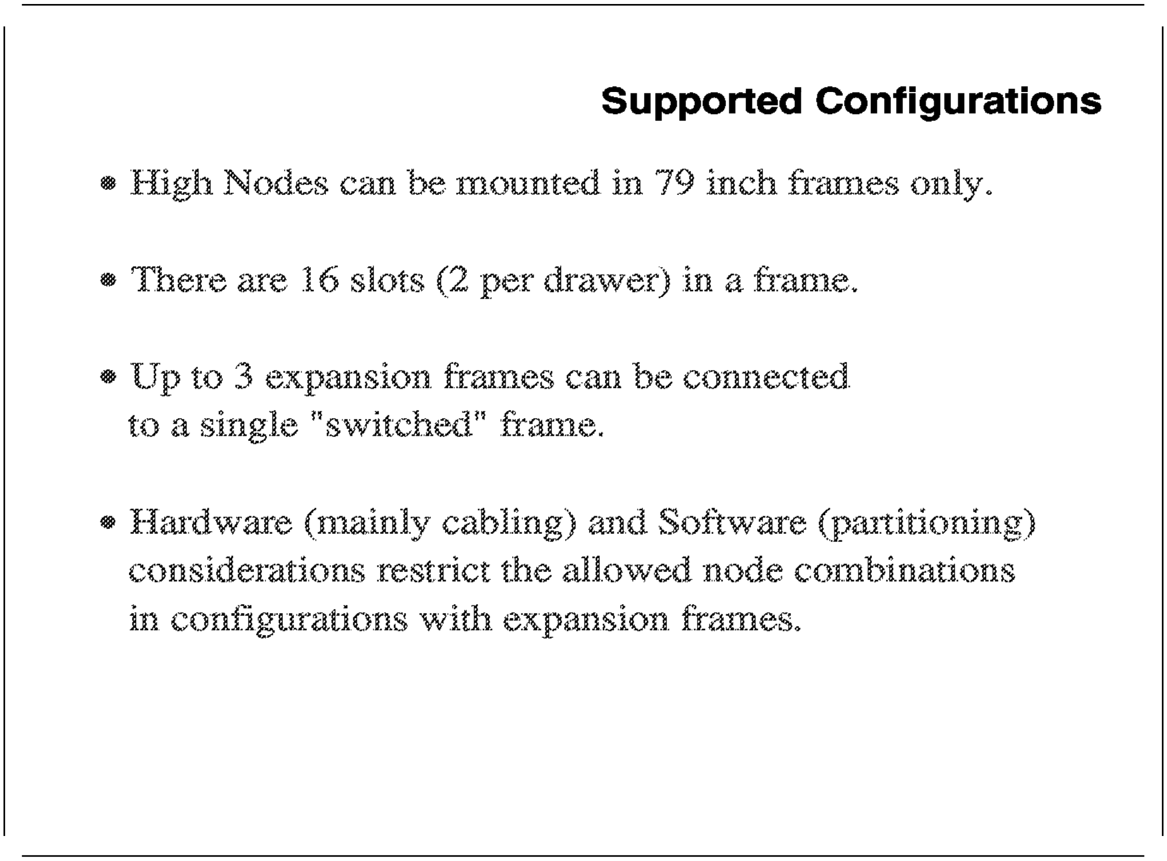


7

The functions of the original R40 operator panel can still be accessed. The *node front panel* display (from SPmon) gives you the same functionality as any other R40 machine.

You still can access the BUMP menus by opening an *s1term* which will connect you to the S1 port when the node is powered off. (The BUMP is then in standby mode and can be accessed.)

1.2.2 Supported Configurations



Supported Configurations

- High Nodes can be mounted in 79 inch frames only.
- There are 16 slots (2 per drawer) in a frame.
- Up to 3 expansion frames can be connected to a single "switched" frame.
- Hardware (mainly cabling) and Software (partitioning) considerations restrict the allowed node combinations in configurations with expansion frames.

With the introduction of High Nodes and the possibility of adding them to existing configurations, you need to understand what configurations are possible and how you number the nodes and the switch ports in such configurations.

The High Nodes are supported only in the high (79″) frames. Each frame has 8 drawers (16 slots). Four slots are occupied by each High Node. Therefore, you can mount up to four High Nodes in each frame.

Each switch board has 16 ports. If your frame is fully populated by Thin nodes, then all of the switch ports will be occupied, because each node is connected to a switch port. If, however, you have Wide or High Nodes in the frame, you will be left with *unused* switch ports.

You can add frames without a switch to the system, and you can connect the nodes within such frames to switch ports in a *switched* frame, where not all switch ports are used.

If there is already a frame with a switch board currently in the system, the next frame that is without a switch board is called an *expansion frame*.

Up to 16 High Nodes are supported in an RS/6000 SP switched configuration system. If you have a full “High Nodes only” configuration, you can connect all the nodes to a single switch board. For that you will need one *switched* frame and three expansion frames. Thus, up to three expansion frames are allowed for each switched frame.

When you have a frame with three expansion frames, you are not forced to put only High Nodes in these frames. However, hardware (mainly cabling and possible connections to switch chips) and software considerations (like switch port numbering, system partitioning, and consistency of topology files) restrict the allowed nodes′ locations and types in configurations with expansion frames.

Here are some terms we will use in the following discussion:

Frame Numbers

Frame numbers are established by the system administrator when the system is installed. Each frame is referenced by the tty port on the Control Workstation to which the frame supervisor cable is attached, and is assigned a numeric ID by the system administrator. Expansion frame numbers must follow the number of the frame with the switch board to which their nodes are connected. Thus, if *n* is the frame number of a frame with a switch and there are 3 expansion frames connected to same switch, their numbers will be *n+1, n+2,* and *n+3*.

Slot Numbers

Each 79-inch frame has 16 slots numbered from 1 to 16. Slots in the same drawer have consecutive numbers. Slot numbers only have meaning within the context of a specific frame.***:***

Node Numbers

A node number is a global ID assigned to a node. It is the primary means by which the administrator will reference a specific node in the system. The node number is computed as follows:

node_number = (16 x (frame_number - 1)) + slot_number

where slot_number is the lowest slot number occupied by the node.

Note

- A Wide or High Node will always have an odd node number.
- Nodes in expansion frames and in the frame with a switch to which expansion frames are connected will have only odd node numbers.

Switch Port Numbering

A switch port number is a global identifier for a port on the switch. It in itself does not identify the physical node that it is attached to. A *switch port* can be thought of as a connector on the back of a switch board in an SP frame. Therefore, the *switch port number* is the global ID assigned to each switch port that is unique across the RS/6000 SP. It is used internally in PSSP software as a direct index into the switch topology file and to determine routes between switch ports.

If a node is connected to the switch, then its *switch node number* is the same as the *switch port number* it is connected to. The SDR contains information for each node to identify the switch port number that it is attached to (its switch node number). Since there is no automatic method for determining how nodes are wired to switch ports, fixed rules have been defined and implemented in SDR_config. For properly configuring and partitioning the system, these rules must be followed. The rules are such that if you add an expansion frame to an existing system, the existing switch node numbers will not be changed; you will only add new switch node numbers to the topology files.

In the rest of this section, switch_number is the ID that has been assigned to the switch board (in the *switched* frame) to which the node is connected.
The slot number is the lowest slot number occupied by the node.

Here are the rules:

Frame with a Switch
In a frame with a switch, the switch node numbers are calculated as follows:

```
          switch_node_number = (16 x (switch_number - 1)) + slot_number - 1
```

First Expansion Frame
In the first expansion frame, the switch node numbers are assigned as follows:

```
          switch_node_number = (16 x (switch_number - 1)) + slot_number
```

Second Expansion Frame
In the second expansion frame, the switch node numbers are assigned as follows:

```
          switch_node_number = (16 x (switch_number - 1)) + slot_number + 2
```

Third Expansion Frame
In the third expansion frame, the switch node numbers are assigned as follows:

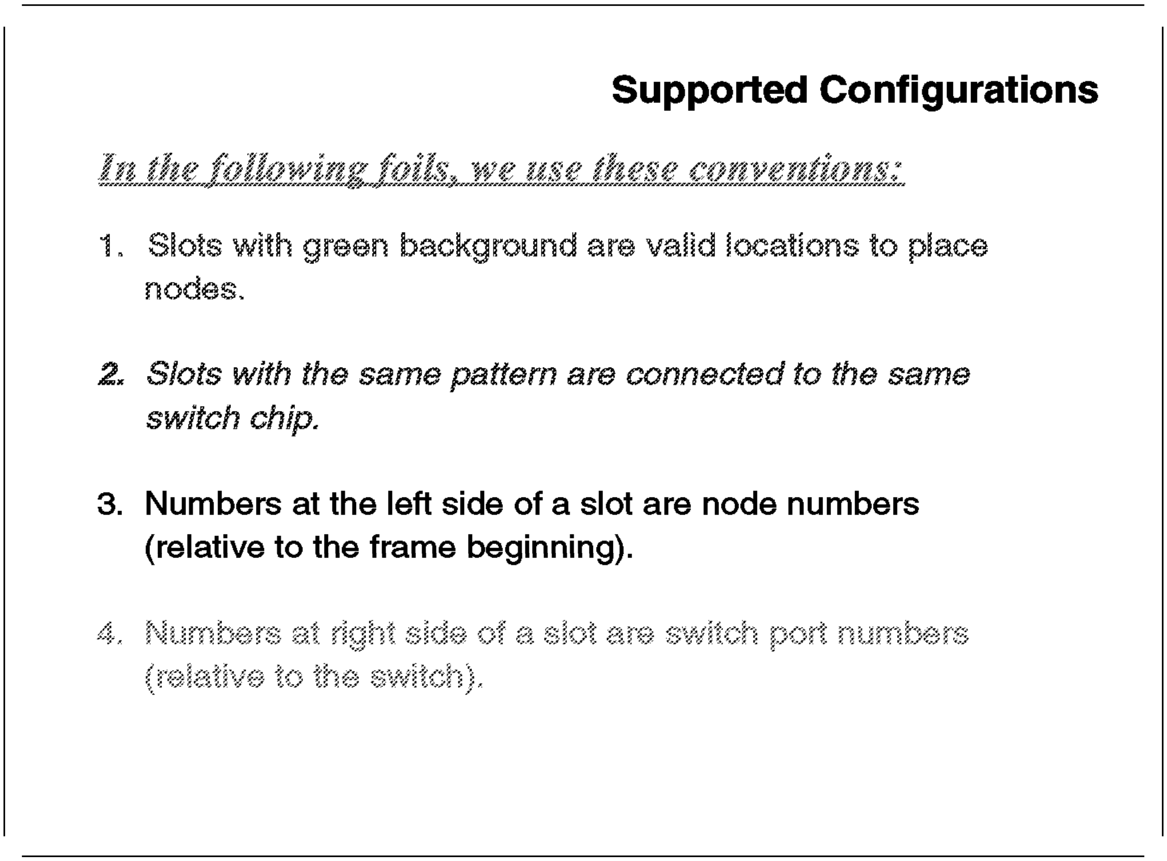
```
          switch_node_number = (16 x (switch_number - 1)) + slot_number + 1
```

Note

If there is a frame with a switch, and expansion frames are connected to the same switch, then the following is true:

The frame with the switch and the third expansion frame will have nodes with even-numbered switch nodes.

The first and second expansion frames will have nodes with odd-numbered switch nodes.



Supported Configurations

In the following foils, we use these conventions:

1. Slots with green background are valid locations to place nodes.

2. *Slots with the same pattern are connected to the same switch chip.*

3. **Numbers at the left side of a slot are node numbers (relative to the frame beginning).**

4. Numbers at right side of a slot are switch port numbers (relative to the switch).

The next few foils show the possible combinations of frames and expansion frames with the mixing of node types.

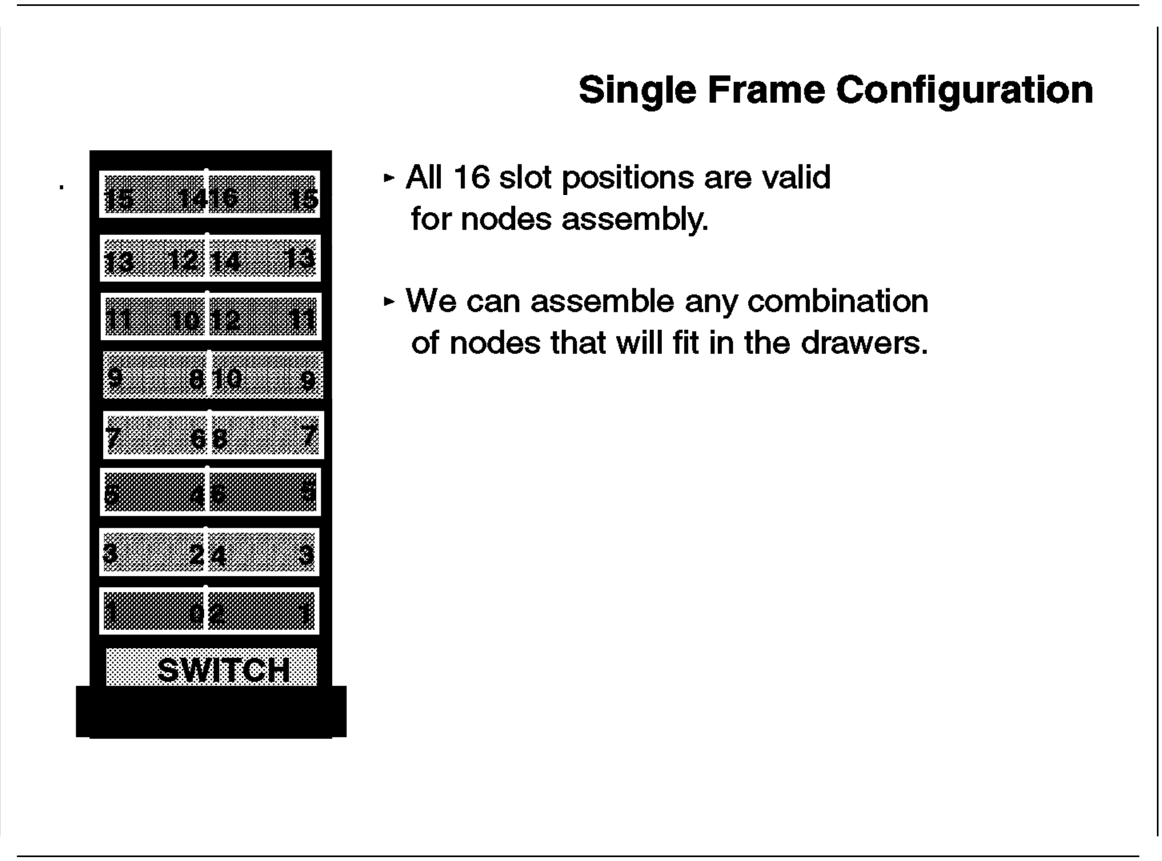
The node numbers and switch numbers in the foils and in the following text assume that the frame is frame number 1 and the switch is switch number 1. To get the general case, add *16 x (frame_no - 1)* to the node numbers in the foils and *16 x (switch_no - 1)* to the switch node numbers.

In following foils, a light gray background in a slot means that one *cannot* mount a node in this slot.

Each switch board has four switch chips and each chip has four ports. Nodes connected to the same switch chip must exist in same partition. The cabling in SP frames is such that the nodes in each group of switch nodes *(0, 1, 4, 5), (2, 3, 6, 7), (8, 9, 12, 13), and (10, 11, 14, 15)* are connected to same switch chip. In the following foils, nodes that are connected to the same switch chip will have the same background pattern.

The number at the left side of a node is the node number; the number at the right side is the switch node number.

1.2.2.1 Single Frame Configuration



Single Frame Configuration

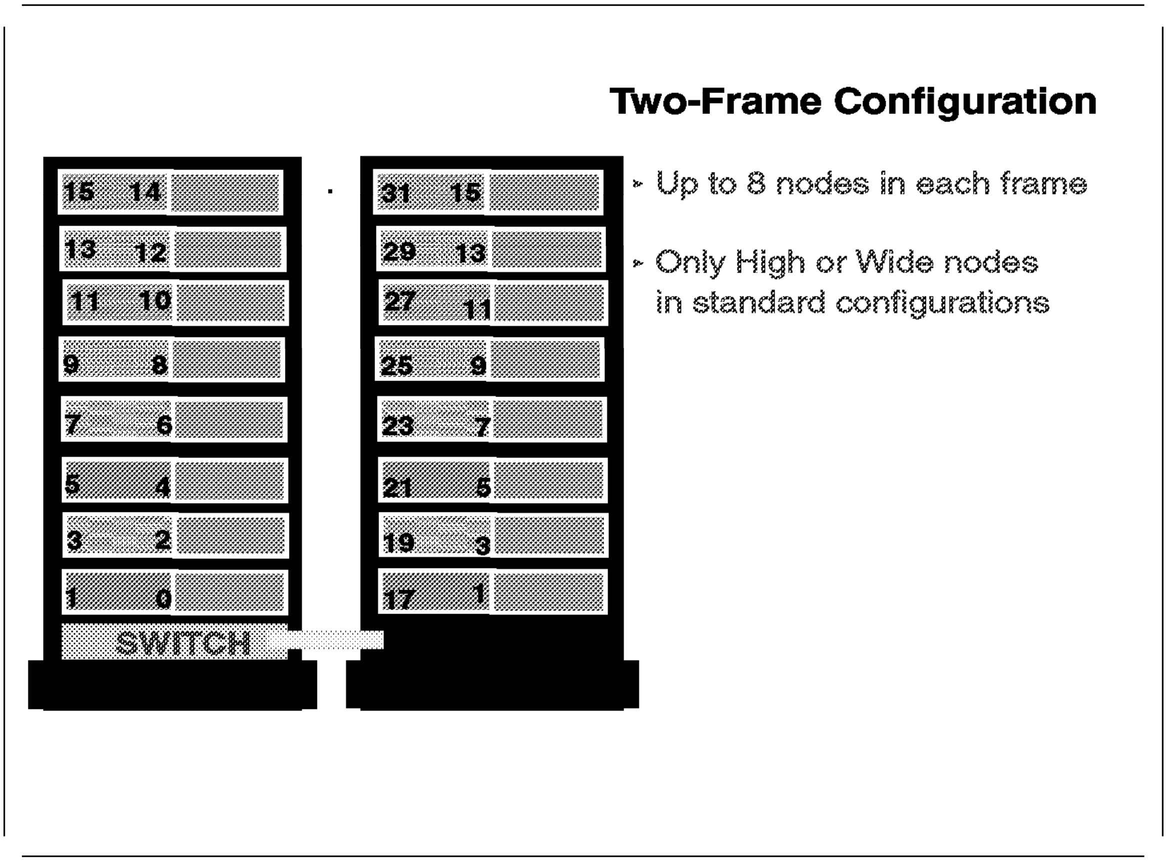
- All 16 slot positions are valid for nodes assembly.
- We can assemble any combination of nodes that will fit in the drawers.

This is the simplest configuration of all. Any combination of nodes is allowed.

If there are Wide or High Nodes in this frame, some switch port numbers will not be used. You can add an expansion frame to expand the system and utilize the free switch ports.

Note: If you have High Nodes in the first frame, some even switch port numbers will not be used. These even switch ports can be used only in the third expansion frame.

1.2.2.2 Two-Frame Configuration



Two-Frame Configuration

- Up to 8 nodes in each frame
- Only High or Wide nodes in standard configurations

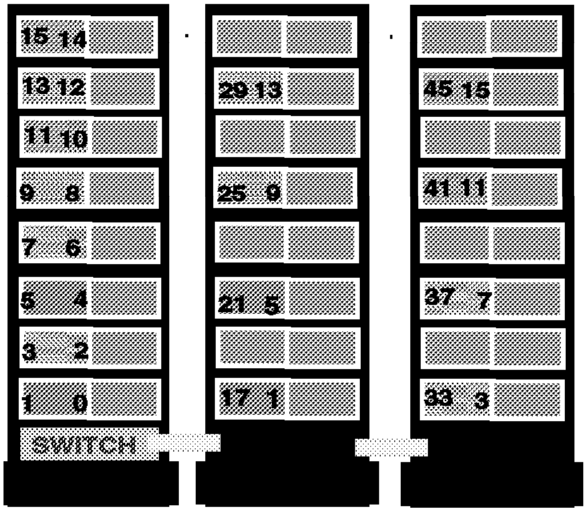
In a configuration with an expansion frame, only odd-numbered slots are used. This means that you can mount Wide or High Nodes.

In standard configurations, Thin Nodes come in pairs, with each pair occupying one drawer. In configurations with expansion frames, if you want Thin Nodes, you have to mount them one above the other. This can be done only in RPQ configurations.

If you have a frame with a mix of High, Wide, and Thin Nodes, you can add an expansion frame and use the positions that do not violate the rules of switch node numbering. However, you could end with an expansion frame that is only partly populated.

1.2.2.3 Three-Frame Configuration

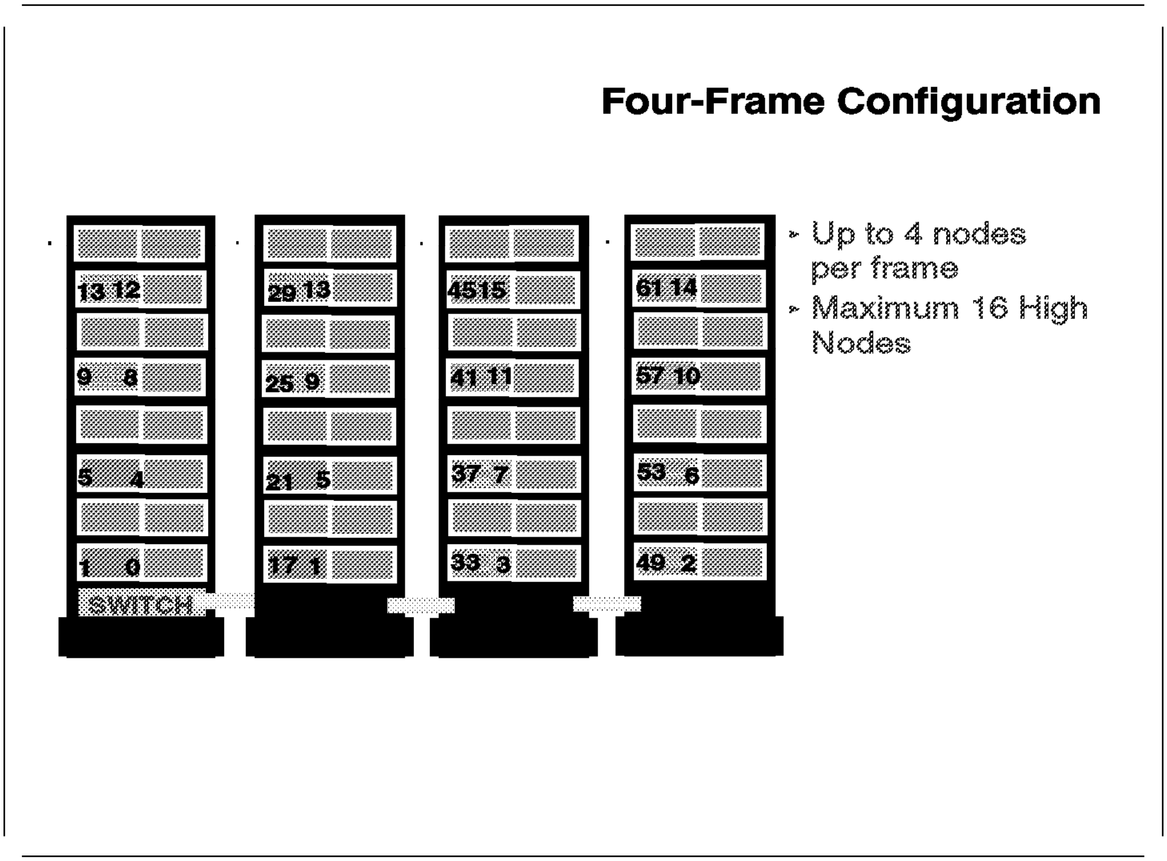
Three-Frame Configuration



- Up to 8 nodes in a switched frame
- Up to 4 nodes in expansion frames

In this case, as in the two-frame configuration, the frame with the switch can contain up to 8 nodes. Up to four nodes can be mounted in the expansion frames. The fact that you can have only four nodes in the expansion frames leads to the conclusion that High Nodes should be mounted in the expansion frames so as to use all of the space, while Wide Nodes should be left in the first frame.

1.2.2.4 Four-Frame Configuration

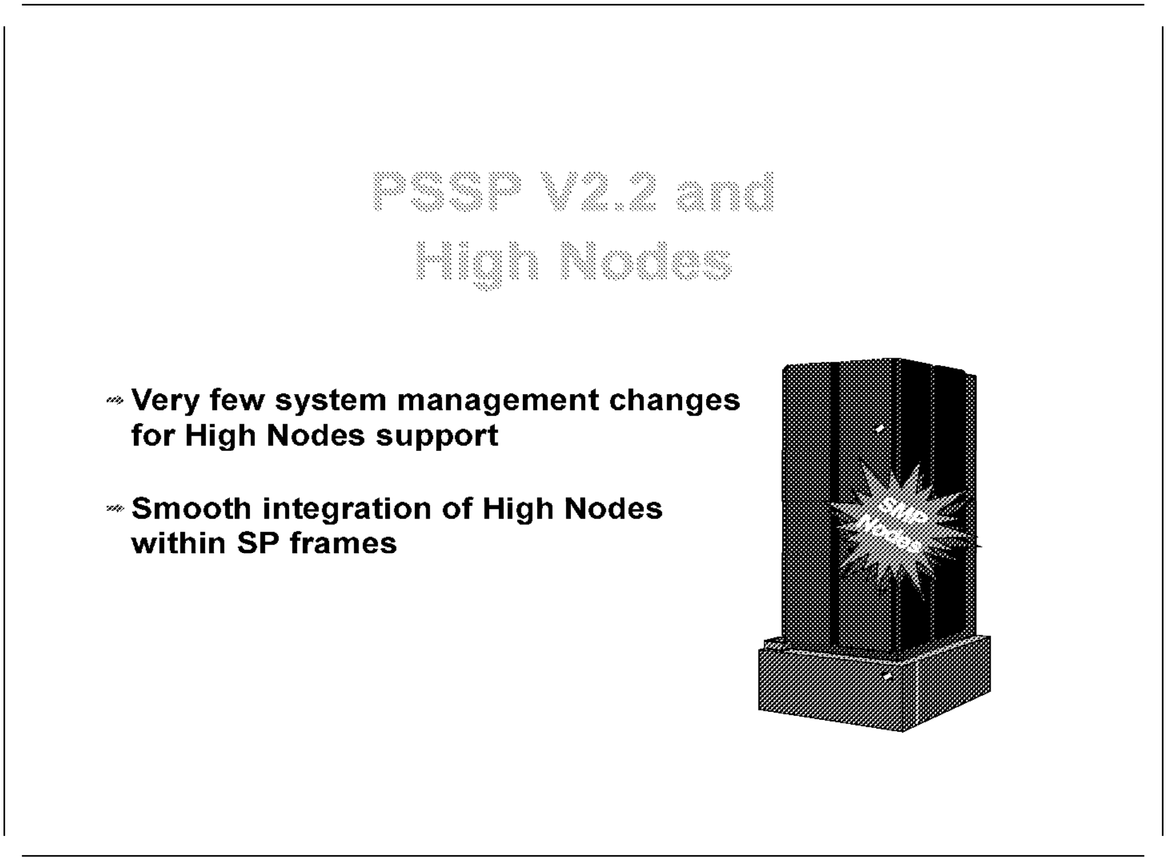


Four-Frame Configuration

- Up to 4 nodes per frame
- Maximum 16 High Nodes

The last possibility is to have three expansion frames connected to one frame with a switch. Four nodes can be configured in each frame. If all of them are High Nodes, you will reach the maximum possible number of High Nodes in this switched configuration system.

1.3 PSSP Version 2 Release 2 and High Nodes



PSSP V2.2 and
High Nodes

- Very few system management changes for High Nodes support
- Smooth integration of High Nodes within SP frames

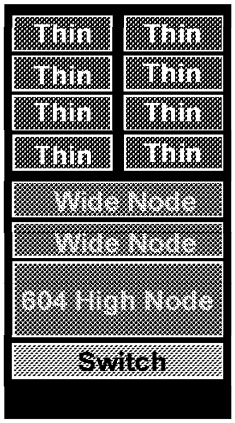
SMP nodes bring a change in architecture, as former nodes were all based on the POWER/POWER2 technologies. PowerPC hardware and SMP concepts are new to the RS/6000 SP. Some level of change in the system management is therefore expected, like the one encountered by RS/6000 administrators with the introduction of desk-side and rack-mounted RS/6000 SMP servers.

The minor changes necessary to integrate the High Nodes within the RS/6000 SP system management structure were made at a perfect time. The introduction of the new version of PSSP is the best foundation for system management and administration improvements from which all type of nodes, Thin, Wide, and High, will benefit.

This section focuses on some PSSP Version 2 Release 2 topics related to SMP nodes in conventional RS/6000 SP System Management tasks.

Overview

- Dedicated hardware to mix SP node supervisor card and SMP BUMP processor
- SDR modifications to incorporate MP node flavor
- Modifications of scripts and commands to work with MP and UP nodes in a flexible way
- UP node management unchanged



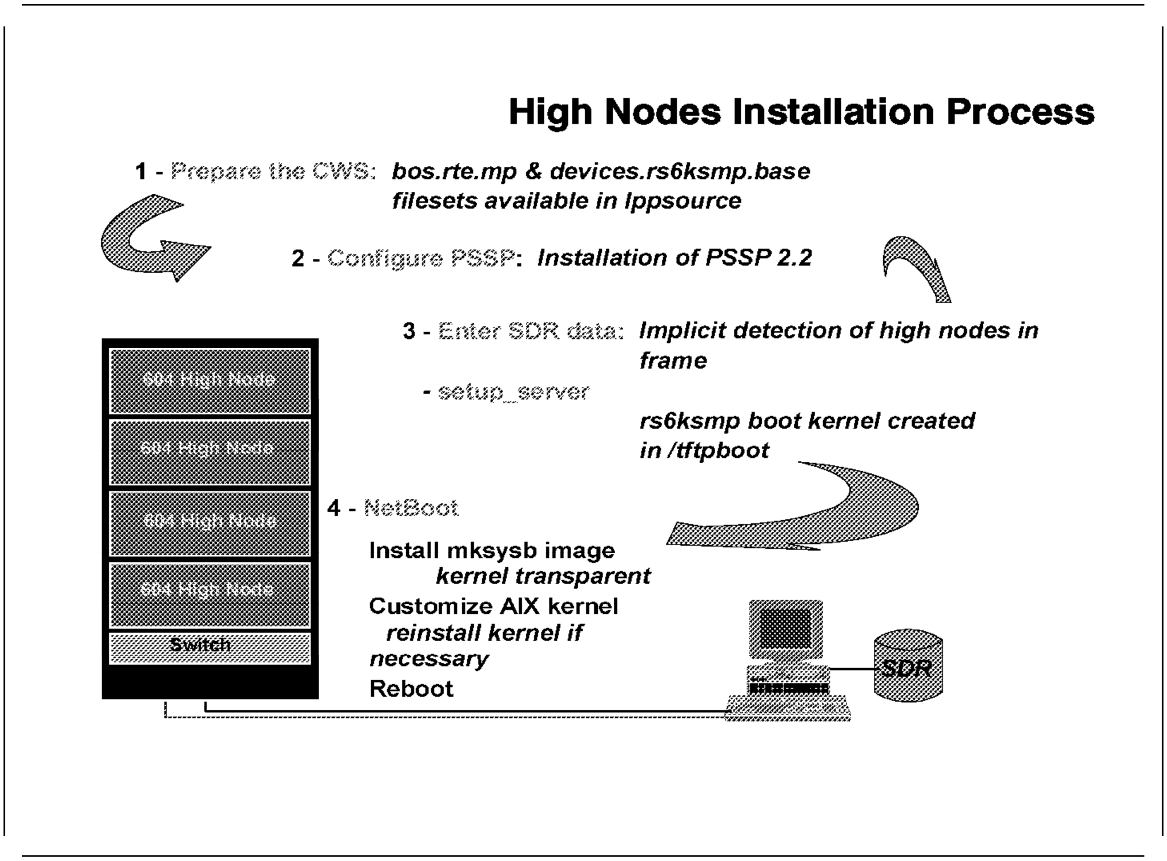
The introduction of the High Nodes brought two types of changes to the RS/6000 SP system management:

- Necessary modifications to deliver the same level of information and control as with the other RS/6000 SP nodes
- Specific improvements to cope with the introduction of nodes with multiple processors

Administrators of previous RS/6000 SP systems will find very few modifications in the way they interface with the system. Developers and users should become acquainted with SMP programming concepts to take advantage of the new nodes. SMP experts who are just starting with RS/6000 SP have to learn about RS/6000 SP concepts like Massively Parallel Processing (MPP) or Single Point of Control to feel comfortable with a *shared nothing* system built up from *shared everything* nodes.

Although the look is the same, the mechanisms involved (which are often hidden behind commands and scripts) are sometimes different for the new nodes. The purpose of this section is to detail some of these changes to develop some understanding of the *undercover details* of the process of installing and monitoring the High Nodes.

1.3.1 High Nodes Installation Process



High Nodes Installation Process

1 - Prepare the CWS: ***bos.rte.mp & devices.rs6ksmp.base filesets available in lppsource***

2 - Configure PSSP: ***Installation of PSSP 2.2***

3 - Enter SDR data: ***Implicit detection of high nodes in frame***

- setup_server ***rs6ksmp boot kernel created in /tftpboot***

4 - NetBoot

Install mksysb image
kernel transparent
Customize AIX kernel
reinstall kernel if necessary
Reboot

The installation of High Nodes follows exactly the same process as other nodes. This process is now very well-known on the RS/6000 SP. Some steps have changed with the introduction of PSSP Version 2 Release 2. These changes, common to all node types, are described in Chapter 2, “New Installation Methodology” on page 73. In the following paragraphs, we concentrate on installation topics directly related to High Nodes.

1.3.1.1 Prepare the Control Workstation

AIX 4.1.4 is required at the Control Workstation. Within the /spdata preparation at step 11, the AIX 4.1 LPP images should be downloaded to the /spdata/sys1/install/*name*/lppsource directory. At this point, it is worth ensuring that the bos.rte.mp and *devices.rs6ksmp.base* are included. This is necessary for the proper creation of the NIM spot and for installation of the High Nodes. With a new installation, you could just copy them with the rest. If you are adding High Nodes to an existing system, it is possible that these filesets are not there.

1.3.1.2 Install PSSP Version 2 Release 2

Install at least PSSP Version 2 Release 2. It is the mandatory level to be able to install and control SMP nodes.

1.3.1.3 Enter Site Environment, Frame, Node, and Switch Information

There are no changes in the way information is entered to the SDR. SMP-specific attributes are automatically gathered from the hardware either at SDR reinitialization (*SDR_config*), or at node boot. The various wrappers called from the setup_server script create all of the required NIM resources and objects related to all network and machine types you have installed.

Creation of the SPOT resource for NIM creates boot files in /tftpboot. The naming convention for these files is:

```
spot_<spot>.<arch>.<net>
```

where:

- *<spot>* is the name of the created spot.
- *<arch>* refers to the machine architecture the boot file is intended for. It can have a value of *rs6k*, *rs6ksmp*, or *rspc*.
- *<net>* refers to the network used with value in *ent*, *tok*, or *fddi*.

In the past, as all nodes were POWER2 nodes, only the *rs6k* boot file was used. In the current version, based on the processor_type found in the SDR for a given node, when turning it into a NIM client and allocating resources to it, setup_server will allocate the right boot file for a node.

Setup_server is now more modular than in previous versions, and calls individual Perl scripts (*wrappers*) to perform specific tasks. The *mknimclient* wrapper that turns the node to a NIM client will launch a command similar to the following example for a High Node:

```
nim -o define -t standalone -a platform=rs6ksmp -a if1=sp_net \
speth21 000043269F78 ent
```

Thus, High Nodes are attached to an MP kernel boot image. This is done by a link established in /tftpboot between a file having the reliable hostname of the node (speth21 in this example) and the proper boot file:

```
speth21 -> /tftpboot/spot_aix414.rs6ksmp.ent
```

1.3.1.4 Power ON and Install the Nodes

The netboot process for High Nodes is the same as for the wide and thin nodes, with a particular emphasis on the part played by the pssp_script file.

Since both the MP node type and UP node type are part of the same system, it is required that the *mksysb* image created on one node type is used to install the other type. Although the MP kernel can execute on the UP system, its performance on a single CPU machine will be less than that of the UP kernel. You need a mechanism that, after installing the mksysb image, will compare the installed kernel type with the required type, and if they differ, will load from the Control Workstation and install the necessary kernel.

This mechanism involves the following steps:

1. After the netboot process is started from Spmon or from the Hardware Perspectives GUI, automatic node conditioning is done. In the case of the High Nodes, it involves activating the BUMP menu by the node supervisor, choosing the set flags option, setting the proper flags (the important ones

being *enable bump console* and *enable fast boot*), then choosing Ethernet as the boot device, and switching the power on.

> **Note**
>
> If you have never seen an SMP install process, you can easily watch the *conversation* between the node supervisor and the BUMP by opening a read mode only *s1term*: to the node immediately after starting the net boot process, using the command:
>
> ```
> s1term <frame> <node>
> ```
>
> where
>
> ```
> <frame>
> ```
>
> and
>
> ```
> <node>
> ```
>
> point to the booting node.
>
> Do not try to open the console, since it is a write-enabled terminal and will interfere with the boot conversation.

2. NIM then takes care of the installation of the node by transferring the appropriate boot kernel. Through the network, this boot kernel installs the image pointed to by the mksysb NIM resource allocated to the node. This mksysb image can have an MP or a UP kernel in it.

3. The *pssp_script* is executed on the node before the first *normal* boot. The pssp_script tests the node type by issuing the

   ```
   bootinfo -T
   ```

   command to the still-running boot kernel. It then checks the type of kernel installed from the mksysb image by using the

   ```
   lslpp
   ```

   command. If the appropriate kernel is not installed, the lppsource directory is net mounted from the Control Workstation and the missing *bos.rte.<processor_type>* fileset is installed by the

   ```
   installp
   ```

   command. At this step, pssp_script also configures the BUMP diagnostics flags to allow a proper reboot of the node. See the next section for more information.

4. pssp_script checks, transfers, and executes user customization files for the system or the network (script.cust and tuning.cust).

1.3.2 High Node Boot Process

The boot process of an RS/6000 SMP system is different for a uniprocessor RS/6000 system due to the part played by the BUMP processor during the *Init* phase. For more information on the complete boot process and the operations occurring at each phase change, refer to *IBM RISC System/6000 SMP Servers Architecture and Implementation*, SG24-2583.

The BUMP processor acts according to the values of parameters called *diagnostic flags* and read from nvram. The flag values can be set during the stand-by phase, when the system is powered off, by direct dialog through the

BUMP console. The BUMP nvram can also be updated from AIX diagnostics menus or by executing the

```
/usr/sbin/mpcfg
```

command. However, when the system is started, the nvram flags are reset to default values. Therefore, any changes made are valid only for the next boot.

The following extract from the *pssp_script* shows how the flags are set to ensure proper boot after installation. The same commands are executed by */etc/rc.sp* every time the node is booted.

Extract from pssp_script:

```
if test $proctype = ”mp”
then
     echo ”Setting mp configuration flags.”

     #disable the autoservice ipl flag to allow the Maintenance menu to appear
     /usr/sbin/mpcfg -cf 2 0

     #enable the bump console
     /usr/sbin/mpcfg -cf 3 1

     #disable the dial-out authorization
     /usr/sbin/mpcfg -cf 4 0

     #set mode to normal when booting
     /usr/sbin/mpcfg -cf 5 1

     #set EMS from service line
     /usr/sbin/mpcfg -cf 6 1

     #disable the multi-user service boot
     /usr/sbin/mpcfg -cf 7 0

     #disable the extended tests
     /usr/sbin/mpcfg -cf 8 0

     #disable the Power On Tests in Trace Mode
     /usr/sbin/mpcfg -cf 9 0

     #disable the Power On Tests in Loop Mode
     /usr/sbin/mpcfg -cf 10 0

     #enable the fast ipl
     /usr/sbin/mpcfg -cf 11 1

     #save the results in /etc/lpp/diagnostics/data/bump
     /usr/sbin/mpcfg -s
fi
```

The assumption is that the user is not changing these values, and thus setting them by /etc/rc.sp will ensure proper next boot.

Special attention should be paid to the

```
#enable the fast ipl
/usr/sbin/mpcfg -cf 11 1
```

command. The full Power-On test sequence of SMP is very long (it may take more than 15 minutes on configurations with large amounts of memory, disks, and other attached devices). Setting the *fast ipl* flag to 1 disables most of these tests, so the boot is much faster.

If something fails during the boot of an SMP node, the flags will be reset to defaults, and that means that the next boot may be much longer.

In such a case, we may turn to direct conversation with the BUMP. This can be achieved with the following steps:

- Power off the node from the front panel display.
- Open a console window to the node.
- Press *Enter*. You should now see a ″>″ prompt on the screen. That means that BUMP is waiting for your commands.
- Type

  ```
  sbb
  ```

  You should now see the *stand-by* menu.
- Choose option 1 *set flags*.
- When the flag menu appears, set the flags you need.
- Exit the menus until the ″>″ prompt appears again.
- Set the *node key* to the boot mode you need.
- Power the node on.

1.3.3 System Monitoring and Control

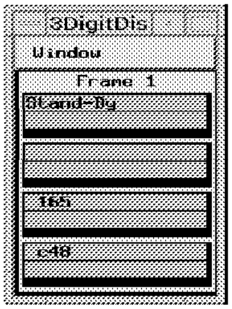
System Monitoring and Control

Hardware Monitoring (SPmon)

- SMP 2x16 LCD display interfaced by 3DigitDisplay
- SMP BUMP console operator panel replaced by node supervisor card: operator panel is front panel
- SMP node environmental variables not accessed

Performance Monitoring

- PTX PE monitors SMP High Nodes



Stand-By

165

c48

The hardware control of the High Node is done entirely from the Control Workstation, as is the monitoring of the other nodes. This can be done from node-related actions of the *Perspectives* GUI or from the different *SPmon* displays.

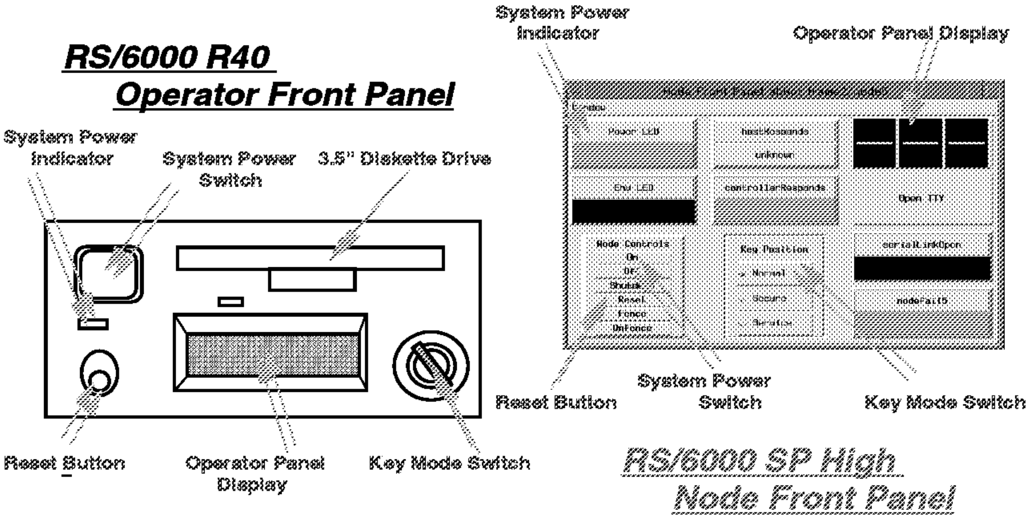
The LCD display from the original SMP operator panel, which is 2 lines by 16 characters in size, is captured by the 3DitDisplay both in Perspectives and in SPmon.

The SPmon *front panel display* replaces all the other aspects of the original operator display, as shown on the next foil.

Network monitoring for High Nodes is the same as for the other types of nodes. Note that the switch is supported only as a TCP/IP network device.

PTX is used to monitor the performance of the nodes. Use PTX/PE to group and average the performance information.

Hardware Control



RS/6000 R40 Operator Front Panel

RS/6000 SP High Node Front Panel

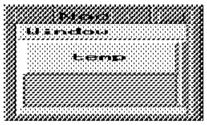
Front Panel Display

The front panel display replaces the functions of the operator panel on the RS/6000 SMP server. Note, however, that if we want to see the full 2x16 LCD SMP display, we have to open the 3DigitDisplay. The front panel′s display LED window can show only three digits.

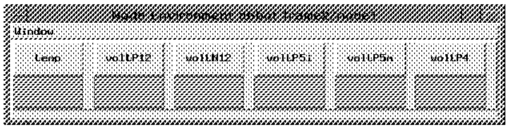
While most of the original SystemGuard functions are preserved on the High Nodes, three of them, namely *call home*, *battery backup*,and *cluster power control*, are not supported.

All of the nonsupported functions can be performed on a system level, since the High Node is part of the whole RS/6000 SP complex. Battery backup should be done for all frames, and call home and power control are provided from the Control Workstation.

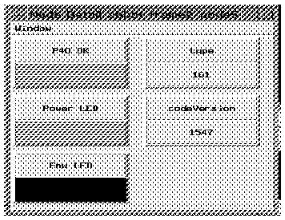
Hardware Monitoring



High Node
environment layout



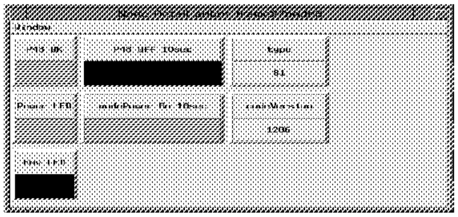
Thin/Wide node
environment layout



type
161

codeVersion
1547

High Node
detail layout



type
81

codeVersion
1206

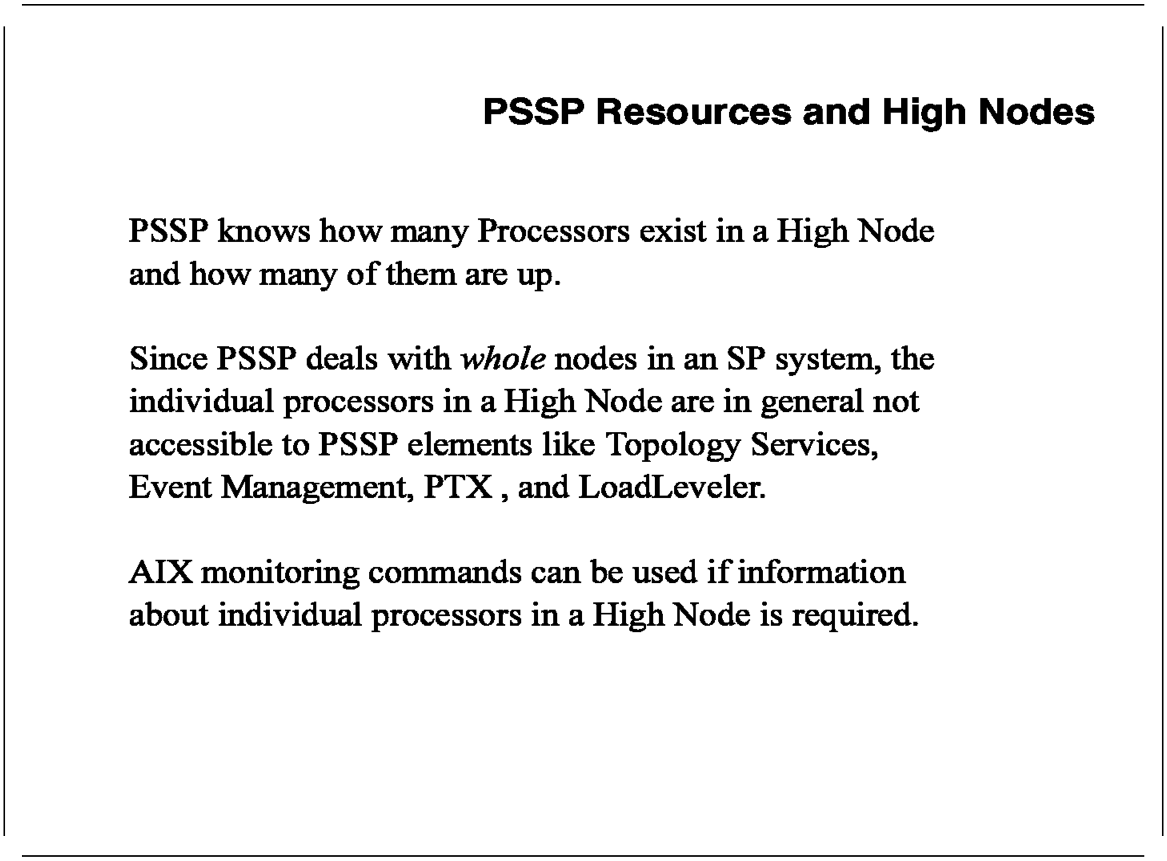
Thin/Wide node
detail layout

Environmental Layouts

The High Node supervisor card interacts with the SystemGuard processor and not directly with the hardware as in the case of the other node types. Therefore, node environmental variables accessed for Thin and Wide Nodes cannot be accessed for the High Nodes.

SystemGuard monitors the operational environment of the node and, if required, takes appropriate action.

1.3.4 PSSP Resources and the High Nodes



PSSP Resources and High Nodes

PSSP knows how many Processors exist in a High Node and how many of them are up.

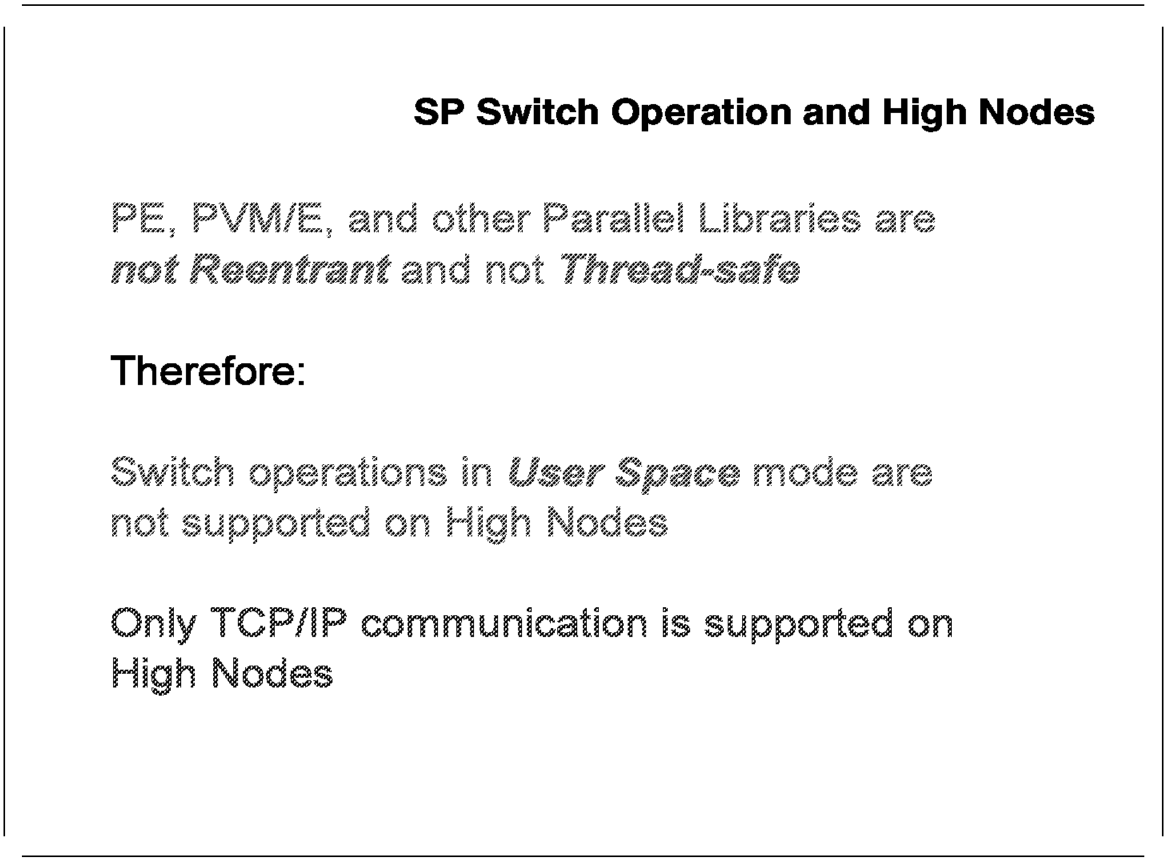
Since PSSP deals with *whole* nodes in an SP system, the individual processors in a High Node are in general not accessible to PSSP elements like Topology Services, Event Management, PTX , and LoadLeveler.

AIX monitoring commands can be used if information about individual processors in a High Node is required.

The High Node, although different in architecture, is a member of the whole RS/6000 SP system. PSSP Version 2 Release 2 treats the High Node the same as it treats any other node. That means that information that is specific to one of the processors inside a High Node is usually not monitored by Parallel System Support Programs.

Although PSSP Version 2 Release 2 knows the type of the node, the number of processors installed, and even the number of running processors, it cannot tell you about, for example, the CPU load of the second processor.

If such information is required, you have to login to the node and use the MP tools that are part of AIX.



SP Switch Operation and High Nodes

PE, PVM/E, and other Parallel Libraries are ***not Reentrant*** and not ***Thread-safe***

Therefore:

Switch operations in ***User Space*** mode are not supported on High Nodes

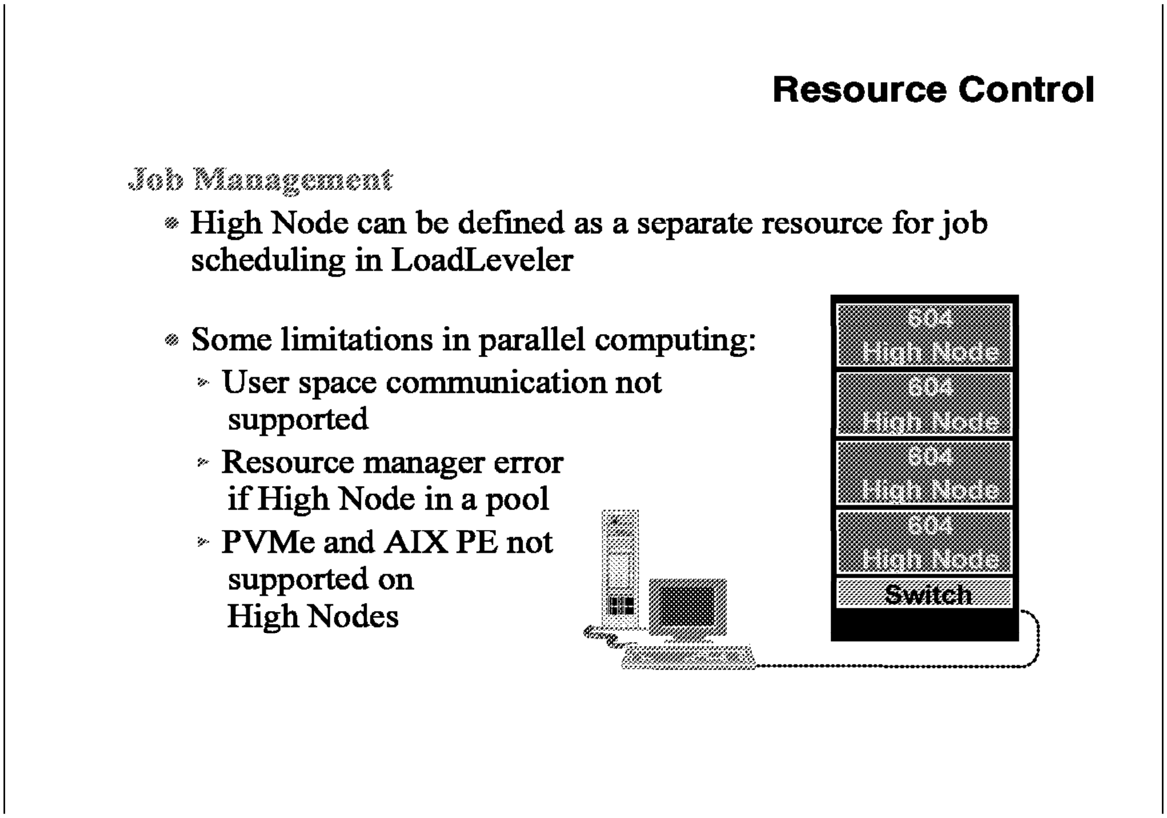
Only TCP/IP communication is supported on High Nodes

As pointed out in the SMP programming introduction, development and API libraries that are not thread-safe cannot be supported on an SMP computer.

Requests like *reentrance* and the use of *thread safeness* have some impact on the performance of functions, especially when used in a UP environment.

The MPP environment libraries included in packages like PE and PVM/E were meant to give the best possible performance in engineering/scientific *floating point* numeric intensive parallel applications, by using the message passing programming model over the High Performance Switch or the new SP Switch. These libraries are not thread-safe, and therefore, the *user space* communication over the switch, accessed through these libraries, is not supported on the High Nodes.

The TCP/IP protocol communication libraries and the corresponding switch drivers are thread-safe and can be used on all nodes, including the High Nodes.



Resource Control

Job Management

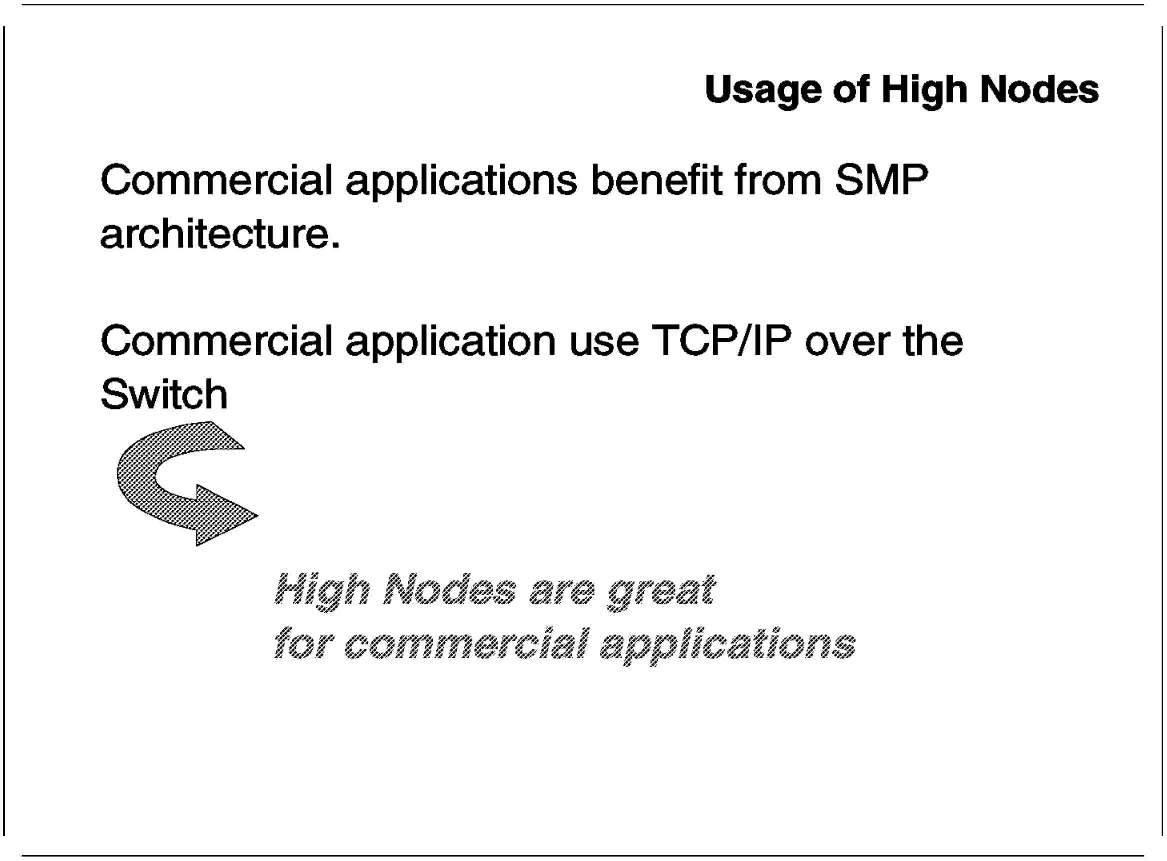
- High Node can be defined as a separate resource for job scheduling in LoadLeveler
- Some limitations in parallel computing:
  - User space communication not supported
  - Resource manager error if High Node in a pool
  - PVMe and AIX PE not supported on High Nodes

604 High Node
604 High Node
604 High Node
604 High Node
Switch

The new types of threaded applications you can run on the High Nodes makes this node a programming resource of a different class. Therefore, High Nodes can be defined as separate resources in LoadLeveler′s job scheduling.

On the other hand, the fact that *user space* communication is not allowed in High Nodes means that they cannot be defined in same *pool* as other node types. The Resource Manager will report an error if you try to add a High Node to a pool.

1.3.5 Conclusion



Usage of High Nodes

Commercial applications benefit from SMP architecture.

Commercial application use TCP/IP over the Switch

High Nodes are great for commercial applications

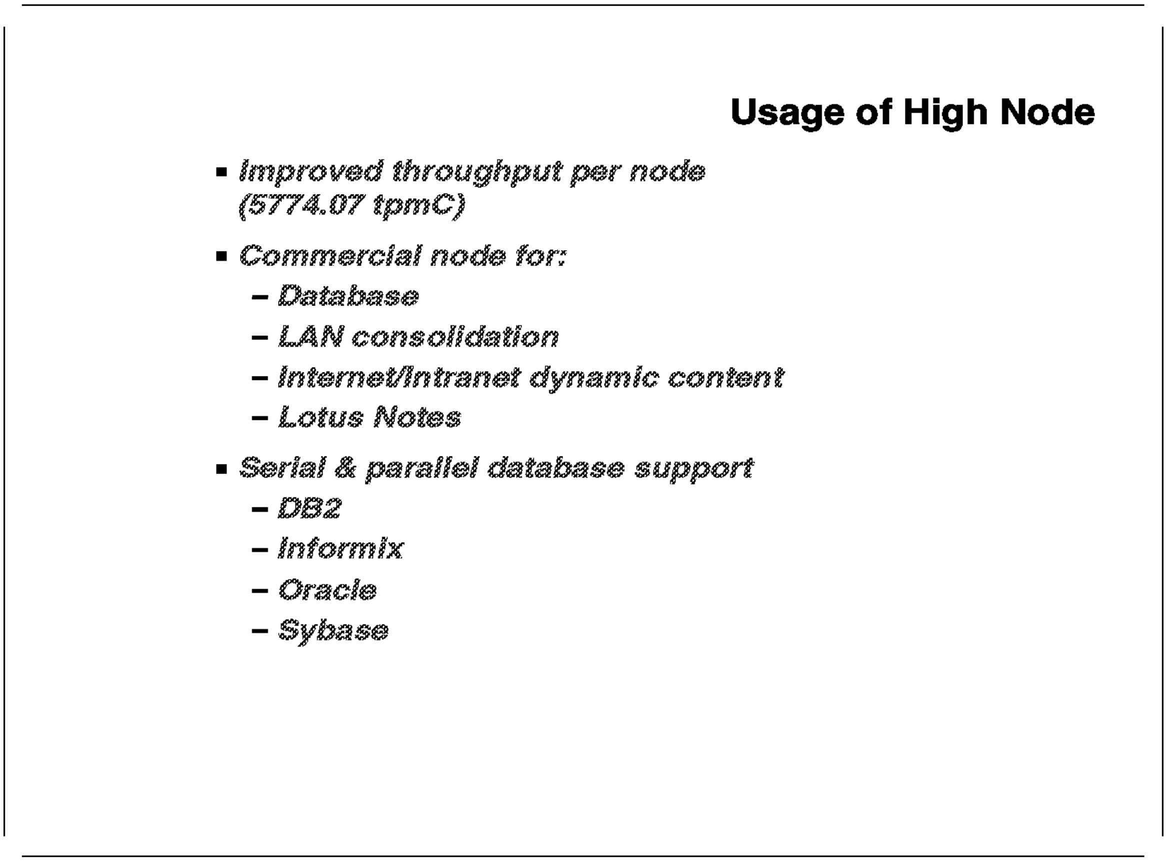
The new High Nodes introduce the SMP architecture and programming model to the RS/6000 SP.

Most commercial applications benefit from the SMP model; the great success of the RS/6000 SMP line of servers proves this.

In many commercial applications, such as databases, Internet type, and others, fast communication between server nodes running the application is crucial. All applications of this type use TCP/IP as the communication protocol.

TCP/IP over the High Performance Switch or the SP Switch, as a communication protocol between RS/6000 SP nodes, produces outstanding performance characteristics.

High Nodes are both SMP computers and members of the RS/6000 SP, taking advantage of its features like high speed communications, VSD, centralized administration, and more. In other words, the RS/6000 SP with High Nodes is a great choice for commercial applications.



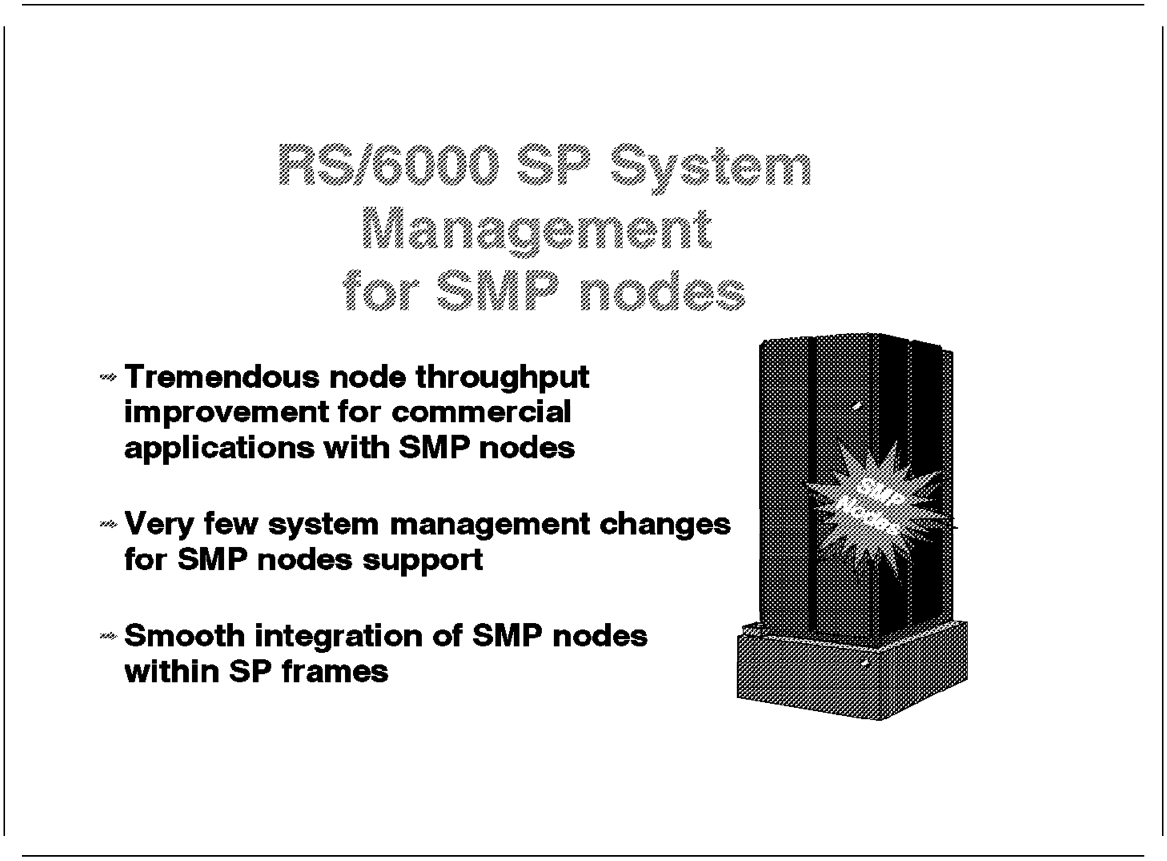
Usage of High Node

- *Improved throughput per node (5774.07 tpmC)*
- *Commercial node for:*
  - *Database*
  - *LAN consolidation*
  - *Internet/Intranet dynamic content*
  - *Lotus Notes*
- *Serial & parallel database support*
  - *DB2*
  - *Informix*
  - *Oracle*
  - *Sybase*

This summarizes and positions the High Node, and reminds us again that the RS/6000 SP is the premium choice solution for the following customers:

- Scientific/engineering customers looking for an MPP type number crunching supercomputer
- Commercial customers requiring a supercomputer for data warehousing and data mining
- Any other customers who need two or more high performance servers closely coupled together with a very fast interconnect layer and centralized administration and management.

1.4 System Management for SMP Nodes



RS/6000 SP System
Management
for SMP nodes

- Tremendous node throughput improvement for commercial applications with SMP nodes
- Very few system management changes for SMP nodes support
- Smooth integration of SMP nodes within SP frames

The introduction of Symmetric Multiprocessor (SMP) nodes on the RS/6000 SP platform opens new market opportunities for this system. It is a major breakthrough within the world of SMPs clustering. In commercial bids, the competitors of the RS/6000 SP often were SMP clusters with poor interconnection. With the introduction of SMP nodes and the new SP switch, the RS/6000 SP now combines both shared memory high computation capacity and massively parallel computing connection efficiency.

SMP nodes bring a change in architecture, as former nodes were all built on the POWER2 concept. PowerPC and its related packaging are different. Sensible changes in the system management were encountered by RS/6000 administrators in the use of desk-side and rack-mounted RS/6000 SMP servers.

Major changes in the system approach, caused by SMP introduction, were not acceptable due to the installed base of the RS/6000 SP systems. Thus user interaction and system administrator tasks still conform to the usual RS/6000 SP philosophy.

Nevertheless, minor changes were necessary to integrate SMP nodes within the RS/6000 SP system management structure. The new hardware announcement also provides the opportunity to announce new software and system management improvements from which all nodes, whether POWER2 or PowerPC-based, will benefit.

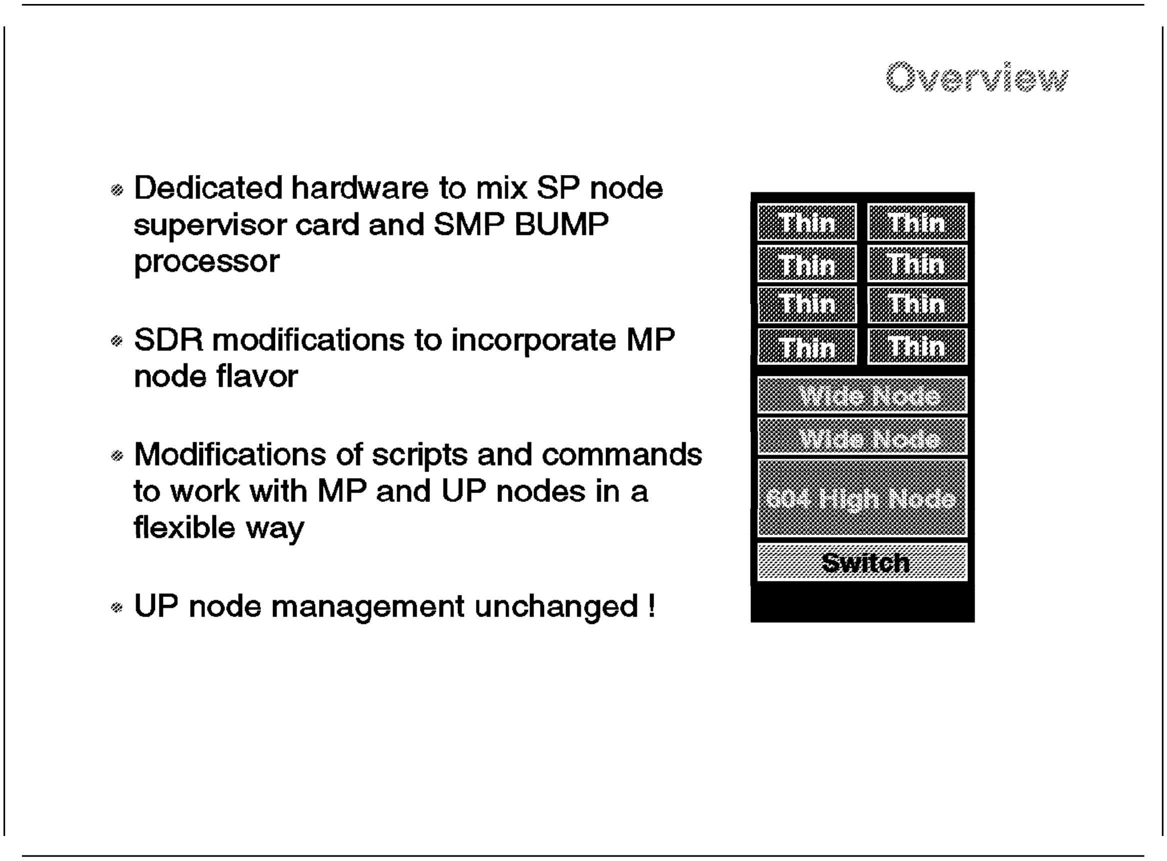
This section focuses on changes and improvements that are related to SMP nodes in conventional RS/6000 SP system management tasks.

System management is not a single application. The operating system and system infrastructure provide the base for a range of system management applications encompassing system administration and operations, system monitoring and control, and resource control. These applications are accessed through standard interfaces. A way of structuring system management is thus infrastructure first, then interfaces, and finally system management tasks.

This structure has been used since the launching of the RS/6000 SP and gives one view on system management. It has one drawback: it does not emphasize that *availability* is the key element within the system management in general and the RS/6000 SP system management in particular. Most monitoring tasks are now performed towards availability, and the implementation of availability relies on a strong infrastructure base.

The following pages describe the improvements brought to the elements of this structure to support SMP nodes.

1.4.1 Overview



Overview

- Dedicated hardware to mix SP node supervisor card and SMP BUMP processor
- SDR modifications to incorporate MP node flavor
- Modifications of scripts and commands to work with MP and UP nodes in a flexible way
- UP node management unchanged !

Thin
Thin
Thin
Thin
Thin
Thin
Thin
Thin
Wide Node
Wide Node
604 High Node
Switch

As mentioned before, the focus will be kept on changes brought to RS/6000 SP system management due to the introduction of SMP nodes. There are two types of changes:

- Necessary modifications to deliver the same level of information and control as with the other POWER2 nodes
- Specific improvements to cope with the introduction of a new multiplicity factor on processors included in a single node

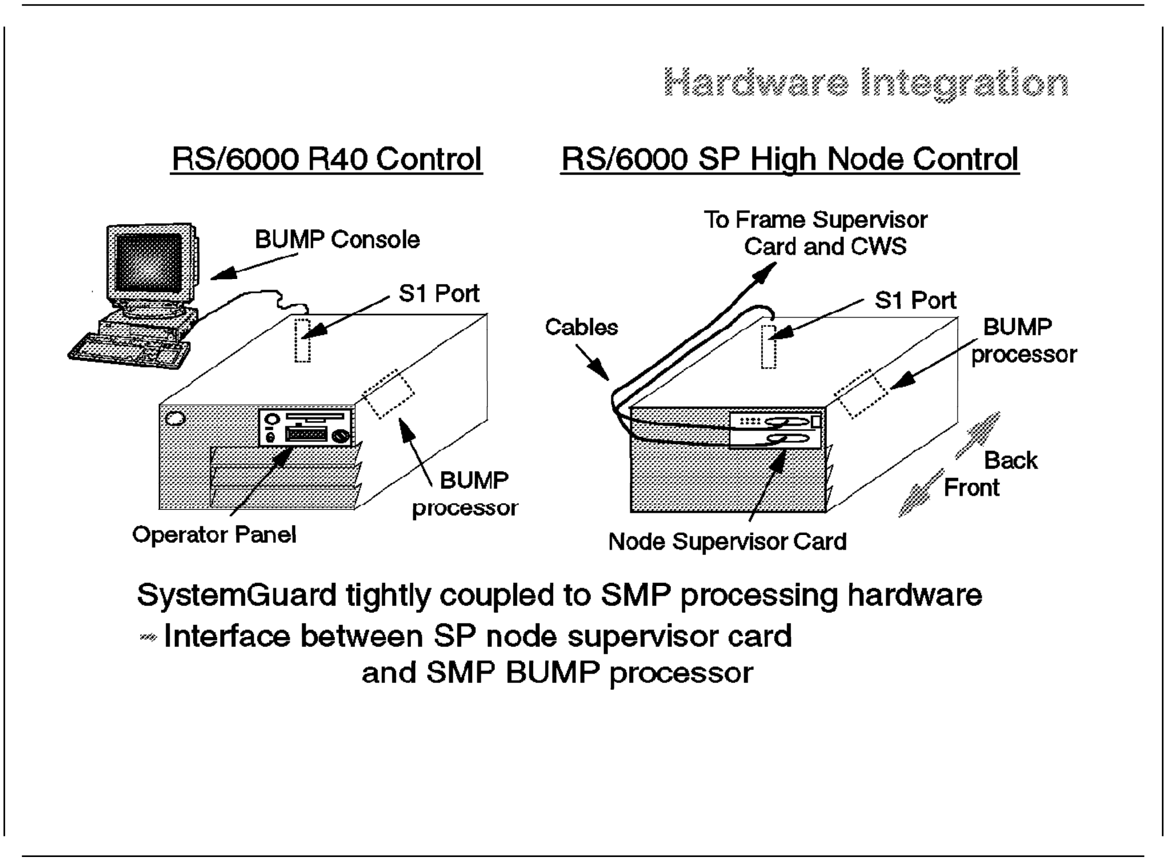
Former users of RS/6000 SP systems will not find tremendous modifications in the way they interface with the system. However, former SMP J40 and R40 users may need to learn about RS/6000 SP concepts like Massively Parallel Processing (MPP) or Single Point of Control to feel comfortable with the RS/6000 SP.

RS/6000 SP system management has not dramatically changed since the introduction of SMP nodes. Mechanisms involved, and hidden behind commands and scripts, are sometimes done in a different way. The purpose of this section is to detail some of these changes that should be transparent but may occasionally appear.

This section describes the hardware set up to allow system management of SMP nodes. Then it describes the software infrastructure where SMP nodes are declared as such (that is SDR in fact). The section ends by browsing the various system management tasks where close interaction with node architecture is concerned. This includes installation, debugging, and hardware monitoring.

1.4.2 System Management Infrastructure

1.4.2.1 Hardware Integration for System Management



Hardware Integration

RS/6000 R40 Control

RS/6000 SP High Node Control

SystemGuard tightly coupled to SMP processing hardware

⇝ Interface between SP node supervisor card and SMP BUMP processor

This foil aims at comparing the hardware elements used to access, for system management purposes, both a rack-mounted RS/6000 SMP system and an RS/6000 SP High Node:

- On the left side are the key elements of the rack-mounted RS/6000 SMP system. They are:
  - The BUMP processor handling pre-boot system management and post-boot monitoring.
  - The operator panel for direct operation on the system hardware. It is further described in 1.3.1, “High Nodes Installation Process” on page 41.
  - The BUMP console hooked to the S1 port allowing remote access to the system. This means BUMP processor management interface before system boot, BUMP processor messages during boot, and AIX access after boot.
- On the right side are the key elements of the RS/6000 SP High Node. Remember of the RS/6000 SP Single Point of Control philosophy and the corresponding elements: Control Workstation, frame supervisor card, and node supervisor card. The picture of the High Node explains that these

elements are there as they are on every other node, and it also explains how they cope with rack-mounted SMP existing elements. It emphasizes the fact that the node supervisor card simply interfaces with the BUMP processor. This is important for future High Node system management. RS/6000 SP software, PSSP, controls the BUMP in the conventional operations, but the operator will have to know more about the BUMP functions to manage the RS/6000 SP.

The usual RS/6000 SP accesses to the node, by the node front panel and s1term, are still performed through the node supervisor card.

RS/6000 SP System Management Hardware Infrastructure: The RS/6000 SP system management philosophy is implemented by the Control Workstation. It is a single point of control that allows remote administration using a Centralized Management Interface. The goal is to access the RS/6000 SP resources in a secure and parallel way. This masks the complexity of the cluster architecture, thus simplifying the tasks of the administrator.

The Control Workstation is hooked to each RS/6000 SP frame by a serial link, allowing access through the frame supervisor card to a node supervisor card in every node. The PSSP software manages this serial link to allow both physical access to hardware resources as well as operations on the hardware. It synthesizes, up to the graphical user interface point of view, the front panel that is available on a normal RS/6000, but masked on an RS/6000 SP node. This is the only available link at early stages of the installation of a node when no operating system code is installed on it.

Once a minimum installation has been performed on the node, allowing TCP/IP to work, a more conventional network, Ethernet by default, is used to transfer an important amount of data to the node. When the node is up and running, distributed commands can be remotely executed through the same conventional network. Maintenance and hardware monitoring are still using the serial link.

Both serial and network links are secured by the use of an authentication mechanism that allows the requester to prove its identity to the service it requests.

SMP Server System Management Hardware Infrastructure: Users of the SMP servers with the RS/6000 product line may already be familiar with remote control. It relies partly on the same requirements and principles that were used when the SystemGuard service processor was designed for SMP systems. This device may not be familiar to RS/6000 SP users, and so a list of its characteristics is given. For more information on SystemGuard, refer to *IBM RISC System/6000 SMP Servers Architecture and Implementation*, SG24-2583.

SystemGuard is a group a hardware components including:

- A microprocessor called BUMP (Bring-Up MicroProcessor) with its EPROM and its RAM
- A couple of EPROM as Flash and Backup
- An operator panel
- Links with system I/Os (MCA bus, serial/parallel ports, storage devices)
- Links with SMP calculation processors and main memory

These components are thus tightly coupled with the SMP processing hardware and they provide functions as:

- Initialization process flow management
- Local as well as remote control of the system
- Console mirroring
- Dial-out to a support center in case of system boot failure
- Run-time surveillance

The basic elements allowing local and remote access are the first level interfaces in basic system management regarding RS/6000 SMP systems:

- The BUMP processor is the ″heart″ of SystemGuard. Service processor acts in different ways depending of the status of the SMP system. When the system is powered off, it gives access to specific tests and configuration for the next boot. When the system is booting, the BUMP controls the hardware tests and loading of the AIX operating system. When AIX is running, a heartbeat mechanism allows the BUMP to fulfill a surveillance function.
- The operator panel is the minimum interface to the BUMP, including:
  - Power button is used to switch on or off the power supplies of the system. The BUMP processor has its own power boundary and is not affected by this Power button.
  - Reset button is used to reset SystemGuard to the Init phase, that is, the reinitialization of the system depending on the key position.
  - LCD display allows you to visualize information depending on the state of the system. It is an improved 3DigitDisplay made of 2 x 16 characters.
  - Physical key is used to manage behavior of the system during initialization. It has the three usual positions: Normal, Secure, and Service.

  The operator panel gives ″physical″ access to the system, but for remote software access to be achieved, through the BUMP console for example, it is recommended that elements stay in a stable position: Power button ON and Physical key on Normal position.
- The S1 Port is the port to which the BUMP console is attached.
- The BUMP console is an ASCII terminal attached to the S1 serial port providing the normal input to the BUMP. Once AIX is running on the system, the BUMP console is then the AIX console allowing you to logon into the system.

The administrator interfaces with SystemGuard through these elements when the system is powered off. But access to SystemGuard parameters for visualization or modification is still possible when AIX is running. This is done using a set of AIX commands. This allows the operator to perform specific tasks without having to stop the system. This is normally done using the BUMP console on the S1 port.

A service console can be hooked to the S2 serial port. It is usually a remote terminal accessing the system through a Hayes compatible modem for remote assistance in case of system failure. Mirroring of this service console to the BUMP console can be achieved by allowing the customer to see what remote assistance is doing on the system. Work with SystemGuard and AIX is possible from the Service Console.

1.4.2.2 Operating System and Software Infrastructure

Operating System and Infrastructure

Operating System

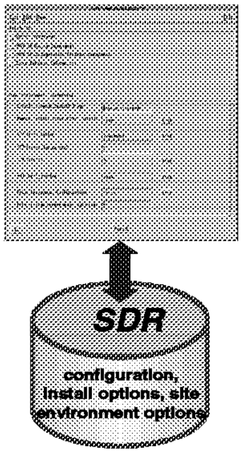
- AIX 4.1.4 and PSSP V2.2 required on SMP node and boot/install server (CWS)

System Data Repository

- "processor_type" and "processors_installed" attributes added to Node class

Reliability / Availability

- new event management and problem management framework supporting SMP nodes extended components granularity ("procs_online" variable)



- PSSP 2.2 is the mandatory level of code to control SMP nodes on the RS/6000 SP. As with the former evolution of the PSSP software, if a given level of AIX and PSSP have to be installed on a node, at least the same level has to be installed first on its boot/install server. The boot/install server can be another node or the Control Workstation.
- Modifications in the *Node* class of the SDR were necessary for proper PSSP management. Two new attributes have been added to the *Node* class:
  - *processor_type* contains the processor type of the node. It can be *UP* or *MP*. This value is used for the management of the node by PSSP commands and scripts.
  - *processors_installed* contains the number of processors actually installed on processor cards inside the node. The field value is *1* for UP nodes and can range from *2* to *8* on SMP nodes.

  Like most of the SDR information, the management of these attributes is transparent to the administrator. The way the attributes values are set varies:

  - *processor_type* is set by the SDR_config routine based on the node supervisor card type. The SDR_config uses the serial connection to the frames to collect hardware information about the component installed. It is executed at system installation to initialize the SDR. It has to be re-executed each time a new hardware is installed in the frame (a new node for example) to create the necessary entries in the SDR.

  - *processors_installed* cannot be set by the SDR_config, as such granularity in hardware configuration is not accessible to the node supervisor card installed in an SMP node when the node is not running. The SDR_config routine sets processors_installed to the default value of *1*. The real value can only be grabbed on the node when IPL is passed. This will be performed at node boot by /etc/rc.sp. This script uses the processor_type value to know if the acquisition of processor_installed is necessary for the node on which it executes. On a UP node, the value is simply kept to 1.

  The processor_type attribute is queried from the SDR by any script that is running on the node or interacting with the node, and that will do things differently on a multiprocessor (MP) than on a uniprocessor (UP). One example already mentioned is the /etc/rc.sp script that searches for the need to query the processors_installed parameter, which is useless for a uniprocessor.

  The processors_installed is stored in the SDR for user information. It can also be used as a reference, to be compared with a dynamic variable named *processors_online* by the event manager. This processors_online variable actually monitors available and functional processors among those installed in a node, allowing failure detection and recovery actions.

  These attributes are neither part of the information displayed by the `splstdata -n` command nor accessible through any SPmon display. There are two ways of accessing these values for a given node:

  - Within the **Hardware Perspectives** of the **Perspectives** tool:

    Choosing a node and displaying its attributes, the Configuration Tab will show the following information:

    ```
    Node number:          21
    Short hostname:       speth21
    Hostname:             speth21
    Internet address:     129.1.1.121
    Partition:            cws
    Processor_type:       MP
    Processors_installed: 4
    Frame number:         2
    Slot:                 5
    Slots used:           4
    Switch node number:   20
    Switch chip:          5
    Switch chip port:     1
    ```

  - Using the `SDRGetObjects` command:

    ```
    $ SDRGetObjects Node processor_type processors_installed
    processor_type processors_installed
    UP                     1
    MP                     4
    ```

- The new event management framework coming with PSSP 2.2 can also take the SMP processor granularity into account. For example, the resource monitor module is using an attribute, *processors_online*, to monitor the number of processors available in an SMP node at a given point in time. This information is reported to the topology services that allow other requesters, such as performance monitoring tools, to take it into account. The event manager is also notified of changes of the processor_online value so that proper actions are taken in case of processor failure within an SMP node.

The processor_online attribute is not permanently stored and is not part of the Node class in the SDR.

1.4.3 Administrative and Operations Interfaces

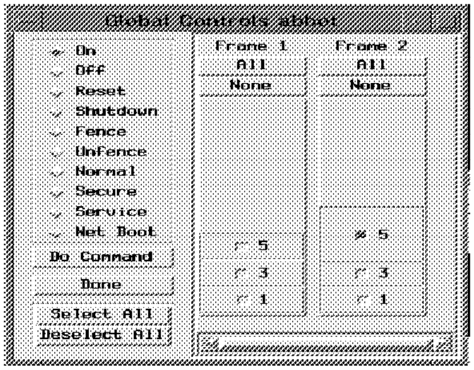
Administrative and Operations Interface

Single point of Control: CWS

- BUMP console interfaced by PSSP
- NO other remote control facility supported (CPC)
- SMP RS/6000 not supported as CWS

Simplified / Remote Administration

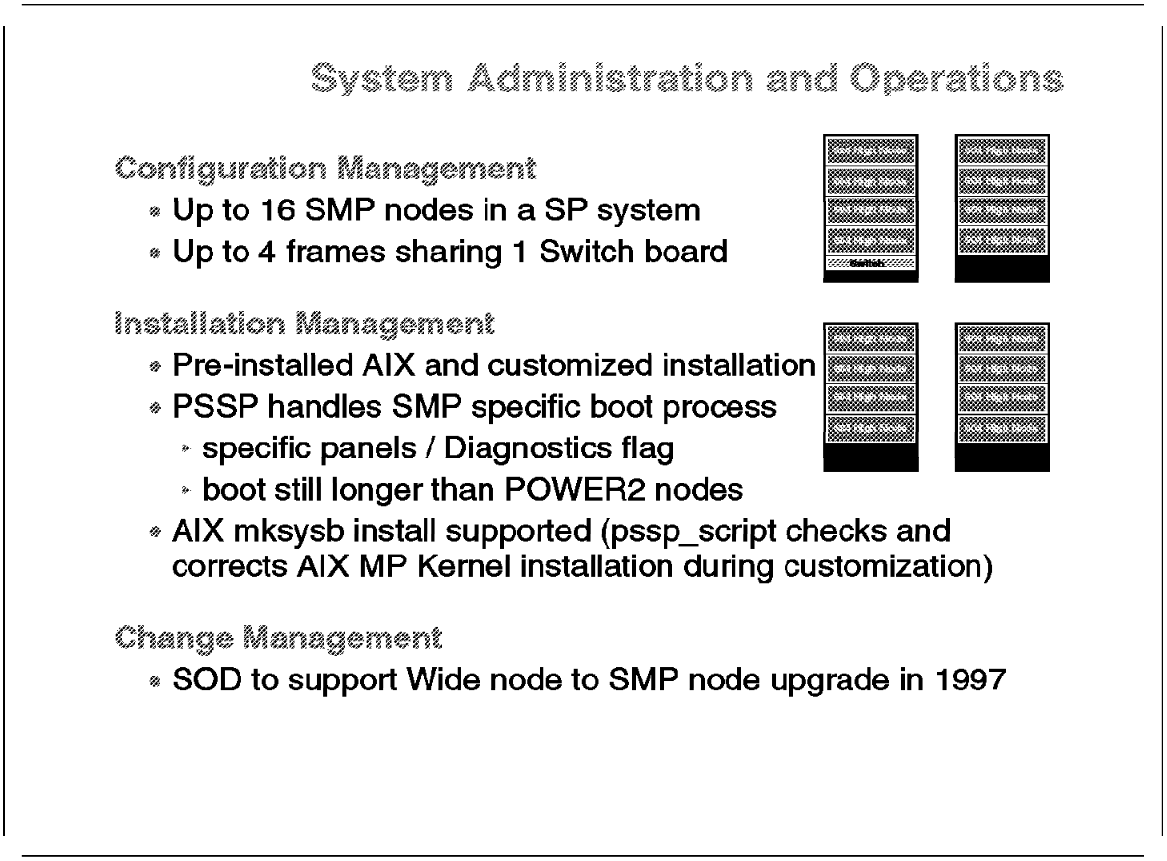
- Maintenance & Diagnostics supported through NIM



Interfaces to a given system encompass both the means to access the system once the system is up and running and the means to bring it to that state.

The RS/6000 SP system management philosophy is implemented by the Control Workstation. It is a single point of control allowing remote administration with the use of a Centralized Management Interface. The goal is to access the RS/6000 SP resources in a secure and parallel way. This masks some of the complexity of the cluster architecture and thus simplifies the tasks of the administrator.

1.4.4 System Administration and Operations



System Administration and Operations

Configuration Management

- Up to 16 SMP nodes in a SP system
- Up to 4 frames sharing 1 Switch board

Installation Management

- Pre-installed AIX and customized installation
- PSSP handles SMP specific boot process
  - specific panels / Diagnostics flag
  - boot still longer than POWER2 nodes
- AIX mksysb install supported (pssp_script checks and corrects AIX MP Kernel installation during customization)

Change Management

- SOD to support Wide node to SMP node upgrade in 1997

1.4.4.1 Configuration Management with SMP Nodes

Due to SMP nodes, there are some configuration constraints when building an RS/6000 SP system. The number of SMP nodes currently allowed in a single system is 16, but it is not a technical constraint, and it should be relieved before the end of 1996.

The obvious constraint is the size of the node itself: it takes up four slots within a frame. A High Node is addressed by the lowest slot number of the four slots that it takes up. Within a single frame system populated by four High Nodes, the nodes are numbered 1, 5, 9 and 13. Switch node numbering in the conventional case follows the same rule. Using the same system, nodes have switch node numbers 0, 4, 8, and 12.

A lot of questions are likely to arise concerning switch topology with the use of High Nodes. Apart from questions on the position used by the single port of the High Node itself, some questions will concern the use of the three unused switch ports for other nodes.

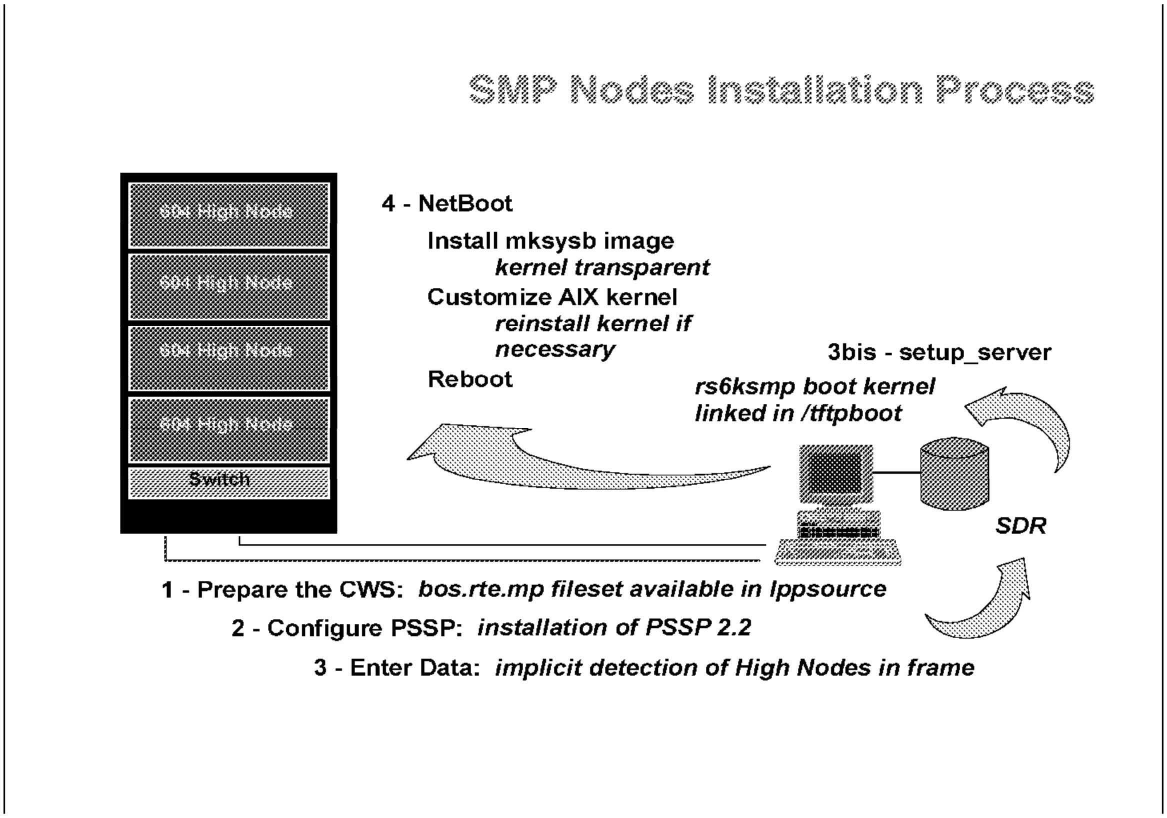
Concerning High Node switch connection, the four slots used by a High Node are in two contiguous drawers that belong to two different switch chips. You could choose to connect the High Node to either chip. This is not supported. The High

Node has to be connected to the switch in the same way that a Thin node occupying the lowest slot would be connected. A general discussion about interaction between node numbering and switch node numbering within an RS/6000 SP system is covered in 1.2.2, “Supported Configurations” on page 31.

The only solution available is to swap the High Node one drawer immediately higher or one drawer immediately lower than its current position. The swap will occur with the drawer contents, which may be two Thin nodes or a Wide node. For example, this applies when the High Node has to be in the same partition, so as to be attached to the same switch chip, as the contents of the drawer immediately below it.

Regarding the use of the other three switch connections left vacant by a High Node, there are some optimization means. It is comparable to the case where two frames of Wide nodes are able to share a single switch board located in one of them. With High Nodes, this configuration can go up to four frames sharing a single switch board located in one of them (4 High Nodes in a frame x 4 frames = 16 necessary connections to the switch).

1.4.4.2 SMP Specific Tasks in Installation Process



SMP Nodes Installation Process

4 - NetBoot

Install mksysb image
kernel transparent

Customize AIX kernel
reinstall kernel if necessary

Reboot

3bis - setup_server

rs6ksmp boot kernel linked in /tftpboot

1 - Prepare the CWS: *bos.rte.mp fileset available in lppsource*

2 - Configure PSSP: *installation of PSSP 2.2*

3 - Enter Data: *implicit detection of High Nodes in frame*

Installation of High Nodes follows exactly the same process as other nodes. This process is now very well-known on the RS/6000 SP. Some steps have changed in the PSSP 2.2 release, as described in Chapter 2, "New Installation Methodology" on page 73. In the following paragraphs, we concentrate on changes directly related to SMP nodes.

Prepare the Control Workstation: Within the /spdata preparation, at step 11, you must download the AIX 4.1 LPP images to the /spdata/sys1/install/*name*/lppsource directory. At this point, it is worth ensuring that the bos.rte.mp fileset is included. This may be necessary for the proper installation of the nodes, at a later step.

Install PSSP: Install at least PSSP version 2.2. It is the mandatory level to be able to install and control SMP nodes.

Enter Site Environment, Frame, Node, and Switch Information: There are no changes in the way information is entered in the SDR. We have already mentioned (see 1.3.1, "High Nodes Installation Process" on page 41) that SMP-specific attributes are automatically gathered from the hardware either at SDR reinitialization or at node boot. There are some changes in the way this information is reformulated in order to enable the future netboot of the nodes. This is traditionally what setup_server does.

setup_server & NIM

There are no specific NIM resources created to support SMP nodes. The creation of the SPOT resource for NIM is used to create seven boot files in /tftpboot. The naming convention for these files is:

```
psspspot.<arch>.<net>
```

where

- *<arch>* refers to the architecture with value in *rs6k*, *rs6ksmp* or *rspc*.
- *<net>* refers to the network used with value in *ent*, *tok* or *fddi*.

The naming convention has changed, because multiple spots can now be defined. It is now:

```
spot_<spot>.<arch>.<net>
```

where

- *<spot>* is the name of the spot.
- *<arch>* and *<ent>* significance and values are unchanged.

In the past, as all nodes were POWER2 nodes, only a rs6k boot file was used. Now, based on the processor_type it finds in the SDR for a given node, when turning it into a NIM client and allocating resources to it, setup_server will choose the right boot file. Because setup_server is now more modular and as such is calling other individual Perl scripts, this declaration happens in one of these Perl scripts: mknimclient. It is setup_server part that makes a node a NIM client of its boot server as specified in the SDR. It launches the following command using the *platform* variable, set before in the same Perl script using the processor_type SDR attribute:

```
nim -o define -t standalone -a platform=$platform -a if1=sp_net \
speth21 000043269F78 ent
```

Thus, SMP nodes are attached to an SMP kernel boot image; this is done by a link established in /tftpboot between a file having the reliable hostname of the node and the proper boot file:

```
speth21 -> /tftpboot/spot_aix414.rs6ksmp.ent
```

Power ON and Install the Nodes: The netboot process is the same as the usual node netboot process with a particular emphasis on the part played by the pssp_script file: it will ensure before the first boot that the node will be able to boot with a correct kernel and the correct system parameters.

There was a strong requirement to simplify image management and increase the flexibility of node cloning by allowing the installation of any mksysb coming from any RS/6000 on any node. This applies to both ways, which means that an mksysb image generated on an SMP RS/6000 or a High Node will also work on a normal POWER2 node. An MP kernel will always work on a UP system, but you may experience performance problems. pssp_script ensures that the right kernel is installed on the right node before allowing the node to reboot after installation.

The description of the mechanism involves four steps:

1. After the netboot button is clicked in spmon, or netboot action is performed on a node from the hardware perspectives GUI, automatic node conditioning

is done. Node conditioning is different with High Nodes. In this step, it is normally done transparently to the user.

2. NIM then takes care of the installation of the node by transferring and installing the mksysb image pointed to by the mksysb NIM resource allocated to the node. This mksysb image can have an MP or a UP kernel installed.
3. NIM finally executes the pssp_script file on the node before the first boot. pssp_script tests the kernel previously installed on the node and compares it to the node type. If the matching is not realized, a new suitable kernel is installed. At that step, pssp_script also configures the MP service flags to allow a proper re-boot of the node. See 1.4.4.3, “High Node Boot Process” for more information.
4. pssp_script checks, transfers, and executes user customization files for the system or the network (script.cust and tuning.cust).

pssp_script checks the installed kernel by using a `lslpp` command, and grabs the platform type by using:

```
bootinfo -T
```

This command returns rs6k, rs6smp, or rspc. pssp_script could have used another way to determine the architecture type of the node by referring the processor_type attribute in SDR. This information is already accessible at the time pssp_script executes on the node, as setup_server is setting a variable named *proctype* and includes an export of that variable in the <node_hostname>.install_info file in /tftpboot.

You can use this proctype variable for your own customization by testing its contents within the script.cust file. This allows you to perform conditional operations only on UP or MP nodes at the end of system customization. If present in /tftpboot, the script.cust file is transferred to the node and executed at the end of pssp_script. The value of proctype is *up* or *mp* in lower case.

The installation of the correct kernel is performed by pssp_script by mounting the correct lppsource directory from the Control Workstation, and by a conventional `installp` command.

1.4.4.3 High Node Boot Process

The boot process of an RS/6000 SMP system is different for a uniprocessor RS/6000 system due to the part played by the BUMP processor during the Init phase. For more information on the complete boot process and operations occurring at each phase change, refer to *IBM RISC System/6000 SMP Servers Architecture and Implementation*, SG24-2583.

The BUMP processor is acting based on the values of parameters set either during the Stand-By phase, when the system is powered off, or the RunTime phase, when AIX is running. The second case is achieved using AIX commands, and parameters changed will be effective for next system reboot.

This is the mechanism pssp_script is using to set SystemGuard parameters at the end of a node installation before the first boot, to allow the node to boot correctly from the RS/6000 SP point of view. The AIX command used here is the `mpcfg` command. It may be useful to know the modified flags and understand the reasons for their modification. In case of abnormal boot, the operator will be able to check and correct these flags. The rest of this section describes the meanings of the flags and then mentions the ways to access and modify them manually.

PSSP-Customized Diagnostics Flags: The following is an extract from pssp_script:

```
if [[ $proctype = "mp" ]]
then
     echo "Setting mp configuration flags."

     #disable the autoservice ipl flag to allow the Maintenance menu to appear
     /usr/sbin/mpcfg -cf 2 0

     #enable the bump console
     /usr/sbin/mpcfg -cf 3 1

     #disable the dial-out authorization
     /usr/sbin/mpcfg -cf 4 0

     #set mode to normal when booting
     /usr/sbin/mpcfg -cf 5 1

     #set EMS from service line
     /usr/sbin/mpcfg -cf 6 1

     #disable the multi-user service boot
     /usr/sbin/mpcfg -cf 7 0

     #disable the extended tests
     /usr/sbin/mpcfg -cf 8 0

     #disable the Power On Tests in Trace Mode
     /usr/sbin/mpcfg -cf 9 0

     #disable the Power On Tests in Loop Mode
     /usr/sbin/mpcfg -cf 10 0

     #enable the fast ipl
     /usr/sbin/mpcfg -cf 11 1

     #save the results in /etc/lpp/diagnostics/data/bump
     /usr/sbin/mpcfg -s
fi
```

All the mpcfg commands included in pssp_script apply to the Diagnostics flags of SystemGuard. Diagnostics flags are used to control the service, diagnostics, and maintenance from a customer point of view.

The syntax of the mpcfg command used here is the following:

```
mpcfg -c -f <index> <value>
```

where:

- *-c* means changing the flags.
- *-f* means the Diagnostics flag list.
- *<index>* means the index of the flag in the list.
- *<value>* means the binary (0 or 1) value set to the flag. If not specified, 0 means disable and 1 means enable.

Here is a list of Diagnostics Flags and some of their meanings:

1. *remote authorization*
2. *autoservice ipl*

   When enabled, booting with the key in Service mode will go to Diagnostics screen. Go to the Maintenance menu so you can disable it.

3. *bump console*

   When enabled, the LED codes and BUMP messages are displayed on the console during the Init phase. This is what you want here to monitor that on the s1term. If disabled, it is like a regular RS/6000: no code and no messages are displayed on the console during the Init phase. Only AIX messages appear when the system starts loading AIX.

4. *dial-out authorization*
5. *normal mode when booting*
6. *EMS from service line*
7. *multi-user service boot*
8. *extended tests*
9. *Power On Tests in Trace mode*
10. *Power On Tests in Loop mode*
11. *fast IPL*

    When enabled, SystemGuard will skip the Power-On Self Tests (POST). By default it is disabled and, if enabled, this will only last one reboot. As we want the boot to be as fast as possible, we enable it.

Finally, once the flags have been reset, the `mpcfg -s` command saves the parameters/flag values from NVRAM to the /etc/lpp/diagnostics/data/bump file.

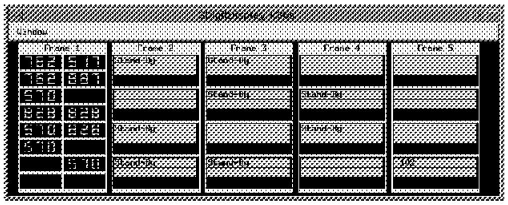
At normal system boot, that is, after system shutdown and not after installation, no one can foresee manipulations that could have changed the BUMP flags during system initialization. This is the reason why the set of commands included in pssp_script detailed earlier have also been added to /etc/rc.sp. This ensures that the flags are reset properly during each boot.

Changing the Diagnostics Flags at Boot Time: If something fails during the boot of an SMP node, the symptoms and ending state may be cross-checked with one of the behaviors described in the former paragraph about Diagnostics flags. It means that the flag does not have the correct value and that it has to be manually changed to allow correct boot.

Chapter 2. New Installation Methodology

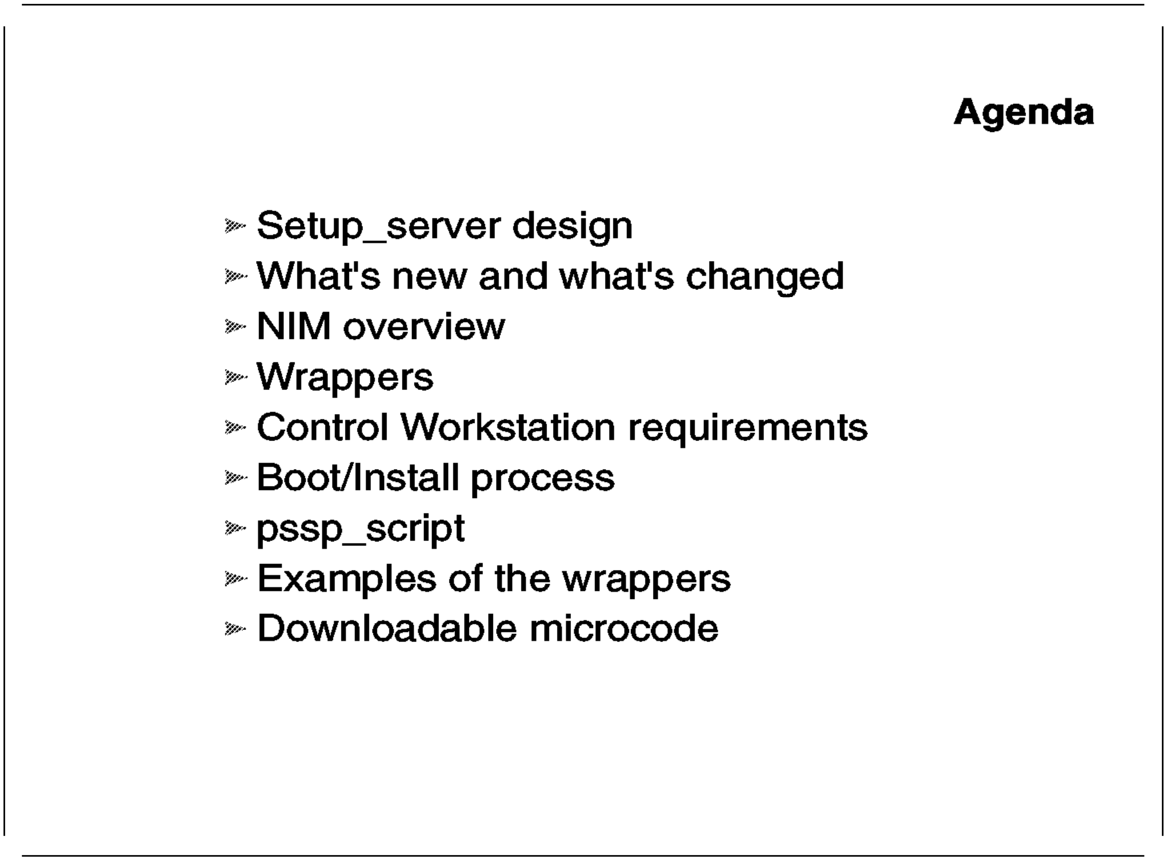
New Installation Methodology

Setup_server



This New Installation Methodology is the summary for all new installation-related topics of PSSP V2.2. This chapter discusses these topics in more detail.

2.1 Agenda

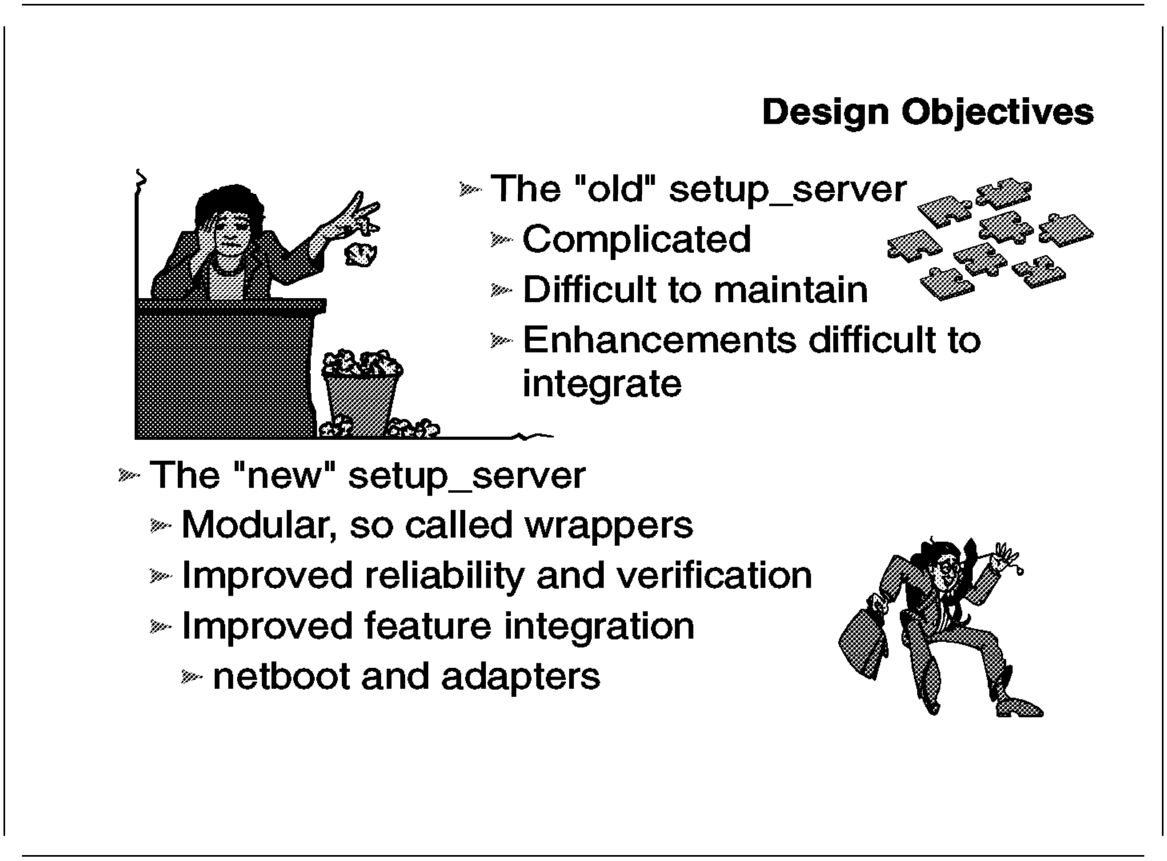


Agenda

- Setup_server design
- What's new and what's changed
- NIM overview
- Wrappers
- Control Workstation requirements
- Boot/Install process
- pssp_script
- Examples of the wrappers
- Downloadable microcode

The section describes the new installation methodology involved with the new PSSP Version 2 release 2.

2.2 Design Objectives



Design Objectives

- The "old" setup_server
  - Complicated
  - Difficult to maintain
  - Enhancements difficult to integrate
- The "new" setup_server
  - Modular, so called wrappers
  - Improved reliability and verification
  - Improved feature integration
    - netboot and adapters

The setup_server command is the main part that was changed with regard to the New Installation Methodology. New functions and new hardware reflect their changes in the setup_server command. The changes in setup_server will be discussed later in this chapter.

Apart from the functional changes, setup_server was thoroughly cleaned up, as far as internal structure was concerned. The “old” setup_server has developed into a complicated script. This led to difficulties in integrating new functions and features in following releases.

The new version of setup_server is chopped into modular sections. The setup_server script is divided in two major sections:

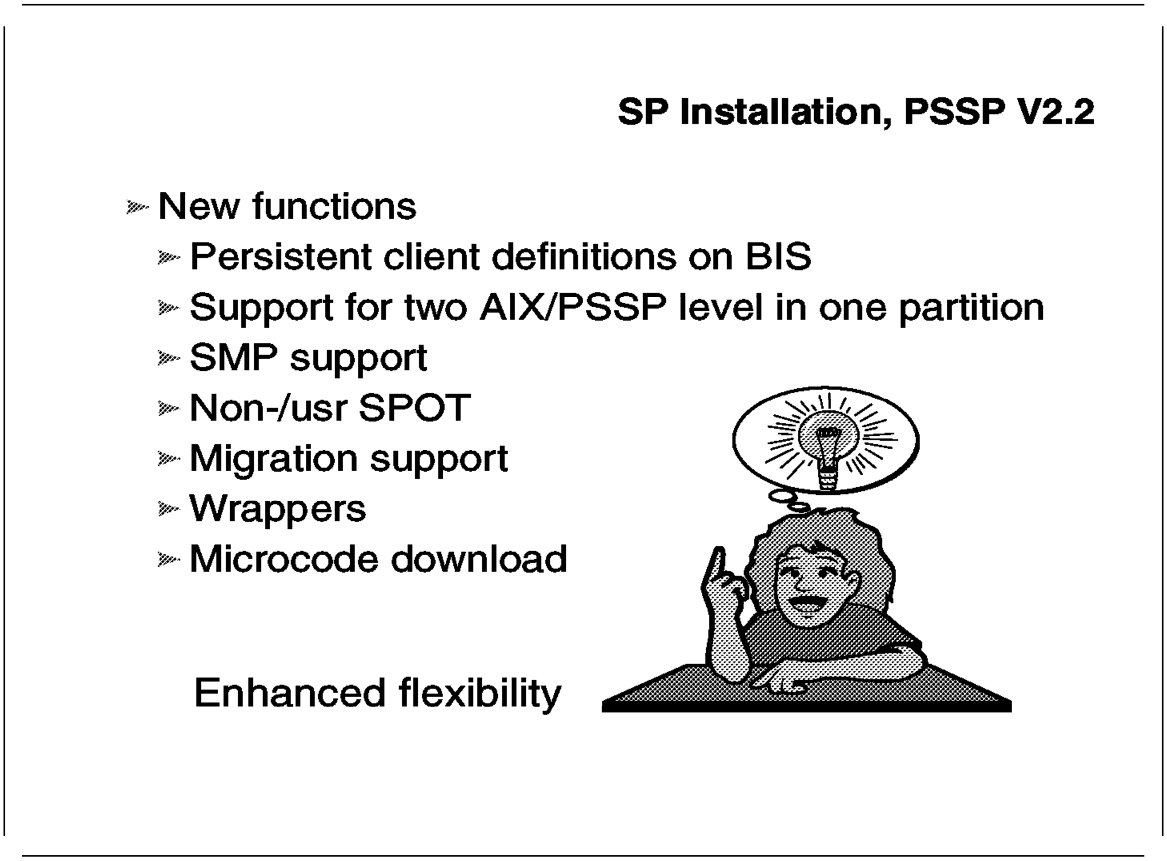
- The section that runs on all nodes and the Control Workstation
- The section that only runs on boot-install servers

The last section in the current version of setup_server is now divided in subroutines, which in effect are new external commands. Each of these subroutines takes care of a particular task, to set up and configure a boot-install server. These external subroutines are called *wrappers*.

The wrapper approach gives us the following big advantages:

- Each wrapper has its own error messages. Previously, when `setup_server` failed, it was difficult to tell where it failed. With the wrapper approach, it is fairly easy to determine where the error occurs, and when found, the task flow can be picked up from there by executing the individual wrapper. More details on how the wrappers can be used are discussed later in this chapter.
- Improved debugging. Since each wrapper is an independent external command, the checking of flags, input parameters, and output is handled better. This means that when errors occur, a customer is capable of reporting the problem more accurately, and IBM is capable of solving the problem quickly.
- Future integration of new functions can be established in a much more graceful way than before. Since the wrappers are divided in a functional way, new features and functions are easier to integrate.

2.3 SP Installation, V2.2



SP Installation, PSSP V2.2

- New functions
  - Persistent client definitions on BIS
  - Support for two AIX/PSSP level in one partition
  - SMP support
  - Non-/usr SPOT
  - Migration support
  - Wrappers
  - Microcode download

Enhanced flexibility

The following new functions are available through setup_server:

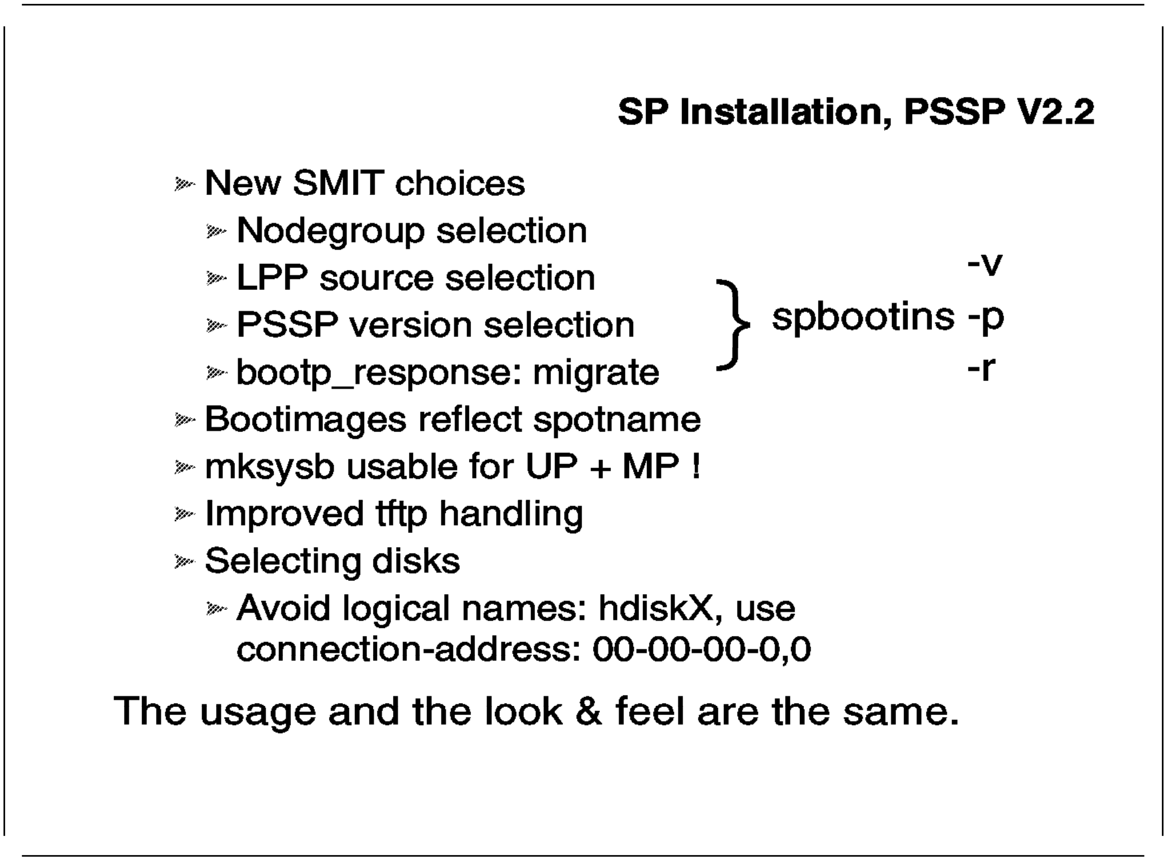
- Migrate install, spbootins -r migrate
- Nodegroup selection, spbootins -N node_group_name
- lppsource selection, spbootins -v lppsource_name
- PSSP Version selection, spbootins -p PSSP-2.2
- SMP support, setup_server, recognizes through configuration in the SDR how to set up the different install environments for the two supported processor types: uniprocessor (UP), and multiprocessor (MP).
- Enhanced tftp handling, setup_server will make sure that the following files and directories are opened for tftp usage:
  - /tftpboot, for the node-new-srvtab, node.install_config, and node.config_info files
  - /usr/lpp/ssp, for install-related commands, not yet available on the newly installed nodes, like rcmdtgt, rsh, and rcp
  - /etc/SDR_dest_info, for the destination of the Control Workstation
  - /etc/krb.conf, for the Kerberos configuration
  - /etc/krb.realms, for the Kerberos environment

  - /usr/sys/inst.images/ssp, for install images

- Persistent NIM definitions, to ensure that the NIM definitions on a boot-install server are permanently available. In the old setup_server, the NIM definitions on boot-install servers (except for the Control Workstations) would disappear after installation of the clients. The persistent definitions save time in the execution of setup_server at a later point in time and allow manipulations of the NIM definitions without having to run setup_server.

- Support for two levels of AIX or PSSP in one partition.

- non-/usr SPOT. The Shared Product Object Tree used to be a /usr-SPOT in the older releases of PSSP. This allows you to serve more than one AIX version in the SP and offers the flexibility to exclude or include packages, independently. A /usr-SPOT inherits the characteristics of the /usr filesystem it is created against. All clients installed from a /usr-SPOT will have access to the same products and optional software as the master. The SPOT contains the skeleton of an AIX installation. A SPOT provides a */usr* filesystem for network boot support. Every basic requirement of a machine to run, such as the AIX kernel, libraries, and configuration methods are located in a SPOT. Also, a SPOT contains the definitions to create the bootimages in /tftpboot. The bootimages are able to boot a machine with the bare minimum effort. The SPOT is used as the process following the initial boot process, where the SPOT is mounted from the master. LED code 612 indicates a successful mount of the SPOT, and LED code 611 indicates a failure mount from the master. When mounted, the SPOT delivers all necessary information to proceed with installations. Anyone familiar with the /usr-client functionality in AIX 3.2.5 will see the familiarities with the SPOT concept.



SP Installation, PSSP V2.2

- New SMIT choices
  - Nodegroup selection
  - LPP source selection
  - PSSP version selection
  - bootp_response: migrate
- Bootimages reflect spotname
- mksysb usable for UP + MP !
- Improved tftp handling
- Selecting disks
  - Avoid logical names: hdiskX, use connection-address: 00-00-00-0,0

} spbootins -v -p -r

The usage and the look & feel are the same.

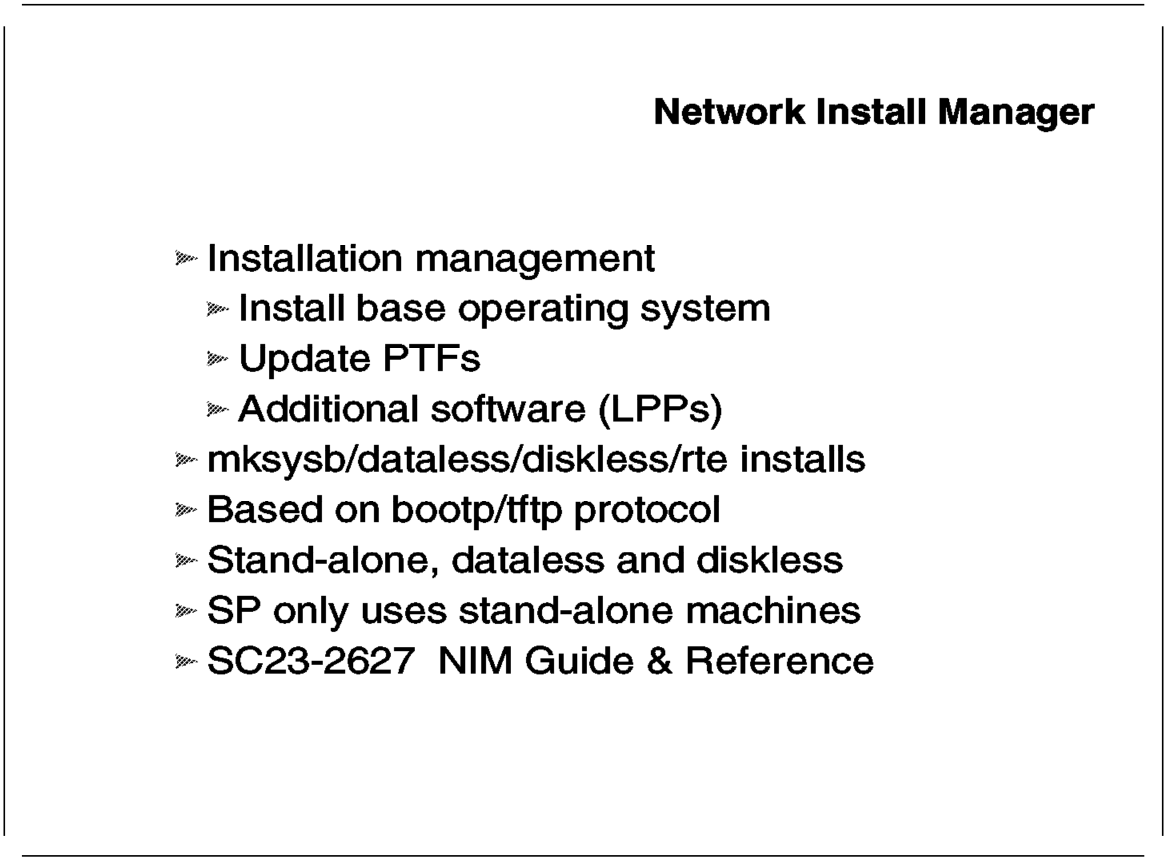
The changes in setup_server and the added functionality are also visible in the RS/6000 SP SMIT panels. For example, new selector fields in the server_dialog commandheader (smitty server_dialog) for setup_server are:

- Node Group
- LPP Source Name
- PSSP Level

In /tftpboot, the two-level version support is shown by the new naming convention of the bootimages. The bootimages reflect the name of the SPOT they belong to. For example, *aix414.rs6ksmp.ent* is the bootimage for a High Node, where the bootimage belongs to the *aix414* SPOT.

The ability to use any system backup for installation of uniprocessor *and* multiprocessor machines was only supported in NIM, but it is now also supported on the RS/6000 SP. The installation process “discovers” the processor type (in the node.install_info file). This information makes sure that the appropriate kernel will be applied to the installed machine, that is a uniprocessor with a uniprocessor kernel (bos.rte.up) and a multiprocessor with a multiprocessor kernel (bos,rte.mp).

2.4 Network Install Manager



Network Install Manager

- Installation management
  - Install base operating system
  - Update PTFs
  - Additional software (LPPs)
- mksysb/dataless/diskless/rte installs
- Based on bootp/tftp protocol
- Stand-alone, dataless and diskless
- SP only uses stand-alone machines
- SC23-2627 NIM Guide & Reference

The Network Install Manager was first introduced in the PSSP code with PSSP Version 2.1. NIM is the software in AIX that manages installations, install environments, and resource management. The basic software components that are associated with NIM are:

- bos.sysmgt.nim.master

  This fileset contains all the NIM management commands and NIM configuration definitions. It should be installed on master nodes.

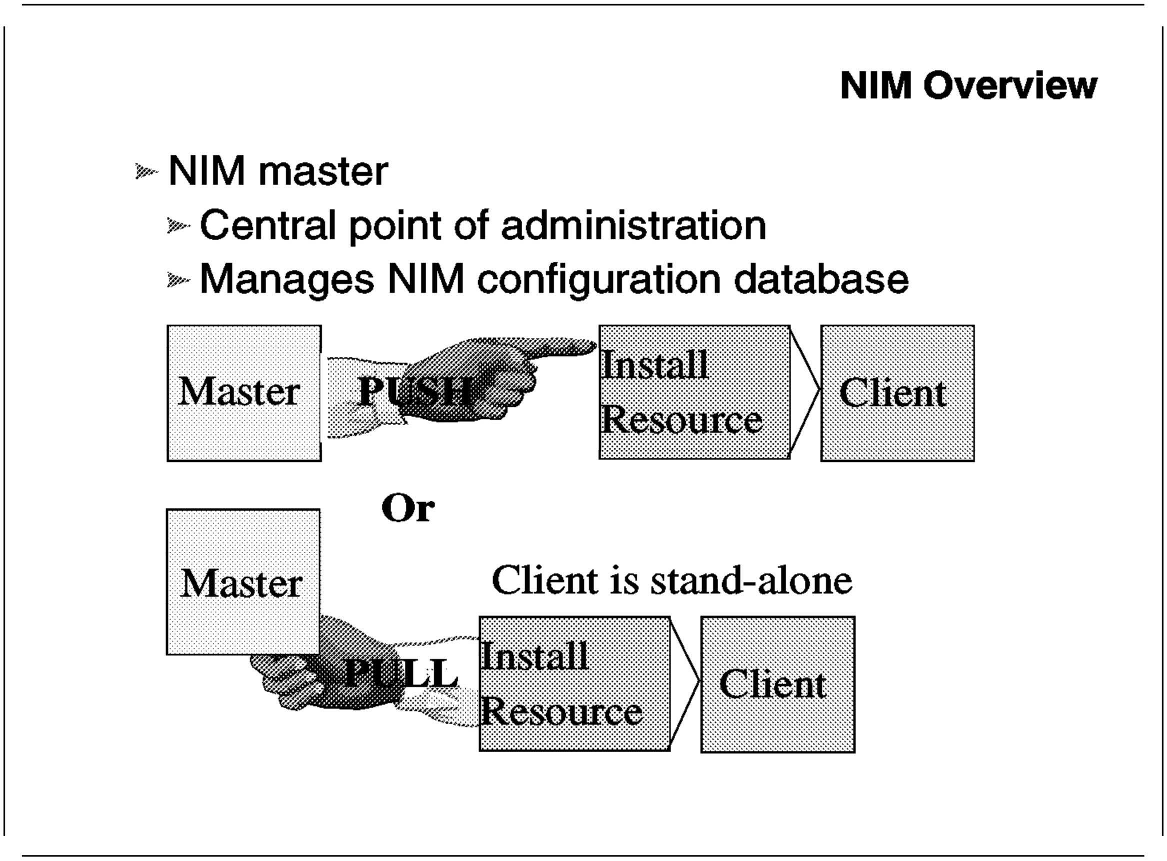
- bos.sysmgt.nim.client

  This fileset contains the NIM commands taking care of the installation of a client and the NIM installation methods. It is installed on all NIM clients.

- bos.sysmgt.nim.spot

  This fileset contains the internal commands invoked by /sbin/rc.boot. It must be installed on boot-install servers.

2.5 NIM Overview



NIM Overview

- NIM master
  - Central point of administration
  - Manages NIM configuration database

Master

PUSH

Install Resource

Client

Or

Master

Client is stand-alone

PULL

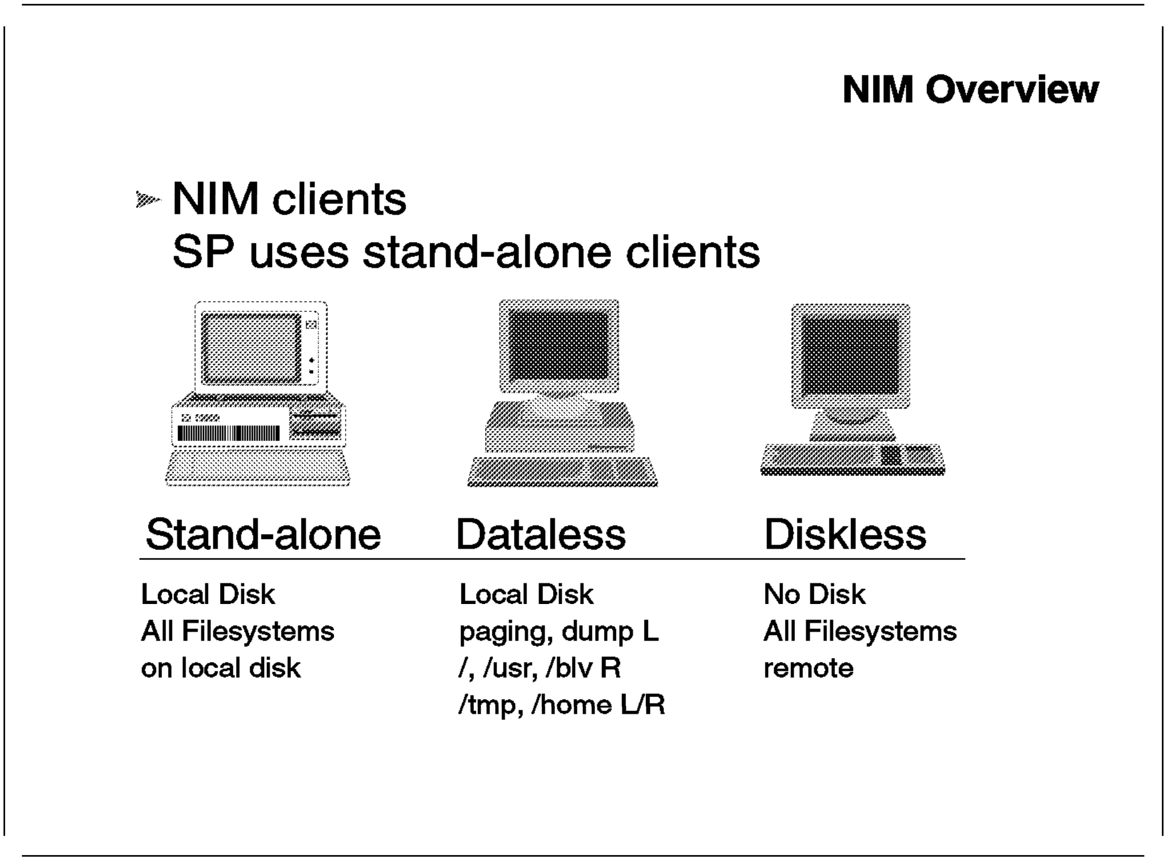
Install Resource

Client

The manager of a NIM environment is the NIM master. The master contains all NIM definitions and is the central point of administration. The RS/6000 SP, however, can host several NIM masters. In the case of multiple NIM masters, there is more than one administrator. The Control Workstation, however, will always remain the source of all information. If more than one NIM master is configured, the Control Workstation, apart from being a NIM master itself, will function as a Resource Server for all other NIM masters.

NIM supports two ways of installing machines:

1. **Push mode** is used when NIM installs dataless or diskless machines. Generally, push mode installations are invoked by the NIM master.
2. **Pull mode** is used when NIM installs stand-alone machines. This is the mode used for RS/6000 SP node installations. Generally, pull mode means that the client invokes the install action.



NIM Overview

- NIM clients
  SP uses stand-alone clients

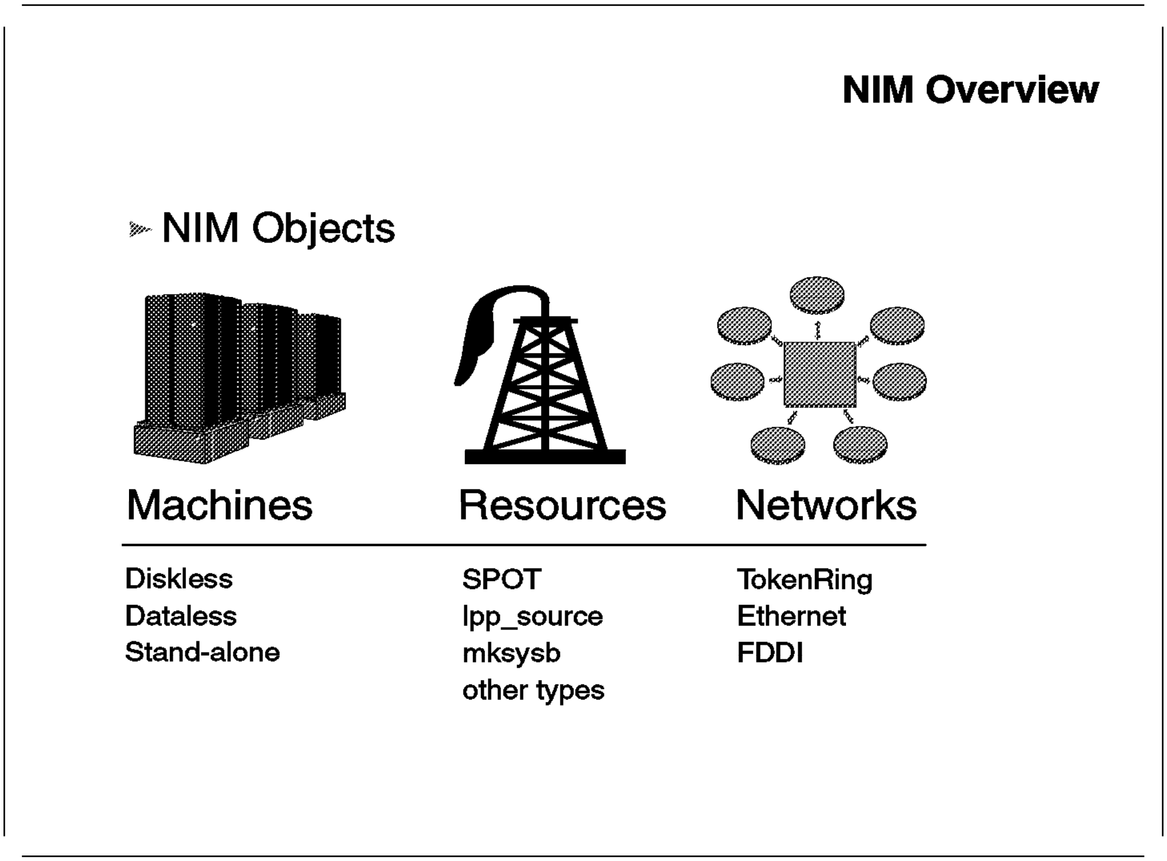
Stand-alone	Dataless	Diskless
Local Disk	Local Disk	No Disk
All Filesystems	paging, dump L	All Filesystems
on local disk	/, /usr, /blv R	remote
	/tmp, /home L/R	

NIM clients can be one of the following:

1. **Diskless**. The machine contains no disk, and all filesystems and logical volumes reside on a remote machine.

2. **Dataless**. The client has a disk, but the root filesystem, the /usr filesystem and the boot-logical volume are remote. Paging space and dump device are on the local disk. All other filesystems can be remote or local.

3. **Stand-alone**. A stand-alone machine has all filesystems on the local disk. In the RS/6000 SP, the NIM environment only uses stand-alone NIM definitions.



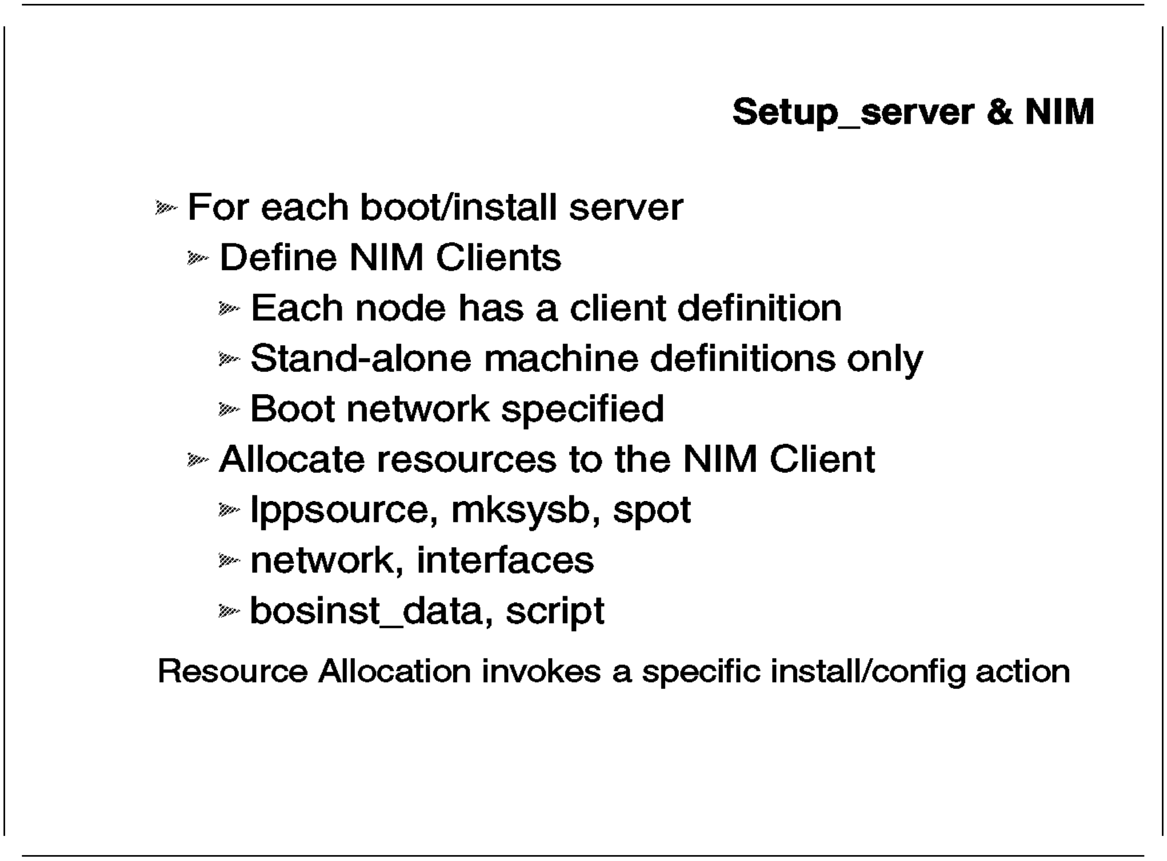
NIM Overview

- NIM Objects

Machines	Resources	Networks
Diskless	SPOT	TokenRing
Dataless	lpp_source	Ethernet
Stand-alone	mksysb	FDDI
	other types	

The NIM configuration contains three different objects:

1. **Machines**. The three different types of machine objects were discussed in the previous foil.
2. **Resources**. Resources are objects in NIM containing vital installation information, filesets, system backups, and SPOTs. In order to install a machine, the particular machine object, or NIM client definition, will be allocated to one or more NIM resources. Each resource has its unique type definition, and each type invokes a specific install action.
3. **Networks**. Network objects in NIM determine how a NIM client can be reached over a network and which interfaces are involved in finding the NIM client. A network resource defines the routing information for a NIM client at the moment the client is booted for installation.



Setup_server & NIM

- For each boot/install server
  - Define NIM Clients
    - Each node has a client definition
    - Stand-alone machine definitions only
    - Boot network specified
  - Allocate resources to the NIM Client
    - lppsource, mksysb, spot
    - network, interfaces
    - bosinst_data, script

Resource Allocation invokes a specific install/config action

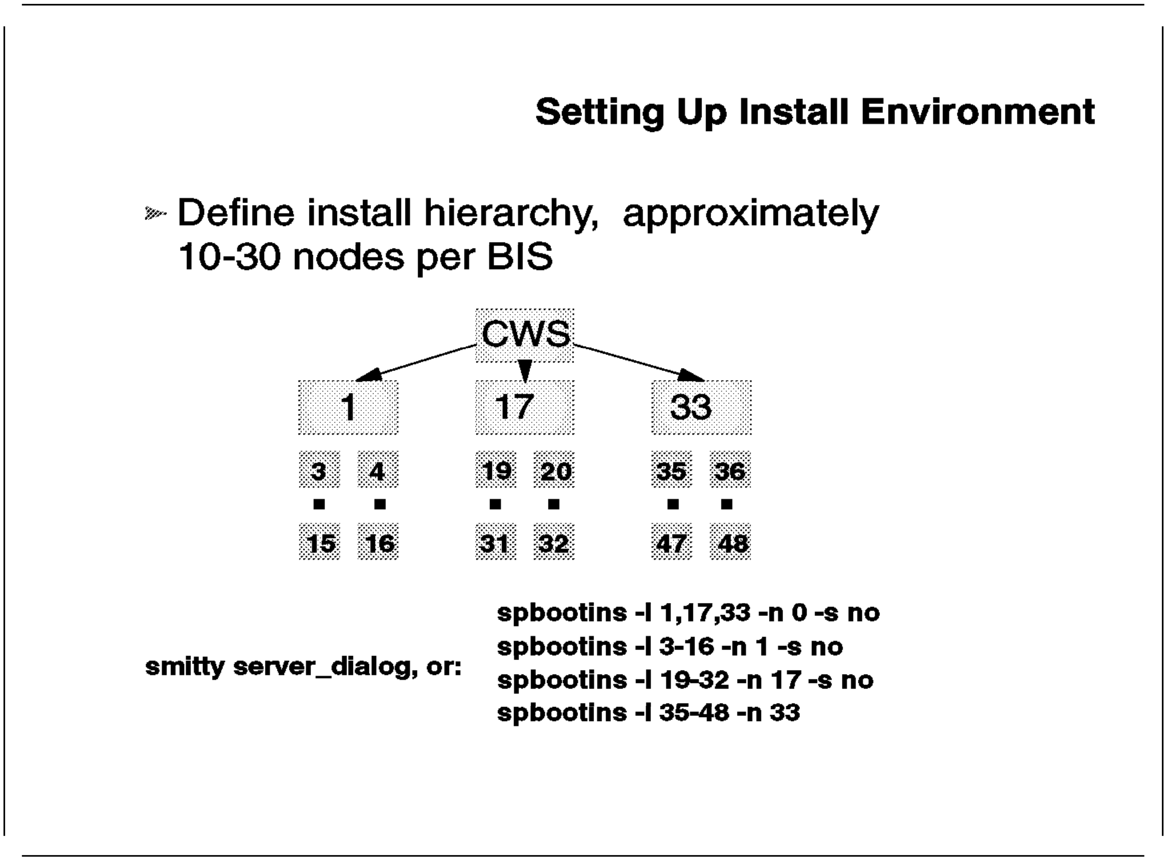
Each node in an RS/6000 SP will appear on a NIM master as a NIM client. The following NIM objects can be allocated to NIM clients in an RS/6000 SP:

- **SPOT**, for installation assistance. The SPOT will be created in the directory /spdata/sys1/install/AIXVERSION/spot.
- **boot**, representing the boot-image in /tftpboot.
- **mksysb**, representing the system backup image in /spdata/sys1/install/images for initial installation of AIX.
- **lppsource**, representing all filesets of AIX necessary to build the final AIX level on the node. The lppsource object is created in /spdata/sys1/install/AIXVERSION/lppsource.
- **noprompt**, the installation data specifying that the installation will happen without keyboard assistance. A noprompt install is used for node installations. The file associated with noprompt installations is stored as /spdata/sys1/install/pssp/bosinst_data.
- **prompt**, the installation data specifying that installation happens with manual input. The prompt resource is allocated to a node when a node needs to be booted for diagnostics or maintenance, as when the bootp_response is diag or maint. The file associated with prompt installations is stored as /spdata/sys1/install/pssp/bosinst_data_prompt.
- **migrate**, the installation data specifying that installation happens without destroying the client′s rootvg. During a migration installation, the installation process installs the filesets of the new AIX version, matching or overriding

the filesets currently installed. Migration installations will remove /usr/lib/microcode, /usr/lib/methods and /dev, so non-AIX drivers must be reinstalled. The file associated with migrate installations is stored as /spdata/sys1/install/pssp/bosinst_data_migrate.

- **nim_script**, representing the customization scripts of NIM. The files associated with the nim_script resource are stored in /exports/nim/script.
- **psspscript**, representing the SP-specific customization script, taking care of SP-specific installation actions.
- **spnet_enX**, representing the network definition to use when installing the NIM client or node. The psspscript′s file is stored as /spdata/sys1/install/pssp/pssp_script.

2.6 Setting Up Install Environment



Setting Up Install Environment

- Define install hierarchy, approximately 10-30 nodes per BIS

smitty server_dialog, or:

```
spbootins -l 1,17,33 -n 0 -s no
spbootins -l 3-16 -n 1 -s no
spbootins -l 19-32 -n 17 -s no
spbootins -l 35-48 -n 33
```

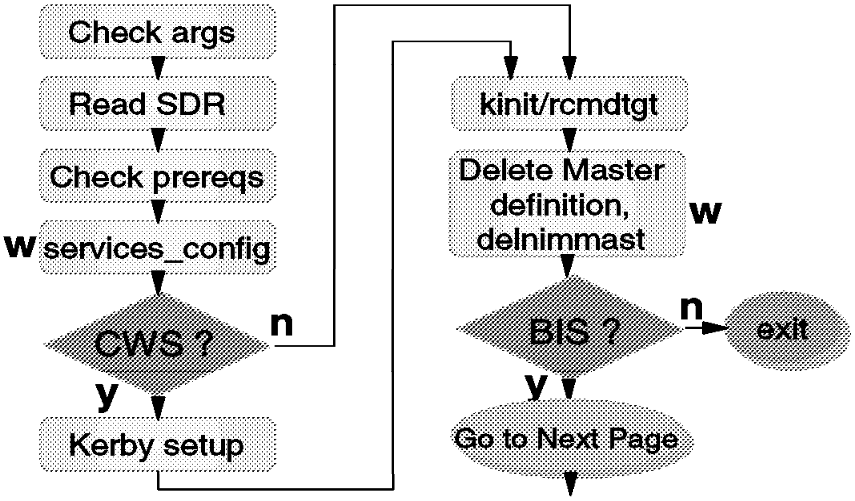
Currently, the installation environment supports up to two levels of install servers. In larger environments, 30 nodes and more, additional boot-install servers are recommended. Also, if subnetting is used to divide the RS/6000 SP Ethernet, each subnet requires a boot-install server. General rules as to how many boot-install servers should be present in an RS/6000 SP are hard to define. Approximately 10 to 30 nodes per boot-install server would be the rough estimate.

Once designed, the hierarchy should be defined in the SDR. To avoid having setup_server run at each definition step, the **-s** flag must be used. At the last step, when all hierarchy members are defined to the SDR, setup_server may start execution.

2.7 Wrappers

Wrappers

➢ Setup_server



As discussed before, setup_server was redesigned in a modular way. The modules that now form the setup_server command are called wrappers. Wrappers are functions or subroutines in the setup_server command that can also be invoked as an individual command from the command line. The setup_server command currently uses eleven new wrappers and one already existing wrapper. The new wrappers will be discussed in the next foils.

The already existing wrapper is services_config. Officially, this command was considered an internal command, but since its function is so useful as an independent command, this function is currently defined as external as well. The services_config command will remain in the /usr/lpp/ssp/install/bin directory, as with this release of PSSP. The services_config command reads the SDR information stored in the SP class. The SP class contains information that determines which services all nodes should run:

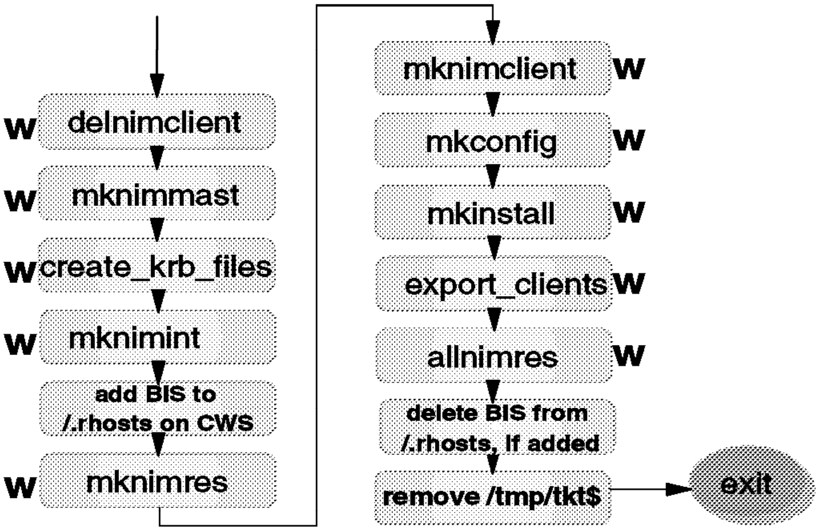
- ntp
- amd
- file collections
- accounting
- printing

The services_config command makes sure that the RS/6000 SP site configuration information will be applied to all nodes and the Control Workstation.

The setup_server command, as shown on the preceding foil, is the first part of setup_server that will run on every CPU in the RS/6000 SP, both Control Workstation and all nodes. Usually setup_server is invoked when a node is booted, but it may be invoked anytime, since setup_server is idempotent. The setup_server command will stop execution when it discovers it is not running on a boot-install server (BIS) or on the Control Workstation (CWS).

Wrappers

- Setup_server, continued...



w delnimclient

w mknimmast

w create_krb_files

w mknimint

add BIS to /.rhosts on CWS

w mknimres

mknimclient w

mkconfig w

mkinstall w

export_clients w

allnimres w

delete BIS from /.rhosts, If added

remove /tmp/tkt$

exit

The second part of setup_server will only run on boot-install servers and on the Control Workstation. Although these wrappers will only run on a BIS and the CWS, the wrappers can be invoked on every node in the RS/6000 SP system. The structure of the wrappers is such that the SDR will be queried first. The information found in the SDR will tell the wrapper where the actions of the wrapper must be executed. Depending on the SDR, the wrapper will eventually run on the applicable boot-install server.

The following wrappers are restricted to running on the appropriate boot-install server only:

- setup_CWS is a wrapper in setup_server and should only be run on the Control Workstation.
- create_krb_files
- mkconfig
- mkinstall
- export_clients

This foil shows the flowchart of all remaining (new) wrappers.

The individual wrappers will be discussed in the following sections.

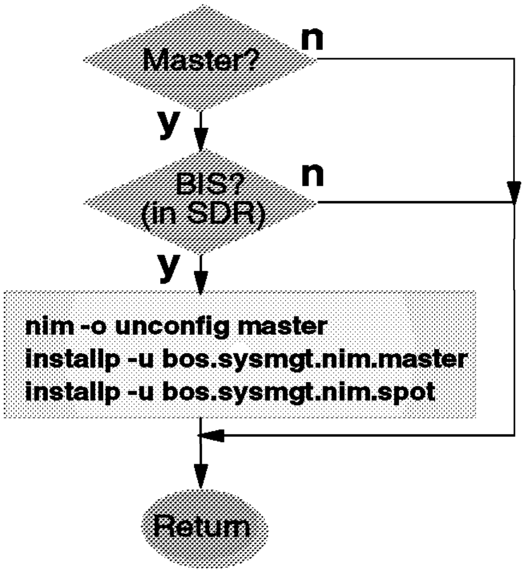
Two functions in the setup_server flow are not available as wrappers. These functions involve the creation and deletion of an entry for boot-install servers in the /.rhosts file on the Control Workstation. Some NIM commands require normal TCP/IP r-command authorization, since these commands call the rcmd() function from libc.a. The rcmd() library call is not authenticated.

The security exposure is limited to the duration of setup_server running. An alternate solution to avoid this exposure is not to use additional boot-install servers.

2.7.1 delnimmast

delnimmast

input: -l nodenumber
verify: lsnim (not found)

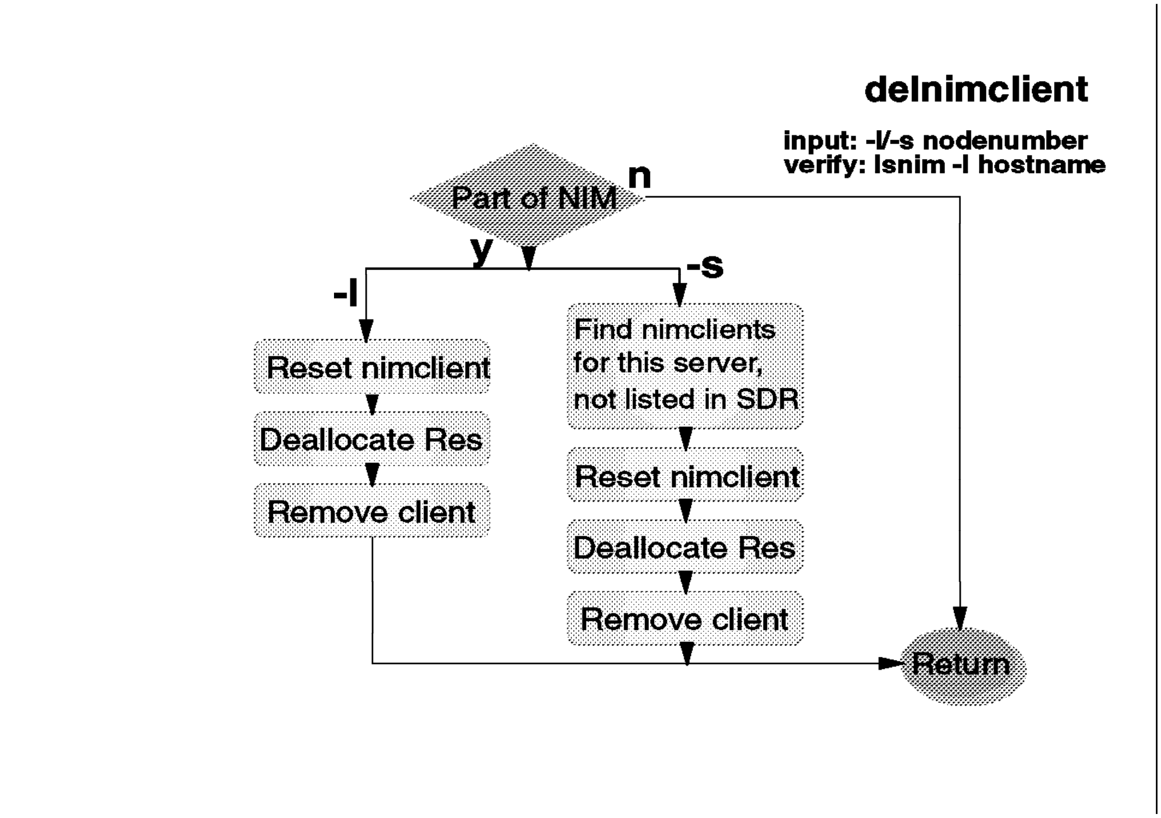


```
nim -o unconfig master
installp -u bos.sysmgt.nim.master
installp -u bos.sysmgt.nim.spot
```

The delnimmast wrapper deletes the NIM master definition of a node or the Control Workstation. The input parameter of delnimmast is the nodenumber representing a NIM master. The delnimmast wrapper deletes the NIM configuration, that is all NIM objects and the SPOT. Also, the NIM filesets are deinstalled.

When run from setup_server, delnimmast checks the SDR for any master node that is not serving clients anymore. So the deletion of a master is not explicitly specified from setup_server.

2.7.2 delnimclient



The `delnimclient` command is used to delete NIM client definitions from a NIM master. When run from the command line, the `delnimclient` command deletes the NIM client definition from the master this client belongs to. For example:

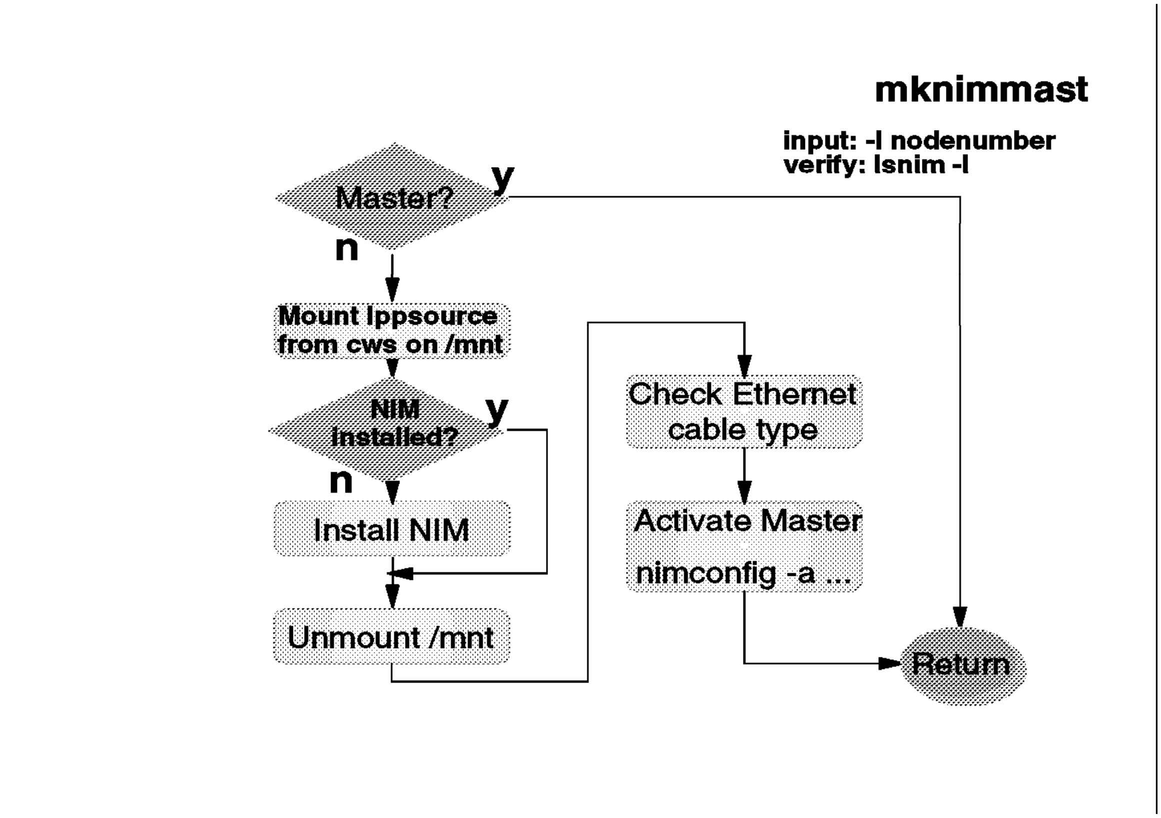
```
delnimclient -l 4-6
```

In this example, the NIM clients corresponding to nodenumbers 4, 5, and 6 will be deleted from their NIM master. The `delnimclient` will query the SDR to find out to which NIM master each of these nodes belongs.

The `delnimclient` command recognizes two input parameters:

- -l 7, specifying nodenumbers, means to delete the NIM client corresponding to nodenumber 7.
- -s 0, means to delete all NIM clients from server 0 that are no longer boot/install clients of server 0, per the SDR.

2.7.3 mknimmast



The mknimmast command creates a NIM master. A NIM master is created by installing the NIM filesets and making the NIM master active by entering the nimconfig command.

The input parameter for mknimmast is a nodenumber corresponding to the node that needs to be configured as a NIM master. A prerequisite for execution is that the SDR should list the node as a boot-install server. The right way to assign a new boot-install server is:

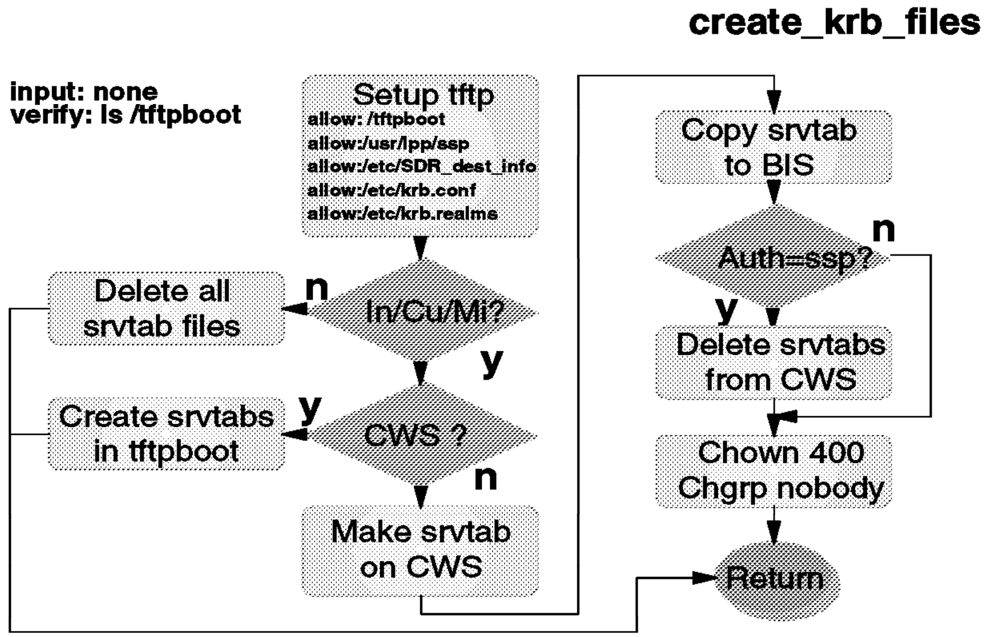
```
spbootins -l 35-48 -n 1 -s no
```

This means that nodenumber 1 is defined as the boot-install server for nodenumbers 2, 3, and 4. The previously defined boot-install server for nodenumbers 2, 3, and 4 is still untouched: spbootins was told not to run setup_server. In this case, two scenarios could happen:

- The original boot-install server has no clients left to serve, according to the SDR. In that case, the next time setup_server runs on node 33, the original boot-install server, setup_server will discover that node 33 does not serve NIM clients anymore, and will delete the NIM master definition with delnimmast -l 33. If this NIM master contained special filesystems for the storage of lppsources and system backups, they will not be deleted by delnimmast.

- The original boot-install server still has NIM clients left to serve. In this case, setup_server will only delete the NIM clients that have changed boot-install server.

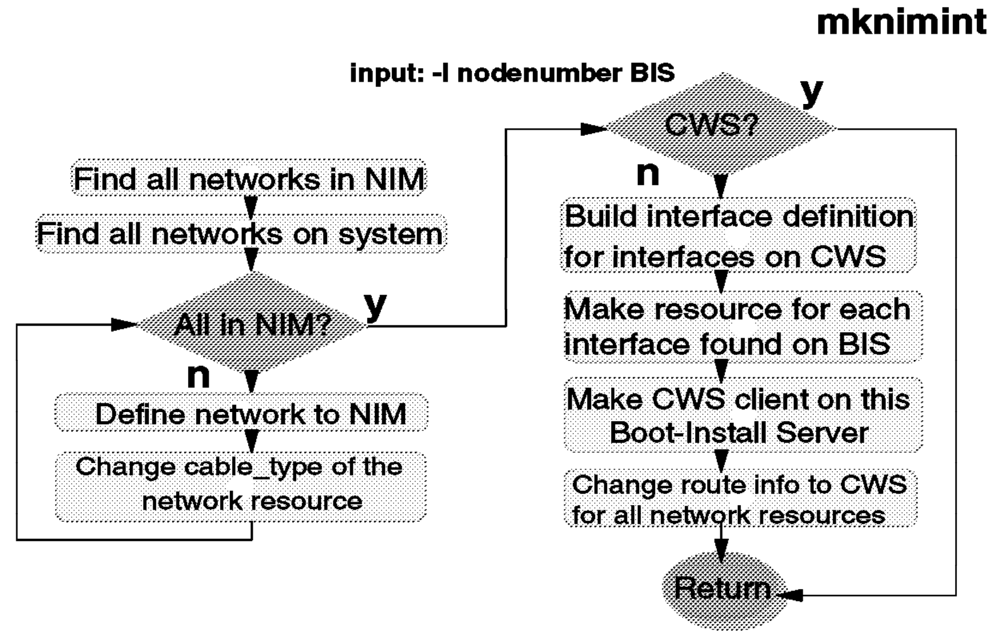
2.7.4 create_krb_files



The create_krb_files wrapper creates the Kerberos definition for a node that has a boot-response of install, customize, or migrate on the associated boot-install server. This wrapper must be run on the boot-install server where the NIM client is being served from.

The create_krb_files wrapper is run without input parameters.

2.7.5 mknimint

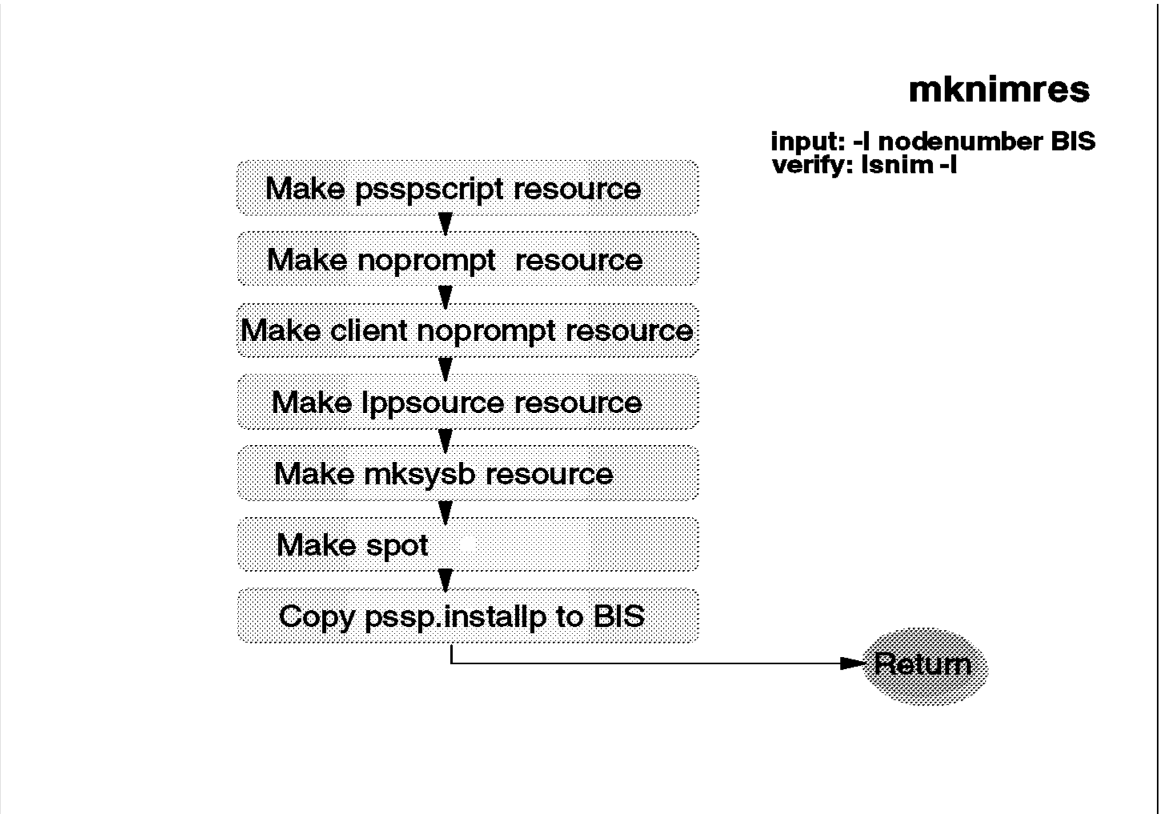

mknimint
input: -l nodenumber BIS
Find all networks in NIM
Find all networks on system
All in NIM?
y
n
Define network to NIM
Change cable_type of the network resource
CWS?
y
n
Build interface definition for interfaces on CWS
Make resource for each interface found on BIS
Make CWS client on this Boot-Install Server
Change route info to CWS for all network resources
Return

The mknimint wrapper creates the network objects on a NIM master. Network objects are defined on the NIM masters to ensure that the install process can find its way to a NIM master from a NIM client and vice versa. So, mknimint will create network objects on the NIM master serving NIM clients, and, if more NIM masters are configured, it will create network objects to find the Control Workstation. The Control Workstation will serve as a Resource Server if more than one NIM master is configured. When mkniminst is run on a boot-install server, not the CWS, it will create the CWS as a NIM client on the boot-install server. This means that a boot-install server is able to access the resources on the Control Workstation. For that purpose, the boot-install server will also create network objects who are related to the interfaces of the CWS, and make sure a route is available on the boot-install server to every network interface on the Control Workstation.

The mknimint command syntax uses the **-l** flag to identify the NIM master on which to create the network objects.

In order to verify the correct results of the mknimint command, the `lsnim -l -c networks` command should be used. On the CWS, the output should include all network objects corresponding to the network interfaces responsible for installation. On boot-install servers, the output should also include network objects that correspond to all network interfaces on the CWS. Excluding the network interface, the boot-install server is booted from itself.

2.7.6 mknimres



The mknimres wrapper creates all necessary resource objects on a boot-install server. The command is very straightforward and creates all resource objects discussed in this chapter.

The input for the mknimres command is a nodenumber.

```
mknimres -l 11
```

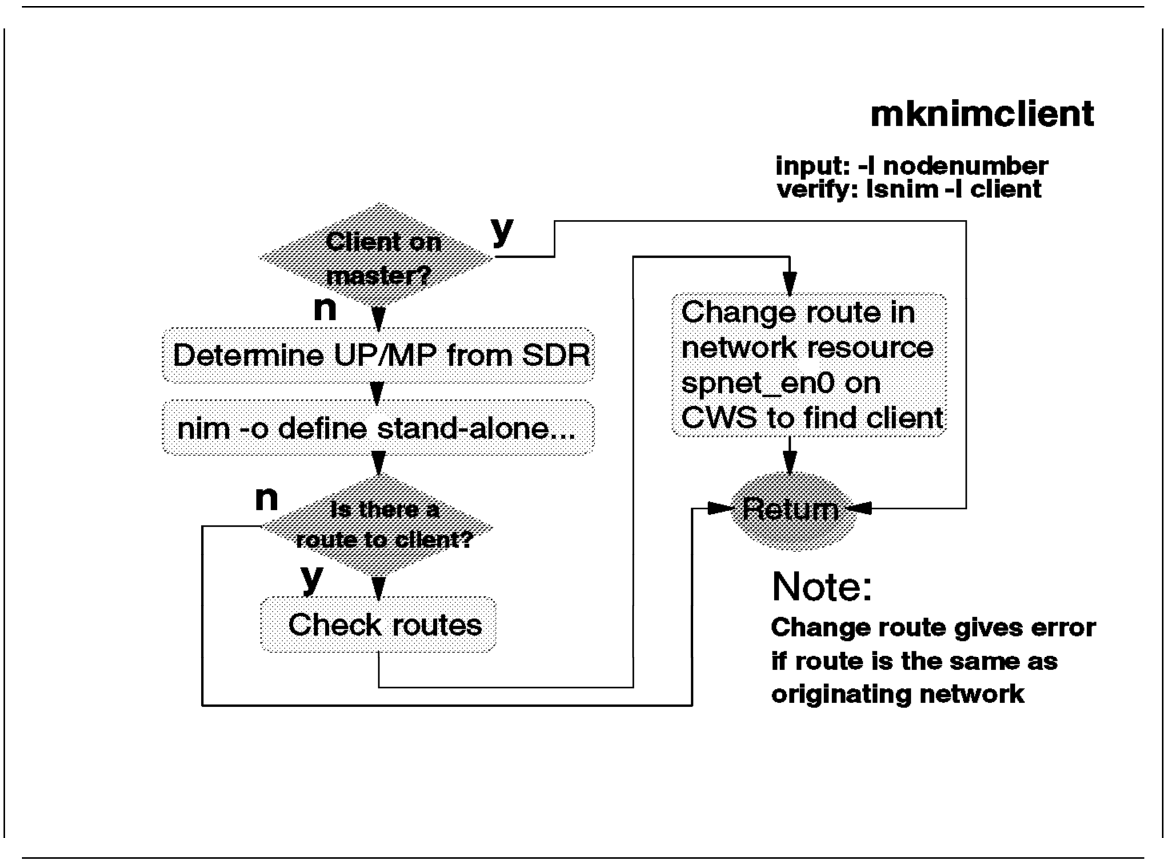
This creates all resource objects on the NIM master corresponding to nodenumber 11.

If the RS/6000 SP contains High Nodes, the SPOT resource object should create boot-images of type rs6ksmp as well. In order to ensure the correct build of the SPOT, make sure that the following images are included in the lppsource object:

- devices.rs6ksmp.base.usr
- bos.rte.mp.usr

If these images are not installed, the SPOT will not serve SMP boot processes.

2.7.7 mknimclient



mknimclient

input: -l nodenumber
verify: lsnim -l client

Client on master?
y
n
Determine UP/MP from SDR
nim -o define stand-alone...
n
Is there a route to client?
y
Check routes
Change route in network resource spnet_en0 on CWS to find client
Return

Note:

Change route gives error if route is the same as originating network

The mknimclient command creates the NIM client definitions on the boot-install server. The mknimclient command queries the SDR to determine the node′s processor type. This characteristic is used later to select the proper boot-image when the installation starts. The input parameter to this command is a nodemumber. For example:

```
mknimclient -l 15
```

This will create the NIM client corresponding to nodenumber 15 on the NIM master, serving nodenumber 15.

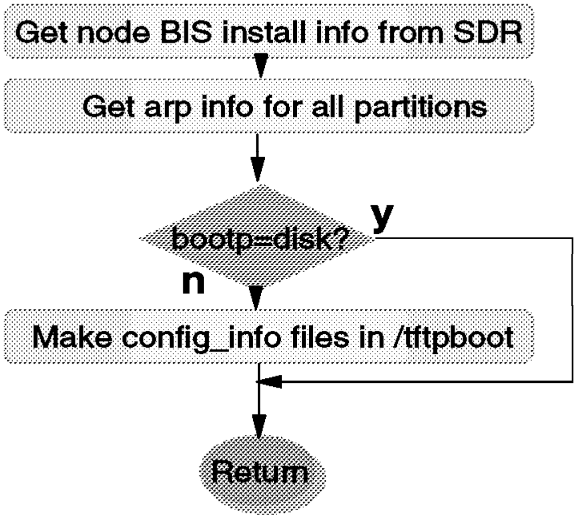
Also, the mknimclient command makes sure that the spnet_en0 network object on the Control Workstation contains a route to reach nodenumber 15.

To verify the completion of this wrapper, use lsnim -l client_name.

2.7.8 mkconfig

mkconfig

input: none
verify: ls -l /tftpboot


Get node BIS install info from SDR
Get arp info for all partitions
bootp=disk?
y
n
Make config_info files in /tftpboot
Return

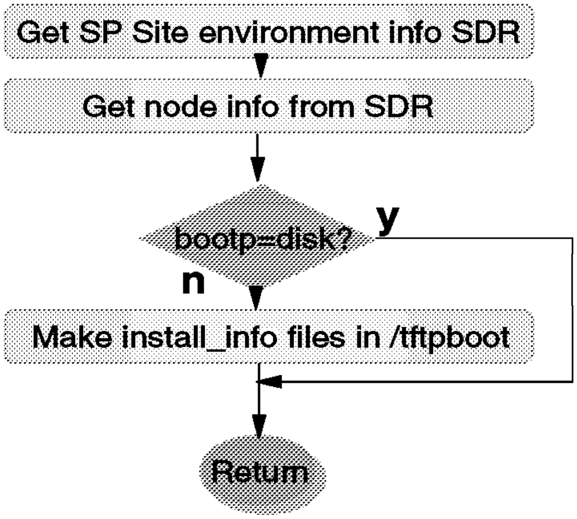
The mkconfig wrapper creates the installation configuration file for each node on the boot-install server in the /tftpboot directory. This wrapper can only be run on the boot-install server it needs to create these files on. The mkconfig wrapper has no input parameters. The configuration information file contains:

- Nodenumber
- Reliable hostname
- Default gateway
- cable_type definitions
- Ethernet characteristics
- Other adapter characteristics

2.7.9 mkinstall

mkinstall

input: none
verify: ls -l /tftpboot


Get SP Site environment info SDR
Get node info from SDR
bootp=disk?
y
n
Make install_info files in /tftpboot
Return

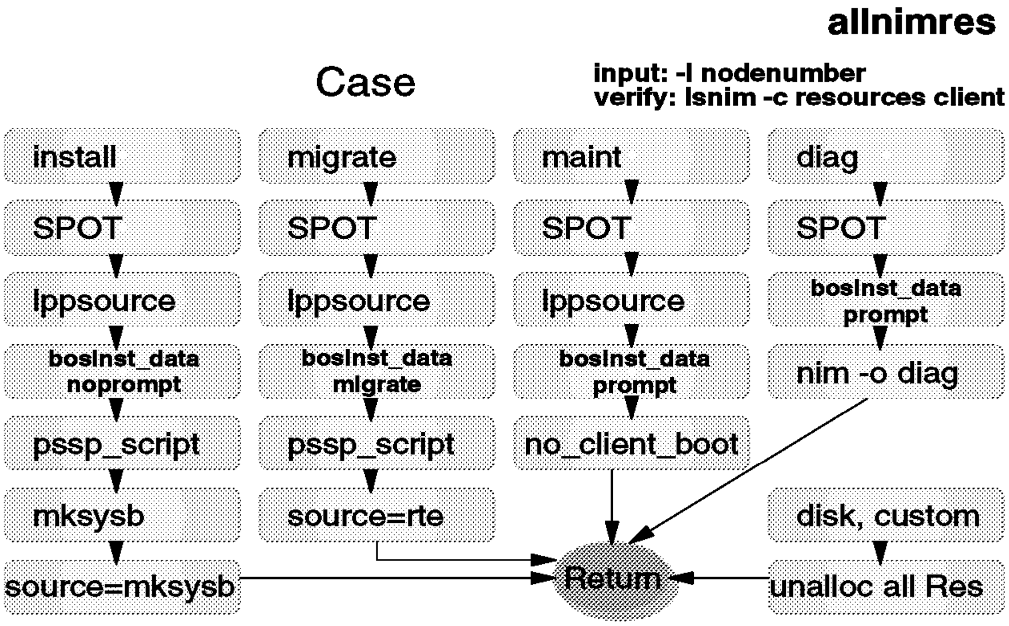
The mkinstall wrapper creates the installation information file for each node on the boot-install server. The mkinstall wrapper should be run on the target boot-install server, without input parameters.

The files that are created by mkinstall look like the following:

```
client-name.install_info
```

The information in these files are environment variables used during the customize phase of a node′s installation.

2.7.10 allnimres



allnimres

input: -l nodenumber
verify: lsnim -c resources client

The allnimres wrapper prepares a NIM client for an installation activity. The allnimres command reads the SDR to determine the bootp-response configured for the nodes it is going to prepare. The bootp_responses currently available in PSSP 2.2 are:

- disk
- customize
- install
- maint
- diag
- migrate

Each of these responses generates a different flow of NIM configuration actions to accommodate the installation action.

The input for this command is a nodenumber. For example:

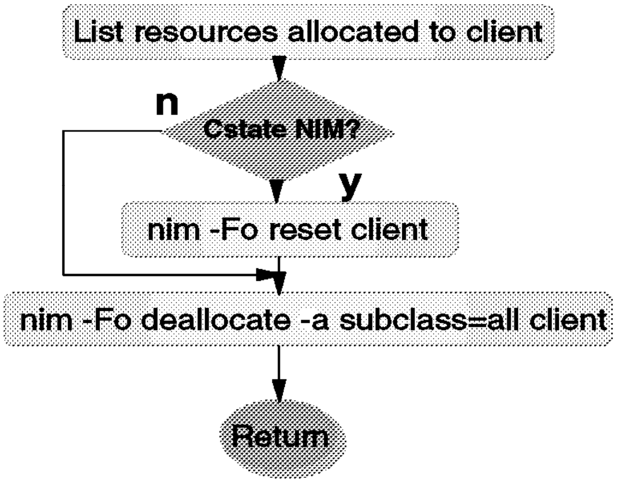
```
allnimres -l 1,2,3,4,5
```

This allocates all resources to nodenumbers 1 to 5, according to the configuration set in the SDR. The flowchart shown in the foil depicts which NIM action is executed for each different bootp_response. When a node is configured with a bootp_responses of disk or customize, the unallnimres wrapper is invoked to deallocate all resources from the node.

2.7.11 unallnimres

unallnimres

input: -l nodenumber
verify: lsnim -l client


List resources allocated to client
n
Cstate NIM?
y
nim -Fo reset client
nim -Fo deallocate -a subclass=all client
Return

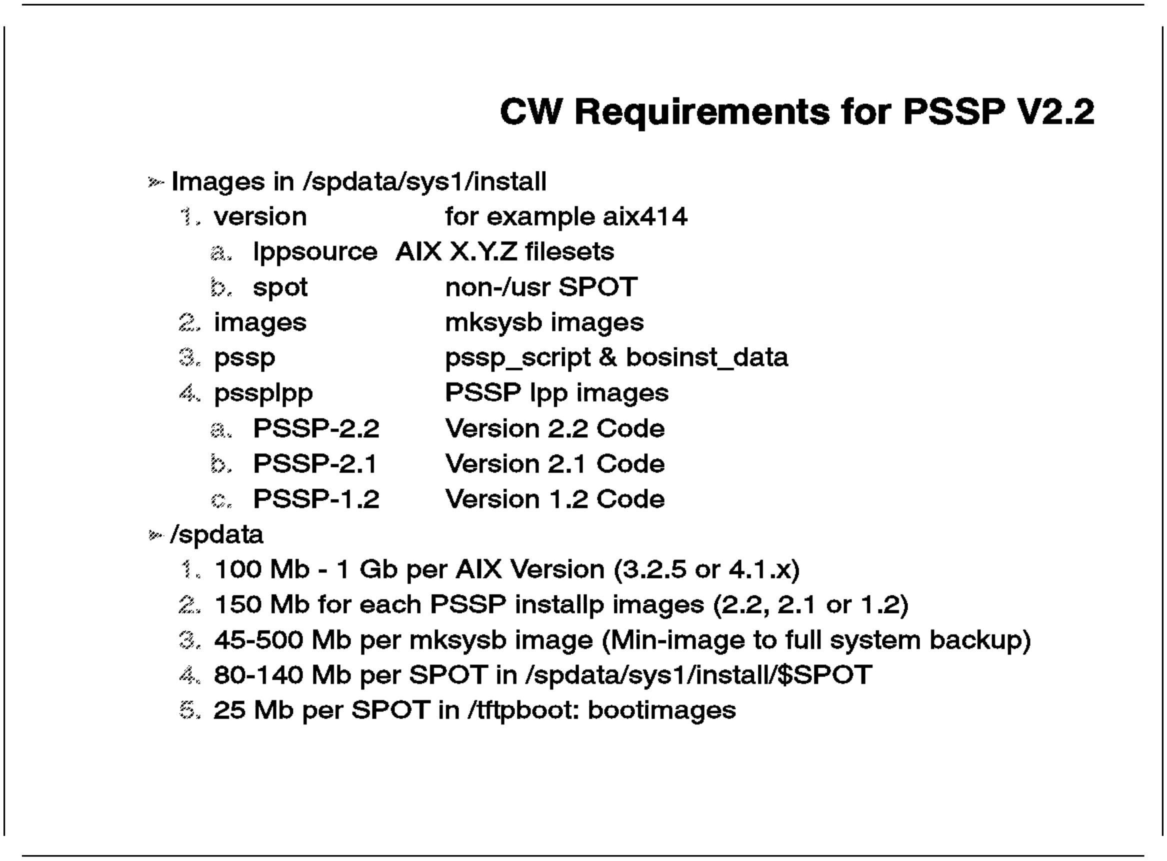
The `unallnimres` wrapper deallocates all NIM resources from a NIM client. This command is not called from setup_server, but from `allnimres`, when the bootp_response for a node is set to disk or customize.

The input for the `unallnimres` command is a nodenumber. For example:

```
unallnimres -l 3
```

This deallocates all resources from NIM client corresponding with nodenumber 3.

2.8 CWS Requirements

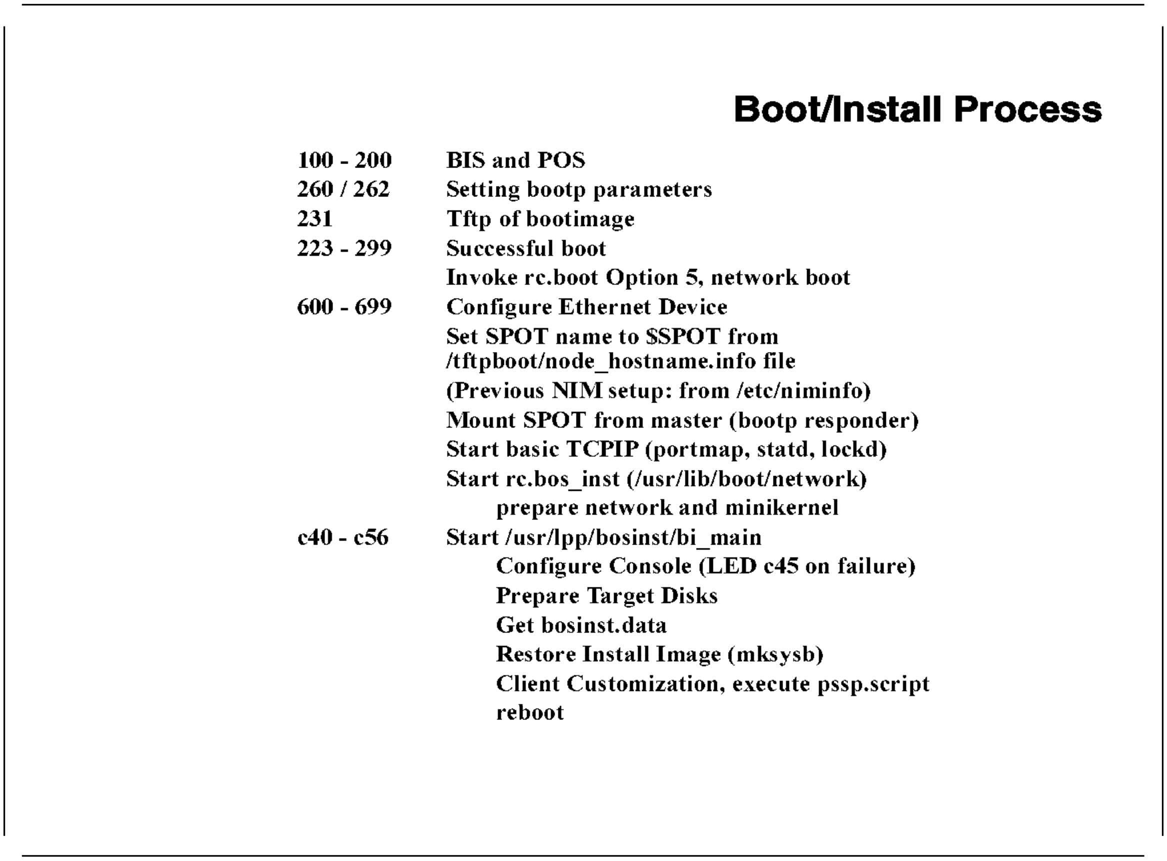


CW Requirements for PSSP V2.2

- Images in /spdata/sys1/install
  1. version — for example aix414
     a. lppsource — AIX X.Y.Z filesets
     b. spot — non-/usr SPOT
  2. images — mksysb images
  3. pssp — pssp_script & bosinst_data
  4. pssplpp — PSSP lpp images
     a. PSSP-2.2 — Version 2.2 Code
     b. PSSP-2.1 — Version 2.1 Code
     c. PSSP-1.2 — Version 1.2 Code
- /spdata
  1. 100 Mb - 1 Gb per AIX Version (3.2.5 or 4.1.x)
  2. 150 Mb for each PSSP installp images (2.2, 2.1 or 1.2)
  3. 45-500 Mb per mksysb image (Min-image to full system backup)
  4. 80-140 Mb per SPOT in /spdata/sys1/install/$SPOT
  5. 25 Mb per SPOT in /tftpboot: bootimages

This foil describes the new Control Workstation disk requirements for each of the different components. PSSP Version 2.2 supports coexistence. This means that more than one AIX level can be supported from the Control Workstation. In that case, usage of a non-/usr SPOT is required. In PSSP Version 2.2, the SPOT location is in the /spdata directory. The foil above provides the disk requirements of the individual directories.

2.9 Boot/Install Process



Boot/Install Process

100 - 200	BIS and POS
260 / 262	Setting bootp parameters
231	Tftp of bootimage
223 - 299	Successful boot
	Invoke rc.boot Option 5, network boot
600 - 699	Configure Ethernet Device
	Set SPOT name to $SPOT from /tftpboot/node_hostname.info file
	(Previous NIM setup: from /etc/niminfo)
	Mount SPOT from master (bootp responder)
	Start basic TCPIP (portmap, statd, lockd)
	Start rc.bos_inst (/usr/lib/boot/network)
	prepare network and minikernel
c40 - c56	Start /usr/lpp/bosinst/bi_main
	Configure Console (LED c45 on failure)
	Prepare Target Disks
	Get bosinst.data
	Restore Install Image (mksysb)
	Client Customization, execute pssp.script
	reboot

This section describes the Boot/Install process flow. The LED-codes associated with the boot-install process can be categorized in six sections:

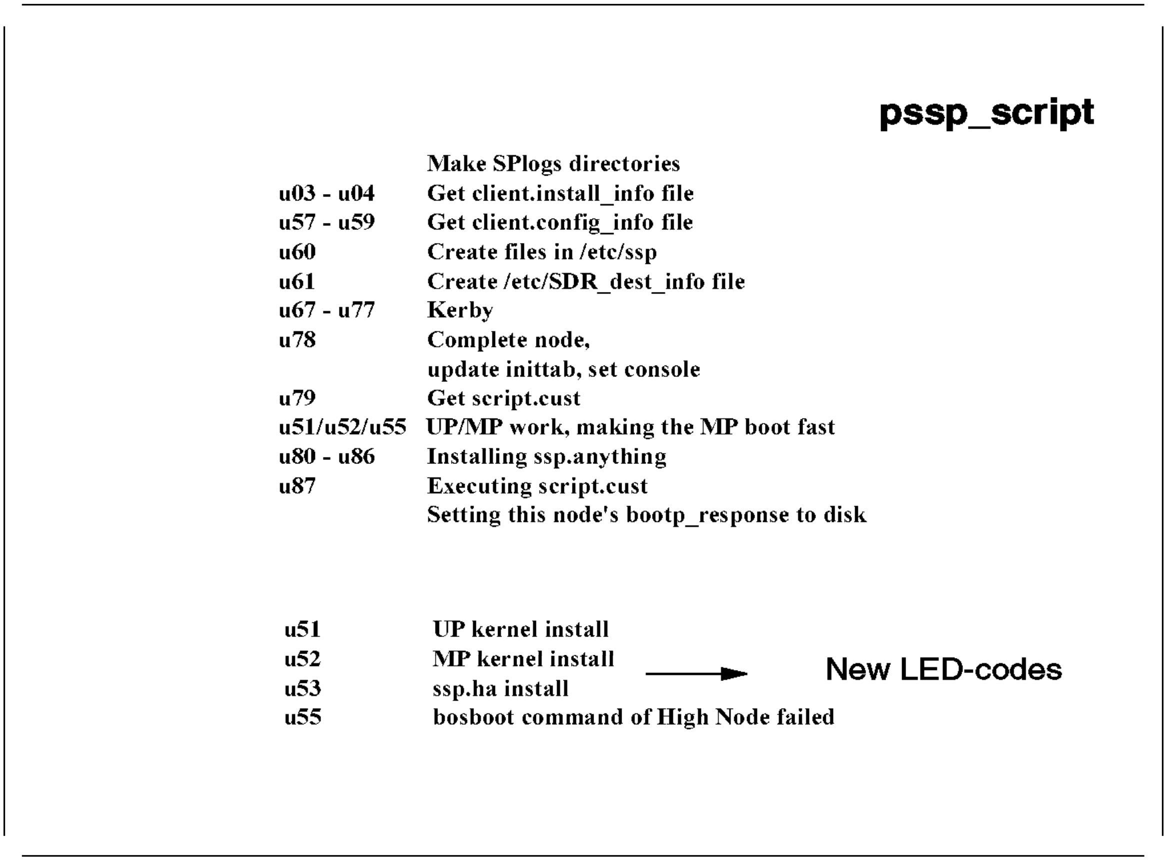
- BIS and POS - Build-In Self-test and Power-On Self-test.
- Setting bootp parameters - Configuration of the parameters provided with the bootp program, invoking the boot/install server. These parameters are:
  - boot-device, for example, Ethernet
  - client IP-address
  - server IP-address
  - gateway IP-address

  In the case that these parameters are provided manually, we refer to *manual nodeconditioning* on the RS/6000 SP. If a node is invoked with the *Netboot* option from the Global Controls panel in spmon, these parameters are automatically retrieved from the SDR and provided to the bootp program.
- tftp of boot-image - Once the bootp parameters are set, the bootp program is invoked and searches a server. If a server is found, the boot image is transferred to the node with the tftp program.
- successful boot - Once the boot image is transferred and the boot image is valid, the initial boot process will complete with LED-code 299.

- LED-codes 600 to 699 - The LED-codes 600 to 699 are associated with building the installation environment. The installation environment is stored in the node_hostname.info file, stored in /tftpboot on the boot/install server. Also, TCP/IP is configured in this section.
- LED-codes c40 to c56 - The LED-codes c40 to c56 are associated with the actual installation. Depending on the bootp-response, the installation either performs a varyonvg of the rootvg disk in the case of a migration install, or a mkvg of rootvg (redefining the rootvg) in the case of a complete overwrite install. The bosinst.data file contains the logical volume definition of all filesystems that need to be defined in rootvg.

After these steps, the actual installation starts. When the installation finishes, the customization of the node is performed by pssp_script.

2.10 pssp_script



pssp_script

	Make SPlogs directories
u03 - u04	Get client.install_info file
u57 - u59	Get client.config_info file
u60	Create files in /etc/ssp
u61	Create /etc/SDR_dest_info file
u67 - u77	Kerby
u78	Complete node, update inittab, set console
u79	Get script.cust
u51/u52/u55	UP/MP work, making the MP boot fast
u80 - u86	Installing ssp.anything
u87	Executing script.cust
	Setting this node's bootp_response to disk

u51	UP kernel install
u52	MP kernel install
u53	ssp.ha install
u55	bosboot command of High Node failed

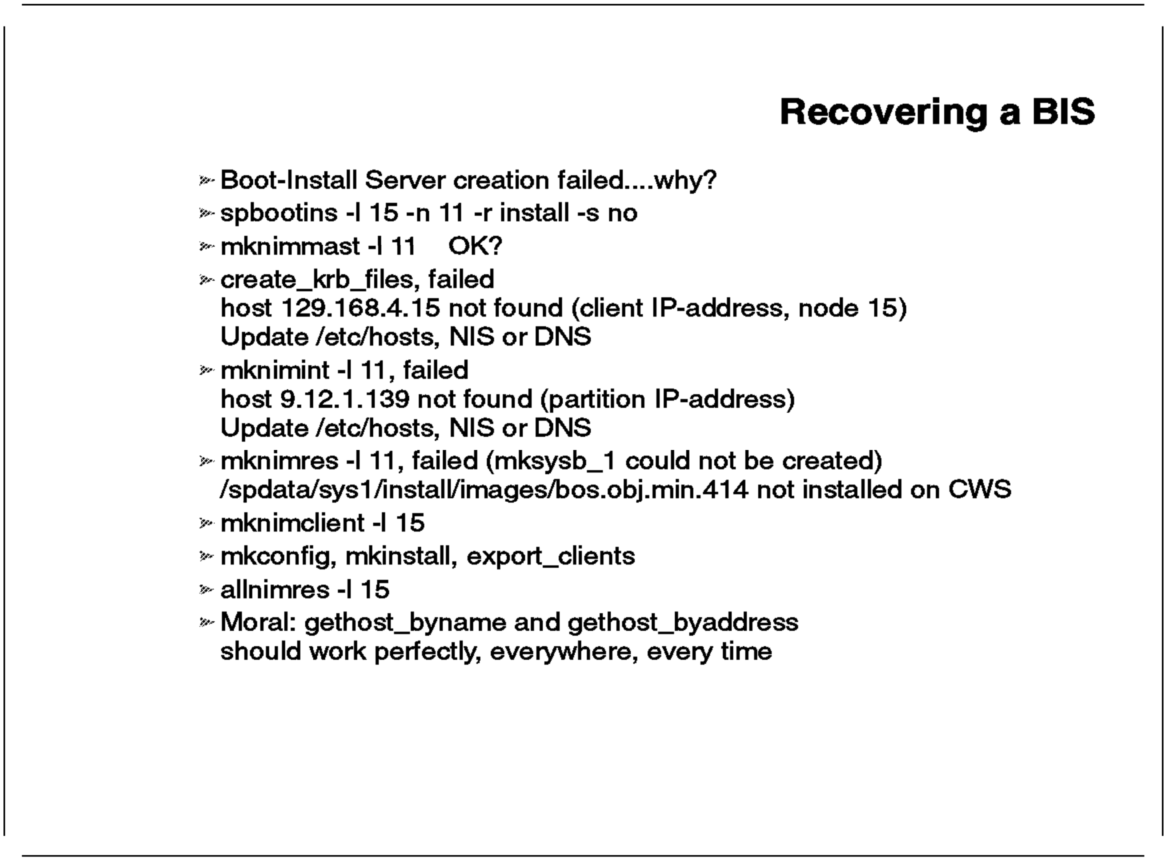
New LED-codes

This section describes the process flow of pssp_script. The pssp_script file is invoked after each installation or when a node is configured with the bootp_response customize. This script finishes the installation of a node according to the configuration of the SDR and optionally invokes a customer script, if required (script.cust).

The basic functions are:

- Configure the node in the Kerberos and SDR environment.
- Install all minimum required PSSP filesets.
- Install the appropriate kernel fileset.
- Execute script.cust

2.11 Recovering a Boot-Install Server

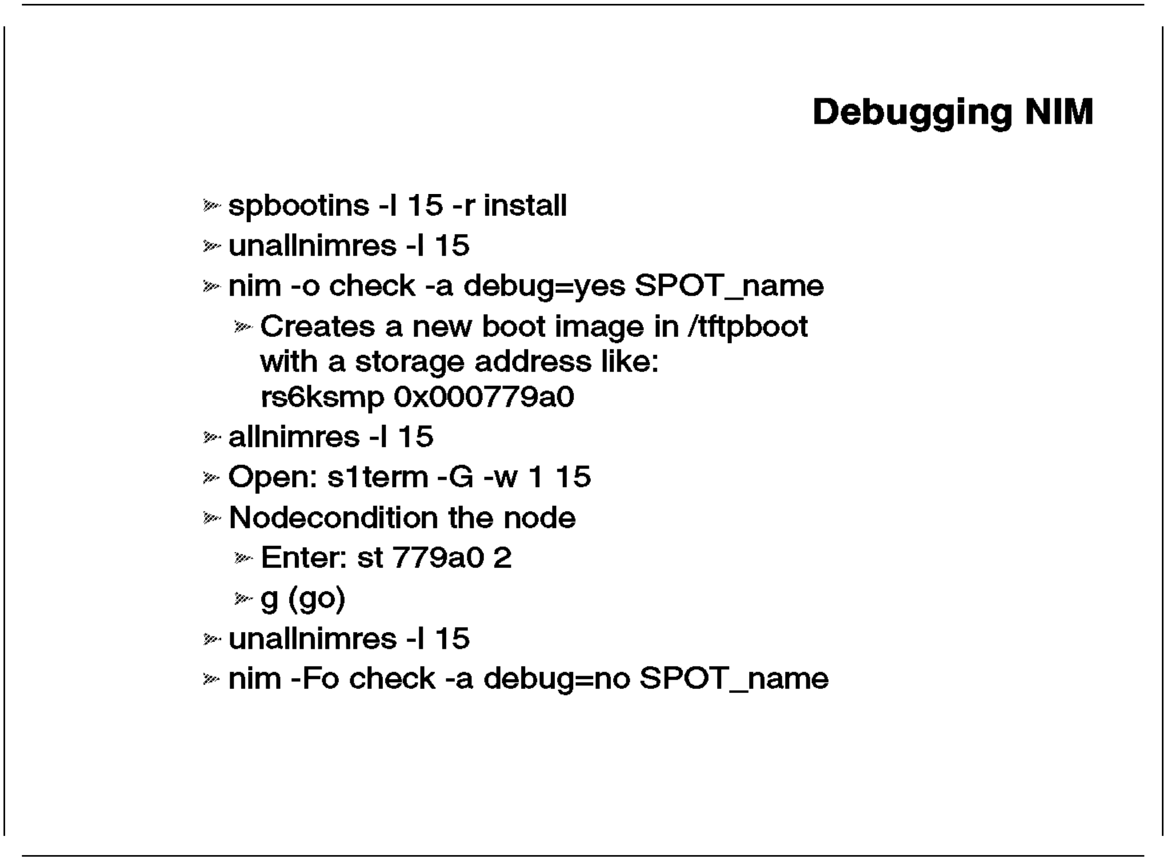


Recovering a BIS

- Boot-Install Server creation failed....why?
- spbootins -l 15 -n 11 -r install -s no
- mknimmast -l 11   OK?
- create_krb_files, failed
  host 129.168.4.15 not found (client IP-address, node 15)
  Update /etc/hosts, NIS or DNS
- mknimint -l 11, failed
  host 9.12.1.139 not found (partition IP-address)
  Update /etc/hosts, NIS or DNS
- mknimres -l 11, failed (mksysb_1 could not be created)
  /spdata/sys1/install/images/bos.obj.min.414 not installed on CWS
- mknimclient -l 15
- mkconfig, mkinstall, export_clients
- allnimres -l 15
- Moral: gethost_byname and gethost_byaddress
  should work perfectly, everywhere, every time

This section describes the usage of wrappers after a failure of setup_server. The wrappers are invoked individually to analyze the progress of the installation setup. The flow of the wrappers used on the command line should be the same as the flow, used in setup_server.

2.12 Debugging NIM



Debugging NIM

- spbootins -l 15 -r install
- unallnimres -l 15
- nim -o check -a debug=yes SPOT_name
  - Creates a new boot image in /tftpboot with a storage address like: rs6ksmp 0x000779a0
- allnimres -l 15
- Open: s1term -G -w 1 15
- Nodecondition the node
  - Enter: st 779a0 2
  - g (go)
- unallnimres -l 15
- nim -Fo check -a debug=no SPOT_name

This section describes the actions required to change the SPOT for debugging the install process and using the wrappers. If the installation process fails, and the symptom is not clear from the LED-codes, the method shown in the foil, can be used to closely follow the installation steps of a node. For that purpose, the SPOT can be configured to debug the installation steps. The characteristics of a NIM objects can not be changed as long as the object is in use by another object. To unallocate all resources, the `unallnimres` command can be used:

```
unallnimres -l 15
```

This command only unallocates the resources that were allocated to node 15. So, if more nodes are allocated to the SPOT-resource, the same command must be applied to the other nodes. The `spbootins` command is used to make sure the SDR reflects the installation characteristics for the node. If the installation failed, the SDR probably still reflects install as the bootp response. After configuring the SPOT for debug usage, the node should be node-conditioned. Node-conditioning is also known as manually netbooting a node.

2.13 The Frame Supervisor

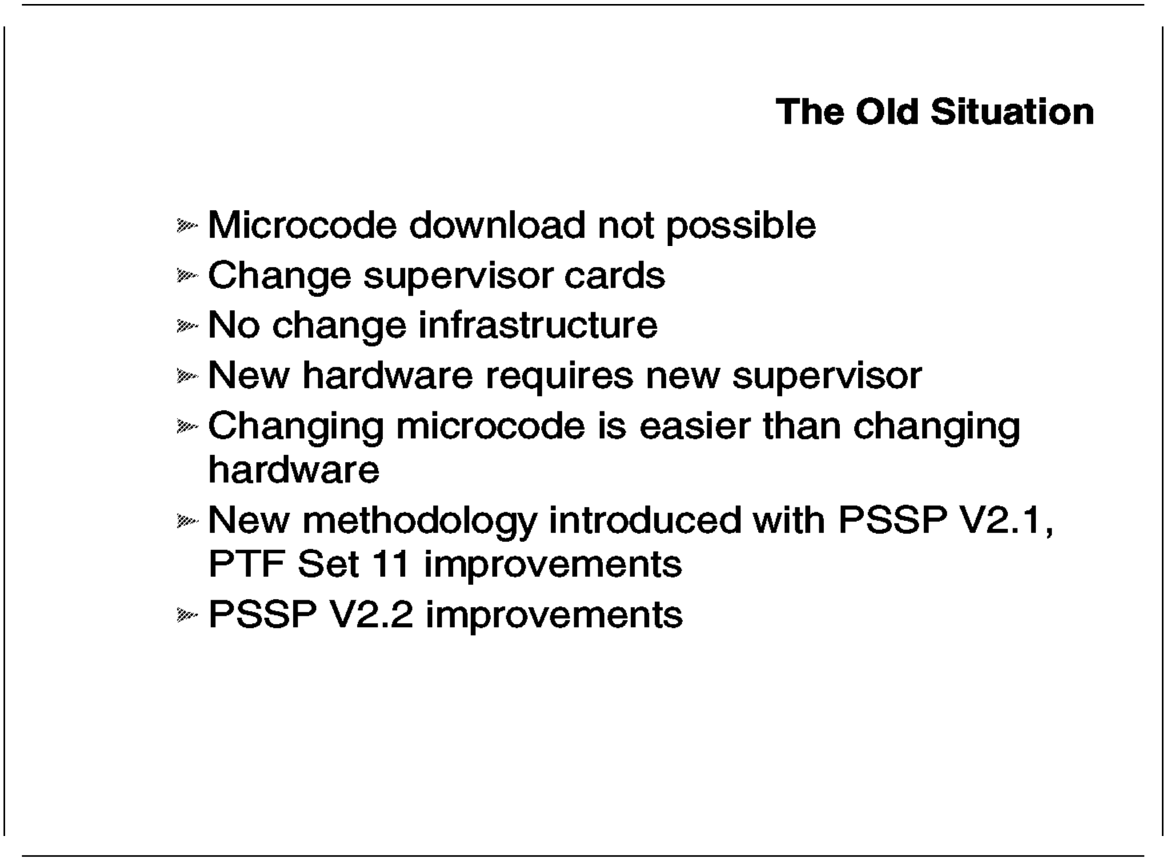
New Installation Methodology

Frame Supervisor

The announcement of PSSP V2.1 PTF Level 11 introduced the possibility of changing the supervisor hardware microcode, the switchboard microcode, and the microcode of the supervisor board in the nodes. When new functionality is required, generally the frame and node supervisor cards need to be replaced.

New functionality, for instance, can be changes in the node conditioning procedure or complete new hardware, like High Nodes. With the introduction of microcode download, the customer and the field engineer can react better to changing environments and requirements.

2.13.1 The Old Situation

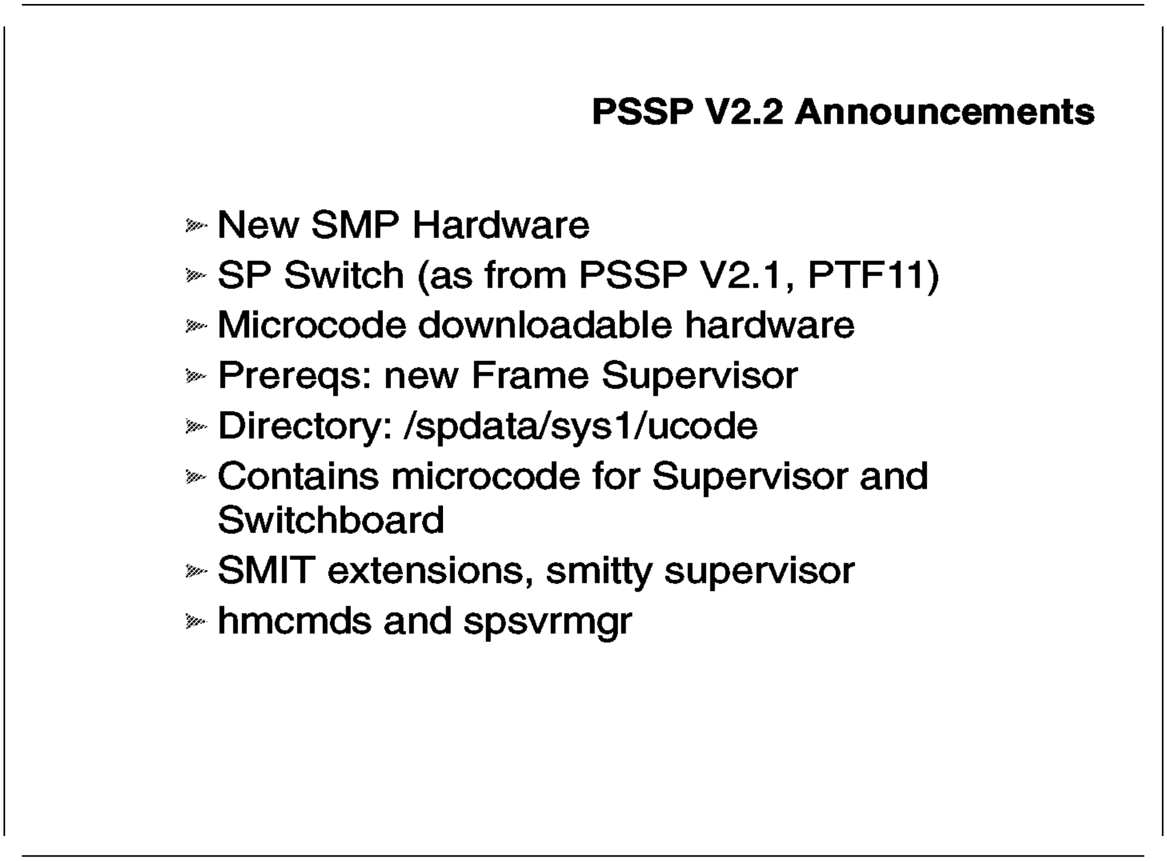


The Old Situation

- Microcode download not possible
- Change supervisor cards
- No change infrastructure
- New hardware requires new supervisor
- Changing microcode is easier than changing hardware
- New methodology introduced with PSSP V2.1, PTF Set 11 improvements
- PSSP V2.2 improvements

In previous releases (before PSSP V2.1, PTF Level 11) microcode download was not possible. With PSSP V2.2, new hardware was introduced. This means that the frame supervisor card should be updated to recognize this hardware.

2.13.2 New with PSSP V2.2



PSSP V2.2 Announcements

- New SMP Hardware
- SP Switch (as from PSSP V2.1, PTF11)
- Microcode downloadable hardware
- Prereqs: new Frame Supervisor
- Directory: /spdata/sys1/ucode
- Contains microcode for Supervisor and Switchboard
- SMIT extensions, smitty supervisor
- hmcmds and spsvrmgr

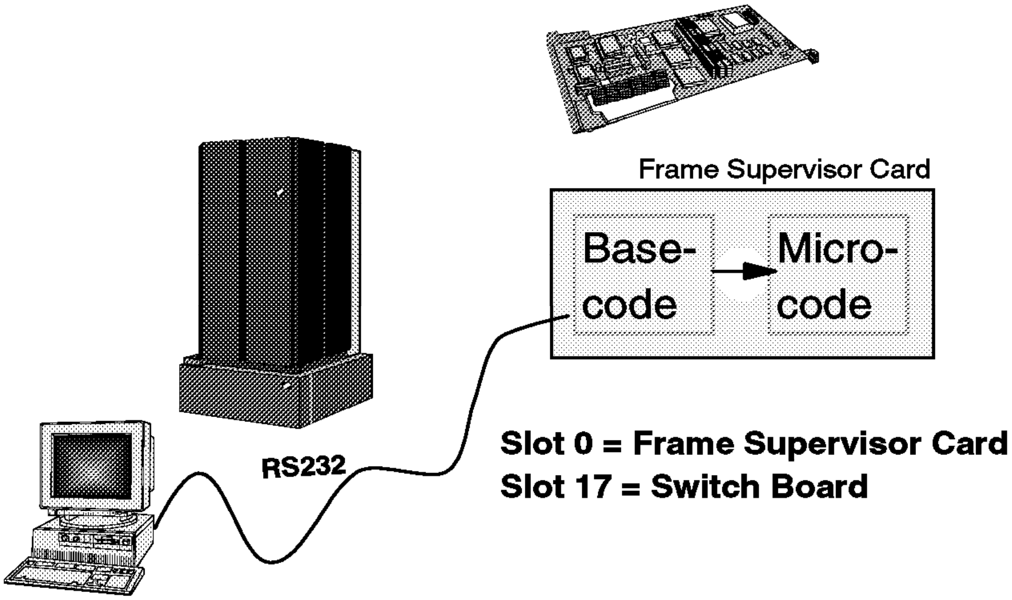
With the announcement of PSSP V2.2, the following functions are included in the supervisor environment:

- SMP hardware
- SMIT panels
- The `spsvrmgr` command
- The /spdata/sys1/ucode directory

In order to operate with high nodes and the new SP switch, the new frame supervisor card is required.

2.13.3 The Supervisor Interface

The Supervisor Interface

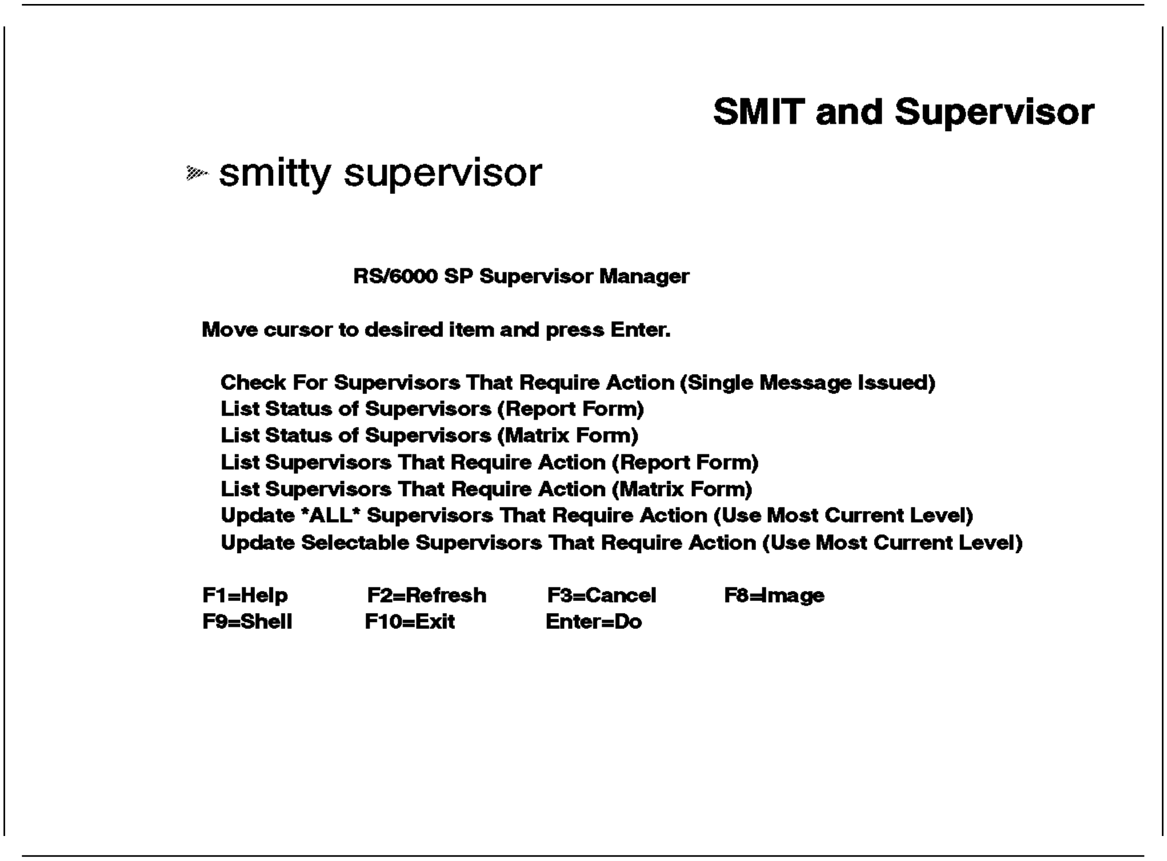


Slot 0 = Frame Supervisor Card
Slot 17 = Switch Board

The supervisor card contains two chips. The first chip is the basecode chip. This chip contains fixed microcode to start the supervisor card in “maintenance mode.” The second chip is an erasable, programmable, read-only memory (EPROM) chip. The supervisor microcode is stored in this chip.

If the microcode chip contains microcode, the supervisor card is able to boot and come up.

2.13.4 SMIT and Supervisor



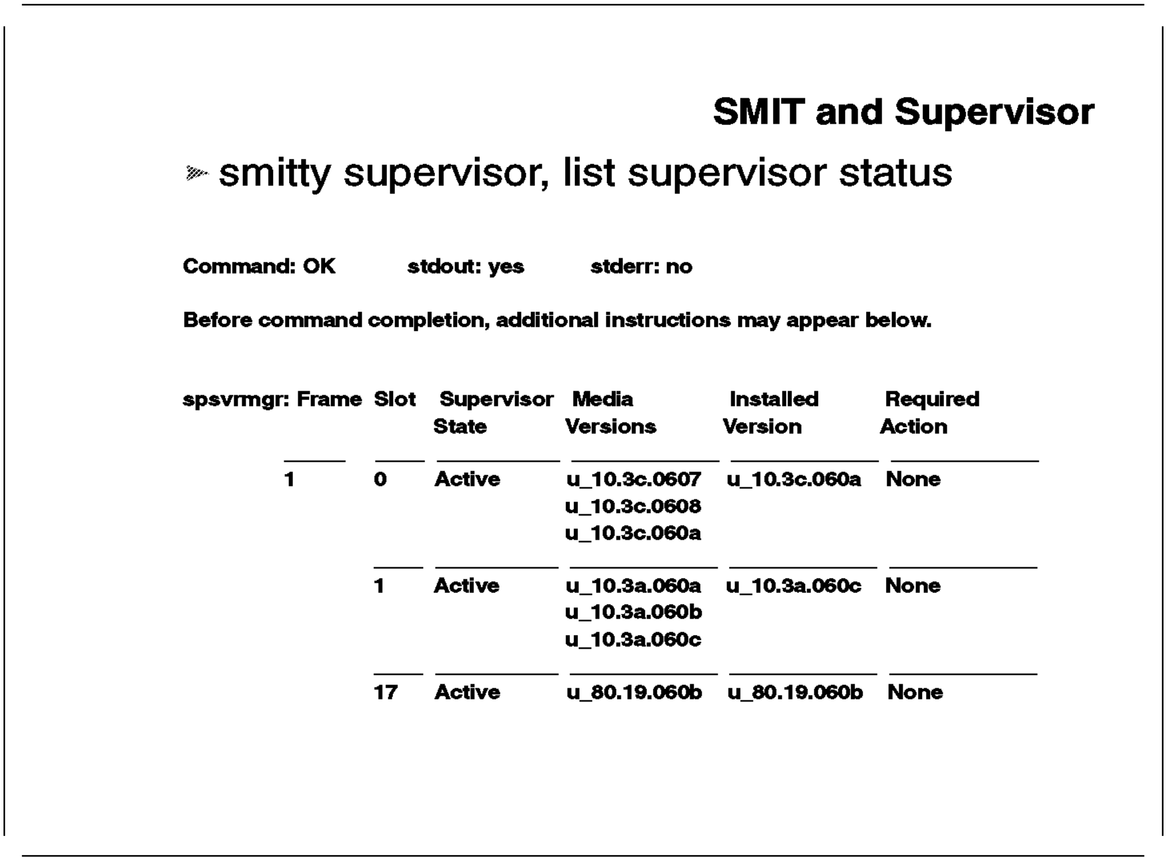
SMIT and Supervisor

smitty supervisor

```
                RS/6000 SP Supervisor Manager

 Move cursor to desired item and press Enter.

   Check For Supervisors That Require Action (Single Message Issued)
   List Status of Supervisors (Report Form)
   List Status of Supervisors (Matrix Form)
   List Supervisors That Require Action (Report Form)
   List Supervisors That Require Action (Matrix Form)
   Update *ALL* Supervisors That Require Action (Use Most Current Level)
   Update Selectable Supervisors That Require Action (Use Most Current Level)

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

The SMIT extensions are the command interface to managing the supervisor card. All relevant actions can be invoked from this panel; query the status and update the supervisor if required.

Note: The supervisor card counts the number of times the card is updated. This counter is a half-word, and thus the number of updates can only be 255.



SMIT and Supervisor

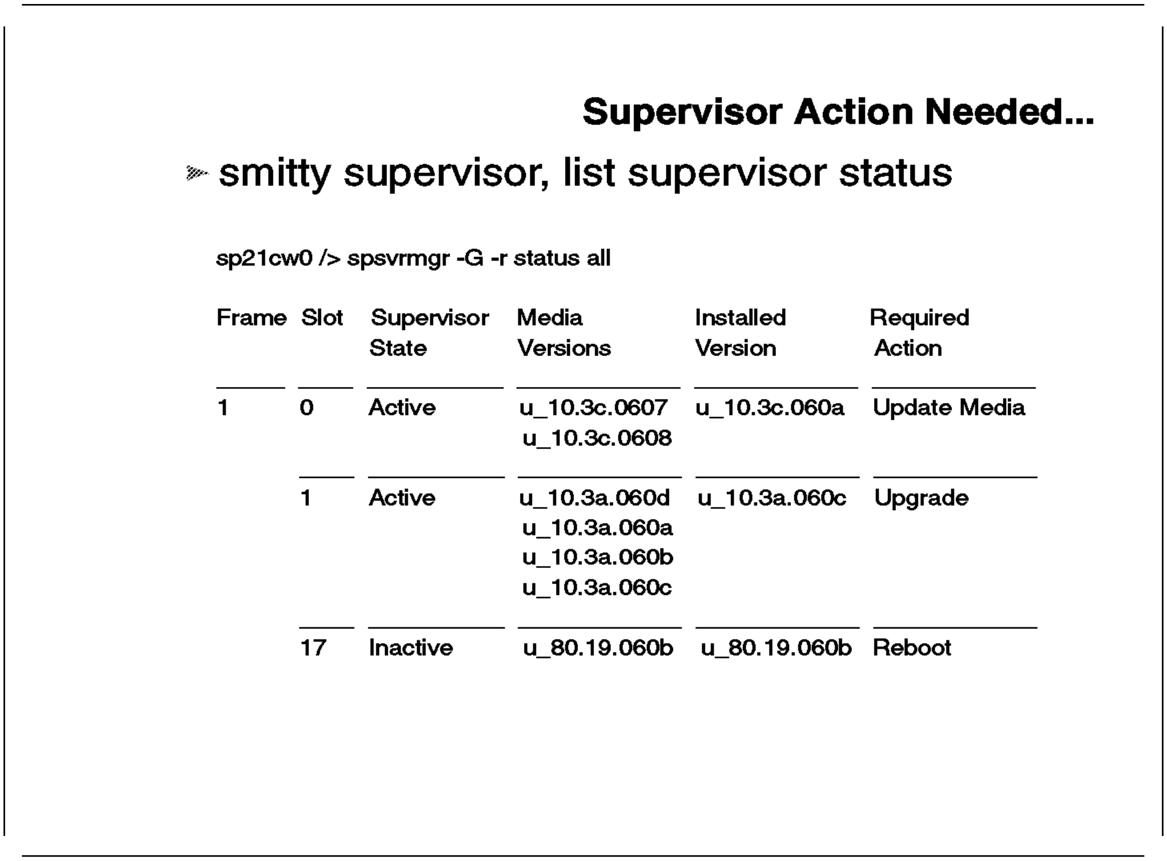
- smitty supervisor, list supervisor status

```
Command: OK          stdout: yes          stderr: no

Before command completion, additional instructions may appear below.

spsvrmgr: Frame  Slot  Supervisor  Media           Installed       Required
                       State       Versions        Version         Action
          _____  ____  __________  ______________  ______________  ______________
          1      0     Active      u_10.3c.0607    u_10.3c.060a    None
                                   u_10.3c.0608
                                   u_10.3c.060a
                 ____  __________  ______________  ______________  ______________
                 1     Active      u_10.3a.060a    u_10.3a.060c    None
                                   u_10.3a.060b
                                   u_10.3a.060c
                 ____  __________  ______________  ______________  ______________
                 17    Active      u_80.19.060b    u_80.19.060b    None
```

When a query of the supervisor card is issued, the output shows the microcode status for each slot in the supervisor card. The query is limited to hardware the query recognizes. This means that old hardware is not listed in the output.

The supervisor slot numbering follows the nodenumbering. This means that nodenumber 1 is represented as supervisor slot 1. The supervisor card is numbered slot number 0. The switchboard is numbered slot 17.

2.13.5 Supervisor Action Needed



Supervisor Action Needed...

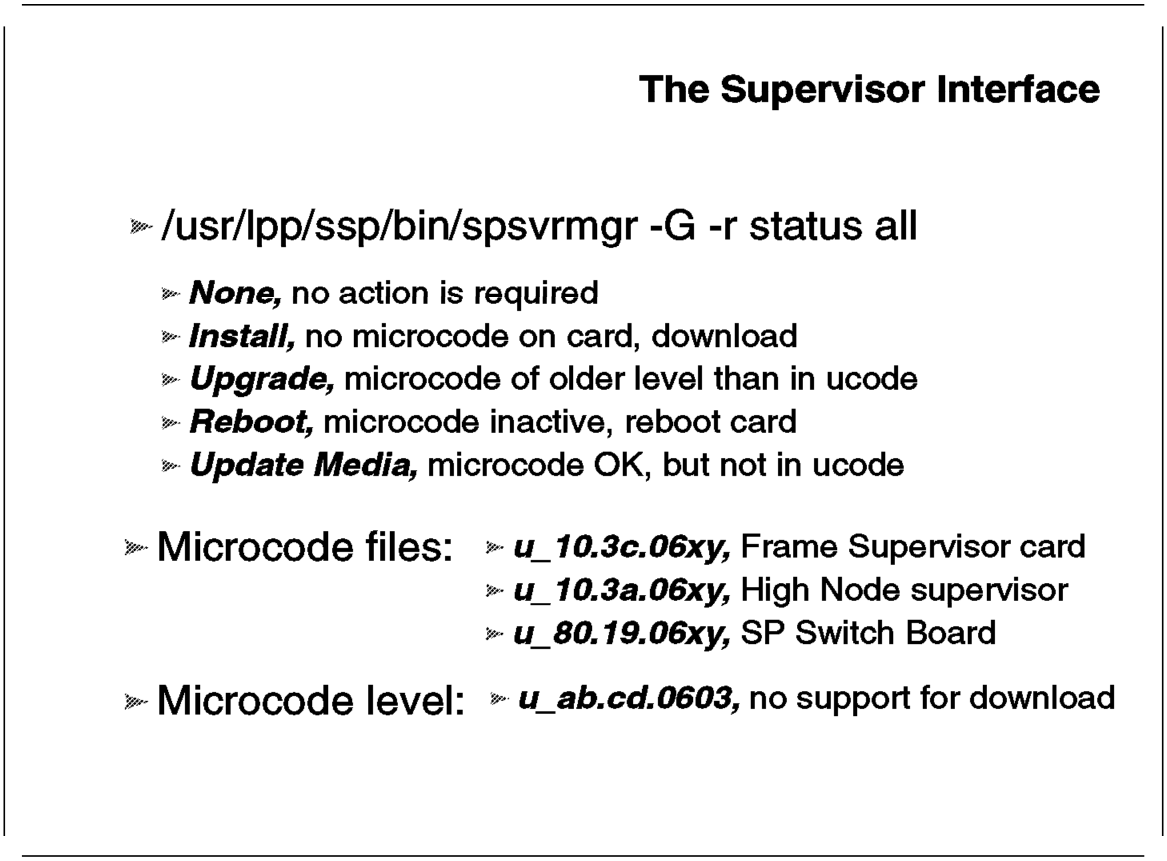
- smitty supervisor, list supervisor status

```
sp21cw0 /> spsvrmgr -G -r status all

Frame  Slot  Supervisor   Media          Installed      Required
             State        Versions       Version        Action
_____  ____  ___________  _____________  _____________  _____________
1      0     Active       u_10.3c.0607   u_10.3c.060a   Update Media
                          u_10.3c.0608
       ____  ___________  _____________  _____________  _____________
       1     Active       u_10.3a.060d   u_10.3a.060c   Upgrade
                          u_10.3a.060a
                          u_10.3a.060b
                          u_10.3a.060c
       ____  ___________  _____________  _____________  _____________
       17    Inactive     u_80.19.060b   u_80.19.060b   Reboot
```

The output shown in this foil lists the most common microcode states. The microcode states are always matched against the available microcode in the */spdata/sys1/ucode* directory.

If the microcode on the card is equal to the highest microcode level in the *ucode* directory, no action is required.

2.13.6 The Supervisor Interface



The Supervisor Interface

- /usr/lpp/ssp/bin/spsvrmgr -G -r status all
  - ***None,*** no action is required
  - ***Install,*** no microcode on card, download
  - ***Upgrade,*** microcode of older level than in ucode
  - ***Reboot,*** microcode inactive, reboot card
  - ***Update Media,*** microcode OK, but not in ucode
- Microcode files:
  - ***u_10.3c.06xy,*** Frame Supervisor card
  - ***u_10.3a.06xy,*** High Node supervisor
  - ***u_80.19.06xy,*** SP Switch Board
- Microcode level:
  - ***u_ab.cd.0603,*** no support for download

The possible states of the microcode are:

- None

  This means that the microcode on the card is equal to the highest microcode level in */spdata/sys1/ucode*.

- Install

  This status indicates that the microcode chip is empty. Action required is to install microcode. Installing microcode can only be done in *basecode* mode. This can be done by executing:

  ```
  hmcmds -G -v -u u_80.19.060b microcode 1:17
  hmcmds -G -v boot_supervisor 1:17
  ```

  In this example, the switchboard of frame 1 is installed with new microcode from the microcode file u_80.19.060b.

- Upgrade

  The microcode level of the hardware is of a lower level than the highest level available in /spdata/sys1/ucode. The upgrade can be done by executing:

  ```
  hmcmds -G -v basecode 2:1
  hmcmds -G -v -u u_10.3a.060c microcode 2:1
  hmcmds -G -v -u boot_supervisor 2:1
  ```

In this example, nodenumber 1 in frame 2 is upgraded. Note that the installation requires the supervisor card to be in the *inactive* state. A supervisor card is inactive when running in basecode mode.

- Reboot

  This status indicates that the new microcode is installed, but the card is still running in basecode mode. Action required is a reboot of the card:

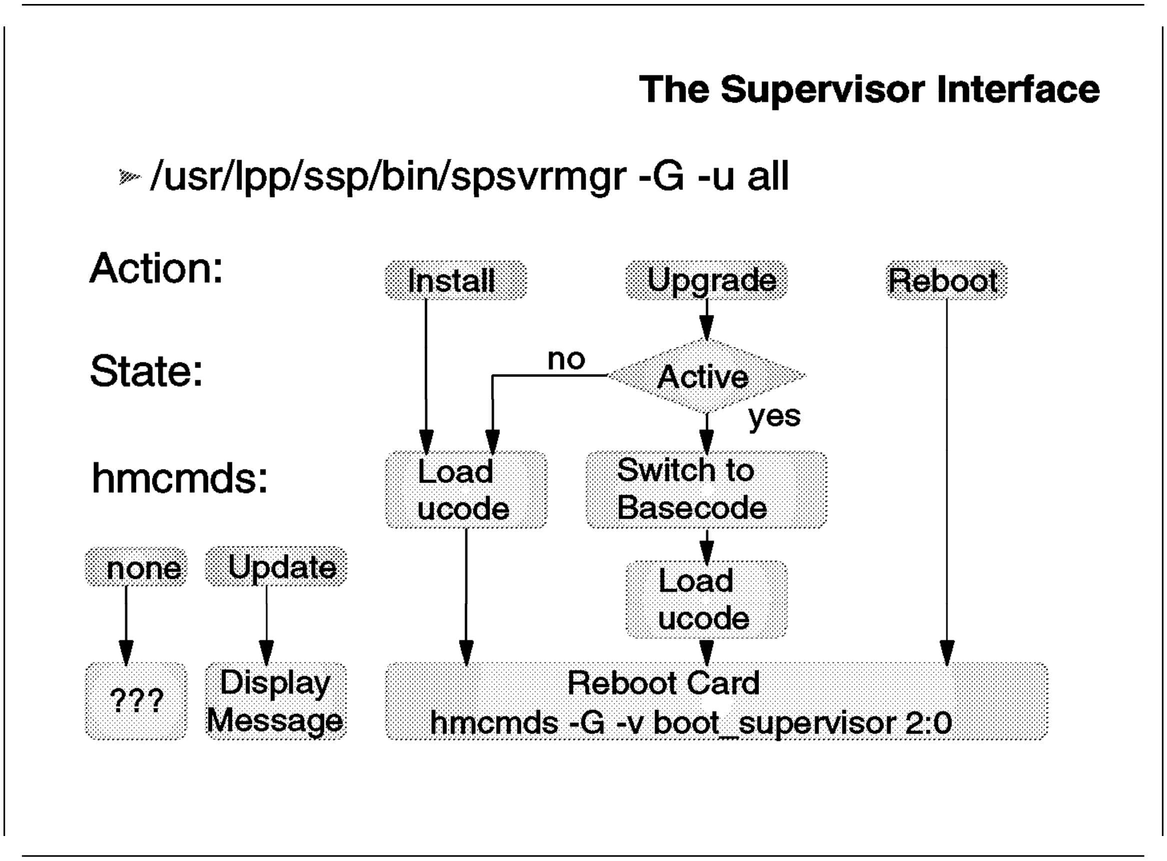
  ```
  hmcmds -G -v boot_supervisor 1:0
  ```

  In this example, the frame supervisor card is rebooted.

- Update Media

  The level of the microcode on the card is higher than the available microcode files in /spdata/sys1/ucode. Action required is to find the microcode files and store them in the ucode directory.

2.13.7 The Supervisor Interface



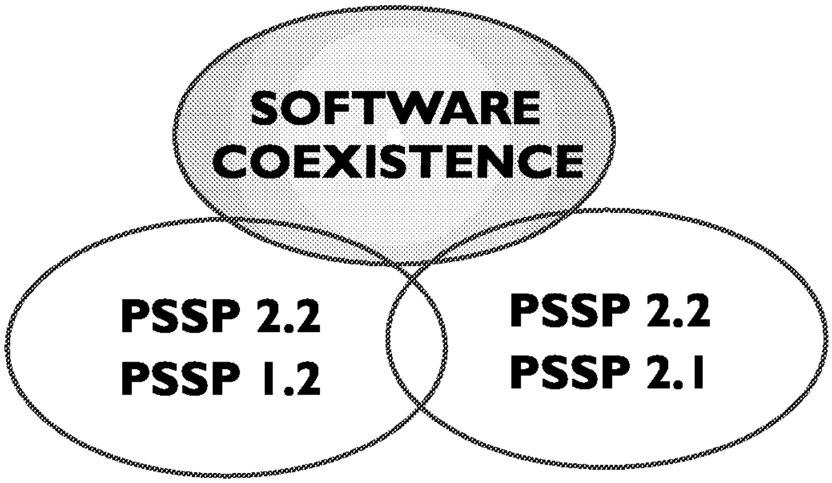
The Supervisor Interface

- /usr/lpp/ssp/bin/spsvrmgr -G -u all

The chart in this foil shows an overview of the flow of actions to follow in order to bring the microcode to the appropriate level and make it operational. The microcode states that require specific action to be taken are install, upgrade, and reboot.

Chapter 3. Software Coexistence

Software Coexistence



SOFTWARE COEXISTENCE

PSSP 2.2
PSSP 1.2

PSSP 2.2
PSSP 2.1

This section describes the new term *software coexistence*. Software coexistence is available with PSSP Version 2 Release 2 and higher. It supports existing RS/6000 SP customers.

Software coexistence means that you can have different versions of AIX and PSSP in one system partition. The following sections describe how to prepare for all different levels of AIX and PSSP, and how the different versions of software work together.

Why have software coexistence? The main reason for coexistence support is that not every customer can use the RS/6000 SP partition concept. Some reasons why coexistence might work better than partitioning follow:

- Most nodes are in production and work perfectly, and no software maintenance should be applied, except for PSSP.
- The AIX 4.1 and AIX 3.2.5 nodes belong to one switch chip.
- Third party software needs AIX 3.2.5, and only some nodes can be upgraded to AIX 4.
- The partition setup is complicated.
- Your location has no IP addresses left for the IP alias.

- With system partitioning, you need almost double the number of SP daemons on the Control Workstation.

With software coexistence support, you can smoothly upgrade the RS/6000 SP nodes to the new software level. There is no need to disturb a working environment.

3.1 Table of Contents for Software Coexistence

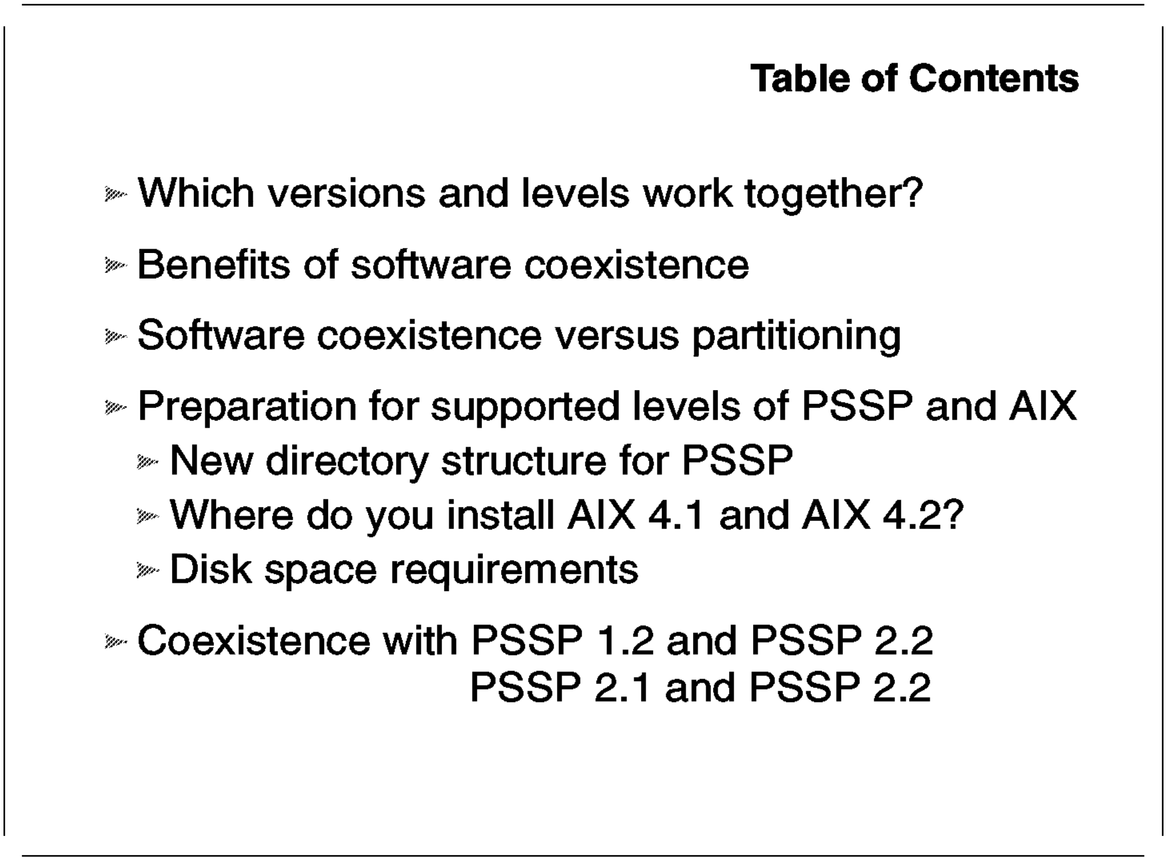


Table of Contents

- Which versions and levels work together?
- Benefits of software coexistence
- Software coexistence versus partitioning
- Preparation for supported levels of PSSP and AIX
  - New directory structure for PSSP
  - Where do you install AIX 4.1 and AIX 4.2?
  - Disk space requirements
- Coexistence with PSSP 1.2 and PSSP 2.2
  PSSP 2.1 and PSSP 2.2

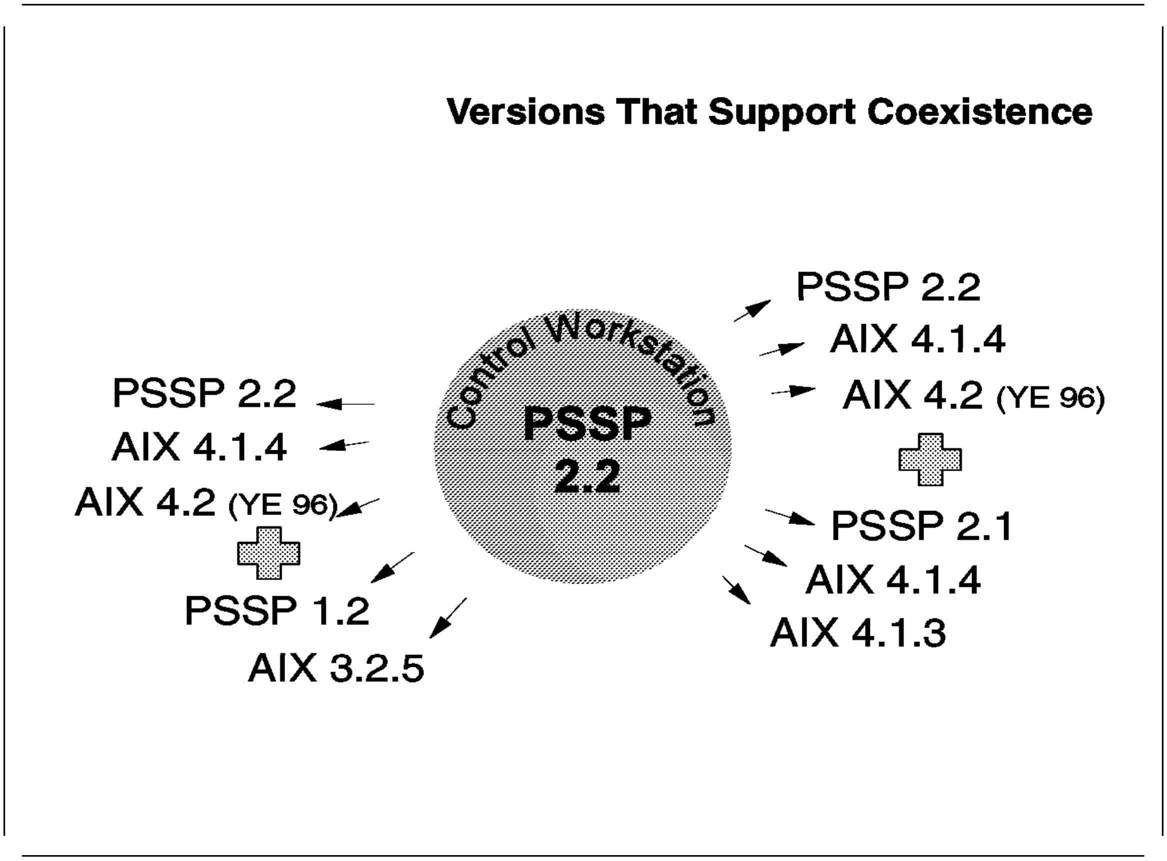
This table of contents gives you an overview of software coexistence. The following sections tell you which software and which levels of that software work in one system partition. You will also learn about what benefits will you see with software coexistence. Furthermore, since coexistence is a new feature with PSSP 2.2, this chapter discusses the features that are different from system partitioning.

Later sections provide answers to the following questions: How do you prepare your system for software coexistence? In which directories do you have to install the different levels of PSSP? Where do you install the different levels of AIX? Since so many different levels of software are installed on your system, how much disk space will you need?

The next section describes coexistence between PSSP 1.2 and PSSP 2.2, and between PSSP 2.1 and PSSP 2.2.

The last sections describe some restrictions for coexistence.

3.2 Versions That Support Coexistence



Versions That Support Coexistence

Control Workstation

PSSP 2.2

PSSP 2.2
AIX 4.1.4
AIX 4.2 (YE 96)
+
PSSP 2.1
AIX 4.1.4
AIX 4.1.3

PSSP 2.2
AIX 4.1.4
AIX 4.2 (YE 96)
+
PSSP 1.2
AIX 3.2.5

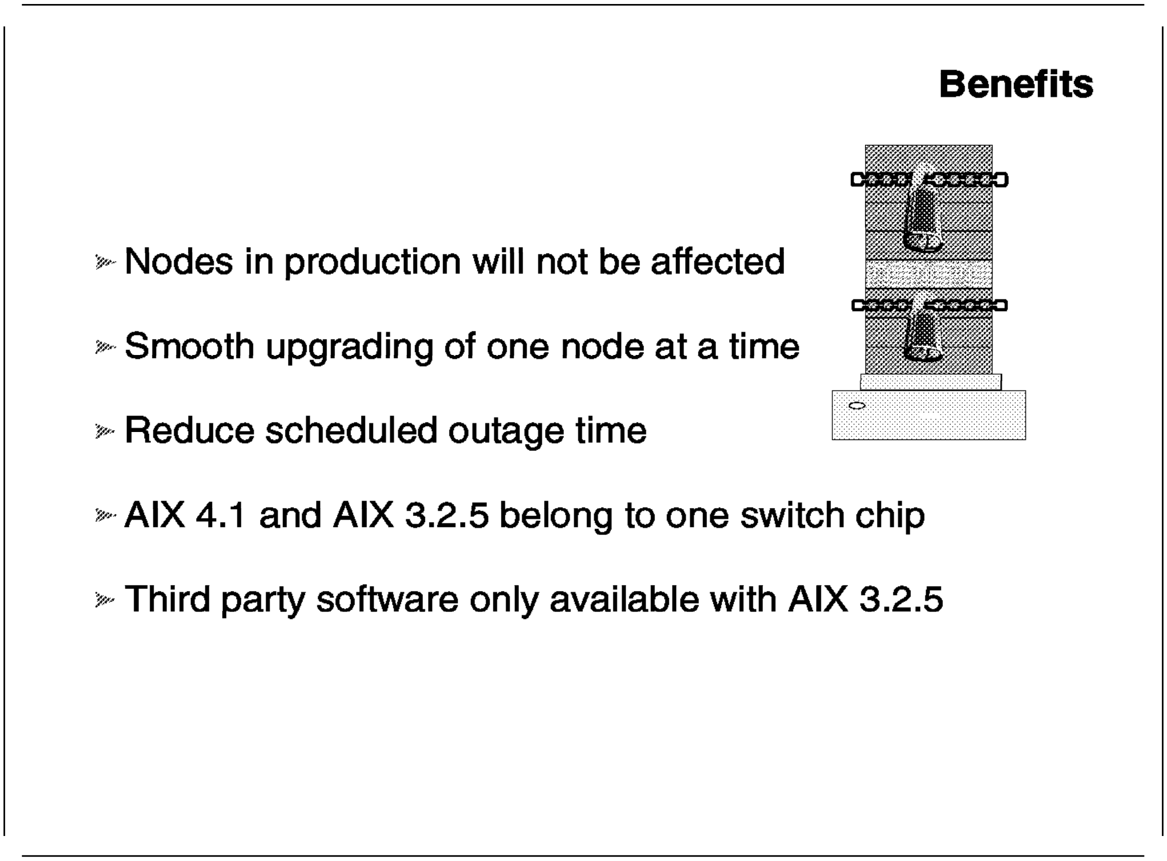
First of all, you must have PSSP 2.2 on the Control Workstation and AIX 4.1.4 installed. Which versions of software work together? On the RS/6000 SP nodes, you can have PSSP 2.2 and PSSP 1.2 in one system partition. AIX 4.1.4 is needed for PSSP 2.2, and AIX 3.2.5 is needed for PSSP 1.2. All PSSP 1.2 nodes must have at least PTF set 23 or later installed.

Alternatively, when you have only AIX 4 on the nodes, you can have PSSP 2.2 and PSSP 2.1 in a different partition or in the default partition. PSSP 2.1 or PSSP 2.2 support the new SP switch systems. AIX 4.1.4 or AIX 4.1.3 is required for PSSP 2.1 with PTF set 16 and AIX 4.1.4 for PSSP 2.2.

The requirement for PTF sets may change; see the Internet page `http://www.rs6000.ibm.com/tech` for the latest information on software coexistence.

At the end of 1996, AIX 4.2 will be supported by PSSP 2.2. When you want to use software coexistence and AIX 4.2, you must install AIX 4.2 on the Control Workstation and upgrade to the latest PTF set for PSSP 2.2 that supports AIX 4.2 binary compatibility. Then you can migrate a node to AIX 4.2 and PSSP 2.2.

3.2.1 Benefits



Benefits

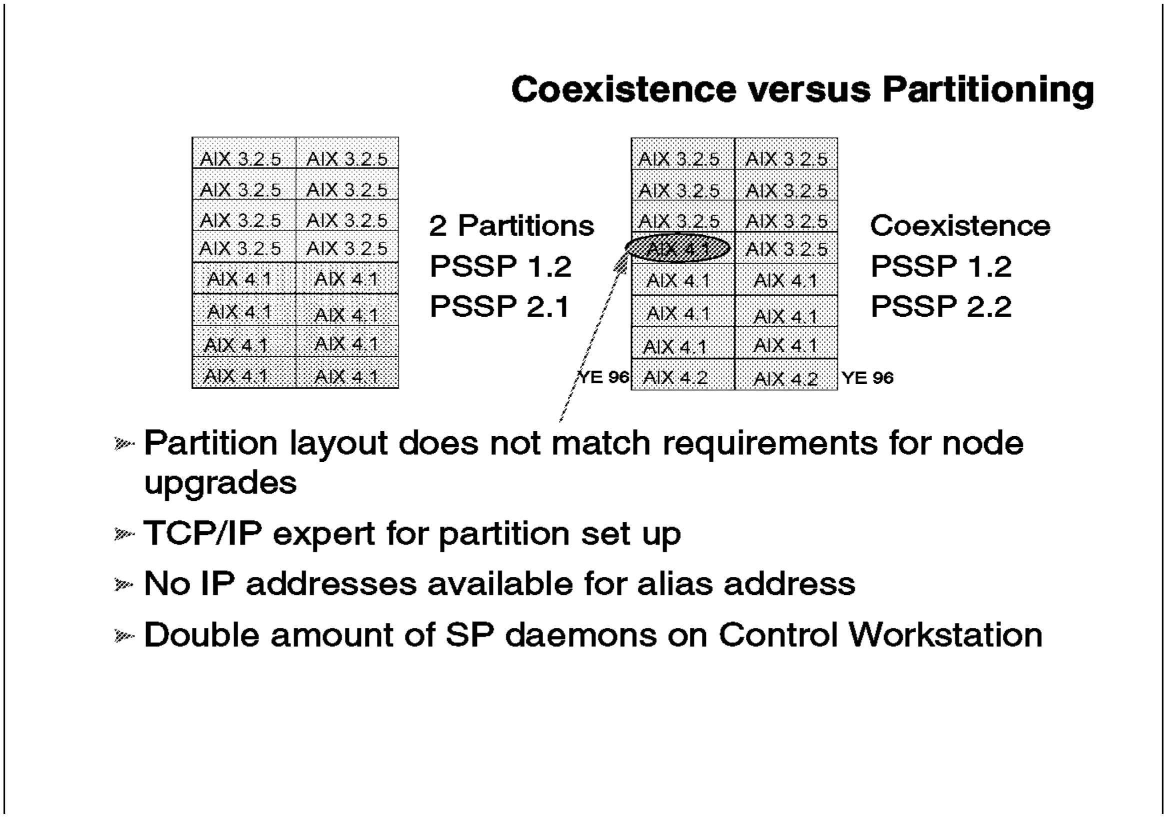
- Nodes in production will not be affected
- Smooth upgrading of one node at a time
- Reduce scheduled outage time
- AIX 4.1 and AIX 3.2.5 belong to one switch chip
- Third party software only available with AIX 3.2.5

The main benefit for PSSP coexistence (or mixed partition support) is for customers that use the RS/6000 SP in a commercial environment. Especially when you have consolidated your LAN, you do not want your RS/6000 SP nodes in production affected when you add new nodes (such as the new RS/6000 SP 604 High Nodes), or when you upgrade some nodes to a new software release.

Software coexistence is intended to be a migration aid, because you can upgrade one node at a time within one system partition. This support is for systems where system partitioning is not a viable solution, especially when you have a small system with the LC8 switch. System partitioning was not possible on small systems. With PSSP 2.2, AIX 4.1 and AIX 3.2.5 can share one switch chip.

Very often it is not possible to upgrade the operation system or the system software because your application software requires a specific level of AIX, or because it is only certified for a specific level of AIX. The software coexistence feature simplifies the installation and migration to new levels of AIX and PSSP, because you can leave the RS/6000 SP nodes at the old software level, and add new nodes (or migrate other nodes) to a new version.

3.2.2 Coexistence versus Partitioning



Coexistence versus Partitioning

AIX 3.2.5	AIX 3.2.5
AIX 3.2.5	AIX 3.2.5
AIX 3.2.5	AIX 3.2.5
AIX 3.2.5	AIX 3.2.5
AIX 4.1	AIX 4.1
AIX 4.1	AIX 4.1
AIX 4.1	AIX 4.1
AIX 4.1	AIX 4.1

2 Partitions
PSSP 1.2
PSSP 2.1

AIX 3.2.5	AIX 3.2.5
AIX 3.2.5	AIX 3.2.5
AIX 3.2.5	AIX 3.2.5
AIX 4.1	AIX 3.2.5
AIX 4.1	AIX 4.1
AIX 4.1	AIX 4.1
AIX 4.1	AIX 4.1
AIX 4.2	AIX 4.2

YE 96
YE 96

Coexistence
PSSP 1.2
PSSP 2.2

- Partition layout does not match requirements for node upgrades
- TCP/IP expert for partition set up
- No IP addresses available for alias address
- Double amount of SP daemons on Control Workstation

Support for system partitions was introduced in PSSP 2.1. With system partitioning, nodes that belong to one or more switch chips can be isolated. But in PSSP 2.1, all nodes in one system partition had to be at the same level of AIX and PSSP. The support was also aimed at isolating the switch traffic between different system partitions.

The preceding example shows a system on the left with two partitions. One partition uses AIX 3.2.5 and PSSP 1.2, and the other partition uses AIX 4.1 and PSSP 2.1. With PSSP 2.2, you can have one (default) partition and use AIX 4 and AIX 3.2.5 at the same time. When you have AIX 4 and AIX 3.2.5 in one partition, you must use PSSP 2.2 for AIX 4 and PSSP 1.2 for AIX 3.2.5. AIX 4.2 will be supported for PSSP 2.2 by the end of 1996.

The RS/6000 SP system on the right shows you a migration scenario, where you want to migrate node 9 from AIX 3.2.5 to AIX 4.1. With PSSP 2.1 and system partitioning, the minimum number of nodes that had to migrate to the new level are four nodes (node 9, node 10, node 13, and node 14). This mixed partition support is very helpful, because very seldom did the partition layout match the requirement for node upgrades, especially for small RS/6000 SP systems.

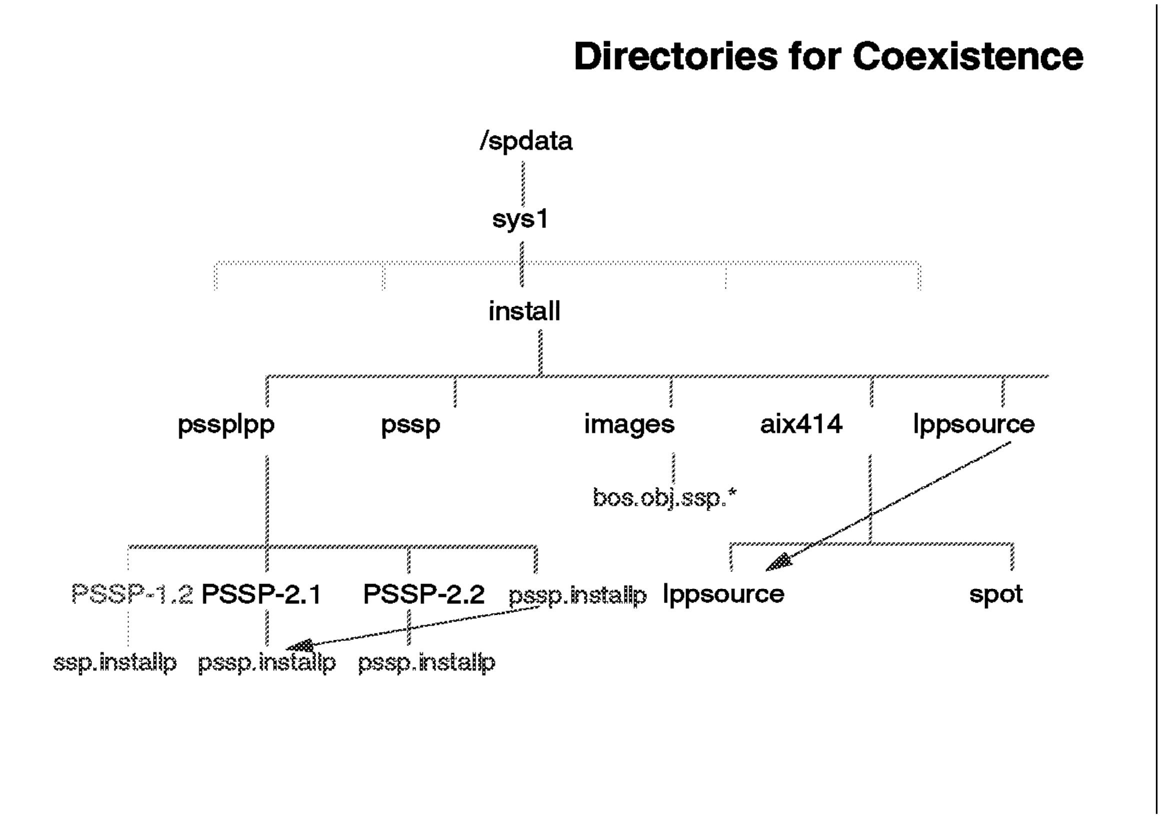
When you partition the RS/6000 SP, you have to add an IP alias to the file /etc/rc.net. If you use TCP/IP as a user and do not understand all TCP/IP terms, adding an IP alias is complicated. When you add this alias and make a mistake or a typo in /etc/rc.net, the Control Workstation may not boot anymore. The

Control Workstation administrator must then boot the Control Workstation in maintenance mode and modify /etc/rc.net.

Universities and other organizations that use the RS/6000 SP on the Internet may have a problem with available IP addresses. These institutions and companies have only a limited number of official addresses, and may have already used a possible alias address for another computer system.

If your RS/6000 SP system uses a RS/6000 model 250 as the Control Workstation, the double amount of daemons leads to lower performance of your system when you have two system partitions.

3.3 Planning and Preparation for Coexistence



This section describes the directory structure needed to support all versions of PSSP and to support AIX 4.1.4 and AIX 4.2.0.

Before installing the AIX and PSSP software images, the following directories under */spadata/sys1/install* must exist:

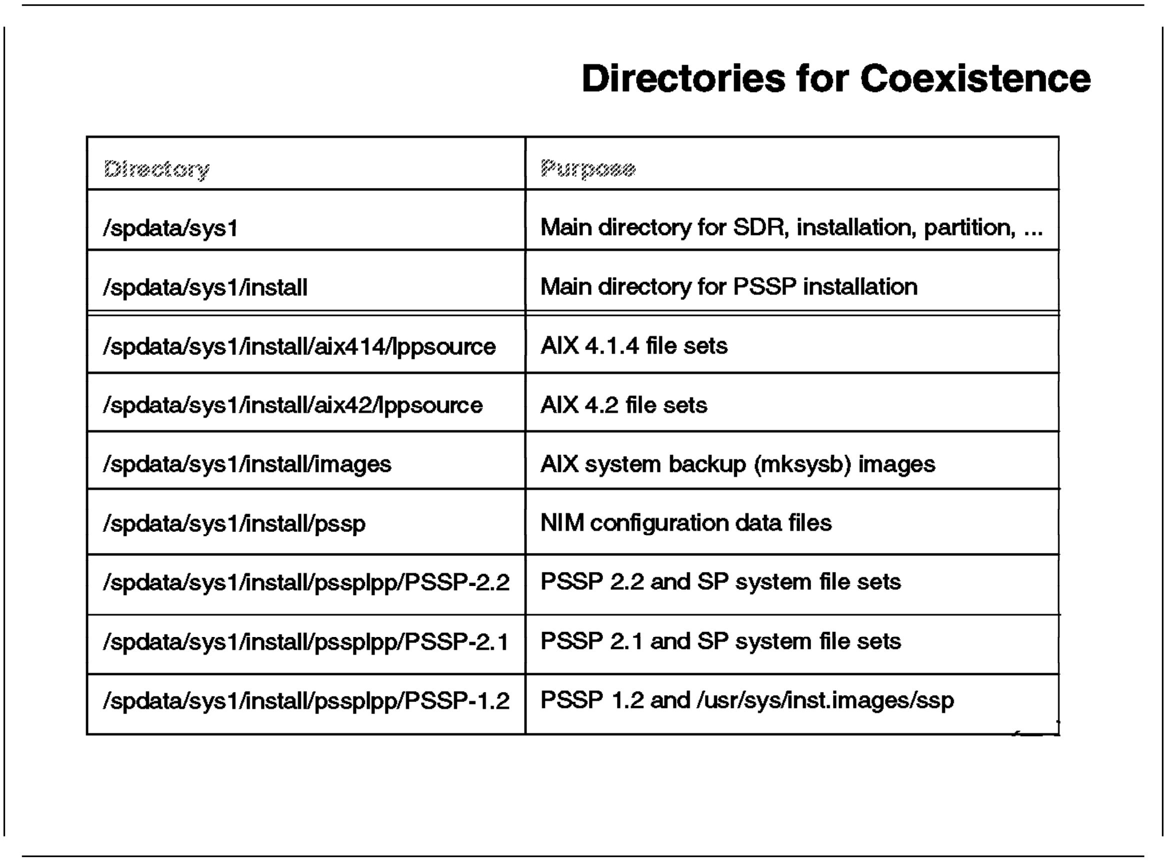
- pssplpp
- pssp
- images
- default

The pssplpp directory contains subdirectories with the name of the PSSP version. For example, version 1.2 of PSSP has to be named "PSSP-1.2" and version 2.2 of PSSP has the name "PSSP-2.2." This directory also contains the file pssp.installp. This file is a symbolic link to the file pssp.installp in the directory PSSP-2.1. Without this symbolic link, software coexistence with version 2.1 of PSSP would not be possible.

The pssp directory contains NIM configuration files.

The images directory contains different system backup versions. In order to support coexistence, you need at least one image for every version of AIX that you want to use on your RS/6000 SP system.

The default directory is the directory for the AIX images and for the NIM SPOT. This example does not have the default directory and uses the name aix414 instead. It is easier to understand that the directory aix414 contains all AIX images for the version 4.1.4. If you want to use a different version of AIX, create an aix42 directory for AIX 4.2. In order to support coexistence with PSSP 2.1, a symbolic link has to be made from directory /spdata/sys1/install/aix414/lppsource to directory /spdata/sys1/install/lppsource. This ensures that all data in lppsource can be accessed by PSSP 2.1.



Directories for Coexistence

Directory	Purpose
/spdata/sys1	Main directory for SDR, installation, partition, ...
/spdata/sys1/install	Main directory for PSSP installation
/spdata/sys1/install/aix414/lppsource	AIX 4.1.4 file sets
/spdata/sys1/install/aix42/lppsource	AIX 4.2 file sets
/spdata/sys1/install/images	AIX system backup (mksysb) images
/spdata/sys1/install/pssp	NIM configuration data files
/spdata/sys1/install/pssplpp/PSSP-2.2	PSSP 2.2 and SP system file sets
/spdata/sys1/install/pssplpp/PSSP-2.1	PSSP 2.1 and SP system file sets
/spdata/sys1/install/pssplpp/PSSP-1.2	PSSP 1.2 and /usr/sys/inst.images/ssp

The directories for software coexistence are:

Directories — **Description**

/spdata/sys1 — Main directory for SDR, SP monitor, log management, partition layout files, microcode, and installation

/spdata/sys1/install — Main directory for PSSP installation

/spdata/sys1/install/aix414/lppsource — Location of required AIX 4.1 file sets (The standard name is "default," not aix414.)

/spdata/sys1/install/aix42/lppsource — Location of required AIX 4.2 file sets (only needed when using AIX 4.2 on the nodes)

/spdata/sys1/install/images — Location of AIX system backup (mksysb) images for all versions of AIX

/spdata/sys1/install/pssp — Location of NIM configuration data files

/spdata/sys1/install/pssplpp/PSSP-2.2 — Location of all PSSP 2.2 and SP system file sets

/spdata/sys1/install/pssplpp/PSSP-2.1 — Location of all PSSP 2.1 and SP system file sets

/spdata/sys1/install/pssplpp/PSSP-1.2 — Location of all PSSP 1.2 and SP system file sets; also contains the /usr/sys/inst.images/ssp files from AIX 3.2.5

The following table shows the directories necessary for software coexistence and the sizes of PSSP and AIX that are needed for software coexistence.

Table 1. Directories for Software Coexistence

Directory	Size
/tftpboot	25 Mb
/spdata/sys1	60 Mb
/spdata/sys1/install/pssplpp/PSSP-1.2	980 Mb
/spdata/sys1/install/pssplpp/PSSP-2.1	310 Mb
/spdata/sys1/install/pssplpp/PSSP-2.2	215 Mb
/spdata/sys1/install/aix414	420 Mb minimum
/spdata/sys1/install/aix420	600 Mb minimum
/spdata/sys1/install/images	480 Mb average

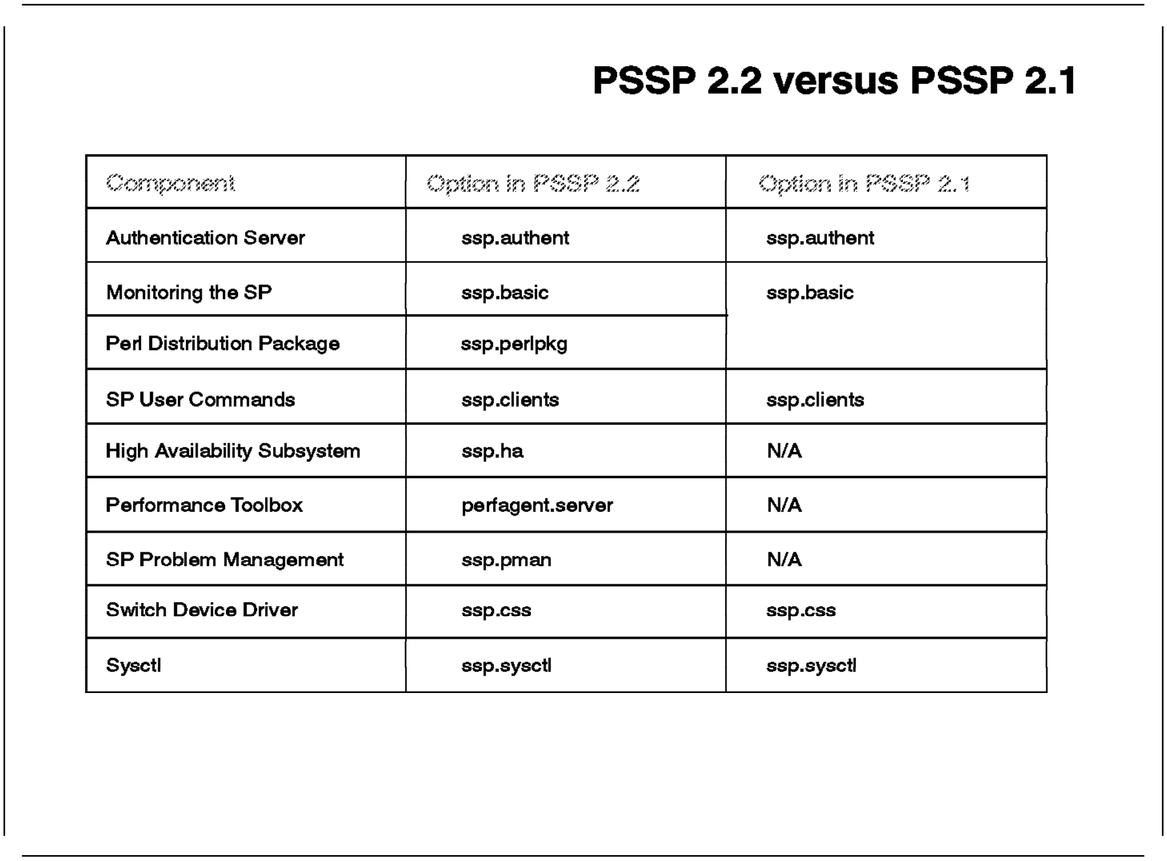
The directory /spdata/sys1/install/pssplpp/PSSP-1.2 is large, because it also contains the system backup from an AIX 3.2.5 node and all other files from the /usr/sys/inst.images/ssp directory.

Coexistence with PSSP 1.2

A symbolic link from /usr/sys/inst.images/ssp to /spdata/sys1/install/pssplpp/PSSP-1.2 is needed when you recreate a boot install server on an AIX 3.2.5 node. Use the following commands to create the symbolic link and to export the file system:

```
# ln -s /spdata/sys1/install/pssplpp/PSSP-1.2
        /usr/sys/inst.images/ssp
# /usr/sbin/mknfsexp -d /usr/sys/inst.images/ssp -t ro -B
```

3.4 PSSP 2.2 Compared to PSSP 2.1



PSSP 2.2 versus PSSP 2.1

Component	Option in PSSP 2.2	Option in PSSP 2.1
Authentication Server	ssp.authent	ssp.authent
Monitoring the SP	ssp.basic	ssp.basic
Perl Distribution Package	ssp.perlpkg	
SP User Commands	ssp.clients	ssp.clients
High Availability Subsystem	ssp.ha	N/A
Performance Toolbox	perfagent.server	N/A
SP Problem Management	ssp.pman	N/A
Switch Device Driver	ssp.css	ssp.css
Sysctl	ssp.sysctl	ssp.sysctl

This section describes the different packing of PSSP 2.2 and PSSP 2.1.

The *ssp.authent* file set must be installed on the Control Workstation when you use the SP Authentication Server. This package contains only minor changes from PSSP 2.1 to PSSP 2.2.

The *ssp.basic* file set was split into two PSSP options. The **Perl** programs were removed from ssp.basic and packed into *ssp.perlpkg*. This package contains Perl, version 4.0 patch level 36 and Perl, version 5.002 beta 2. This software package can be installed on any other system that you use.

The *ssp.clients* file set contains RS/6000 SP monitor command line interfaces, user authentication commands, libraries, and logging daemons. This package was enhanced to support the new RS/6000 SP 604 High Node, and the new high availability structure.

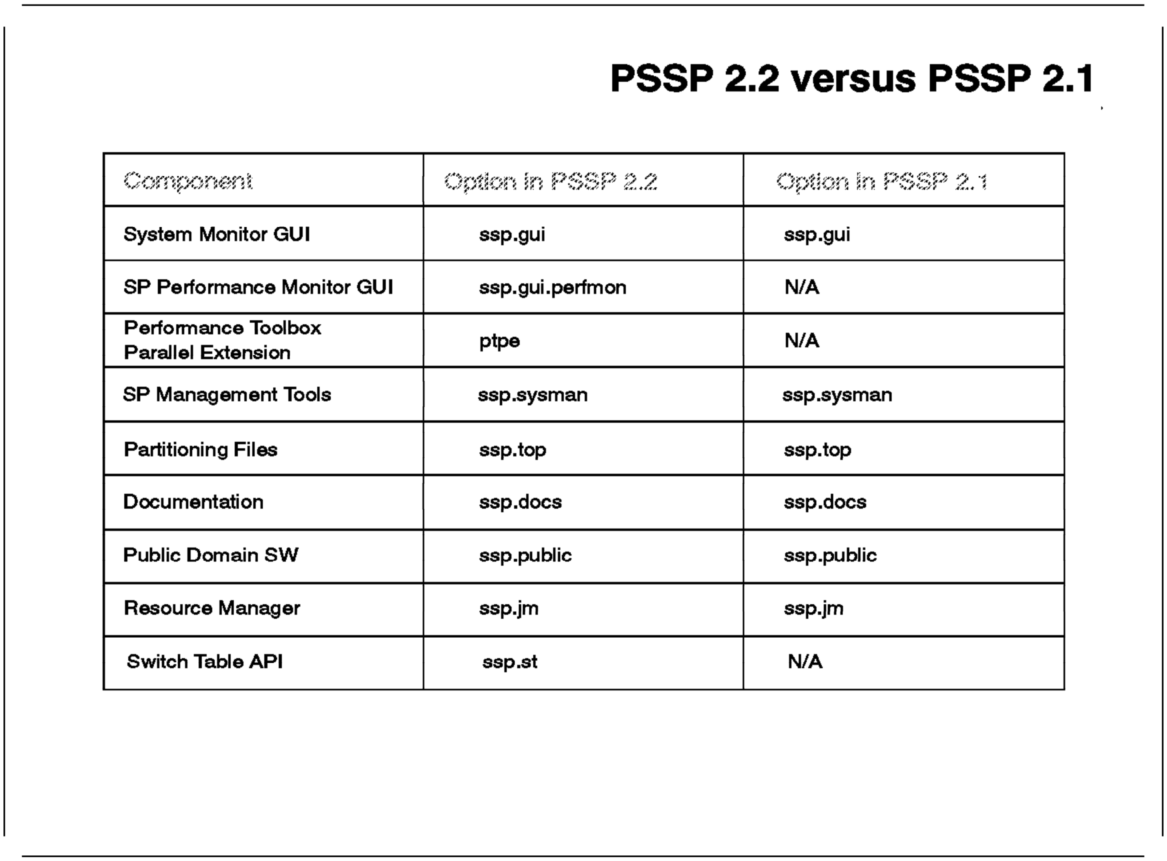
The *ssp.ha.* file set is new with PSSP 2.2. This package contains the availability subsystems, which include group services, event management, and the new heartbeat.

The *perfagent.server* file set is not part of PSSP 2.2, but it is required for ssp.ha. This package is part of the IBM Performance Toolbox for AIX (product number 5765-654).

The *ssp.pman* file set is also new with PSSP 2.2. This package contains the interface to the Event Management subsystem and creates start up scripts for events. It writes entries to error log and syslog. The ssp.pman package can also send SNMP (Simple Network Management Protocol) traps to any SNMP manager (like TME 10 Netview).

The *ssp.css* file set contains the device driver for the High Performance Switch and for the SP Switch. This package was enhanced to support software coexistence across the switch.

The *ssp.sysctl* file set contains system management scripts. These scripts (like pdf) use Kerberos for authentication. The ssp.sysctl package is required by ssp.ha.



PSSP 2.2 versus PSSP 2.1

Component	Option in PSSP 2.2	Option in PSSP 2.1
System Monitor GUI	ssp.gui	ssp.gui
SP Performance Monitor GUI	ssp.gui.perfmon	N/A
Performance Toolbox Parallel Extension	ptpe	N/A
SP Management Tools	ssp.sysman	ssp.sysman
Partitioning Files	ssp.top	ssp.top
Documentation	ssp.docs	ssp.docs
Public Domain SW	ssp.public	ssp.public
Resource Manager	ssp.jm	ssp.jm
Switch Table API	ssp.st	N/A

The *ssp.gui* file set contains in PSSP 2.2 the RS/6000 SP system monitor GUI and the new *Perspectives* GUI. The Perspectives program is the consolidated system interface for the RS/6000 SP. It provides a common launch for PSSP system management applications, including hardware monitoring and control capabilities. This interface is tightly integrated with the problem management infrastructure.

The *ssp.gui.perfmon* file set is the Perspectives part of Performance Toolbox Parallel Extensions for AIX (PTPE). The *ptpe* (PTPE) file set is a feature of PSSP 2.2, and it is required for ssp.gui.perfmon. PTPE is an extension to IBM Performance Toolbox for AIX that has the capability to performance monitor SP-specific subsystems. It provides the utilities to monitor, store, and retrieve performance information collected on RS/6000 SP subsystems, for example, the High Performance Switch, LoadLeveler, and Virtual Shared Disk.

The *ssp.sysman* file set contains the RS/6000 SP system management tools, including:

- File collections
- Login control
- Accounting support
- Network Time Protocol (NTP)
- User management support, such as the BSD automount (AMD)
- Print support

This package includes some new commands (like lppdiff) in PSSP 2.2.

The *ssp.top* file set was enhanced with a new function called the “System Partitioning Aid.” This program allows more flexible configurations of system partitions and allows you to create your own partition layout.

The *ssp.docs* file set contains the documentation and *man* pages for all commands.

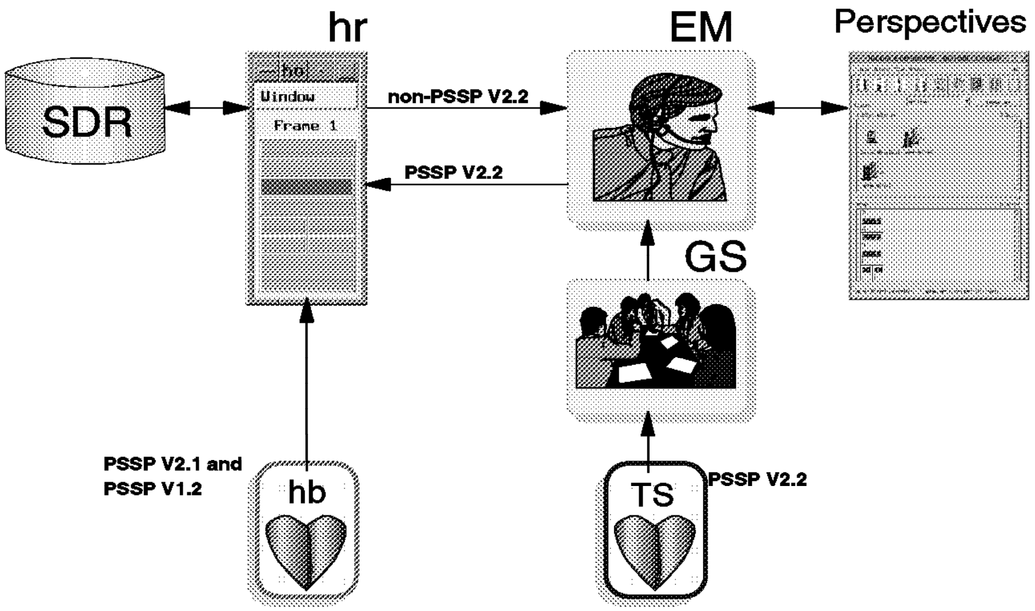
The *ssp.public* file set contains the public domain source code for AMD, Perl 4 and Perl 5, SUP, NTP, Tcl, TclX, TK-X11, and Expect. The Perl 5 source code is new in PSSP 2.2. The ssp.public can be installed on any other system and you can compile the programs.

The *ssp.jm* file set is the Resource Manager for parallel application scheduling. This package was enhanced to support LoadLeveler 1.3.

The *ssp.st.* file set is new with PSSP 2.2. This package is the new switch API. It contains the low level application programming interface for loading, unloading and querying the job switch resource table. This package is mutually exclusive with ssp.jm.

3.5 How Does PSSP Coexistence Work?

How Does PSSP Coexistence Work?

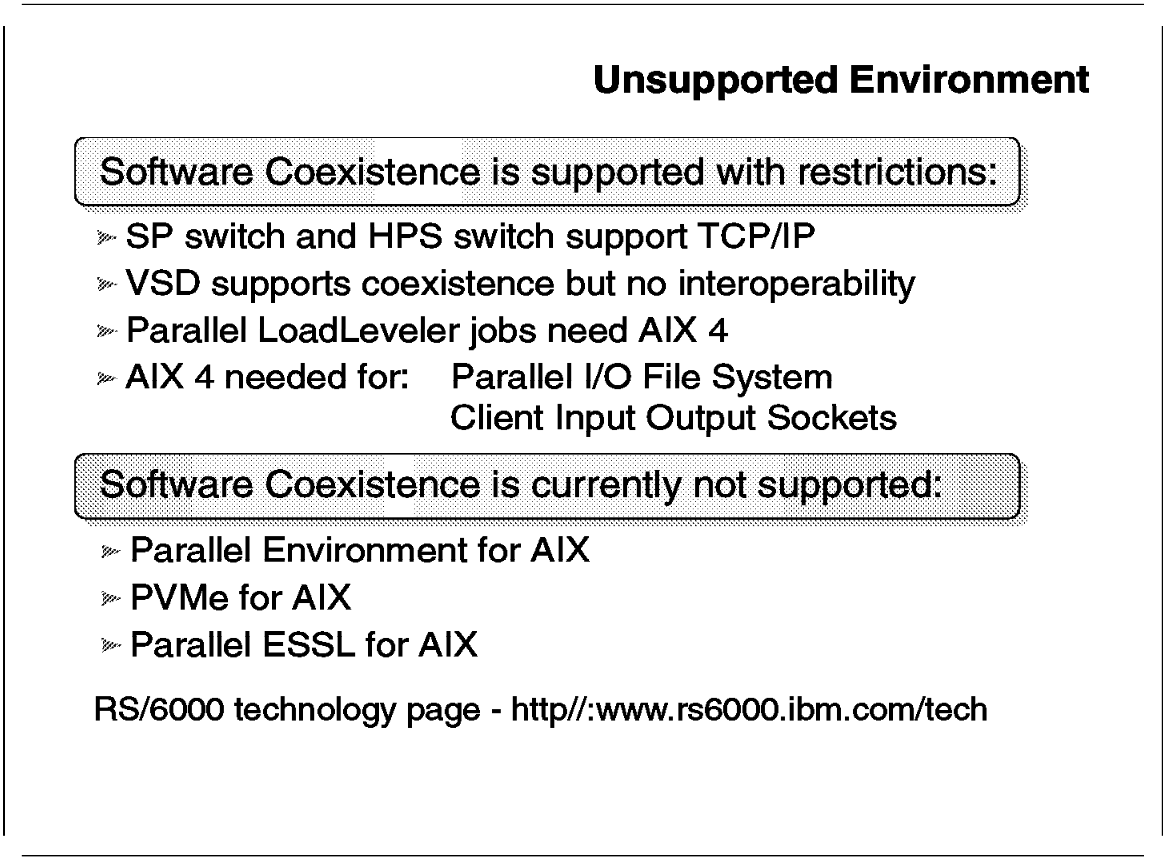


The above foil illustrates how the coexistence of heterogeneous PSSP softwares and AIX Operating Systems are implemented in PSSP 2.2. Within a system partition, you can have PSSP 2.2 and PSSP 2.1 or PSSP 2.2 and PSSP 1.2. The host response of PSSP 2.1 and PSSP 1.2 are provided directly from their respective heartbeat to host response daemon. With PSSP 2.2, Topology Services is responsible for providing the heartbeat function. Although, Topology Services heartbeats through the switch (CSS) and Ethernet, the host response is still done through the Ethernet. The host response information about PSSP 2.2 is provided to the host response daemon from Topology Services through Group Services to Event Management and this information is made available to the host response daemon. Also, information about PSSP Version 2.1 or PSSP 1.2 is available to Event Management from host response for use by Perspectives.

Coexistence with AIX 3.2.5

If you use AIX 3.2.5 nodes then you must define the *netinst* user ID on the Control Workstation. This user ID is needed by the AIX 3.2.5 boot install servers.

3.6 Unsupported Environment



Unsupported Environment

Software Coexistence is supported with restrictions:

- SP switch and HPS switch support TCP/IP
- VSD supports coexistence but no interoperability
- Parallel LoadLeveler jobs need AIX 4
- AIX 4 needed for: Parallel I/O File System
  Client Input Output Sockets

Software Coexistence is currently not supported:

- Parallel Environment for AIX
- PVMe for AIX
- Parallel ESSL for AIX

RS/6000 technology page - http//:www.rs6000.ibm.com/tech

The following programs support software coexistence but have some restrictions:

- The SP switch and the High Performance Switch switch support TCP/IP between nodes with different levels of PSPP. User space communication must use the same level of PSSP within one partition.
- The IBM Virtual Shared Disk (VSD) can be configured within the same PSSP level. You can define multiple VSDs for every level of PSSP and for every level of AIX. Interoperability across different levels of PSSP is not supported.
- The Recoverable Virtual Shared Disk (RVSD) treats nodes running earlier releases as “down.” With RVSD 1.2, you can change the disk quorum setting, but you cannot do so with earlier releases. So when you upgrade more than half of the VSD nodes to PSSP 2.2, the VSD nodes with PSSP 1.2 or PSSP 2.1 will become inactive.
- LoadLeveler 1.2.1 supports AIX 4.1, and LoadLeveler 1.2.0 supports AIX 3.2.5. These two levels of LoadLeveler work together and enable you to schedule serial jobs across AIX 3.2.5 and AIX 4.1. LoadLeveler 1.3 does not coexist with LoadLeveler 1.2.1; however, with LoadLeveler 1.3 and LoadLeveler 1.2.1, you have software coexistence with PSSP 2.1 and PSSP 2.2. That implies that you can run parallel jobs within a mixed partition with PSSP 2.1 and PSSP 2.2, but you can use only one level of LoadLeveler.

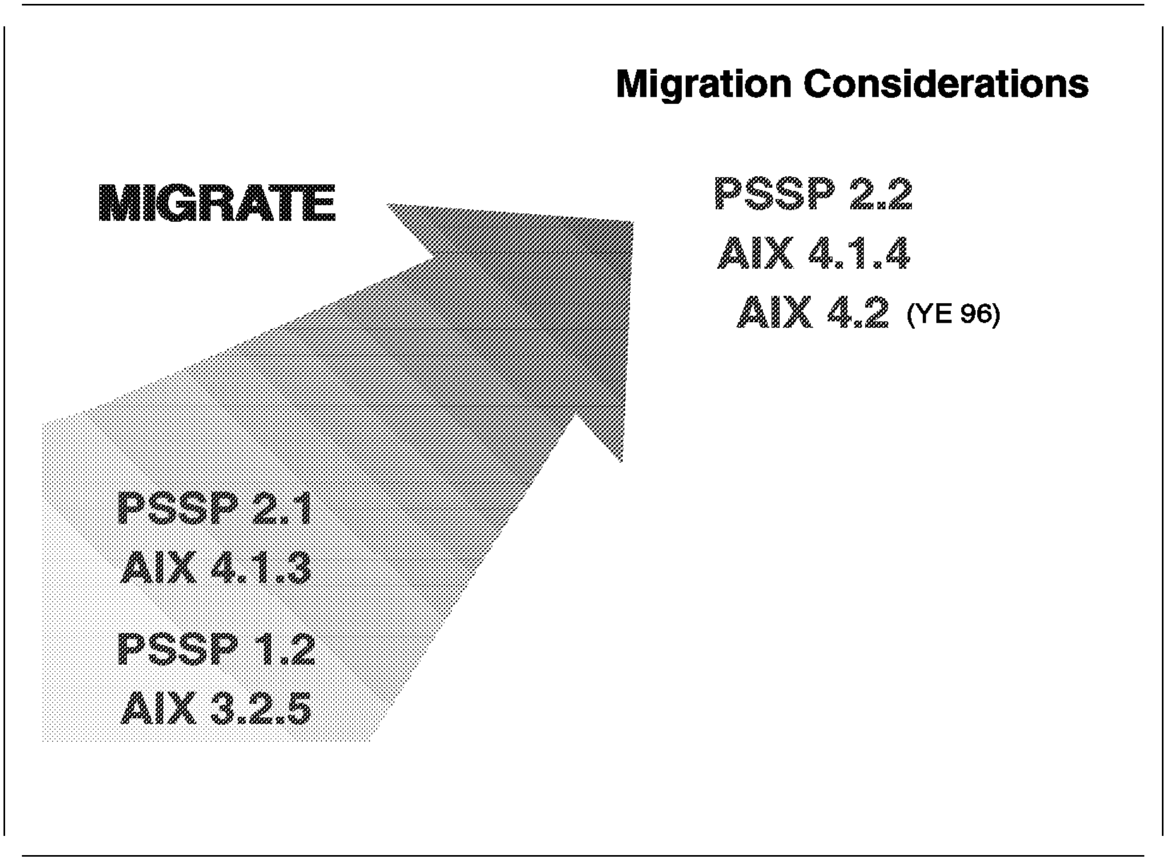
- The Parallel I/O File System program needs AIX 4.1.
- The Client Input Output Sockets program also requires AIX 4.1.

The following programs do not support software coexistence:

- Parallel Environment for AIX
- PVMe for AIX
- Parallel ESSL for AIX

For more information, see the RS/6000 technology page (`http://www.rs6000.ibm.com/tech`) on the Internet.

Chapter 4. Migration Considerations



Migration Considerations

MIGRATE

PSSP 2.1
AIX 4.1.3

PSSP 1.2
AIX 3.2.5

PSSP 2.2
AIX 4.1.4
AIX 4.2 (YE 96)

For RS/6000 SP, the term *migrate* has different meanings. The Control Workstation and the RS/6000 SP nodes can be migrated. If the nodes will be upgraded to a new version of AIX or PSSP, the Control Workstation must be at the latest software level.

4.1 Overview

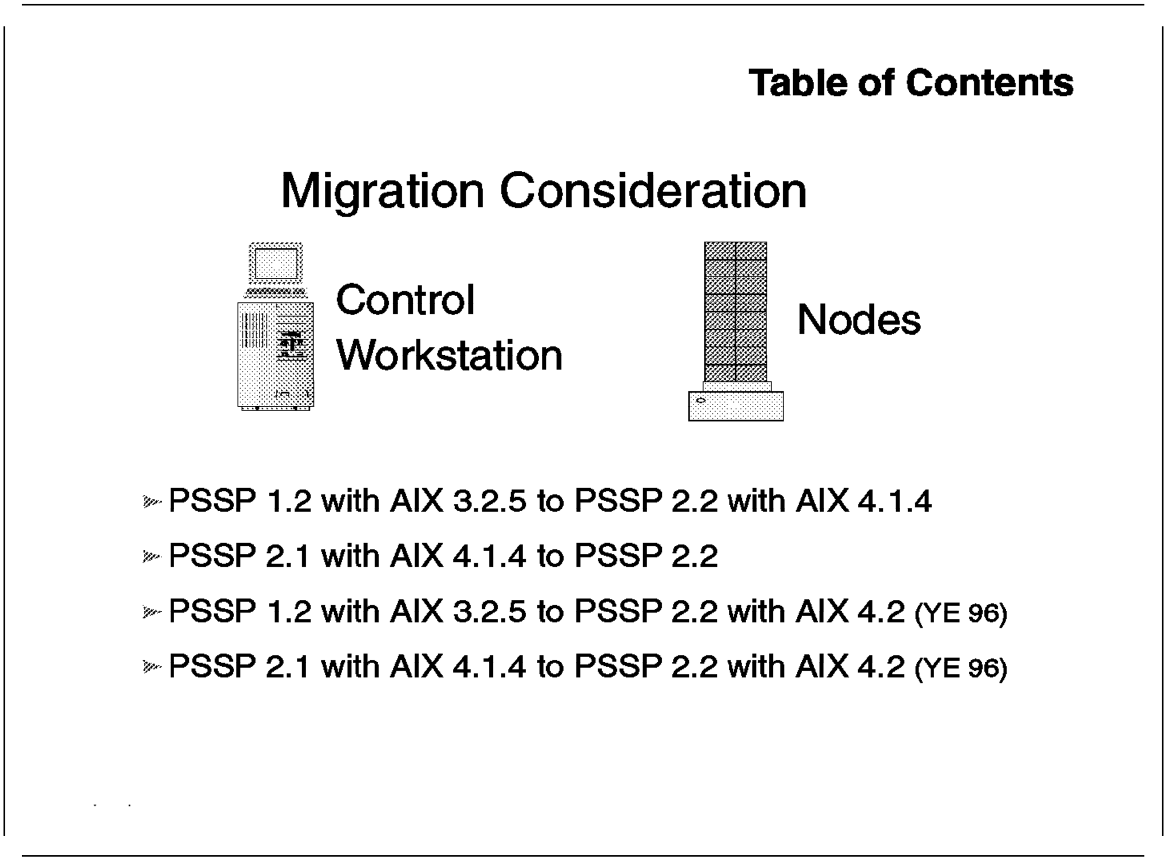


Table of Contents

Migration Consideration

Control Workstation

Nodes

- PSSP 1.2 with AIX 3.2.5 to PSSP 2.2 with AIX 4.1.4
- PSSP 2.1 with AIX 4.1.4 to PSSP 2.2
- PSSP 1.2 with AIX 3.2.5 to PSSP 2.2 with AIX 4.2 (YE 96)
- PSSP 2.1 with AIX 4.1.4 to PSSP 2.2 with AIX 4.2 (YE 96)

This table of contents shows different scenarios when migrating from one level of PSSP to PSSP 2.2, and from one level of AIX to AIX 4.1.4.

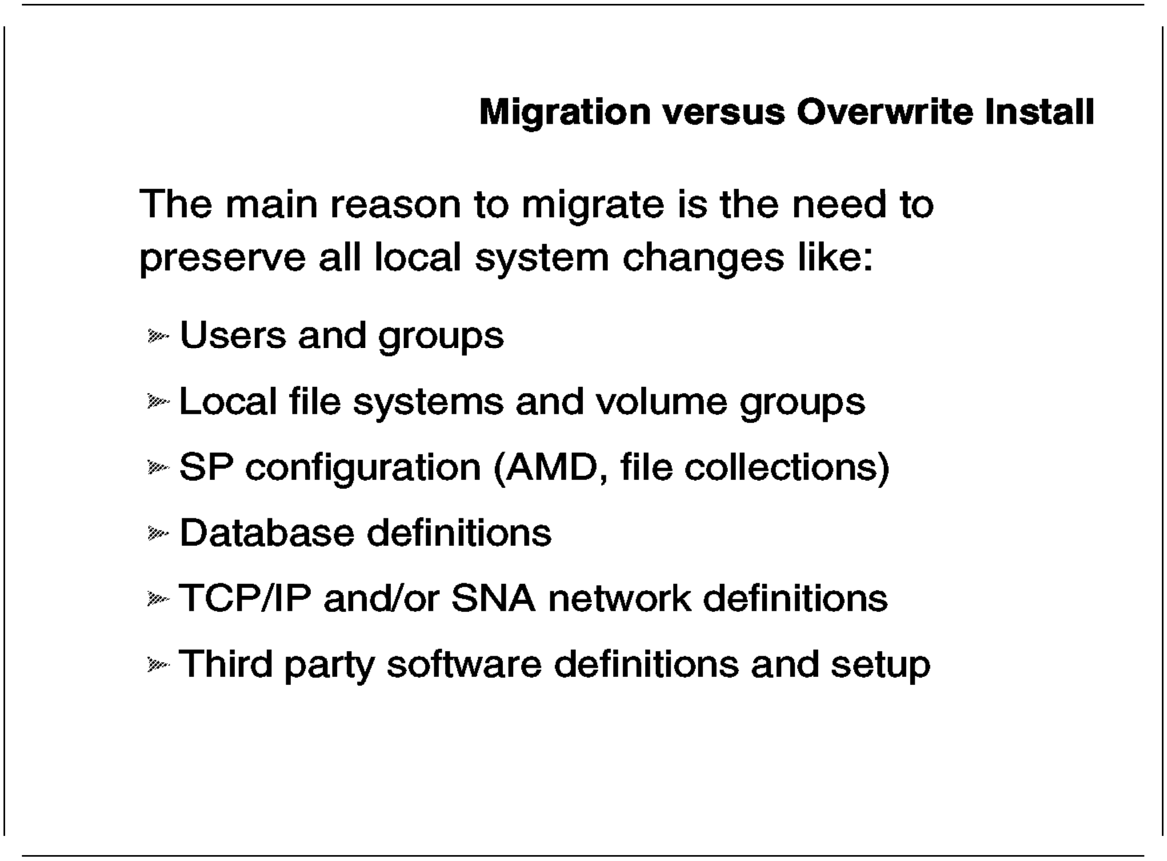
The following sections describe the process of migrating the Control Workstation to AIX 4.1.4. AIX 4.2 will be supported at the end of 1996. The migration to AIX 4.2 is very similar to the migration of AIX 4.1.4. For more information refer to chapter Chapter 5, “AIX 4.2 Support” on page 189, or see the *AIX Version 4.2 Installation Guide*, SC23-1924, for the AIX 4.2 migration process.

Following, the migration process from PSSP 1.2 and AIX 3.2.5 to PSSP 2.2 and AIX 4.1.4 is explained in detail.

The migration process from PSSP 2.1 and AIX 4.1.4 to PSSP 2.2 is much easier, because you upgrade only PSSP. An example of this process is installing PTF sets of PSSP.

If your operating system is at level AIX 4.1.3, follow the procedures as described in the “Read Me First” manual with AIX 4.1.4 to upgrade your Control Workstation to AIX 4.1.4.

4.2 Reasons for Migration



Migration versus Overwrite Install

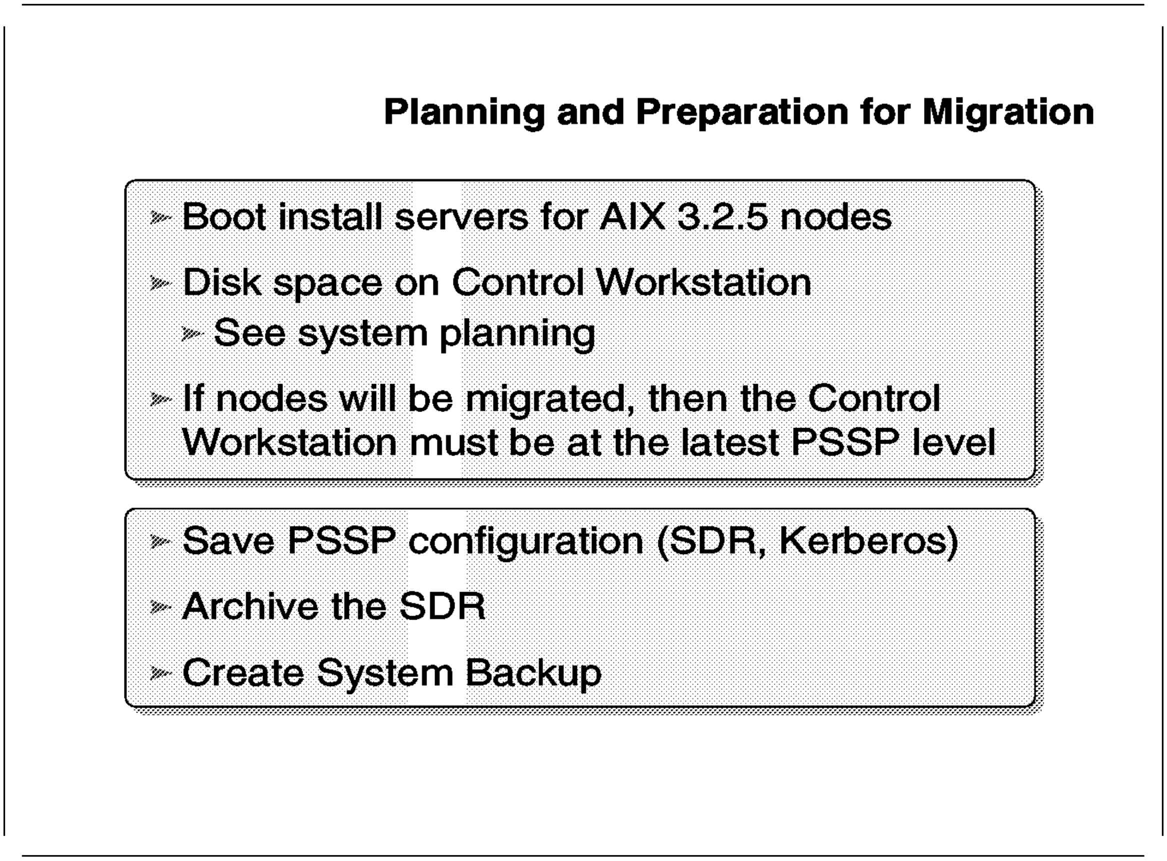
The main reason to migrate is the need to preserve all local system changes like:

- Users and groups
- Local file systems and volume groups
- SP configuration (AMD, file collections)
- Database definitions
- TCP/IP and/or SNA network definitions
- Third party software definitions and setup

Why should you migrate? The main reason to migrate is to preserve all local system changes, such as:

- Users and groups
- Local file systems and volume groups
- RS/6000 SP setup (AMD, file collections)
- Database definitions
- TCP/IP and SNA network setup
- Third party software definitions and setup
- Reduced outage time

4.3 Planning and Preparation for Migration



Planning and Preparation for Migration

- Boot install servers for AIX 3.2.5 nodes
- Disk space on Control Workstation
  - See system planning
- If nodes will be migrated, then the Control Workstation must be at the latest PSSP level

- Save PSSP configuration (SDR, Kerberos)
- Archive the SDR
- Create System Backup

Before you start to migrate the RS/6000 SP system, you have to plan the following:

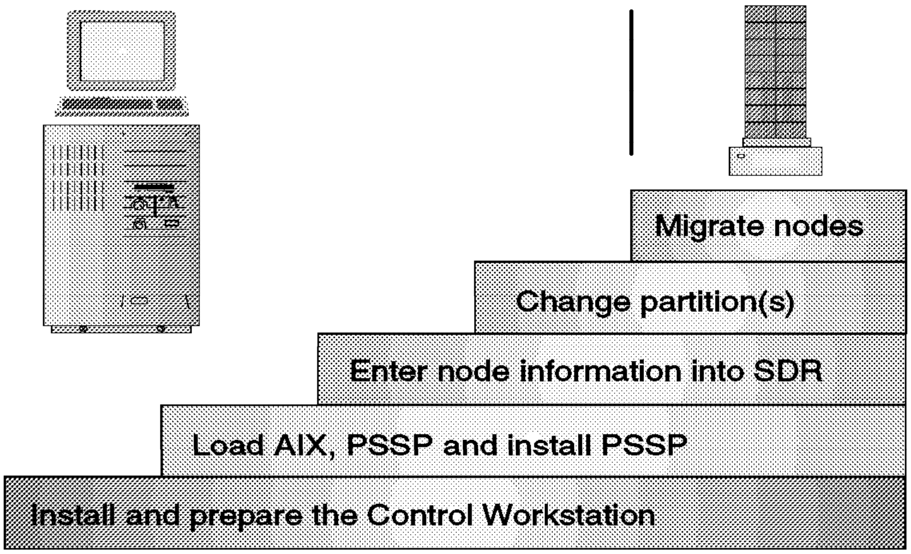
- Where your boot install servers for the AIX 3.2.5 nodes will reside
- How much disk space on the Control Workstation will be needed

When you plan to migrate only some nodes, you must first migrate the Control Workstation to the latest level of PSSP. The Control Workstation must also be at the same level of AIX (or at a higher level of AIX) as an RS/6000 SP node.

Note: For RS/6000 SP High Nodes, you need the PTF for APAR IX57164.

4.4 CWS Migration Steps

CWS Migration Steps



Migrate nodes

Change partition(s)

Enter node information into SDR

Load AIX, PSSP and install PSSP

Install and prepare the Control Workstation

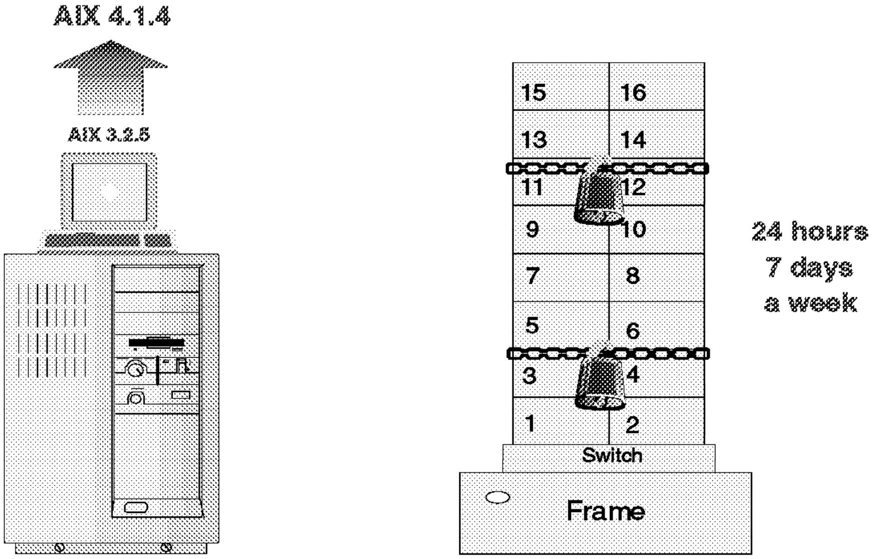
When migrating an RS/6000 SP system, you have to complete five major checkpoints. All checkpoints have one or more steps to fulfill. Following are the major checkpoints:

1. Install and prepare the Control Workstation.
2. Load the AIX and PSSP image on the hard disk of the Control Workstation and install PSSP 2.2.
3. Restore the saved System Data Repository (SDR), or enter the node information into the SDR.
4. Change the default partition, or create a new partition.
5. Prepare the SDR to migrate a node, and network boot the node for the migration process.

The greatest amount of work has to be done on the Control Workstation. The migration of the Control Workstation requires a lot of planning, because you upgrade one node at a time without disturbing the other nodes.

4.5 Migrate the CWS from AIX 3.2.5 to AIX 4.1.4

Migrate CWS to AIX 4.1.4



This section describes the migration scenario for the CWS when changing the operating system from AIX 3.2.5 to AIX 4.1.4.

The most important fact in this scenario is that the nodes must be available 24 hours 7 days a week, and the switch has to work. Availability is very important, because the nodes are in production and will be migrated later.

This scenario is not a typical one. It assumes a Control Workstation with 888 flashing in the LED display, which indicates the Control Workstation is not functioning right. Since you want to use new levels of AIX on the RS/6000 SP nodes, this represents a good opportunity to upgrade the Control Workstation software. The Control Workstation will therefore be installed with the latest software. The following sections illustrate how the Control Workstation can be upgraded without touching the nodes.

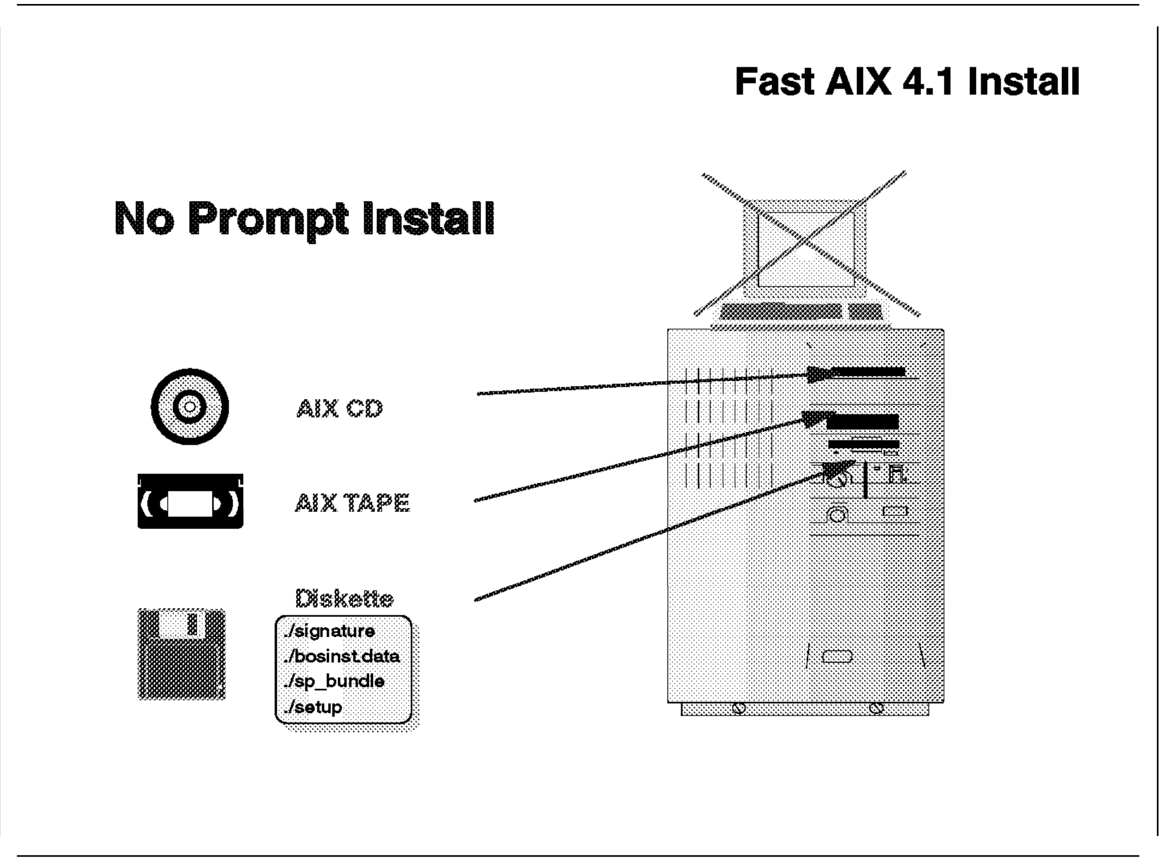
First we will look at the steps to upgrade the Control Workstation from PSSP 1.2 to PSSP 2.2.

A later section describes the migration path from PSSP 1.2 to PSSP 2.1.

4.5.1 Migration from PSSP 1.2 to PSPP 2.2

This section describes a different way to upgrade the Control Workstation from PSSP 1.2 to PSSP 2.2. (The default way to migrate from PSSP 1.2 to the latest level of PSSP is described in the *Installation and Migration Guide*, GC23-3898.)

4.5.1.1 Fast AIX 4.1 Install



Fast AIX 4.1 Install

No Prompt Install

AIX CD

AIX TAPE

Diskette

./signature
./bosinst.data
./sp_bundle
./setup

Install AIX 4.1.4 on a different disk from AIX 3.2.5. This enables you to boot later from the AIX 3.2.5 system disk. The fastest way to install AIX 4.1.4 is to create a diskette with the “no prompt” options. The diskette contains at least four files:

- ./signature
- ./bosinst.data
- ./sp_bundle
- ./setup

To create the ./signature file, use:

```
# echo ″data″ > signature
```

Then use your favorite editor to create the bosinst.data file:

```
control_flow:
    CONSOLE = /dev/lft0
    INSTALL_METHOD = overwrite
    PROMPT = no
    EXISTING_SYSTEM_OVERWRITE = yes
    INSTALL_X_IF_ADAPTER = yes
    RUN_STARTUP = no
    RM_INST_ROOTS = no
    ERROR_EXIT =
    CUSTOMIZATION_FILE = /../setup
    TCB = no
    INSTALL_TYPE = full
    BUNDLES = /../sp_bundle

target_disk_data:
    LOCATION =
    SIZE_MB =
    HDISKNAME = hdisk0

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

The file sp_bundle lists the additional software that should be installed on the Control Workstation:

```
bos.diag
bos.net
bos.rte.up
bos.perf
bos.sysmgt
bos.terminfo
X11.Dt
X11.apps
X11.base
X11.fnt.coreX
X11.fnt.defaultFonts
X11.fnt.fontServer
X11.fnt.iso1
X11.fnt.util
X11.loc.En_US
X11.loc.en_US
X11.motif
X11.msg.En_US
X11.msg.en_US
X11.vsm
```

The setup script is a generic script that executes some configuration steps for the setup of the Control Workstation. This script is similar to the script.cust file. You can add most of the script.cust file to the setup script, or you can even define your complete Control Workstation setup. Execute the following setup steps:

```
#
# step 1
#
cd /
restore -xqvf /dev/rfd0 ./.profile
#
# step 2
#
mknfs -B
#
# step 5
#
```

```
mkdev -c tty -t tty -s rs232 -p sa0 -w s1 -a speed=19200
#
# step 8
#
chdev -l sys0 -a maxuproc=′256′
chdev -l sys0 -a maxmbuf=′16384′
#
# step 9
#
mklv -y′tftplv′ rootvg 6
crfs -v jfs -d′tftplv′ -m′/tftpboot′ -A′yes′ -p′rw′ -t′no′ \
 -a frag=′4096′ -a nbpi=′4096′ -a compress=′no′
#
# add some additional setup steps here
#
```

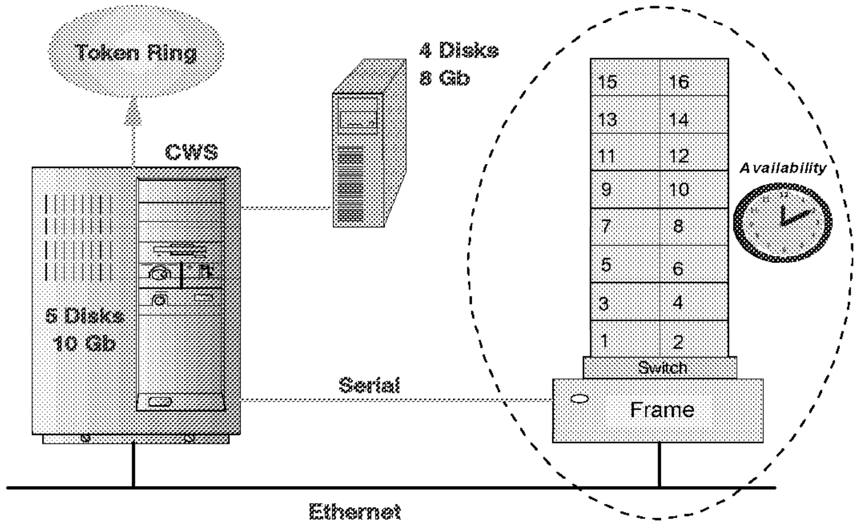
When you have created all files, back those files up to a diskette. Use the following command:

```
# ls ./bosinst.data ./signature ./sp_bundle ./setup ./.profile
   | backup -ivq
```

Then insert the AIX 4.1.4 Server CD or the AIX 4.1.4 Server tape. Turn the key to the service position and boot the Control Workstation in maintenance mode. AIX starts the installation and reads the additional information from the diskette. After one hour, your Control Workstation should be ready for the next steps. (The length of the installation time is dependent on the speed of your machine.)

4.5.1.2 Prepare the Control Workstation

Prepare Control Workstation



Since the System Data Repository (SDR) was not preserved, the network setup, the Kerberos setup, the Control Workstation configuration, and “enter the node information into the SDR” steps have to be done.

For the network setup see the *Installation and Migration Guide*, GC23-3898, Chapter 4, “Step 10: Configure Ethernet Adapters.”

Change the network tunables on the Control Workstation interface and add the following lines to the end of /etc/rc.net:

```
case `hostname` in
sp2cw*)
# tunable parameter for the CWS
        /usr/sbin/no -o thewall=16384
        /usr/sbin/no -o sb_max=163840
        /usr/sbin/no -o tcp_sendspace=65536
        /usr/sbin/no -o tcp_recvspace=65536
        /usr/sbin/no -o tcp_mssdflt=1448
        /usr/sbin/no -o udp_sendspace=32768
        /usr/sbin/no -o udp_recvspace=65536
        /usr/sbin/no -o ipforwarding=1
        /usr/sbin/no -o rfc1323=1
        /usr/sbin/no -o subnetsarelocal=1
;;
esac
```

The space for the Network Installation Management (NIM) Boot Images was created from the setup script at installation time.

In this example, the customer environment has grown over the years, so we have nine hard disks. We will use three disks for /spdata and three disks for the rootvg. The layout is listed in Table 2.

Table 2. Hard Disk Layout

hdisk	Volume Group	Size	Description
hdisk0	rootvg	2.0 Gb	rootvg for AIX 4.1.4
hdisk1	Not active	2.0 Gb	rootvg for AIX 4.2
hdisk2	db2vg	2.0 Gb	DB2 volume group
hdisk3	spaix4vg	2.0 Gb	VG for AIX 4.1.4 lppsource, PSSP 2.1 and PSSP 2.2
hdisk4	spaix32vg	1.0 Gb	PSSP 1.2 volume group
hdisk5	Not active	1.0 Gb	Empty
hdisk6	Not active	2.0 Gb	rootvg for AIX 3.2.5
hdisk7	lotusvg	2.0 Gb	Volume group for Lotus software and paging space
hdisk8	spdatavg	2.0 Gb	spdata volume group
hdisk9	Not active	2.0 Gb	Empty

Import all SP volume groups, create or change file systems, and load AIX 4.1.4 and PSSP 2.2, and use the new PSSP 2.2 directory layout. For detailed information, refer to the *Installation and Migration Guide*, GC23-3898, Chapter 4, step 13 to step 18.

Following are the file systems we use:

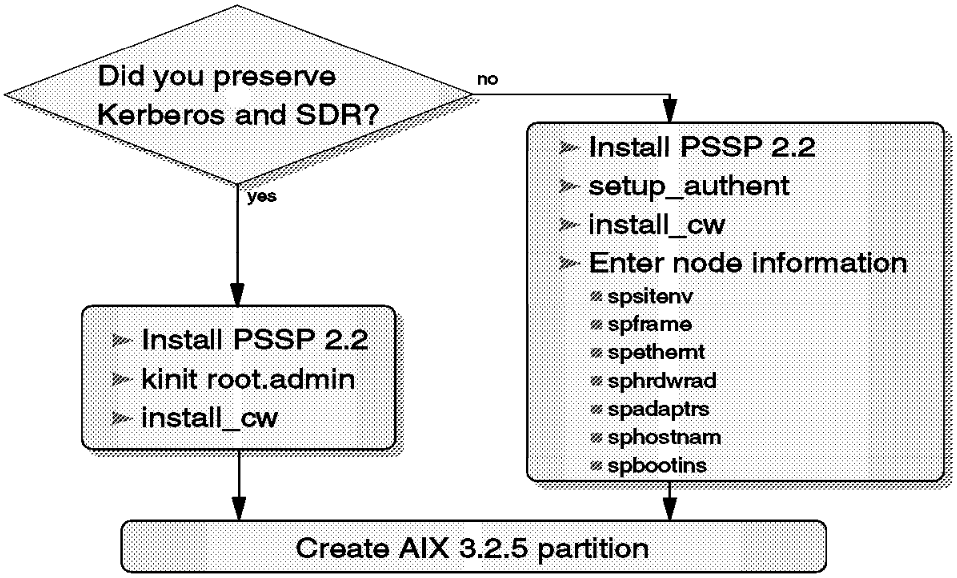
Table 3. CWS File Systems

Directory	Size	Volume Group
/tftpboot	25 Mb	rootvg
/spdata	40 Mb	spdatavg
/spdata/sys1/install/aix414	420 Mb	spaix4vg
/spdata/sys1/install/images	480 Mb	spdatavg
/spdata/sys1/install/pssplpp/PSSP-2.2	215 Mb	spaix4vg
/spdata/sys1/install/pssplpp/PSSP-1.2	980 Mb	spaix32vg

The /spdata/sys1/install/pssplpp/PSSP-1.2 file system was the /usr/sys/inst.images/ssp file system on AIX 3.2.5.

4.5.1.3 Restore Kerberos and SDR Information

Kerberos and SDR Information



Did you preserve Kerberos and SDR?

no

- Install PSSP 2.2
- setup_authent
- install_cw
- Enter node information
  - spsitenv
  - spframe
  - spethernt
  - sphrdwrad
  - spadaptrs
  - sphostnam
  - spbootins

yes

- Install PSSP 2.2
- kinit root.admin
- install_cw

Create AIX 3.2.5 partition

Restore the files that have been saved to the /tmp directory. The files are:

- /etc/bootptab.info
- /etc/passwd
- /etc/security/passwd
- /etc/group
- /etc/security/group
- /etc/amd/amd-maps/amd.u

Use the `installp` command or use `smit install_latest` to install PSSP 2.2. The example shows the `installp` command:

```
# cd /spdata/sys1/install/aix414/lppsource
# installp -acgXd. perfagent.server
# cd /spdata/sys1/install/pssplpp/PSSP-2.2
# mv ssp.usr.2.2.0.0 pssp.installp
# inutoc .
# installp -acgXd . ssp.basic      \
                    ssp.clients    \
                    ssp.authent    \
                    ssp.css        \
                    ssp.gui        \
                    ssp.ha         \
                    ssp.perlpkg    \
                    ssp.pman       \
                    ssp.sysctl     \
                    ssp.sysman      \
                    ssp.top
```

For the next steps, run setup_authent. We are setting up Kerberos, because the TCP/IP addresses on the Control Workstation and nodes have changed. Enable the ~root/.rhosts file, so that normal rcp, rsh, and rlogin will work. Use this setup only for the migration process, and remove all .rhosts files when you have finished. The ~root/.rhosts file has the following entries:

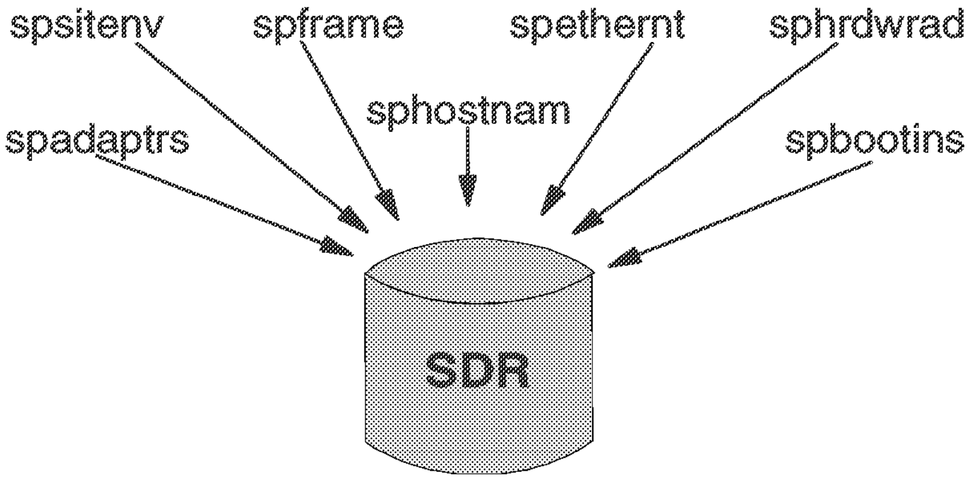
```
sp2cw0.msc.itso.ibm.com        root
sp2en0.msc.itso.ibm.com        root
sp2n01.msc.itso.ibm.com        root
sp2n02.msc.itso.ibm.com        root
sp2n03.msc.itso.ibm.com        root
sp2n04.msc.itso.ibm.com        root
sp2n05.msc.itso.ibm.com        root
sp2n06.msc.itso.ibm.com        root
sp2n07.msc.itso.ibm.com        root
sp2n08.msc.itso.ibm.com        root
sp2n09.msc.itso.ibm.com        root
sp2n10.msc.itso.ibm.com        root
sp2n11.msc.itso.ibm.com        root
sp2n12.msc.itso.ibm.com        root
sp2n13.msc.itso.ibm.com        root
sp2n14.msc.itso.ibm.com        root
sp2n15.msc.itso.ibm.com        root
sp2n16.msc.itso.ibm.com        root
```

Include all network interfaces on the Control Workstation. In this way, you can have the same .rhosts file on the Control Workstation and on the nodes.

Next you should run install_cw on the Control Workstation and configure the SDR.

4.5.1.4 Enter Node Information into SDR

Node Information into SDR



spsitenv

spframe

spethernt

sphrdwrad

sphostnam

spadaptrs

spbootins

SDR

To enter the site environment information, use:

```
# spsitenv cw_lppsource_name=aix414
```

To enter the frame information and to reinitialize the SDR, use:

```
# spframe -r yes 1 1 /dev/tty0
```

The spframe needs 30 seconds to a couple of minutes to complete. Verify the system with the spmon -d command:

```
# spmon -d
1.  Checking server process
    Process 22006 running 0:0
    Check ok

2.  Opening connection to server
    Connection opened
    Check ok

3.  Querying frame(s)
    1 frame(s)
    Check ok

4.  Checking frames

    This step was skipped because the -G flag was omitted.
```

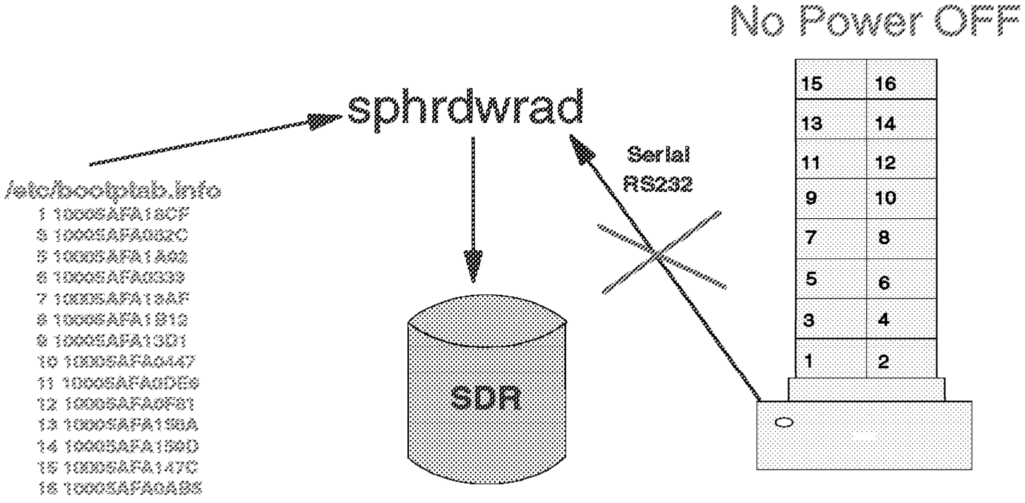
```
5.  Checking nodes

--------------------------------- Frame 1 ------------------------------------
Frame  Node    Node         Host      Switch    Key     Env    Front Panel
Slot   Number  Type  Power  Responds  Responds  Switch  Fail       LEDs
------------------------------------------------------------------------------
  1       1    thin    on      no        no        normal   no     LEDs are blank
  2       2    thin    on      no        no        normal   no     LEDs are blank
  3       3    thin    on      no        no        normal   no     LEDs are blank
  4       4    thin    on      no        no        normal   no     LEDs are blank
  5       5    thin    on      no        no        normal   no     LEDs are blank
  6       6    thin    on      no        no        normal   no     LEDs are blank
  7       7    thin    on      no        no        normal   no     LEDs are blank
  8       8    thin    on      no        no        normal   no     LEDs are blank
  9       9    thin    on      no        no        normal   no     LEDs are blank
 10      10    thin    on      no        no        normal   no     LEDs are blank
 11      11    thin    on      no        no        normal   no     LEDs are blank
 12      12    thin    on      no        no        normal   no     LEDs are blank
 13      13    thin    on      no        no        normal   no     LEDs are blank
 14      14    thin    on      no        no        normal   no     LEDs are blank
 15      15    thin    on      no        no        normal   no     LEDs are blank
 16      16    thin    on      no        no        normal   no     LEDs are blank
```

You can see that the nodes are powered on and the clients use the 3.2.5 system. The host responds and the switch responds show not active, but when you log on to a node you see that the switch is also up and running. The nodes are active because we did not reboot or shutdown any node. The switch will stay active until a switch fault occurs.

To enter the required node information, use the following command:

```
# spethernt -s 'yes' 1 1 16 sp2n01 255.255.255.0 sp2en0
```

Ethernet Hardware Addresses



```
/etc/bootptab.info
1 10005AFA18CF
3 10005AFA082C
5 10005AFA1A92
6 10005AFA0339
7 10005AFA13AF
8 10005AFA1B12
9 10005AFA13D1
10 10005AFA0447
11 10005AFA0DE9
12 10005AFA0F81
13 10005AFA159A
14 10005AFA150D
15 10005AFA147C
16 10005AFA0A95
```

The next step is to acquire the Ethernet hardware addresses. Copy from /tmp the file bootptab.info to the /etc directory. Use the /etc/bootptab.info file, because the sphrdwrad command will power off any node that is not listed in that file.

How can you get the Ethernet hardware address when you do not have it in the /etc/bootptab.info file? There are a lot of different ways to get to the hardware address of an Ethernet card. This example shows how to get the address for the node “sp2n01” with the arp command:

```
# ping -c1 sp2n01
PING sp2n01.msc.itso.ibm.com: (192.168.3.1): 56 data bytes
64 bytes from 192.168.3.1: icmp_seq=0 ttl=255 time=1 ms

----sp2n01.msc.itso.ibm.com PING Statistics----
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 1/1/1 ms
# arp -a | grep sp2n01
  sp2n01 (192.168.3.1) at 10:0:5a:fa:18:cf
```

Now write down the hardware address and make the modification to the /etc/bootptab.info file.

In this example, the address is: “10:0:5a:fa:18:cf,” and this will translate into the following line of /etc/bootptab.info: “1 10005afa18cf”

The command to acquire the Ethernet hardware addresses for the SDR is:

```
# sphrdwrad 1 1 rest
Acquiring hardware ethernet address for node 1 from /etc/bootptab.info
Acquiring hardware ethernet address for node 2 from /etc/bootptab.info
Acquiring hardware ethernet address for node 3 from /etc/bootptab.info
Acquiring hardware ethernet address for node 4 from /etc/bootptab.info
Acquiring hardware ethernet address for node 5 from /etc/bootptab.info
Acquiring hardware ethernet address for node 6 from /etc/bootptab.info
Acquiring hardware ethernet address for node 7 from /etc/bootptab.info
Acquiring hardware ethernet address for node 8 from /etc/bootptab.info
Acquiring hardware ethernet address for node 9 from /etc/bootptab.info
Acquiring hardware ethernet address for node 10 from /etc/bootptab.info
Acquiring hardware ethernet address for node 11 from /etc/bootptab.info
Acquiring hardware ethernet address for node 12 from /etc/bootptab.info
Acquiring hardware ethernet address for node 13 from /etc/bootptab.info
Acquiring hardware ethernet address for node 14 from /etc/bootptab.info
Acquiring hardware ethernet address for node 15 from /etc/bootptab.info
Acquiring hardware Ethernet address for node 16 from /etc/bootptab.info
```

Now add the switch interface to the SDR:

```
# spadaptrs -s ’yes’ -a ’yes’ -n ’no’ 1 1 16 css0 sp2sw01 255.255.255.0
```

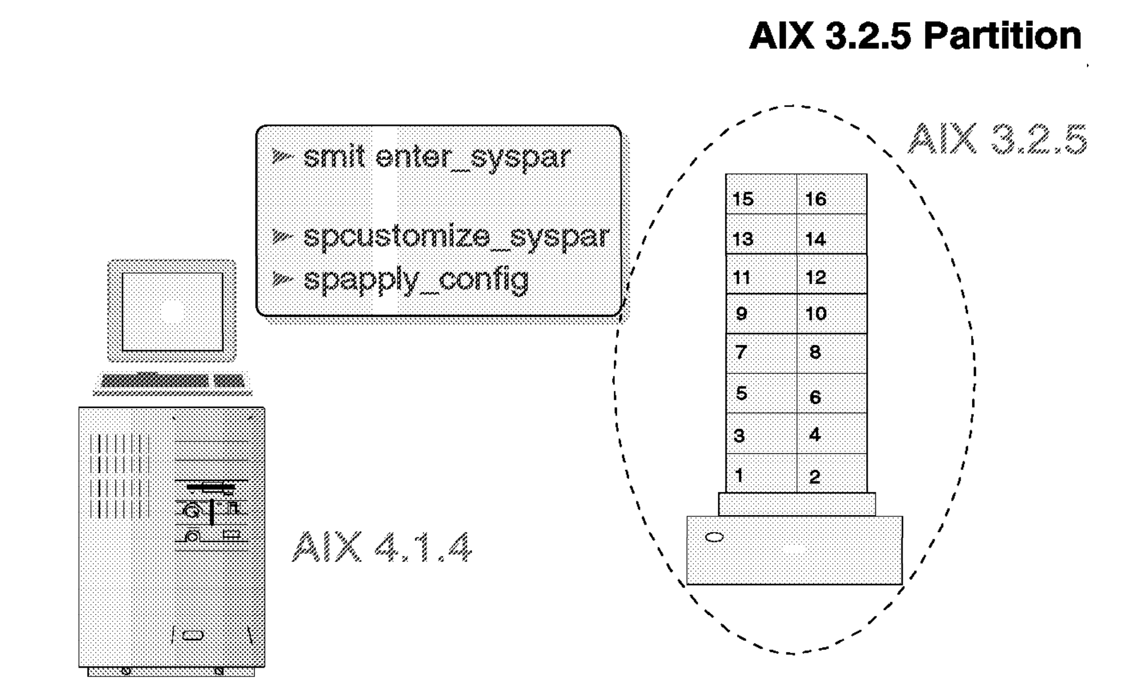
Use the following command to set the short host name:

```
# sphostnam -f ’short’ 1 1 16
```

At this time, you do not want to install any of the nodes. Set the “response from server” flag to “disk.” This means that each node will boot from its local disk, and not start the installation process after the next “Netboot.”

```
# spbootins  -r ’disk’ -s ’no’ 1 1 16
```

4.5.1.5 Create AIX 3.2.5 Partition



AIX 3.2.5 Partition

- ➤ smit enter_syspar
- ➤ spcustomize_syspar
- ➤ spapply_config

The AIX 3.2.5 partition has to be defined. The RS/6000 SP will have only one partition, but PSSP 2.2 creates the default partition as AIX 4.1 system.

First you should save the System Data Repository (SDR), and then define all parameters for PSSP 1.2 and AIX 3.2.5. Use the following commands:

Note: The AIX version will be set to AIX 4.1 because the nodes will be upgraded in the near future to AIX 4.1.4.

```
# SDRArchive PSSP22.def
SDRArchive: SDR archive file name is
   /spdata/sys1/sdr/archives/backup.96222.1153.PSSP22.def
# spbootins -p PSSP-1.2 -s no 1 1 16
spbootins:  0022-176 Advi..
 . . .
# spbootins -v aix414   -s no 1 1 16
# splstdata -p
List System Partition Information

System Partitions:
------------------
sp2cw0

Syspar: sp2cw0
-------------------------------------------------------------------------
```

```
syspar_name     sp2cw0
ip_address      9.12.1.37
install_image   default
syspar_dir      ″″
code_version    PSSP-1.2
haem_cdb_version 839471489,112968960,0
# splst_versions -G -t
1 PSSP-1.2
2 PSSP-1.2
3 PSSP-1.2
4 PSSP-1.2
5 PSSP-1.2
6 PSSP-1.2
7 PSSP-1.2
8 PSSP-1.2
9 PSSP-1.2
10 PSSP-1.2
11 PSSP-1.2
12 PSSP-1.2
13 PSSP-1.2
14 PSSP-1.2
15 PSSP-1.2
16 PSSP-1.2
```

Configure the PSSP Services and set up the daemons.

```
# /usr/lpp/ssp/install/bin/services_config
rc.ntp . . .
# syspar_ctrl -c hr
stopping ″hr.sp2cw0″
0513-044 The stop of the hr.sp2cw0 Subsystem was completed successfully.
removing ″hr.sp2cw0″
0513-083 Subsystem has been Deleted.
# syspar_ctrl -c hb
# syspar_ctrl -A -G
The hats.sp2cw0 subsystem must be stopped before remaking it.
syspar_ctrl:  0022-233 SP_NAME=sp2cw0  /usr/lpp/ssp/bin/hatsctrl -a
  returned with a bad return code, rc = 1.
making SRC object ″hb.sp2cw0″ at system level PSSP-2.2
0513-071 The hb.sp2cw0 Subsystem has been added.
hagsctrl: 2520-208 The hags.sp2cw0 subsystem must be stopped.
syspar_ctrl:  0022-233 SP_NAME=sp2cw0  /usr/lpp/ssp/bin/hagsctrl -a
  returned with a bad return code, rc = 1.
0513-071 The haem.sp2cw0 Subsystem has been added.
wrote 0 objects to class EM_Resource_Variable
 . . .
```

Run setup_server to create the Network Installation Management resources. Use smit syspar_cust or the spcustomize_syspar command to define the partition.

Following is the spcustomize_syspar command:

```
# spcustomize_syspar -n sp2cw0 -l PSSP-1.2 -d bos.obj.ssp.325.mini
  -e 15 config.16/layout.1/syspar.1
Custom file has been created.
```

This command defines one system partition across all 16 nodes, defines node 15 as the primary switch node, and defines that PSSP 1.2 and AIX 3.2.5 are used on the nodes.

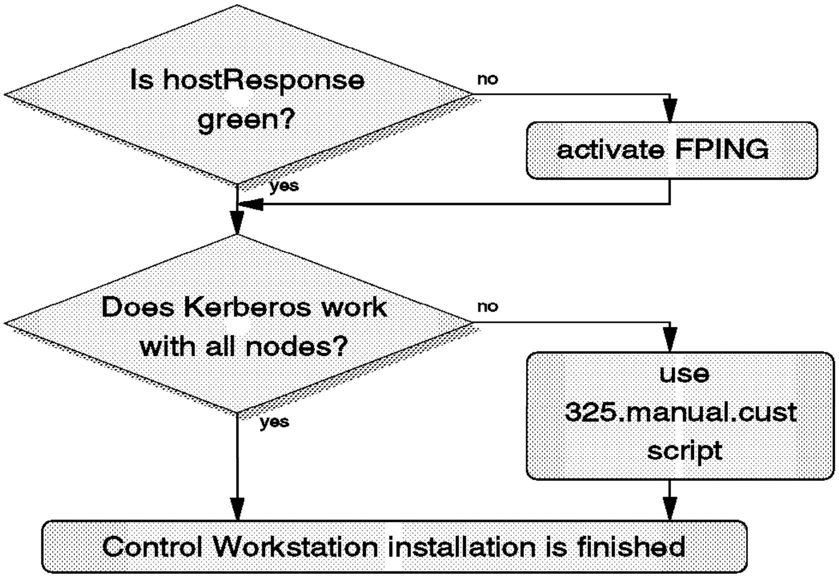
Now it is time to apply the PSSP 1.2 system partition. Use the following command:

```
# spapply_config config.16/layout.1
spapply_config:  Reading and verifying system partition layout information
spapply_config:  Checking System Data Repository...
spapply_config:  Reading current system partition information from SDR...
spapply_config:  Unchanged system partitions:
spapply_config:  Changed system partitions:
  sp2cw0
 . . . .
spapply_config:  Deleting partition sensitive subsystems (hats,hb,hr,etc.)
                 from old (about to be deleted) system partitions...
0513-044 The stop of the sp_configd Subsystem was completed successfully.
0513-004 The Subsystem or Group, pman.sp2cw0, is currently inoperative.
Stopping hr.sp2cw0 has been Deleted.
0513-044 The stop of the hr.sp2cw0 Subsystem was completed successfully.
0513-004 The Subsystem or Group, haem.sp2cw0, is currently inoperative.
0513-044 The stop of the hags.sp2cw0 Subsystem was completed successfully.
0513-044 The stop of the hb.sp2cw0 Subsystem was completed successfully.
0513-044 The stop of the hats.sp2cw0 Subsystem was completed successfully.

spapply_config:  Creating necessary new system partitions and
partition sensitive subsystems (hats,hb,hr,etc.)

0513-071 The hats.sp2cw0 Subsystem has been added.
making SRC object ″hb.sp2cw0″ at system level PSSP-2.2
0513-071 The hb.sp2cw0 Subsystem has been added.
0513-071 The hags.sp2cw0 Subsystem has been added.
0513-071 The hagsglsm.sp2cw0 Subsystem has been added.
0513-071 The haem.sp2cw0 Subsystem has been added.
wrote 0 objects to class EM_Resource_Variable
 . . .
0513-059 The hats.sp2cw0 Subsystem has been started. Subsystem PID is 16141
0513-059 The hb.sp2cw0 Subsystem has been started. Subsystem PID is 24428
0513-059 The hags.sp2cw0 Subsystem has been started. Subsystem PID is 22448
0513-059 The hagsglsm.sp2cw0 Subsystem has been started. Subsystem PID is 16441.
0513-059 The haem.sp2cw0 Subsystem has been started. Subsystem PID is 15367.
0513-059 The hr.sp2cw0 Subsystem has been started. Subsystem PID is 18740.
0513-059 The pman.sp2cw0 Subsystem has been started. Subsystem PID is 17557.
0513-029 The pmanrm.sp2cw0 Subsystem is already active.
0513-059 The sp_configd Subsystem has been started. Subsystem PID is 39595.

spapply_config:  Deleting unused system partitions...

spapply_config:  Preparing switch for changed and new system partitions...

spapply_config:  Verifying contents of SDR...

spapply_config:  Command complete.
```

The command worked fine, but host_responds might not show green on the spmon GUI.

4.5.1.6 Control Workstation Is Installed

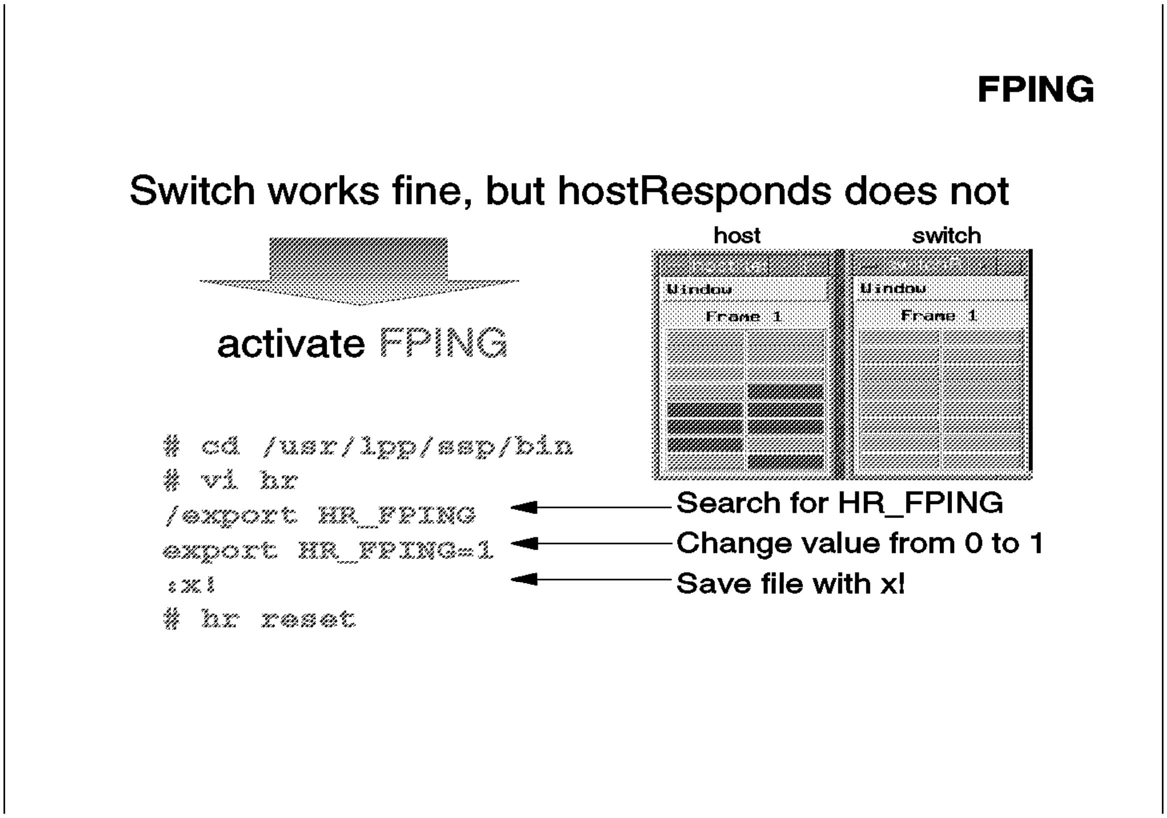
Control Workstation Is Installed


Is hostResponse green?
no
activate FPING
yes
Does Kerberos work with all nodes?
no
use 325.manual.cust script
yes
Control Workstation installation is finished

This flowchart presents the actions′ sequence that need to be followed in case something goes wrong with the installation on the Control Workstation.

The following section presents a discussion on how to use FPING and how to repair Kerberos.

4.5.1.7 How to Use FPING and Repair Kerberos



host
switch
Window
Frame 1
Window
Frame 1

FPING

Switch works fine, but hostResponds does not

activate FPING

```
# cd /usr/lpp/ssp/bin
# vi hr
/export HR_FPING
export HR_FPING=1
:x!
# hr reset
```

Search for HR_FPING

Change value from 0 to 1

Save file with x!

The Control Workstation and all RS/6000 SP nodes will be migrated to AIX 4 and PSSP 2.2. The reason why host_responds does not show green is unknown at this point.

Note: VSD does not support FPING.

Activate the FPING protocol for host_responds and use the following commands:

```
# cd /usr/lpp/ssp/bin
# vi hr
/export HR_FPING
```

Now change the value of HR_FPING from 0 to 1. Save the file and use the hr reset command to restart the hr deamon. The host_responds field now shows solid green, and now the switch has to be activated. Use the Estart command to activate the switch.

At this point, the switch should be active, and the switch response should show green in the spmon GUI. If this is not the case, it is possible that you may have a Kerberos problem, and the switch cannot be activated.

First use the Eprimary command to verify that the primary switch node uses the correct node number. The switch can be restarted with Estart when a switch fault occurs. The steps to disable part of Kerberos have to be done on every node. The parallel commands, for example, dsh or pcp, do not work anymore.

Log in to every node and transfer the ~root/.rhosts file from the Control Workstation. This example shows you the procedure for node 1:

```
# rlogin sp2n01
root′s Password:
*******************************************************************
*                                                                 *
*  Welcome to AIX Version 3.2.5                                   *
*                                                                 *
*******************************************************************
Last login: Fri Aug  2 11:06:27 1996 on /dev/pts/0 from sp2en0

# rcp sp2en0:/.rhosts /.rhosts
# exit
```

Now every node can be accessed with /bin/rsh and /bin/rcp. Create the 325.manual.cust script file to transfer the Kerberos files to the node and to execute /etc/rc.sp manually on the nodes.

Note: This script is only needed when your AIX 3.2.5 boot install servers do not work anymore.

Here are the contents of the 325.manual.cust file:

```
#!/bin/ksh
#
# Needed Environment Variables
#
# primary IP address of the control workstation
export CW_hostaddr=9.12.1.37
# Ethernet IP address of the control workstation
export NIM_MASTER=192.168.3.37
# long host name of the Ethernet interface
export NIM_MASTER_HN=sp2en0.msc.itso.ibm.com
# short host name of the Ethernet interface
export NIM_MASTER_SHN=sp2en0
# use all NIM_M* variables and create /tmp/server_name file
echo ″$NIM_MASTER $NIM_MASTER_HN $NIM_MASTER_SHN″ > /tmp/server_name
#
# all nodes will be set to customize
spbootins -r customize 1 1 16
#
# execute commands for all nodes
for i in 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16
do
   # copy all three kerberos files to the node
   /bin/rcp /tftpboot/sp2n$i-new-srvtab sp2n$i:/etc/krb-srvtab
   /bin/rcp /etc/krb.conf               sp2n$i:/etc
   /bin/rcp /etc/krb.realms             sp2n$i:/etc
   # copy the SDR_dest_info file to the node
   /bin/rcp /etc/SDR_dest_info          sp2n$i:/etc
   # copy the server_name to the nodes
   /bin/rcp /tmp/server_name            sp2n$i:/etc/ssp
   # create the primary IP address of the Control Workstation
   /bin/rsh sp2n$i ″echo $CW_hostaddr >/etc/ssp/cw_name″
   # obtain a ticket-granting-ticket
   /bin/rsh sp2n$i /usr/lpp/ssp/rcmd/bin/rcmdtgt
   # start customize on the node
   /bin/rsh sp2n$i ″nohup /etc/rc.sp &″
done
```

First define the environment variable for the IP address of the Control Workstation. If your Control Workstation has more than one network interface, then use the primary address that corresponds to the hostname command. The spbootins command changes the SDR and starts setup_server. The setup_server command creates the files in /tftpboot. The files will be transferred with the non-Kerberos version of rcp to the nodes. When all files are at the node, the /etc/rc.sp script will be executed for a manual customization.

The Control Workstation migration is finished. Use the Estart command to restart the switch:

```
# Estart
Switch initialization started on sp2n15.
Initialized 16 nodes(s).
Switch initialization completed.
```

In the previous steps you could see that the Control Workstation can be upgraded without affecting the RS/6000 SP. Since the RS/6000 SP and the Control Workstation are fully operational, enable the strict Kerberos security. Use:

```
# dsh -a -G ″rm /.rhosts″
```

With this command, the non-Kerberos versions of rsh, rcp, and rlogin do not work anymore for the root user. You have to supply a password before you can log in remotely.

4.5.2 Migration from PSSP 1.2 to PSPP 2.1

This section describes a different way to upgrade the Control Workstation from PSSP 1.2 to PSSP 2.1. The default way to migrate from PSSP 1.2 to the latest level of PSSP is described in the *Installation and Migration Guide*, GC23-3898.

4.5.2.1 AIX 4 No Prompt Install

Install AIX 4.1.4 on a different disk from AIX 3.2.5. This enables you to boot later from the AIX 3.2.5 system disk. The fastest way to install AIX 4.1.4 is to create a diskette with the “no prompt” options. Create the diskette as described in chapter 4.5.1.1, “Fast AIX 4.1 Install” on page 143.

Then insert the AIX 4.1.4 Server CD or the AIX 4.1.4 Server tape. Turn the key to the service position and boot the Control Workstation in maintenance mode. AIX starts the installation and reads the additional information from the diskette. After one hour, your Control Workstation should be ready for the next steps. The installation time is dependent on the speed of your machine.

4.5.2.2 Prepare the Control Workstation

Since we have not preserved the System Data Repository (SDR), we have to set up the network, set up Kerberos, configure the Control Workstation, and enter the node information into the SDR.

For the network setup, see the *Installation and Migration Guide*, GC23-3898, Chapter 4, “Step 10: Configure Ethernet Adapters.”

Add the following lines to the end of /etc/rc.net:

```
case `hostname` in
sp2cw*)
# tunable parameter for the CWS
        /usr/sbin/no -o thewall=16384
        /usr/sbin/no -o sb_max=163840
        /usr/sbin/no -o tcp_sendspace=65536
        /usr/sbin/no -o tcp_recvspace=65536
        /usr/sbin/no -o tcp_mssdflt=1448
        /usr/sbin/no -o udp_sendspace=32768
        /usr/sbin/no -o udp_recvspace=65536
        /usr/sbin/no -o ipforwarding=1
        /usr/sbin/no -o rfc1323=1
        /usr/sbin/no -o subnetsarelocal=1
;;
esac
```

The space for the Network Installation Management (NIM) Boot Images was created from the setup script at installation time.

Since this customer environment has grown over the years, we have nine hard disks. We will use three disks for /spdata and three disks for the rootvg. The layout is listed in Table 4 on page 162.

Table 4. Hard Disk Layout

hdisk	Volume Group	Size	Description
hdisk0	rootvg	2.0 Gb	rootvg for AIX 4.1.4
hdisk1	not active	2.0 Gb	rootvg for AIX 4.2
hdisk2	db2vg	2.0 Gb	DB2 volume group
hdisk3	spaix4vg	2.0 Gb	VG for AIX 4.1.4 lppsource, PSSP 2.1 and PSSP 2.2
hdisk4	spaix32vg	1.0 Gb	PSSP 1.2 volume group
hdisk5	not active	1.0 Gb	Empty
hdisk6	not active	2.0 Gb	rootvg for AIX 3.2.5
hdisk7	lotusvg	2.0 Gb	Volume group for Lotus software and paging space
hdisk8	spdatavg	2.0 Gb	spdata volume group
hdisk9	not active	2.0 Gb	Empty

Import all SP volume groups, create or change filesystems, and load AIX 4.1.4 and PSSP 2.1 using the new PSSP 2.2 directory layout. For detailed information, refer to the *Installation and Migration Guide*, GC23-3898, Chapter 4, step 13 to step 18. Here are the filesystems that you will use:

Table 5. CWS File Systems

Directory	Size	Volume Group
/tftpboot	25 Mb	rootvg
/spdata	60 Mb	spdatavg
/spdata/sys1/install/aix414	420 Mb	spaix4vg
/spdata/sys1/install/images	480 Mb	spdatavg
/spdata/sys1/install/pssplpp	540 Mb	spaix4vg
/spdata/sys1/install/pssplpp/PSSP-1.2	980 Mb	spaix32vg

The /spdata/sys1/install/pssplpp/PSSP-1.2 file system was the /usr/sys/inst.images/ssp file system on AIX 3.2.5.

Restore the files that have been saved to the /tmp directory. The files are:

- /etc/bootptab.info
- /etc/passwd
- /etc/security/passwd
- /etc/group
- /etc/security/group
- /etc/amd/amd-maps/amd.u

For the next steps, run setup_authent. We are setting up Kerberos, because the TCP/IP addresses on the Control Workstation and nodes have changed. Enable the ~root/.rhosts file, so that normal rcp, rsh, and rlogin will work. Use this setup only for the migration process, and remove all .rhosts files when you have finished. The ~root/.rhosts file has the following entries:

```
sp2cw0.msc.itso.ibm.com         root
sp2en0.msc.itso.ibm.com         root
sp2n01.msc.itso.ibm.com         root
sp2n02.msc.itso.ibm.com         root
sp2n03.msc.itso.ibm.com         root
sp2n04.msc.itso.ibm.com         root
sp2n05.msc.itso.ibm.com         root
sp2n06.msc.itso.ibm.com         root
sp2n07.msc.itso.ibm.com         root
sp2n08.msc.itso.ibm.com         root
sp2n09.msc.itso.ibm.com         root
sp2n10.msc.itso.ibm.com         root
sp2n11.msc.itso.ibm.com         root
sp2n12.msc.itso.ibm.com         root
sp2n13.msc.itso.ibm.com         root
sp2n14.msc.itso.ibm.com         root
sp2n15.msc.itso.ibm.com         root
sp2n16.msc.itso.ibm.com         root
```

Include all network interfaces on the Control Workstation. In this way, you can have the same .rhosts file on the Control Workstation and on the nodes.

The next step is to run install_cw on the Control Workstation and to configure the SDR.

4.5.2.3 Enter Node Information into SDR

To enter the side environment information, use:

```
# spsitenv install_image=′bos.obj.ssp.325.mini′
```

To enter the frame information and to reinitialize the SDR, use:

```
# spframe -r yes 1 1 /dev/tty0
```

The spframe needs 30 seconds to a couple of minutes to complete. Verify the system with the spmon -d command:

```
# spmon -d
1.  Checking server process
    Process 22006 running 0:0
    Check ok

2.  Opening connection to server
    Connection opened
    Check ok

3.  Querying frame(s)
    1 frame(s)
    Check ok

4.  Checking frames

    This step was skipped because the -G flag was omitted.

5.  Checking nodes

-------------------------------- Frame 1 -----------------------------------
Frame  Node    Node         Host      Switch    Key     Env    Front Panel
Slot   Number  Type  Power  Responds  Responds  Switch  Fail       LEDs
----------------------------------------------------------------------------
  1       1    thin    on     no         no     normal   no    LEDs are blank
  2       2    thin    on     no         no      normal   no    LEDs are blank
  3       3    thin    on     no         no      normal   no    LEDs are blank
  4       4    thin    on     no         no      normal   no    LEDs are blank
  5       5    thin    on     no         no      normal   no    LEDs are blank
  6       6    thin    on     no         no      normal   no    LEDs are blank
  7       7    thin    on     no         no      normal   no    LEDs are blank
  8       8    thin    on     no         no      normal   no    LEDs are blank
  9       9    thin    on     no         no      normal   no    LEDs are blank
 10      10    thin    on     no         no      normal   no    LEDs are blank
 11      11    thin    on     no         no      normal   no    LEDs are blank
 12      12    thin    on     no         no      normal   no    LEDs are blank
 13      13    thin    on     no         no      normal   no    LEDs are blank
 14      14    thin    on     no         no      normal   no    LEDs are blank
 15      15    thin    on     no         no      normal   no    LEDs are blank
 16      16    thin    on     no         no      normal   no    LEDs are blank
```

You can see that the nodes are powered on and the clients use the 3.2.5 system. The host responds and the switch responds show not active, but when you log on to a node you see that the switch is up and running. The nodes are active because you did not reboot or shutdown any node. The switch will stay active until a switch fault occurs.

To enter the required node information, use the following command:

```
# spethernt -s ′yes′ 1 1 16 sp2n01 255.255.255.0 sp2en0
```

The next step is to acquire the Ethernet hardware addresses. Copy the file bootptab.info from /tmp to the /etc directory. Use the /etc/bootptab.info file because the sphrdwrad command will power off any node that is not listed in that file. The command to acquire the Ethernet hardware addresses is:

```
# sphrdwrad 1 1 rest
Acquiring hardware ethernet address for node 1 from /etc/bootptab.info
Acquiring hardware ethernet address for node 2 from /etc/bootptab.info
 . . .
Acquiring hardware ethernet address for node 15 from /etc/bootptab.info
Acquiring hardware ethernet address for node 16 from /etc/bootptab.info
```

Now add the switch interface to the SDR:

```
# spadaptrs -s ′yes′ -a ′yes′ -n ′no′ 1 1 16 css0 sp2sw01 255.255.255.0
```

Use the following command to set a short host name:

```
# sphostnam -f ′short′ 1 1 16
```

At this time, you do not want to install any of the nodes. Set the "response from server" flag to "disk." This means that each node will boot from its local disk.

```
# spbootins  -r ′disk′ -s ′no′ 1 1 16
```

4.5.2.4 Create AIX 3.2.5 Partition

The AIX 3.2.5 partition has to be created. The RS/6000 SP will have only one partition, but PSSP 2.1 creates the default partition as an AIX 4.1 system. Use smit syspar_cust or the spcustomize_syspar command to define the partition. Here is the spcustomize_syspar command:

```
# spcustomize_syspar -n sp2cw0 -l PSSP-1.2 -d bos.obj.ssp.325.mini
  -e 15 config.16/layout.1/syspar.1
```

This command defines one system partition across all 16 nodes, defines node 15 as the primary switch node, and defines that PSSP 1.2 and AIX 3.2.5 will be used on the nodes.

Now apply the PSSP 1.2 system partition. Use the following command:

```
# spapply_config config.16/layout.1
spapply_config:  Reading and verifying system partition layout information
spapply_config:  Checking System Data Repository...
spapply_config:  Reading current system partition information from SDR...
spapply_config:  Unchanged system partitions:
spapply_config:  Changed system partitions:
  sp2cw0
 . . .
spapply_config:  Deleting hb and hr subsystems for changed system partition.

0513-044 The stop of the hb.sp2cw0 Subsystem was completed successfully.
removing SRC object ″hb.sp2cw0″
0513-083 Subsystem has been Deleted.
0513-044 The stop of the hr.sp2cw0 Subsystem was completed successfully.
removing SRC object ″hr.sp2cw0″
0513-083 Subsystem has been Deleted.

spapply_config:  Creating necessary new system partitions and new hb and hr
systems...
making SRC object ″hb.sp2cw0″ at system level PSSP-1.2
0513-071 The hb.sp2cw0 Subsystem has been added.
0513-059 The hb.sp2cw0 Subsystem has been started. Subsystem PID is 22226
making SRC object ″hr.sp2cw0″
0513-071 The hr.sp2cw0 Subsystem has been added.
0513-059 The hr.sp2cw0 Subsystem has been started. Subsystem PID is 46098

spapply_config:  Updating VSD (partitionVSDdata)...

3 commands executed, 0 failed.

spapply_config:  Preparing switch for changed and new system partitions...
kshd: 0041-005 Kerberos rsh or rcp failed: Too many links
/usr/lpp/ssp/rcmd/bin/rsh: 0041-004 Kerberos rcmd failed: rcmd protocol failed

trying normal rsh (/usr/bin/rsh)
Permission denied.
Eprimary:  0028-039 Warning: Fault service worm not up on oncoming primary
 Eprimary: Eprimary will continue. sp2n15.msc.itso.ibm.com

spapply_config:  Verifying contents of SDR...

spapply_config:  Command complete.
```

In this example, the command worked fine, but there is a Kerberos problem. The Kerberos problem can be fixed, but in the meantime Kerberos will be partly disabled, so the switch can be started from the Control Workstation. The reason to disable part of Kerberos is that the nodes and the switch must be available around the clock. For the details on repairing Kerberos, read 4.5.1.7, “How to Use FPING and Repair Kerberos” on page 158.

The Control Workstation migration is finished. Use the Estart command to restart the switch:

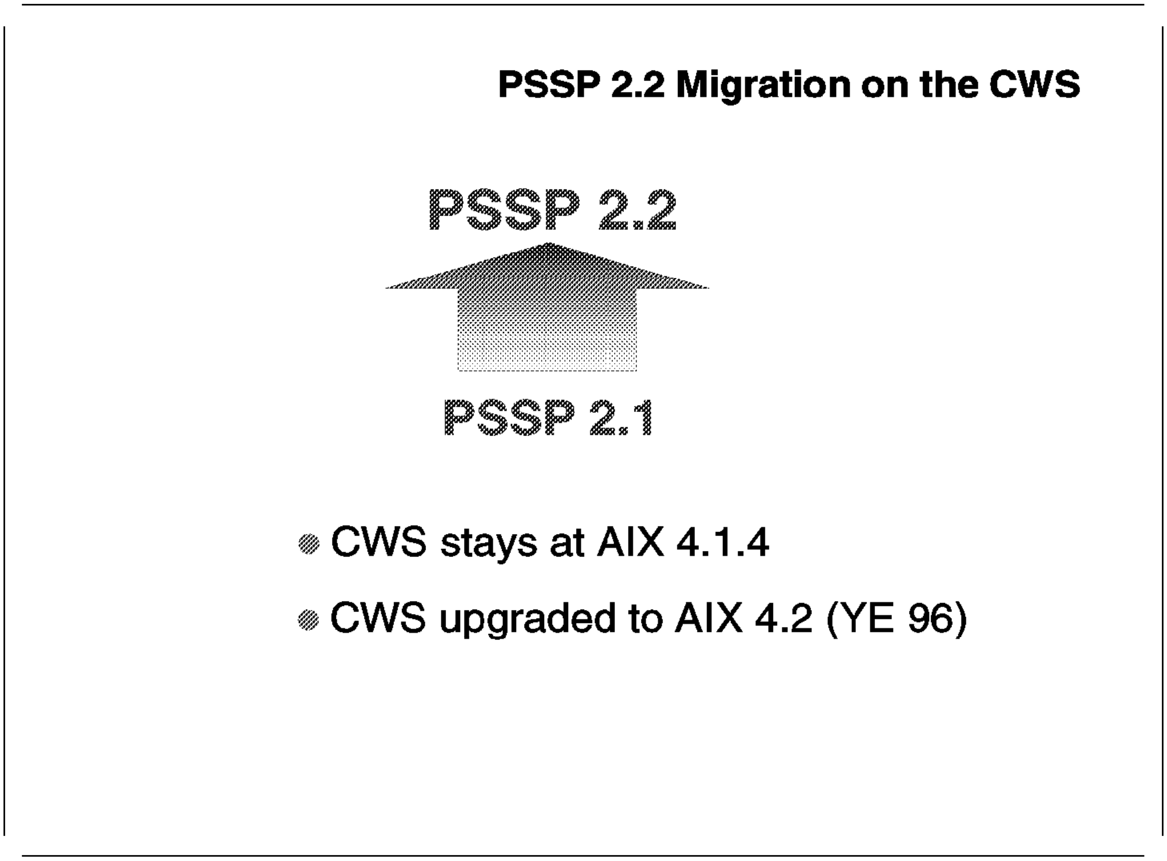
```
Estart
Switch initialization started on sp2n15.
Initialized 16 nodes(s).
Switch initialization completed.
```

In the previous steps, you see that the Control Workstation can be upgraded without affecting the RS/6000 SP. Since the RS/6000 SP and the Control Workstation are fully operational, enable the strict Kerberos security. Use:

```
# dsh -a ″rm /.rhosts″
```

With this command, the normal rsh, rcp, and rlogin do not work anymore for the root user. You have to supply a password before you can login remotely.

4.6 Migrating the CWS from PSSP 2.1 to PSSP 2.2



PSSP 2.2 Migration on the CWS

PSSP 2.2

PSSP 2.1

- CWS stays at AIX 4.1.4
- CWS upgraded to AIX 4.2 (YE 96)

This section describes the migration process for the CWS when upgrading from PSSP 2.1 to PSSP 2.2.

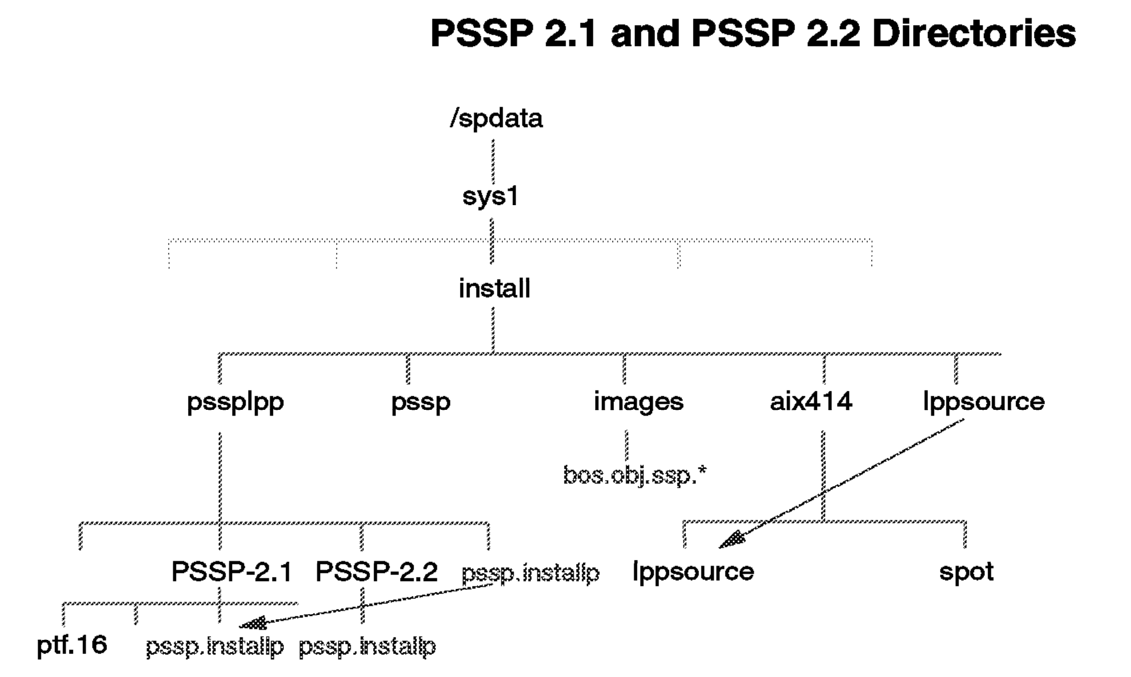
The Control Workstation (CWS) will stay at AIX level 4.1.4. AIX 4.2 will be supported at the end of 1996. For detailed information, refer to 5.4, “Nodes with AIX 4.2” on page 195, or see the *AIX Version 4.2 Installation Guide*, SC23-1924, for the AIX 4.2 migration process.

4.6.1 AIX Is at Level AIX 4.1.4

This section describes a different way to upgrade the Control Workstation from PSSP 2.1 to PSSP 2.2. The default upgrade path to the latest level of PSSP is described in the *Installation and Migration Guide*, GC23-3898.

The most important thing for this scenario is the fact that the nodes must be available 24 hours 7 days a week, and the switch has to work. The availability is very important because the nodes are in production and will be migrated later. These steps show how the Control Workstation can be migrated without rebooting the nodes.

4.6.1.1 PSSP 2.2 and PSSP 2.1 Options



PSSP 2.1 and PSSP 2.2 Directories

For this example, AIX PSSP 2.1 is working well with the RS/6000 SP nodes and AIX will stay at level AIX 4.1.4.

See if you have ample disk space for the new PSSP 2.2 options. The following table shows the directories and disk space requirement for PSSP 2.1, PSSP 2.2, AIX 4.1.4 and AIX 4.2 installed:

Table 6. PSSP 2.x and AIX 4.x Directories

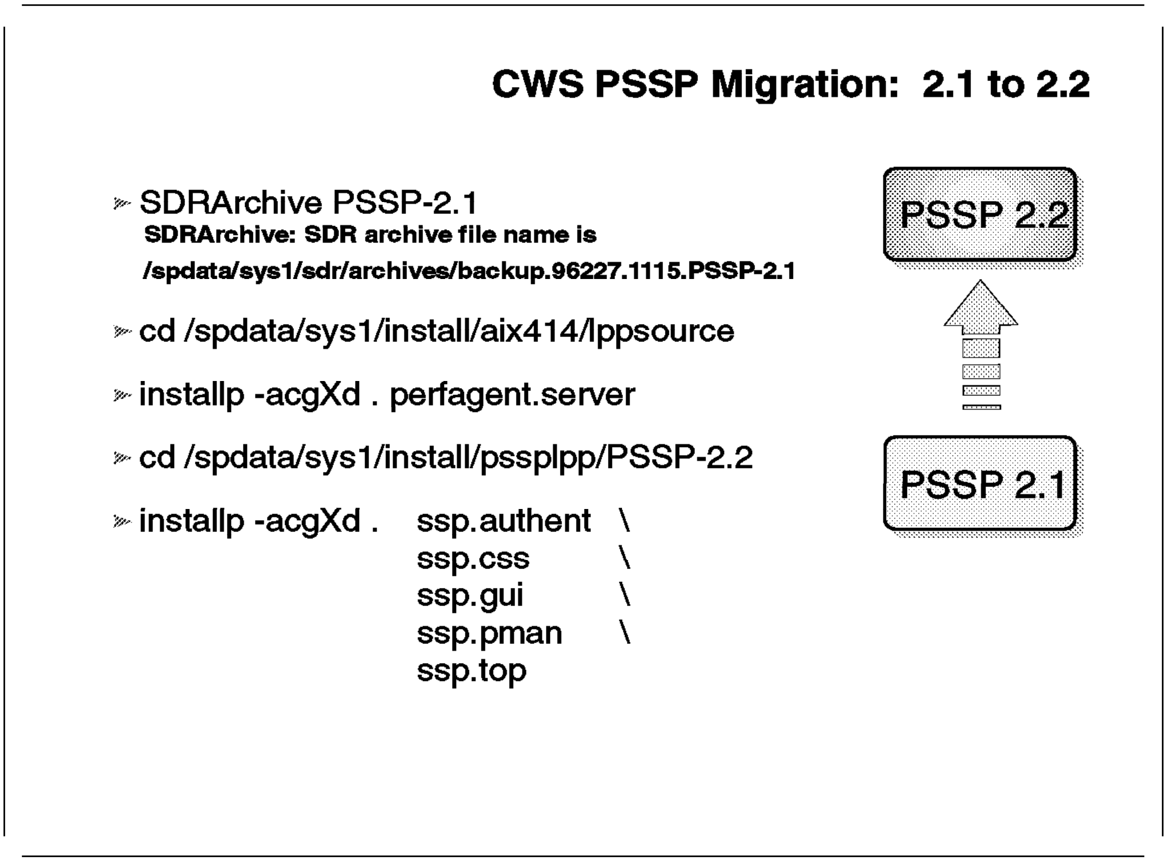
Directory	Size
/spdata/sys1/install/pssplpp/PSSP-2.1	310 Mb total
/spdata/sys1/install/pssplpp/PSSP-2.1/ptf.11	53 Mb
/spdata/sys1/install/pssplpp/PSSP-2.1/ptf.12	59 Mb
/spdata/sys1/install/pssplpp/PSSP-2.1/ptf.13	12 Mb
/spdata/sys1/install/pssplpp/PSSP-2.1/ptf.14	8 Mb
/spdata/sys1/install/pssplpp/PSSP-2.1/ptf.15	52 Mb
/spdata/sys1/install/pssplpp/PSSP-2.1/ptf.16	38 Mb
/spdata/sys1/install/pssplpp/PSSP-2.2	275 Mb total
/spdata/sys1/install/pssplpp/PSSP-2.2/ptf.1	85 Mb
/spdata/sys1/install/aix414	420 Mb minimum
/spdata/sys1/install/aix420	600 Mb minimum

Each ptf.* directory in /spdata/sys1/install/pssplpp/PSSP-2.1 contains PTFs. For example, directory ptf.14 contains all PSSP 2.1 PTFs that belong to PTF tape number 14.

Move the AIX and PSSP 2.1 images to the new directories. This step reflects “Step 16: Move the Existing LPP Images” and “Step 18: Move the PSSP Images for PSSP V2R1” in the *Installation and Migration Guide*, GC23-3898. Make sure that the directory structure previously listed exists. Move the old directories to the new PSSP 2.2 standard place, using the following commands:

```
# cd /spdata/sys1/install
# ls aix414
# mv lppsource aix414/
# ln -s /spdata/sys1/install/aix414/lppsource .
# cd pssplpp
# pwd
/spdata/sys1/install/pssplpp/PSSP-2.1
# mv * PSSP-2.1/
mv: 0653-401 Cannot rename PSSP-2.1 to PSSP-2.1/PSSP-2.1:
             A system call received a parameter that is not valid.
# ln -s /spdata/sys1/install/pssplpp/PSSP-2.1 pssp.installp .
```

4.6.1.2 CWS PSSP Migration: 2.1 to 2.2



CWS PSSP Migration: 2.1 to 2.2

- SDRArchive PSSP-2.1
  SDRArchive: SDR archive file name is
  /spdata/sys1/sdr/archives/backup.96227.1115.PSSP-2.1
- cd /spdata/sys1/install/aix414/lppsource
- installp -acgXd . perfagent.server
- cd /spdata/sys1/install/pssplpp/PSSP-2.2
- installp -acgXd .

```
installp -acgXd .   ssp.authent  \
                    ssp.css      \
                    ssp.gui      \
                    ssp.pman     \
                    ssp.top
```

PSSP 2.2
PSSP 2.1

Stop all RS/6000 SP-related daemons on the Control Workstation. Use the following commands:

```
# stopsrc -g sdr
0513-044 The stop of the sdr.sp2cw0 Subsystem was completed successfully.
# stopsrc -s sysctld
0513-044 The stop of the sysctld Subsystem was completed successfully.
# stopsrc -s hardmon
0513-044 The stop of the hardmon Subsystem was completed successfully.
# stopsrc -s splogd
0513-044 The stop of the splogd Subsystem was completed successfully.
# stopsrc -g hr
0513-044 The stop of the hr.sp2cw0 Subsystem was completed successfully.
# stopsrc -g hb
0513-044 The stop of the hb.sp2cw0 Subsystem was completed successfully.
```

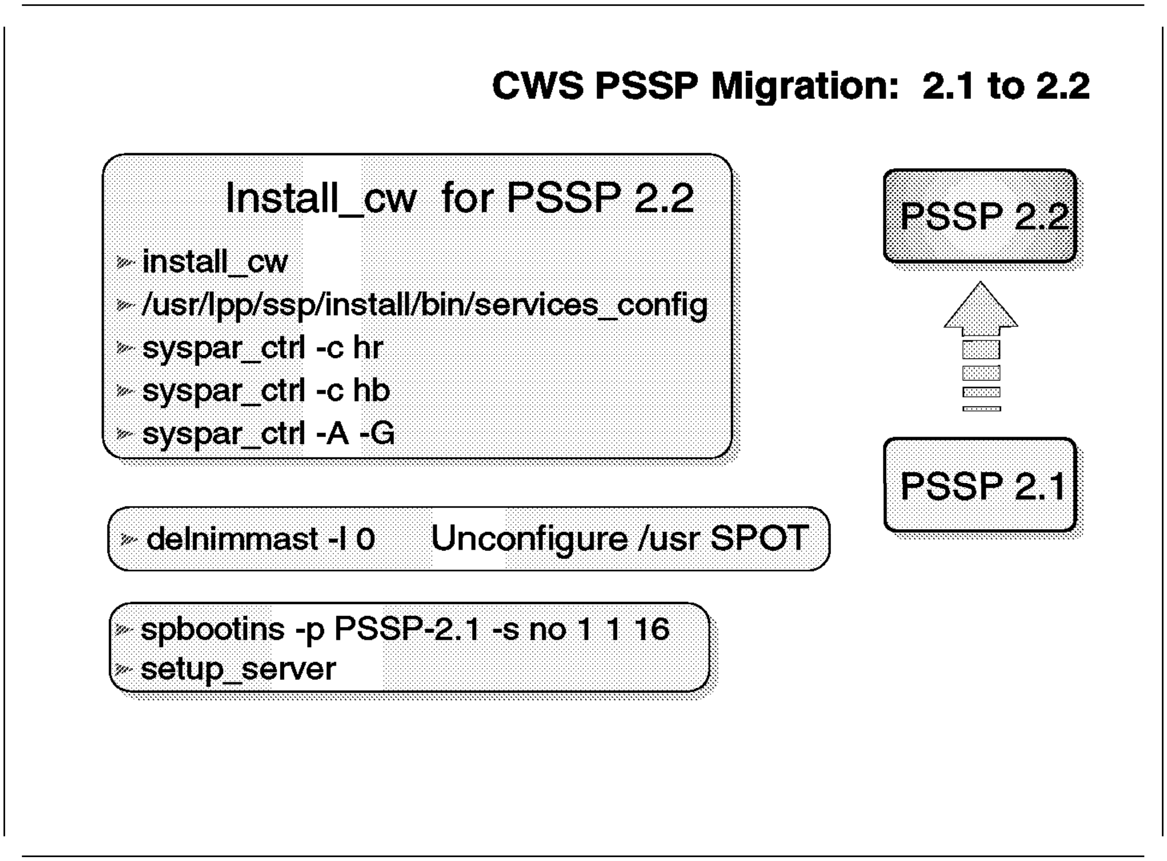
Then check with the `lssrc -a` command to see if all daemons are inoperative. Now use the `installp` command or use `smit install_latest` to install PSSP 2.2. The example shows the `installp` command:

```
# cd /spdata/sys1/install/aix414/lppsource
# installp -acgXd. perfagent.server
# cd /spdata/sys1/install/pssplpp/PSSP-2.2
# mv ssp.usr.2.2.0.0 pssp.installp
# inutoc .
# installp -acgXd . ssp.basic      \
                    ssp.clients    \
                    ssp.authent     \
                    ssp.css         \
                    ssp.gui         \
                    ssp.ha          \
                    ssp.perlpkg     \
                    ssp.pman        \
                    ssp.sysctl      \
                    ssp.sysman      \
                    ssp.top
Warning: the /usr filesystem on this machine has been converted into
a Network Install Manager (NIM) SPOT.  If there are any diskless or
dataless NIM clients using this SPOT, you should perform the following
NIM operations when this invocation of installp finishes:
        nim -o sync_roots psspspot
        nim -Fo check psspspot
 . . .
sysck: 3001-045 WARNING:  A file which is being installed already has an
   entry in the inventory database but is not owned by any installed
   fileset.  The file is:
   /usr/lpp/ssp/perl/lib/syslog.pl
 . . .
```

Ignore the warning messages and use the command lslpp -l ″ssp.*″ to verify that all options are installed. Check if all PSSP option are in the COMMITTED state. If not all options are in the COMMITTED state, issue the installp -u command. The uninstall option will be fully supported with the fix for APAR IX55009.

```
# lslpp -l ssp.*
  Fileset              Level  State      Description
  ------------------------------------------------------------------
Path: /usr/lib/objrepos
  ssp.authent         2.2.0.0  OBSOLETE   SP Authentication Server
  ssp.basic           2.2.0.0  COMMITTED  SP System Support Package
 . . .
# installp -u ″ssp.authent″
 . . .
# cd /spdata/sys1/install/pssplpp/PSSP-2.2
# installp -acgXd . ssp.authent
 . . .
```

Now all PSSP options are installed correctly.



CWS PSSP Migration: 2.1 to 2.2

Install_cw for PSSP 2.2

- install_cw
- /usr/lpp/ssp/install/bin/services_config
- syspar_ctrl -c hr
- syspar_ctrl -c hb
- syspar_ctrl -A -G

- delnimmast -l 0 Unconfigure /usr SPOT

- spbootins -p PSSP-2.1 -s no 1 1 16
- setup_server

The SDR information and the new daemons have to be activated. Use the following commands:

```
# install_cw
# /usr/lpp/ssp/install/bin/services_config
# syspar_ctrl -c hr
# syspar_ctrl -c hb
# syspar_ctrl -A -G
```

Delete the Network Installation Management (NIM) definitions for the PSSP 2.1 nodes:

```
# delnimast -l 0
```

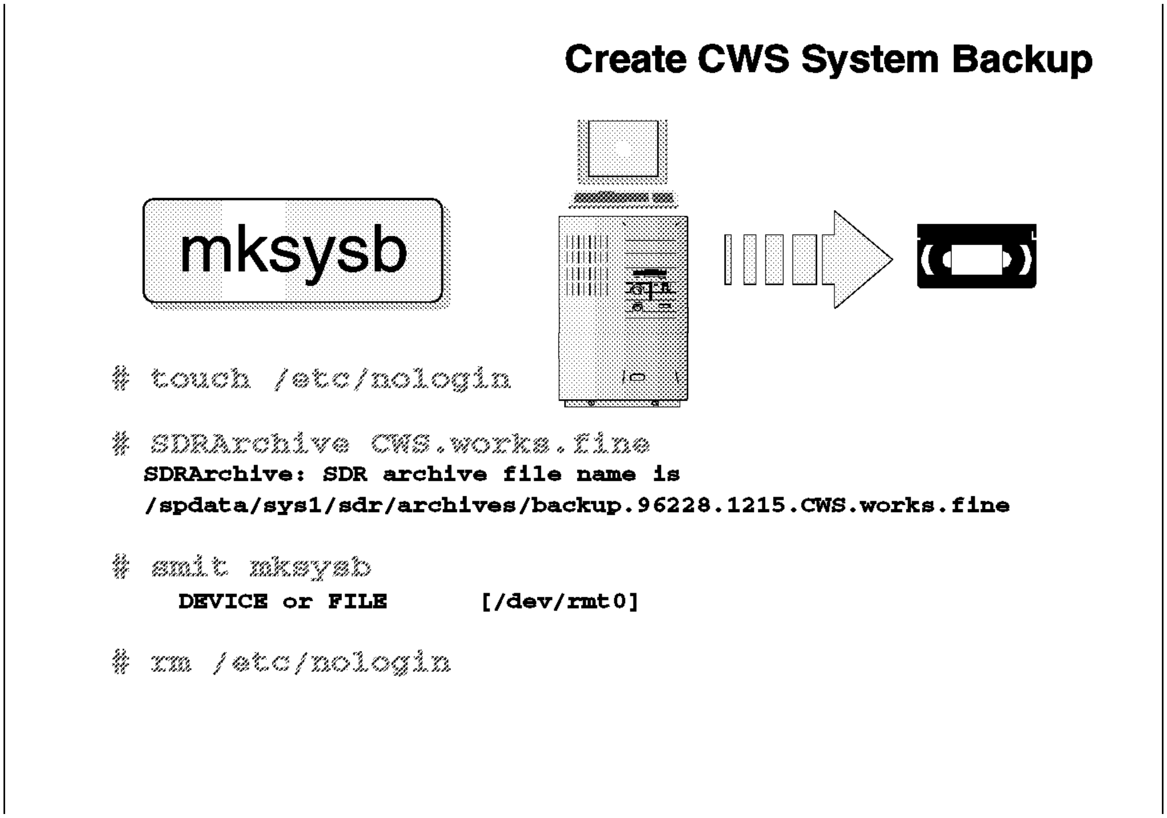
Change the PSSP level for all nodes to 2.1:

```
# spbootins -p PSSP-2.1 -s no 1 1 16
```

Redefine the NIM definitions for the nodes:

```
# setup_server
```

4.6.2 Create CWS System Backup



Create CWS System Backup

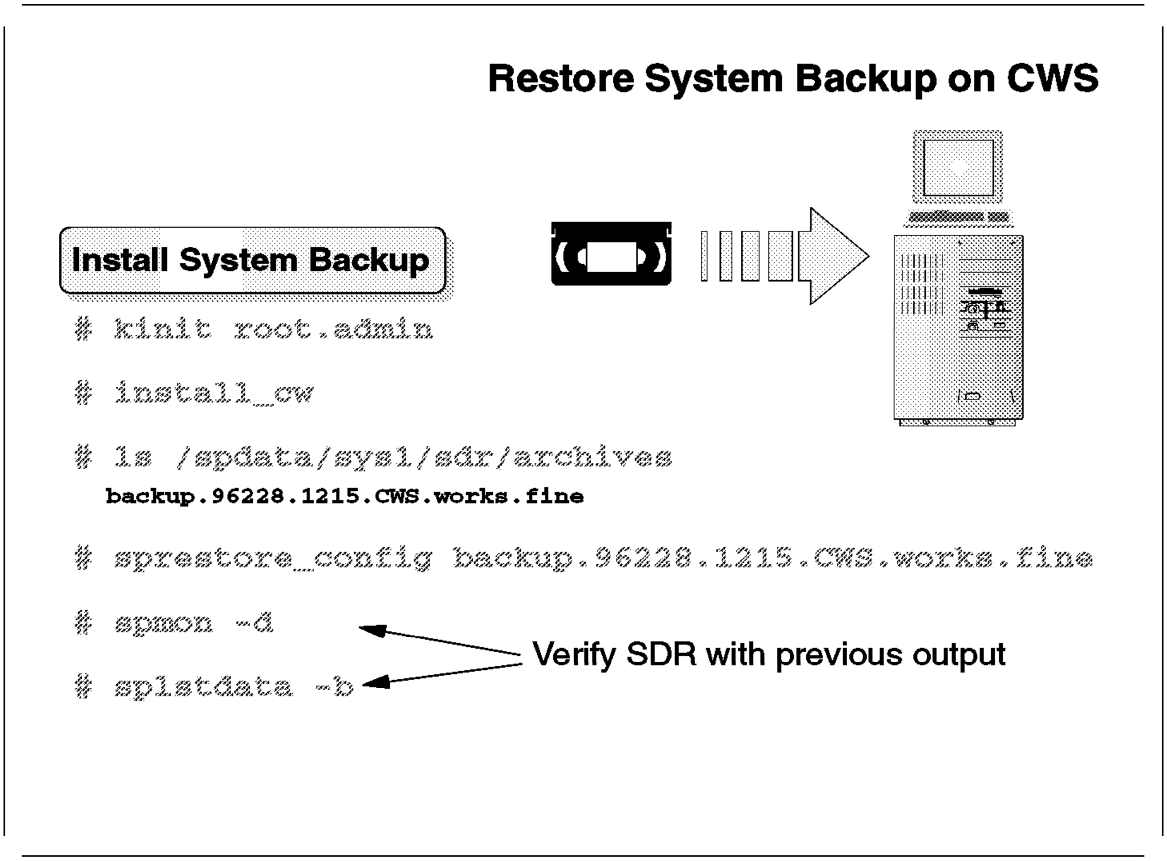
```
# touch /etc/nologin

# SDRArchive CWS.works.fine
  SDRArchive: SDR archive file name is
  /spdata/sys1/sdr/archives/backup.96228.1215.CWS.works.fine

# smit mksysb
    DEVICE or FILE       [/dev/rmt0]

# rm /etc/nologin
```

This section describes the process of the Control Workstation system backup. It tells you which steps are needed when you create the backup.

Perform the following steps:

1. Use the `touch /etc/nologin` command to prevent users from logging into your system. Check to make sure the following conditions are true:
   - No jobs are running.
   - The Resource Manager is stopped.
   - All users are logged off.
   - All user programs are stopped.
2. Create an archive of the SDR with the `SDRArchive <archive_name>` command, where <archive_name> is the name you want to use for this archive. The archive will be stored in the directory /spdata/sys1/sdr/archives, and it will have a name like backup.<date>.<time>.<archive_name>. If the /spdata/sys1/sdr directory is not on your rootvg volume group, copy this file to the /var/adm/SPlogs/sdr directory. In this way, you save this file with your system backup of the rootvg volume group.
3. Perform the system backup with the `smit mksysb` command. If you have the "AIX System Backup & Recovery/6000" (sysback) product, use `smit sysback` to save all your volume groups.
4. When the system backup is finished, remove the /etc/nologin file.

4.6.3 Restore System Backup on CWS



Restore System Backup on CWS

Install System Backup

```
# kinit root.admin

# install_cw

# ls /spdata/sys1/sdr/archives
  backup.96228.1215.CWS.works.fine

# sprestore_config backup.96228.1215.CWS.works.fine

# spmon -d

# splstdata -b
```

Verify SDR with previous output

This section describes how to restore a system backup on the Control Workstation. It tells you which steps are needed when you restore the system backup.

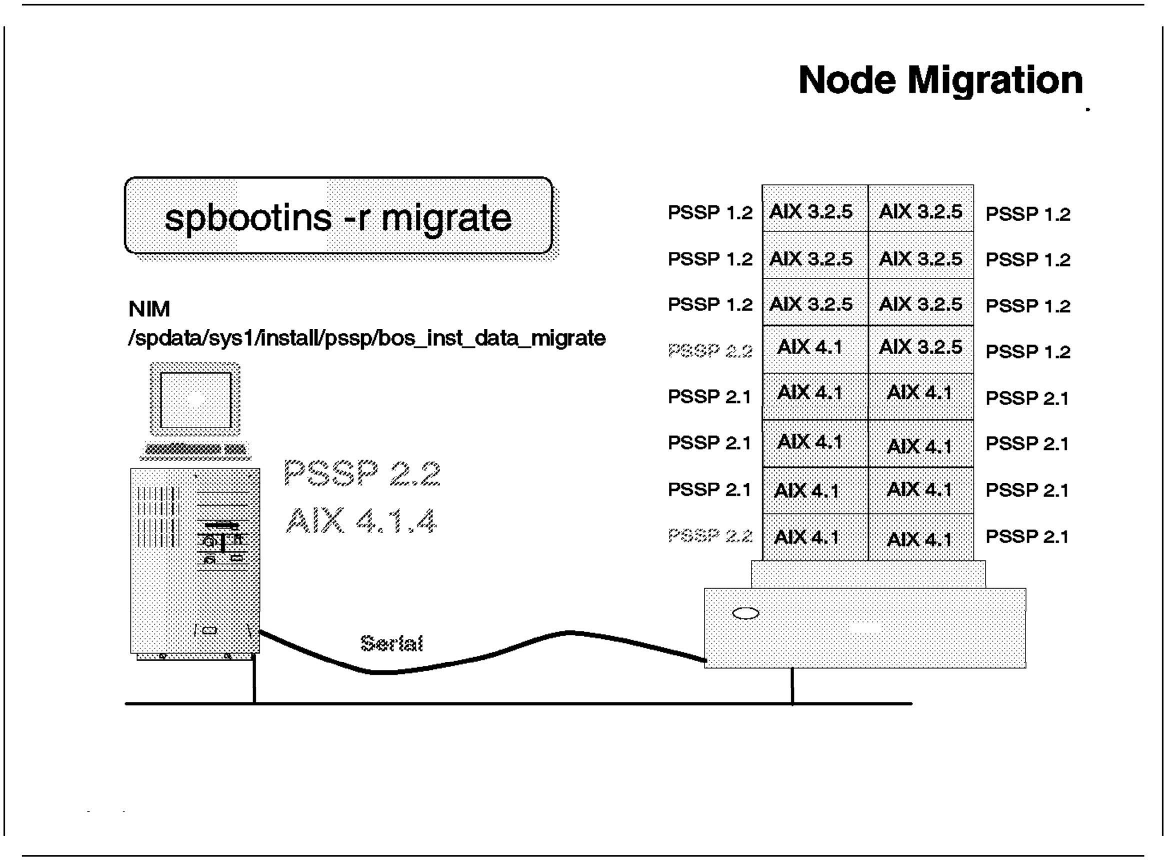
Perform the following steps:

1. Insert the backup tape into the tape drive.
2. Change the key to the service position. If your Control Workstation is a PCI-based RS/6000, press the F2 key at boot time. This will bring up a menu with the devices. Select the tape as boot device and press Enter to boot in service mode.
3. Select the correct disk to install the rootvg volume group.
4. Wait until the restore of the tape has finished.
5. Log in as root user.
6. Authenticate as the Kerberos administrative with the `kinit root.admin` command.
7. Issue the `install_cw` command to set the node_number for the Control Workstation to 0.
8. List the SDR archives.
9. Issue the `sprestore_config <archive_name>` command, where the <archive_name> is the name of your last SDR archive.

10. Check with the `spmon -d` command and with the `splstdata` command to see if your SDR is correct.

When all the information is correct, the system backup is restored correctly.

4.7 Node Migration



Node Migration

spbootins -r migrate

NIM
/spdata/sys1/install/pssp/bos_inst_data_migrate

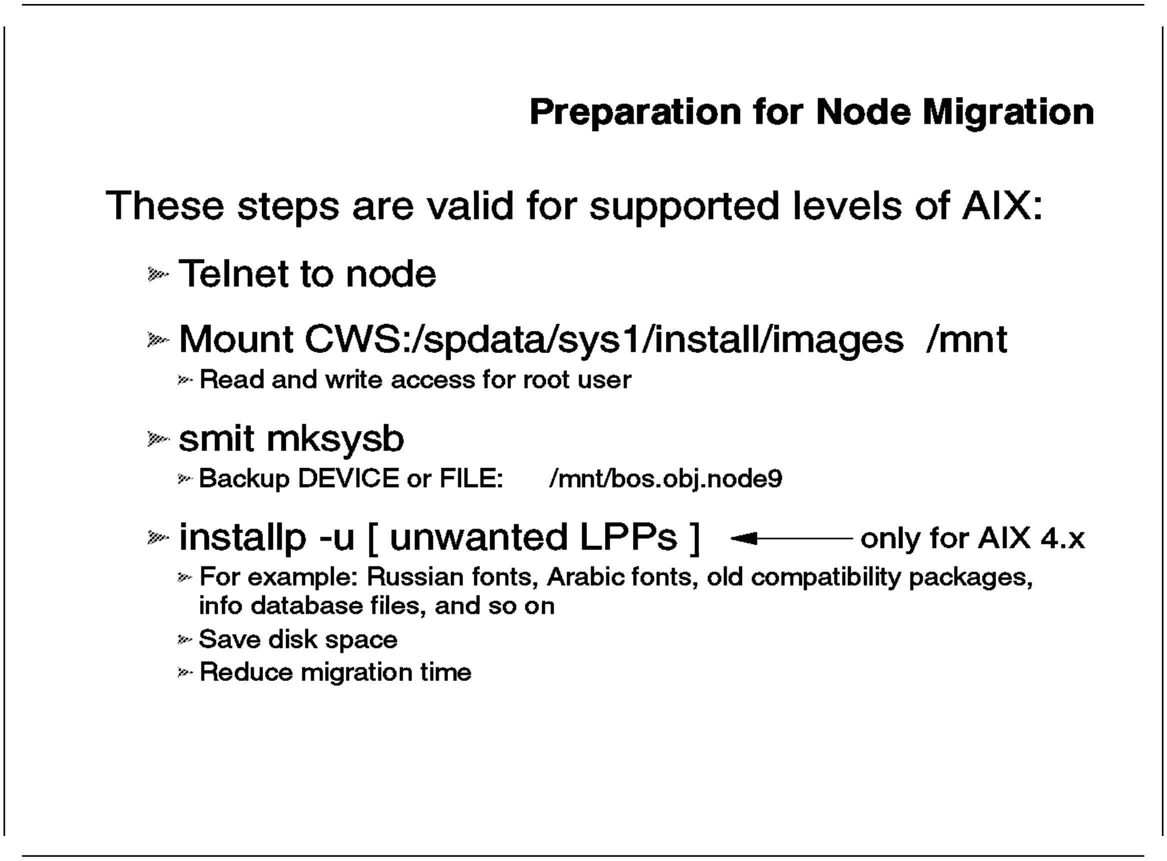
PSSP 1.2	AIX 3.2.5	AIX 3.2.5	PSSP 1.2
PSSP 1.2	AIX 3.2.5	AIX 3.2.5	PSSP 1.2
PSSP 1.2	AIX 3.2.5	AIX 3.2.5	PSSP 1.2
PSSP 2.2	AIX 4.1	AIX 3.2.5	PSSP 1.2
PSSP 2.1	AIX 4.1	AIX 4.1	PSSP 2.1
PSSP 2.1	AIX 4.1	AIX 4.1	PSSP 2.1
PSSP 2.1	AIX 4.1	AIX 4.1	PSSP 2.1
PSSP 2.2	AIX 4.1	AIX 4.1	PSSP 2.1

This section describes the migration process for the RS/6000 SP nodes to PSSP 2.2 and to AIX 4.1.4.

The node migration is a new feature with PSSP 2.2. With this release of PSSP, you do not need to perform an overwrite install on the nodes anymore. Now you can define the boot/install server′s response to the bootp request from the nodes to “migrate.” Use `spbootins -r migrate` or `smit server_dialog` to change the response to migrate. When you start setup_server, Network Installation Management (NIM) allocates resources and points to the /spdata/sys1/install/pssp/bosinst_data_migrate file. In this file, the install method is changed from *overwrite* to *migrate*. The file contains the following:

```
control_flow:
    CONSOLE = /dev/tty0
    INSTALL_METHOD = migrate
    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 =
    HDISKNAME = hdisk0

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

4.7.1 Preparation for Node Migration



Preparation for Node Migration

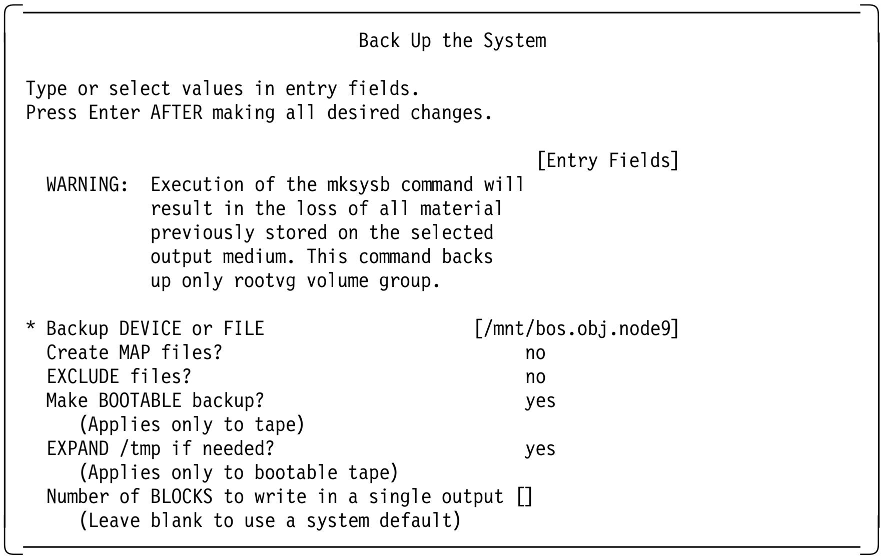
These steps are valid for supported levels of AIX:

- **Telnet to node**
- **Mount CWS:/spdata/sys1/install/images /mnt**
  - **Read and write access for root user**
- **smit mksysb**
  - **Backup DEVICE or FILE: /mnt/bos.obj.node9**
- **installp -u [unwanted LPPs]** ◀─── **only for AIX 4.x**
  - **For example: Russian fonts, Arabic fonts, old compatibility packages, info database files, and so on**
  - **Save disk space**
  - **Reduce migration time**

This section discusses what preparation is needed on the RS/6000 SP nodes before migrating the nodes to PSSP 2.2 and AIX 4.1.4.

Before migrating your RS/6000 SP nodes, perform the following steps:

1. Export the /spdata/sys1/install/images directory for the node that will be migrated. Verify that the root user has read/write permissions for that directory. You can use the `spbootins -r install` command to export that directory for the node, or you can modify the /etc/exports file. Verify the exported directories with the `exportfs -va` command.
2. Telnet to the node.
3. Mount the /spdata/sys1/install/images directory from the Control Workstation over the directory /mnt.
4. Use the `smit mksysb` command to perform the system backup. Set the backup name so that the backup will be performed over the network onto the Control Workstation.



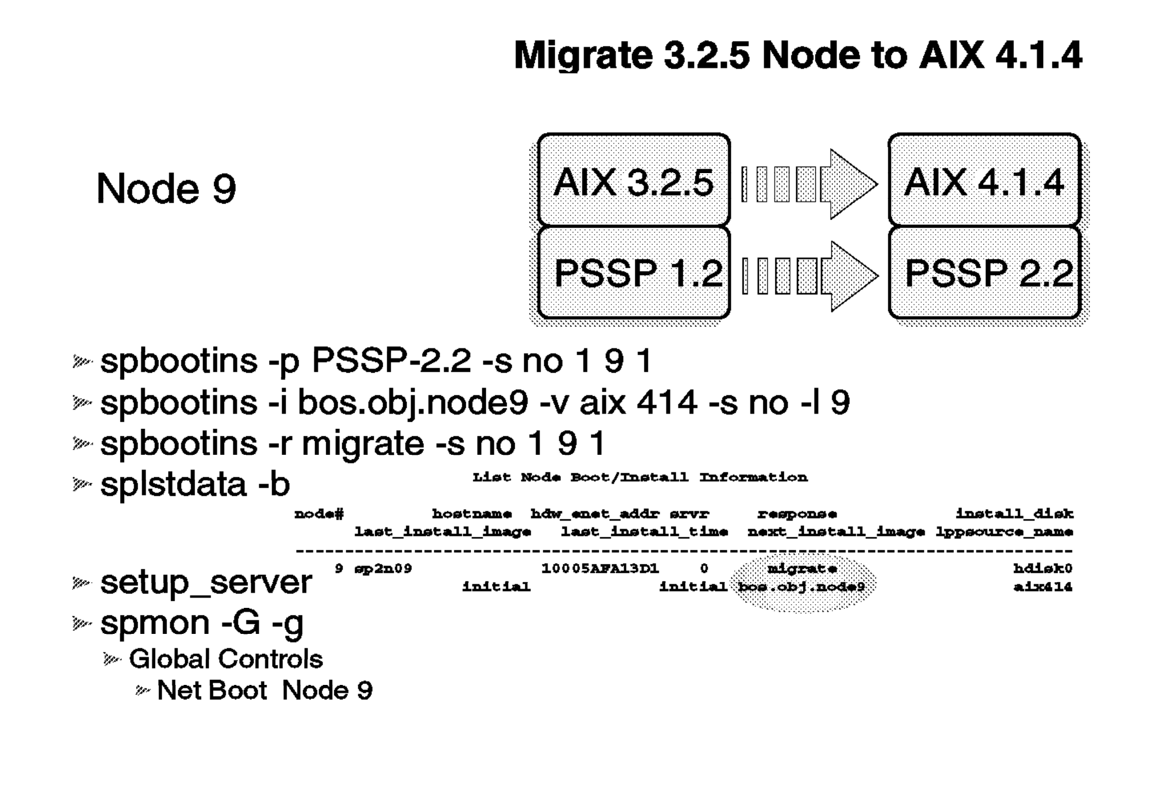
```
                              Back Up the System

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

                                                  [Entry Fields]
  WARNING:  Execution of the mksysb command will
            result in the loss of all material
            previously stored on the selected
            output medium. This command backs
            up only rootvg volume group.

* Backup DEVICE or FILE                     [/mnt/bos.obj.node9]
  Create MAP files?                              no
  EXCLUDE files?                                 no
  Make BOOTABLE backup?                          yes
     (Applies only to tape)
  EXPAND /tmp if needed?                         yes
     (Applies only to bootable tape)
  Number of BLOCKS to write in a single output []
     (Leave blank to use a system default)
```

5. Before starting the migration, perform some cleanup on the node. Remove software packages that you will not need anymore. Uninstall software that is not required on your site, such as:

   - Exotic fonts
   - Exotic languages
   - Old compatibility packages
   - Device drivers

6. Now the migration for that node can start.

4.7.2 Migrate 3.2.5 Node to AIX 4.1.4



Migrate 3.2.5 Node to AIX 4.1.4

Node 9

AIX 3.2.5 → AIX 4.1.4

PSSP 1.2 → PSSP 2.2

- spbootins -p PSSP-2.2 -s no 1 9 1
- spbootins -i bos.obj.node9 -v aix 414 -s no -l 9
- spbootins -r migrate -s no 1 9 1
- splstdata -b

```
                         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
-------------------------------------------------------------------------------
    9 sp2n09                10005AFA13D1     0     migrate                hdisk0
             initial                  initial bos.obj.node9               aix414
```

- setup_server
- spmon -G -g
  - Global Controls
    - Net Boot Node 9

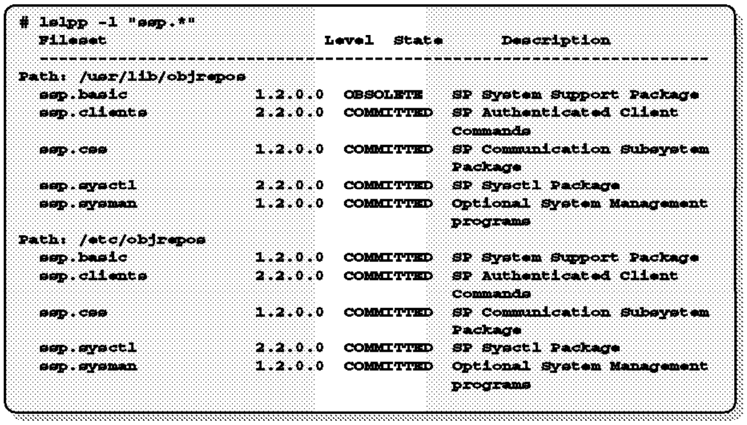
This section discusses what steps are needed to migrate an AIX 3.2.5 node to AIX 4.1.4 and PSSP 2.2.

Following are the migration steps for node 9:

1. Use the spbootins command to change the level of PSSP from 1.2 to 2.2.
2. Change the LPP source name with the “-v” parameter to aix414.
3. Specify the boot/install server′s response to the bootp request from the nodes to “migrate.”
4. Verify the SDR with the splstdata -b command.
5. Start the setup_server script to create the Network Installation Management (NIM) definitions for the node.
6. Netboot the node with SPmon or Perspectives.

After 45 minutes to 60 minutes, the installation will finish and you can restart the switch with Estart. The switch is active for 30 minutes and then stops working on this node.

Migrate 3.2.5 Node to AIX 4.1.4



```
# lslpp -l "ssp.*"
  Fileset                      Level  State      Description
  ----------------------------------------------------------------------
Path: /usr/lib/objrepos
  ssp.basic              1.2.0.0  OBSOLETE   SP System Support Package
  ssp.clients            2.2.0.0  COMMITTED  SP Authenticated Client
                                             Commands
  ssp.css                1.2.0.0  COMMITTED  SP Communication Subsystem
                                             Package
  ssp.sysctl             2.2.0.0  COMMITTED  SP Sysctl Package
  ssp.sysman             1.2.0.0  COMMITTED  Optional System Management
                                             programs
Path: /etc/objrepos
  ssp.basic              1.2.0.0  COMMITTED  SP System Support Package
  ssp.clients            2.2.0.0  COMMITTED  SP Authenticated Client
                                             Commands
  ssp.css                1.2.0.0  COMMITTED  SP Communication Subsystem
                                             Package
  ssp.sysctl             2.2.0.0  COMMITTED  SP Sysctl Package
  ssp.sysman             1.2.0.0  COMMITTED  Optional System Management
                                             programs
```

PSSP 1.2
and
PSSP 2.2
mixed on
one node

- installp -ug ssp
- mount CWS:/spdata/sys1/install/pssplpp/PSSP-2.2 /mnt
- installp -acgXd /mnt ssp.basic
- ksh /usr/lpp/ssp/install/bin/pssp_script &

To analyze the cause of the problem, log into the node and use the following commands to analyze the problem:

```
# telnet sp2n09
*****************************************************************
*                                                               *
*  Welcome to AIX Version 4.1!                                  *
*                                                               *
*****************************************************************
# oslevel
4.1.4.0
# lslpp -l ″ssp.*″
  Fileset                Level  State      Description
  ------------------------------------------------------------------
Path: /usr/lib/objrepos
  ssp.basic             1.2.0.0  OBSOLETE   SP System Support Package
  ssp.clients           2.2.0.0  COMMITTED  SP Authenticated Client
                                            Commands
  ssp.css               1.2.0.0  COMMITTED  SP Communication Subsystem
                                            Package
  ssp.sysctl            2.2.0.0  COMMITTED  SP Sysctl Package
  ssp.sysman            1.2.0.0  COMMITTED  Optional System Management
                                            programs
Path: /etc/objrepos
  ssp.basic             1.2.0.0  COMMITTED  SP System Support Package
  ssp.clients           2.2.0.0  COMMITTED  SP Authenticated Client
                                            Commands
  ssp.css               1.2.0.0  COMMITTED  SP Communication Subsystem
                                            Package
  ssp.sysctl              2.2.0.0  COMMITTED  SP Sysctl Package
  ssp.sysman              1.2.0.0  COMMITTED  Optional System Management
                                              programs
```

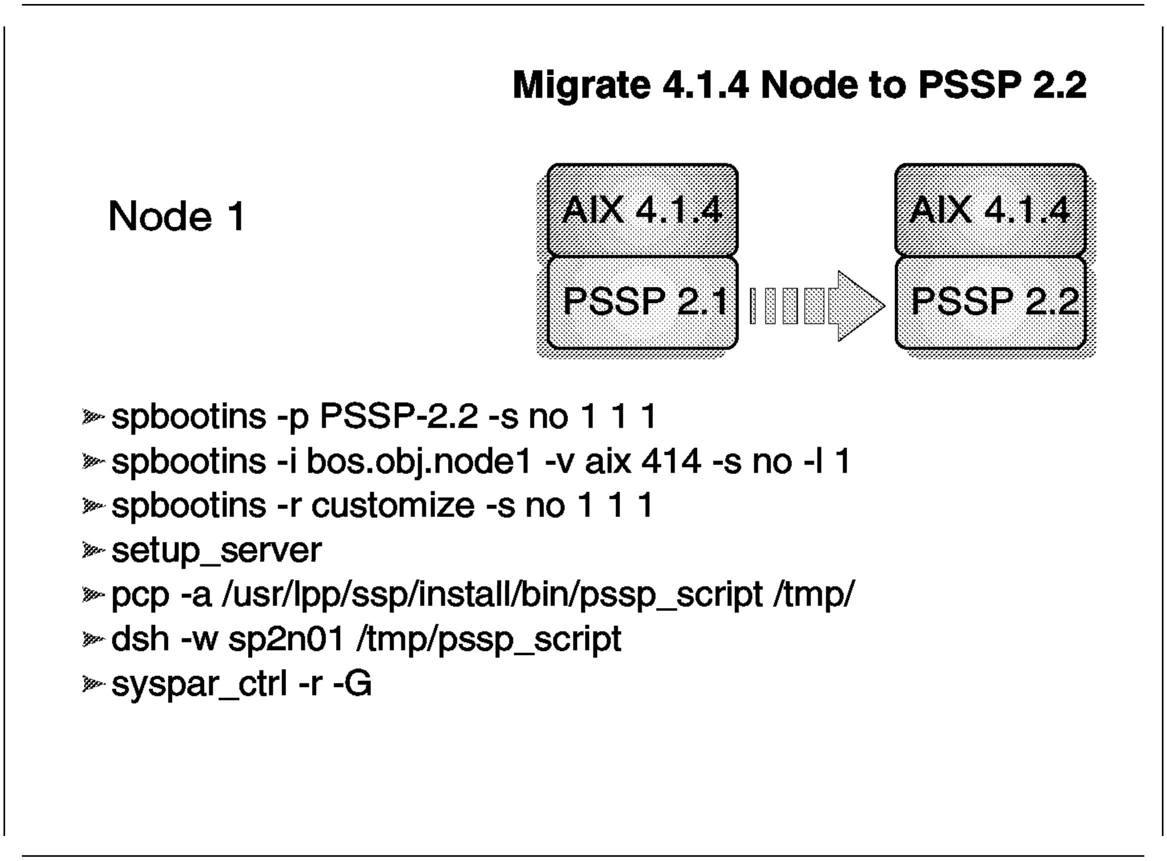
This mix of PSSP version is not valid. Uninstall all PSSP file sets and use the following commands.

Note: The uninstall option will be fully supported with the fix for APAR IX55009.

```
# installp -ug ssp
 . . .
# mount sp2cw0:/spdata/sys1/install/aix414/lppsource /mnt
# installp -acgXd /mnt xlC.rte
 . . .
# umount /mnt
# mount sp2cw0:/spdata/sys1/install/pssplpp/PSSP-2.2 /mnt
# installp -acgXd /mnt ssp.basic
 . . .
installp:  APPLYING software for:
        ssp.perlpkg 2.2.0.0
        ssp.clients 2.2.0.0
        ssp.basic 2.2.0.0
 . . .
# ksh -x /usr/lpp/ssp/install/bin/pssp_script &
 . . .
+ 3> /var/adm/SPlogs/sysman/sp2n09.config.log.13620
# tail -f /var/adm/SPlogs/sysman/sp2n09.config.log.13620
 . . .
# lslpp -l ″ssp.*″
  Fileset                 Level  State      Description
  ------------------------------------------------------------------
Path: /usr/lib/objrepos
  ssp.basic             2.2.0.0  COMMITTED  SP System Support Package
  ssp.clients           2.2.0.0  COMMITTED  SP Authenticated Client
                                            Commands
  ssp.css               2.2.0.0  COMMITTED  SP Communication Subsystem
                                            Package
  ssp.ha                2.2.0.0  COMMITTED  SP High Availability Services
  ssp.perlpkg           2.2.0.0  COMMITTED  SP PERL Distribution Package
  ssp.pman              2.2.0.0  COMMITTED  SP Problem Management
  ssp.sysctl            2.2.0.0  COMMITTED  SP Sysctl Package
  ssp.sysman            2.2.0.0  COMMITTED  Optional System Management
                                            programs
Path: /etc/objrepos
 . . .
```

The node migration is finished. Use the Estart command to restart the switch.

4.7.3 Migrate 4.1.4 Node to PSSP 2.2



Migrate 4.1.4 Node to PSSP 2.2

Node 1

AIX 4.1.4
PSSP 2.1
AIX 4.1.4
PSSP 2.2

- spbootins -p PSSP-2.2 -s no 1 1 1
- spbootins -i bos.obj.node1 -v aix 414 -s no -l 1
- spbootins -r customize -s no 1 1 1
- setup_server
- pcp -a /usr/lpp/ssp/install/bin/pssp_script /tmp/
- dsh -w sp2n01 /tmp/pssp_script
- syspar_ctrl -r -G

This section discusses what steps are needed to migrate an AIX 4.1.4 node from PSSP 2.1 to PSSP 2.2.

When you use PSSP 2.1 on the nodes, you have either AIX 4.1.3 or AIX 4.1.4 installed on the nodes. If you have AIX 4.1.3 installed, you must upgrade to AIX 4.1.4. You can use the dsh command to perform the steps on multiple nodes. This example shows you how to upgrade node sp2n01:

```
# dsh -w sp2n01
dsh> mount CWS:/spdata/sys1/install/aix414/lppsource /mnt
dsh> installp -acgNXd /mnt bos.rte.install 4.1.4.0
 . . .
dsh> installp -acgBXd /mnt all
 . . .
dsh> shutdown -Fr
```

Verify with the oslevel command if AIX is at level 4.1.4.

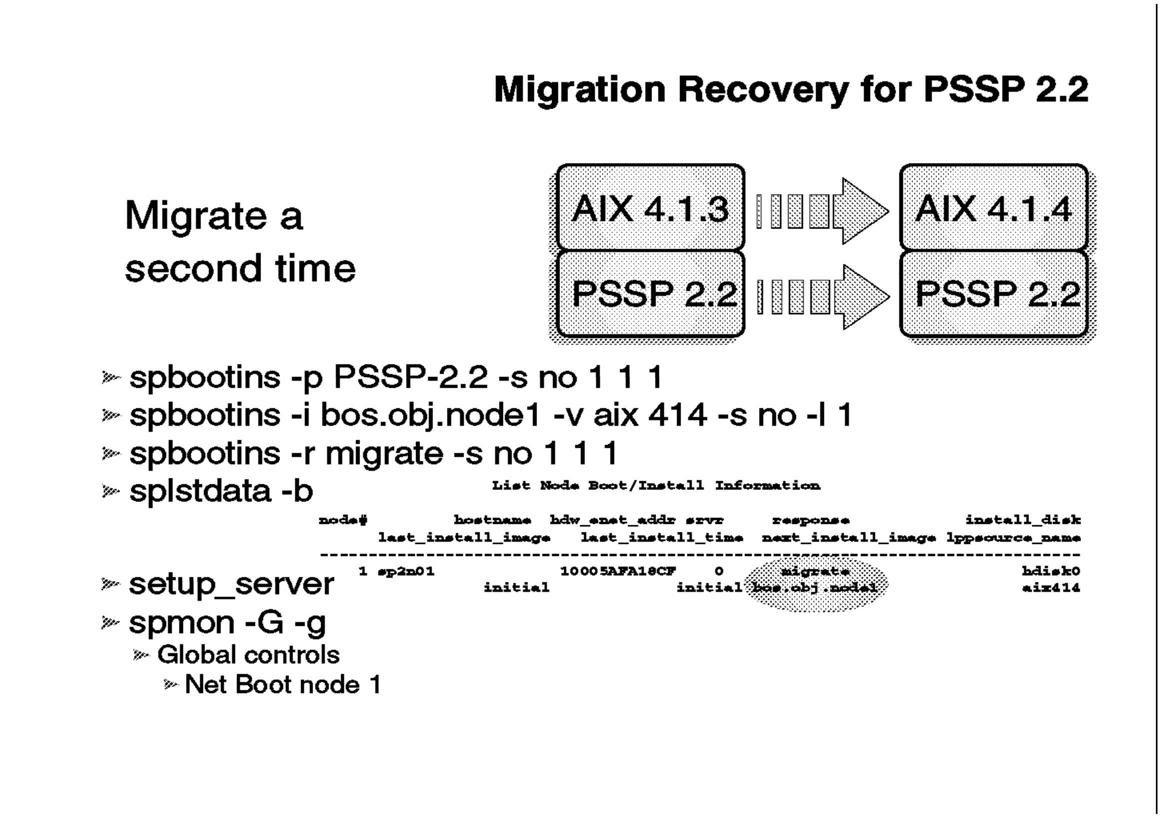
Now upgrade PSSP to level 2.1. This example shows the commands for node 1:

- Define the PSSP code version with the spbootins command.
- Change the LPP source name to aix414 and define the system backup from node 1 as your network install image.
- Set the response from server to bootp request to *disk* or to *customize*.
- Start the setup_server script.

Now there are two ways to upgrade. You can either install the new PSSP 2.2 software over PSSP 2.1 or uninstall the old PSSP 2.1 software and install PSSP 2.2. To install PSSP 2.2 over PSSP 2.1, follow these steps:

```
# pcp -a /spdata/sys1/install/bin/pssp_script /tmp
# telnet sp2n01
# ksh -x /tmp/pssp_script &
```

4.7.4 Preservation Installation for PSSP 2.2



Migration Recovery for PSSP 2.2

Migrate a second time

AIX 4.1.3 → AIX 4.1.4

PSSP 2.2 → PSSP 2.2

- spbootins -p PSSP-2.2 -s no 1 1 1
- spbootins -i bos.obj.node1 -v aix 414 -s no -l 1
- spbootins -r migrate -s no 1 1 1
- splstdata -b

```
                          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
--------------------------------------------------------------------------------
    1 sp2n01                10005AFA18CF     0    migrate                 hdisk0
                 initial                initial bos.obj.node1             aix414
```

- setup_server
- spmon -G -g
  - Global controls
    - Net Boot node 1

This section describes what happens when you perform the migration a second time and how you can use the *migrate* option to upgrade from AIX 4.1.3 to AIX 4.1.4.

In general, a migration can only be performed once. But suppose your AIX 4.1 node has a lot of problems and you want to reinstall it. You can either choose to perform an overwrite install or a migration install. You do not want to use overwrite because you have added some file systems to your node, and recreating the file systems is time consuming. Another fact is that AIX 4.1.4 does not support the migrate option from AIX 4.1.3, and you cannot perform migration installation over AIX 4.1.4 a second time.

The next paragraphs will show you how to perform a **Preservation Installation** for node 1.

First define all values for the System Data Repository (SDR) on the Control Workstation with the spbootins command:

```
# spbootins -p PSSP-2.2 -s no -l 1
# spbootins -i bos.obj.node1 -v aix414 -s no -l 1
# spbootins -r migrate -s no 1 1 1
```

You can also use smit server_dialog to set the bootp response to migrate. Before you start setup_server, use the splstdata -b command to check if all fields are set correctly. Then use spmon command or the sphardware command to

Netboot node 1. Later the program /usr/lpp/ssp/expect/bin/expect will start the script /usr/lpp/ssp/bin/nodecond2 to perform the node conditioning. This command will fail and time out after 240 seconds.

It fails because the migration install option is not possible. Since it does not work, you must consider your options and know the difference between migration install and preserve installation.

The migration installation will preserve all files in your rootvg volume group, except device drivers files. It will delete all files in:

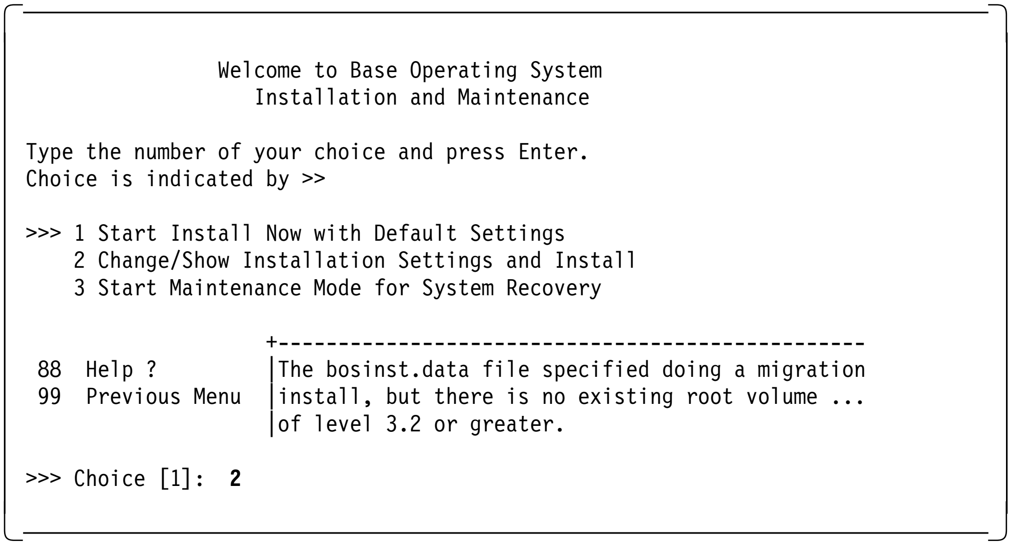
- /usr/lib/drivers
- /usr/lib/microcode
- /usr/lib/methods
- /dev

The preservation installation will preserve all user data in the rootvg volume group. This method overwrites all files in the following file systems:

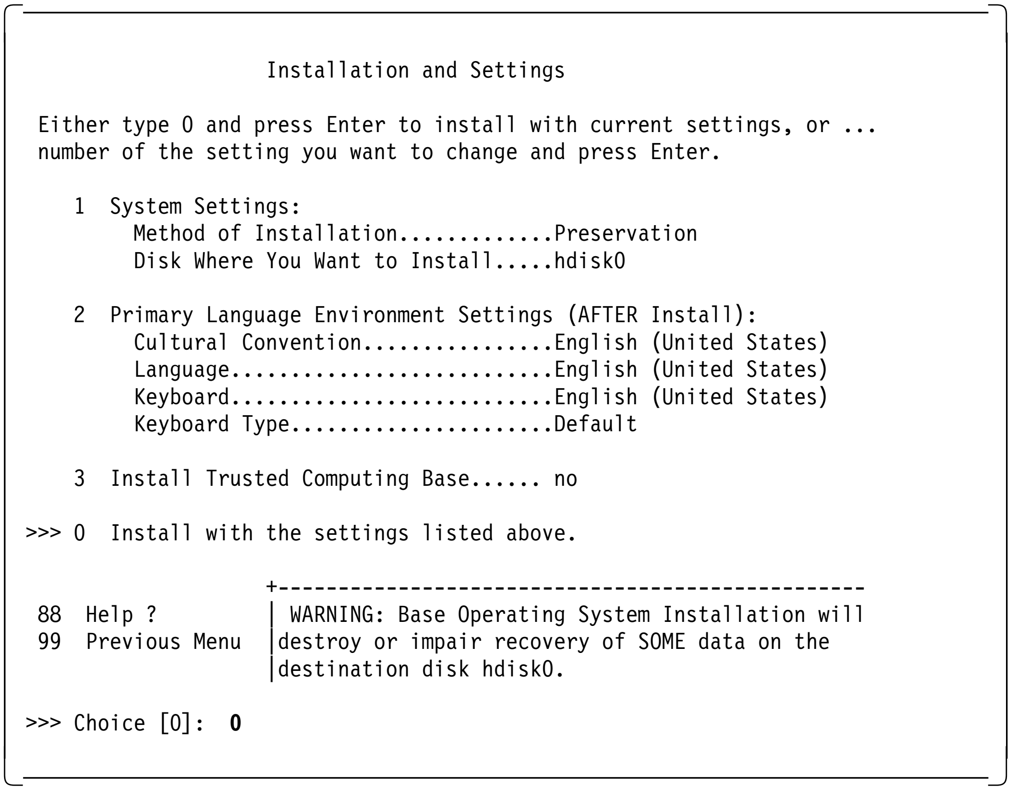
- /usr
- /tmp
- /var
- / (root)

Use the s1term -w command to perform a preservation installation:

```
# s1term -w 1 1
1
```



```
                 Welcome to Base Operating System
                    Installation and Maintenance

 Type the number of your choice and press Enter.
 Choice is indicated by >>

 >>> 1 Start Install Now with Default Settings
     2 Change/Show Installation Settings and Install
     3 Start Maintenance Mode for System Recovery

                      +------------------------------------------------
  88  Help ?          |The bosinst.data file specified doing a migration
  99  Previous Menu   |install, but there is no existing root volume ...
                      |of level 3.2 or greater.

 >>> Choice [1]:  2
```



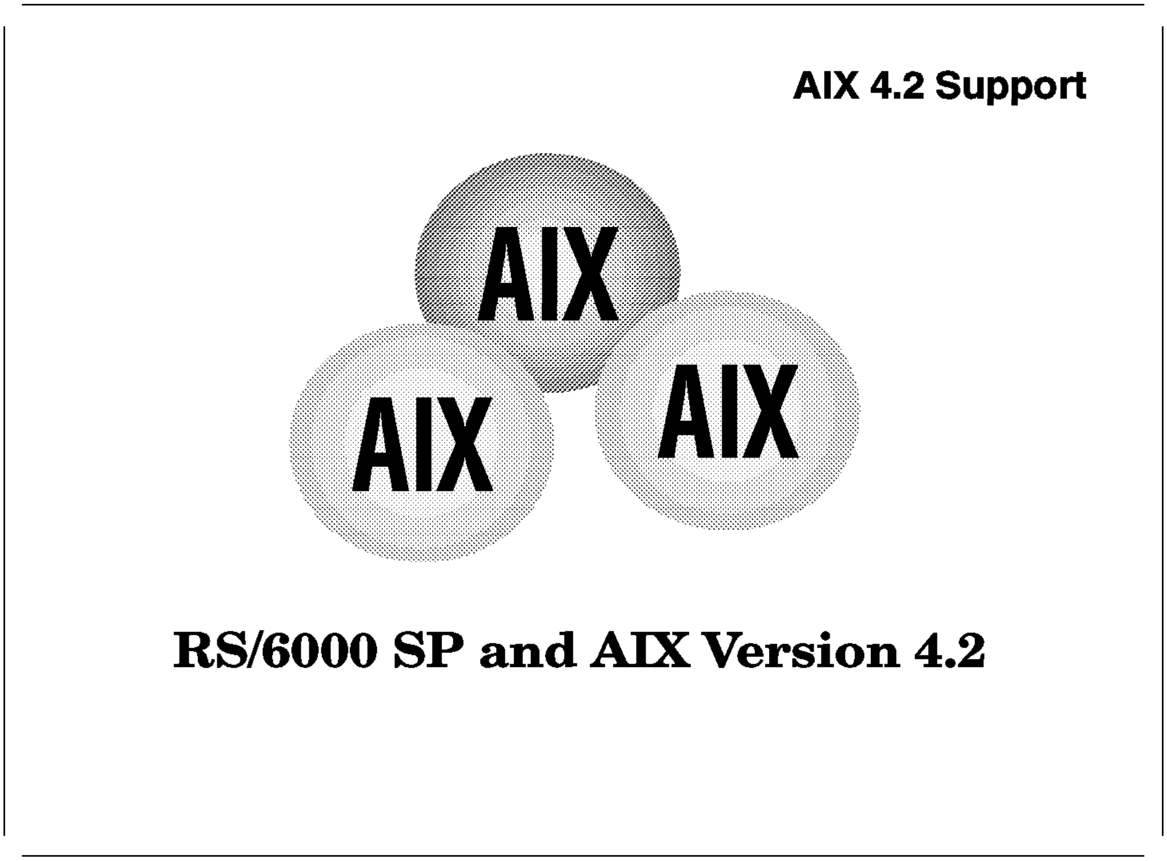
```
                    Installation and Settings

 Either type 0 and press Enter to install with current settings, or ...
 number of the setting you want to change and press Enter.

    1  System Settings:
         Method of Installation.............Preservation
         Disk Where You Want to Install.....hdisk0

    2  Primary Language Environment Settings (AFTER Install):
         Cultural Convention................English (United States)
         Language...........................English (United States)
         Keyboard...........................English (United States)
         Keyboard Type......................Default

    3  Install Trusted Computing Base...... no

>>> 0  Install with the settings listed above.

                    +-------------------------------------------------
 88  Help ?         | WARNING: Base Operating System Installation will
 99  Previous Menu  |destroy or impair recovery of SOME data on the
                    |destination disk hdisk0.

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

With

`s1term`

you can follow the installation process until the installation is finished. At the end, the pssp_script will be executed and the node will be rebooted.

Chapter 5. AIX 4.2 Support



AIX 4.2 Support

RS/6000 SP and AIX Version 4.2

IBM currently intends to support the AIX 4.2 operating system in binary compatibility mode for:

- IBM Parallel System Support Programs for AIX V2.2
- IBM Recoverable Virtual Shared Disk V1.2
- IBM Parallel Environment for AIX V2.2
- IBM PVMe for AIX V2.2
- IBM LoadLeveler 1.3.0

by the end of 1996.

5.1 Table of Contents

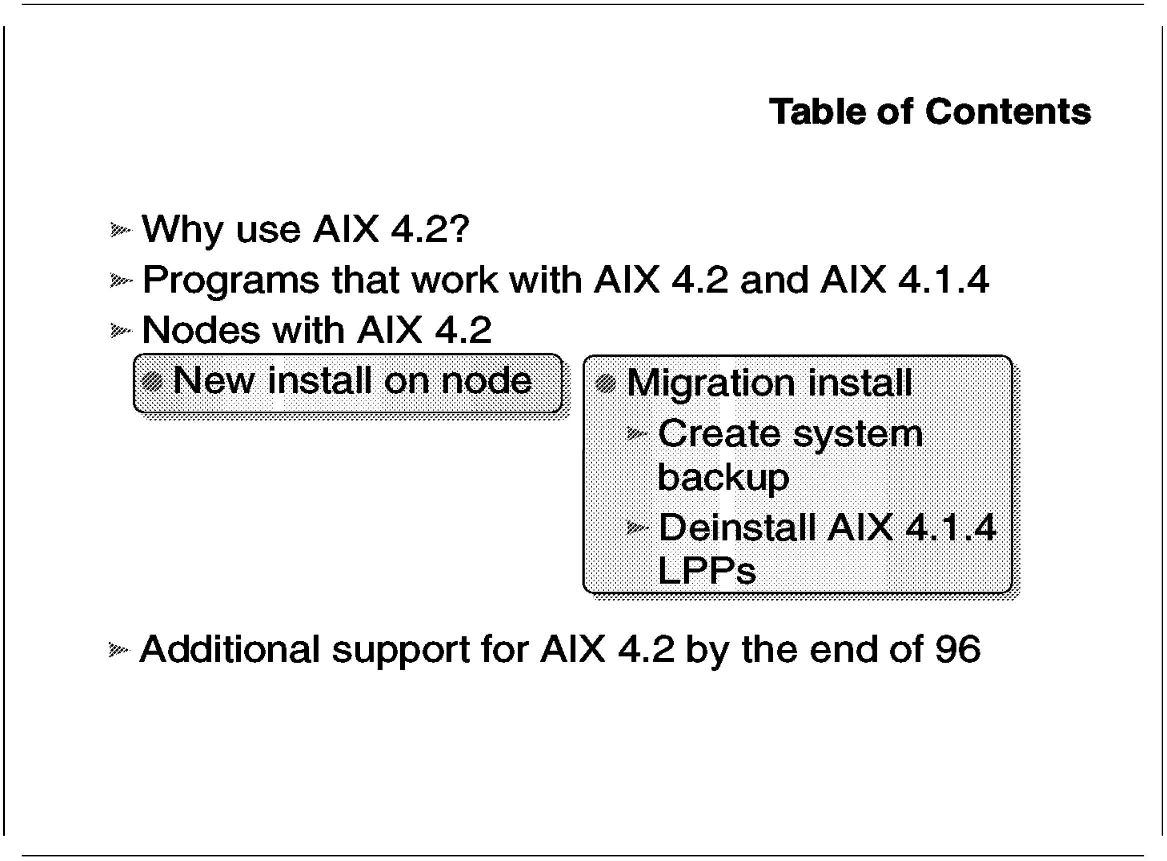


Table of Contents

- Why use AIX 4.2?
- Programs that work with AIX 4.2 and AIX 4.1.4
- Nodes with AIX 4.2
  - New install on node
  - Migration install
    - Create system backup
    - Deinstall AIX 4.1.4 LPPs
- Additional support for AIX 4.2 by the end of 96

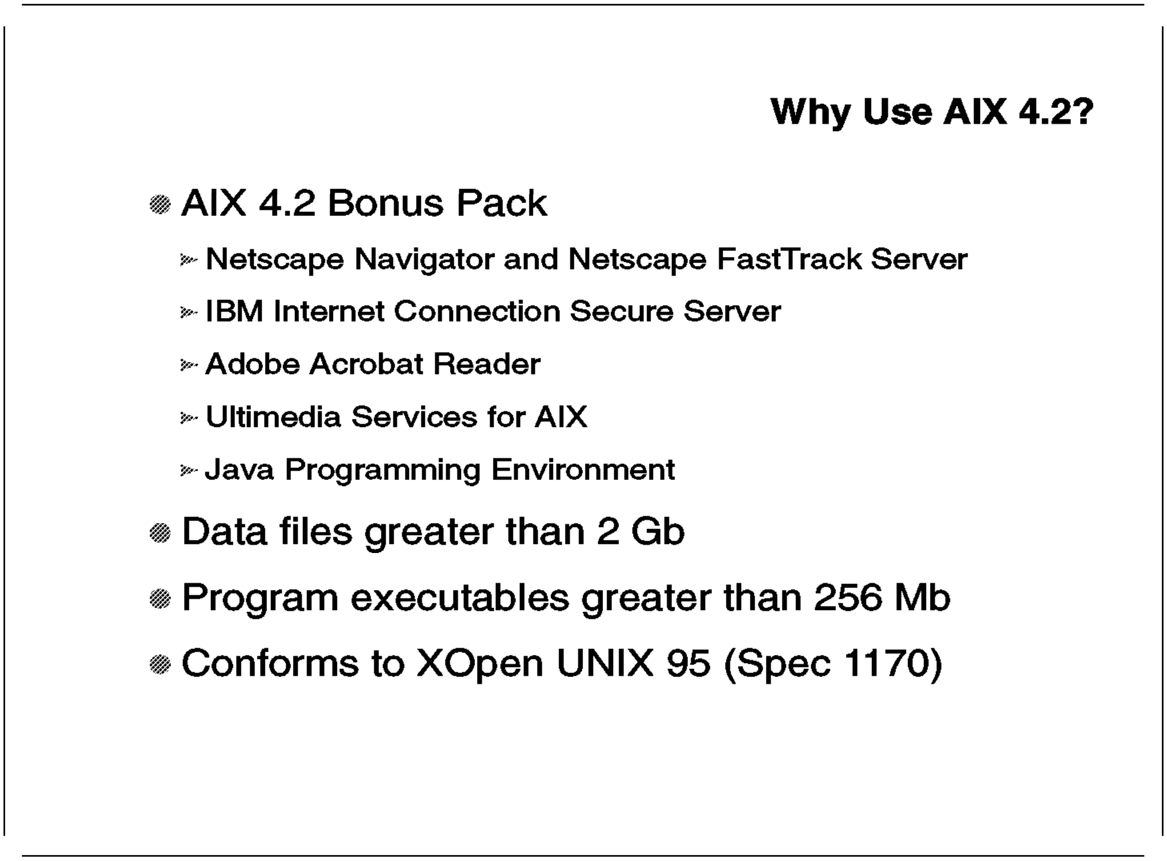
This table of contents gives you an overview of AIX 4.2. The following section tells you the important issues for AIX 4.2 and why to use AIX 4.2 on the RS/6000 SP.

A later section describes which programs are compatible with AIX 4.1.4 and AIX 4.2.

You will also learn how to install a node with AIX 4.2. You can either choose to perform an overwrite install or to perform a migration install.

The last section describes which programs will be supported on AIX 4.2 by the end of 1996.

5.2 Why Use AIX 4.2?



Why Use AIX 4.2?

- AIX 4.2 Bonus Pack
  - Netscape Navigator and Netscape FastTrack Server
  - IBM Internet Connection Secure Server
  - Adobe Acrobat Reader
  - Ultimedia Services for AIX
  - Java Programming Environment
- Data files greater than 2 Gb
- Program executables greater than 256 Mb
- Conforms to XOpen UNIX 95 (Spec 1170)

This section discusses the reasons to have AIX 4.2 on the RS/6000 SP. One of the greatest benefits with AIX 4.2 is the no-charge Bonus Pack that contains:

- Netscape Navigator, Version 2.01
- Netscape FastTrack Server
- IBM Internet Connection Secure Server for AIX
- Adobe Acrobat Reader, Version 2.1
- Ultimedia Services for AIX, Version 2.1.4
- IBM′s implementation for AIX of Sun′s Java programming environment, Version 1.0

The Netscape Navigator program is a Web browser. It allows Web pages to be accessed from the Internet and viewed locally on the client desktop. This Web browser supports JavaScript, a built-in scripting language. JavaScript allows a developer to create an HTML (Web) document that can display a scrolling advertisement banner on a Web page.

The Netscape FastTrack Server is a Web server. It is very easy to use and you can create your own Web Site on the RS/6000 SP. This server is for non-programmers who want a simpler, low-cost means to publish on the Internet or on internal Intranets. This server replaced the Netscape Commerce Server.

The IBM Internet Connection Secure Server for AIX is the Web server from IBM. It has both similar features to the Netscape FastTrack Server and additional features, such as:

- Integration of DB2 gateway
- Integration of CICS gateway
- National Language Support
- Internet security, such as secure sockets layer and Secure Hypertext Transfer Protocol (S-HTTP)
- Support for multiple IP addresses to maintain multiple Web sites on a single server.

The Adobe Acrobat Reader lets you view, distribute, print, and save documents in Portable Document Format (PDF) regardless of the computer, operating system, fonts, or application used to create the original file. Virtually any PostScript document can be converted into a PDF file.

Ultimedia Services for AIX enables the RS/6000 SP system for multimedia. With this software you can create and edit MPEG-1 movies. Ultimedia Services can be integrated with Netscape Navigator to provide utilities such as a media-player to view PhotoCD, TIFF, GIF, and JPEG image formats.

The IBM implementation of Sun′s Java is an object-oriented programming environment that operates independently of any operating system or microprocessor. Java programs (called applets) enable WWW users to deliver more visually compelling Web content, such as using animation.

One other reason to use AIX 4.2 is the need to have files greater than 2 Gb in one filesystem. AIX 4.2 supports data files up to 64 Gb and this is very important for databases.

The support for executables with initialized data of sizes larger than 256 Mb is important for scientific customers and customers who use data mining.

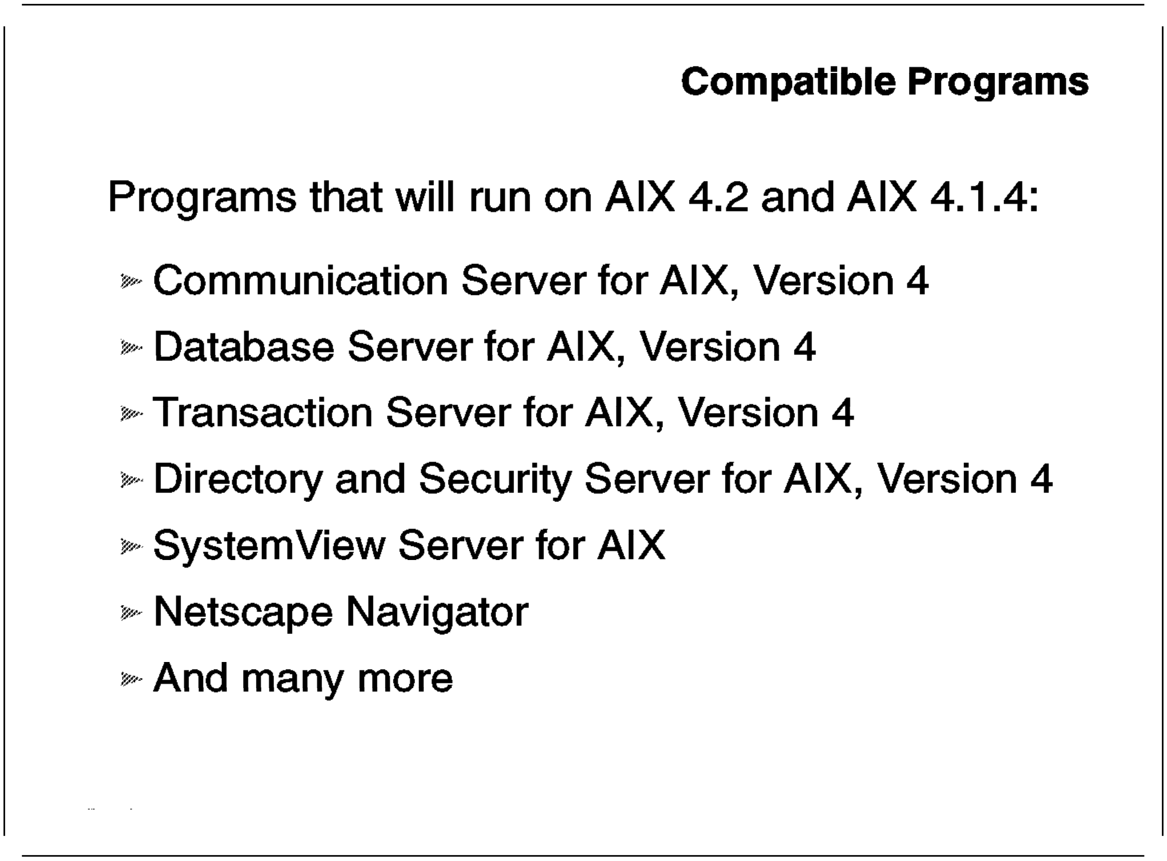
AIX 4.2 is designed to conform to XOpen′s UNIX 95 specification (also known as Spec 1170). This is important for software companies and programmers.

There are other advantages to using AIX 4.2, such as:

- The RS/6000 Welcome Center is a user-friendly introduction to the RS/6000 and AIX product families.

- AIX Connections provides a workgroup solution for a network of PC systems. This server software package provides PC-to-UNIX connectivity.

5.3 Compatible Programs



Compatible Programs

Programs that will run on AIX 4.2 and AIX 4.1.4:

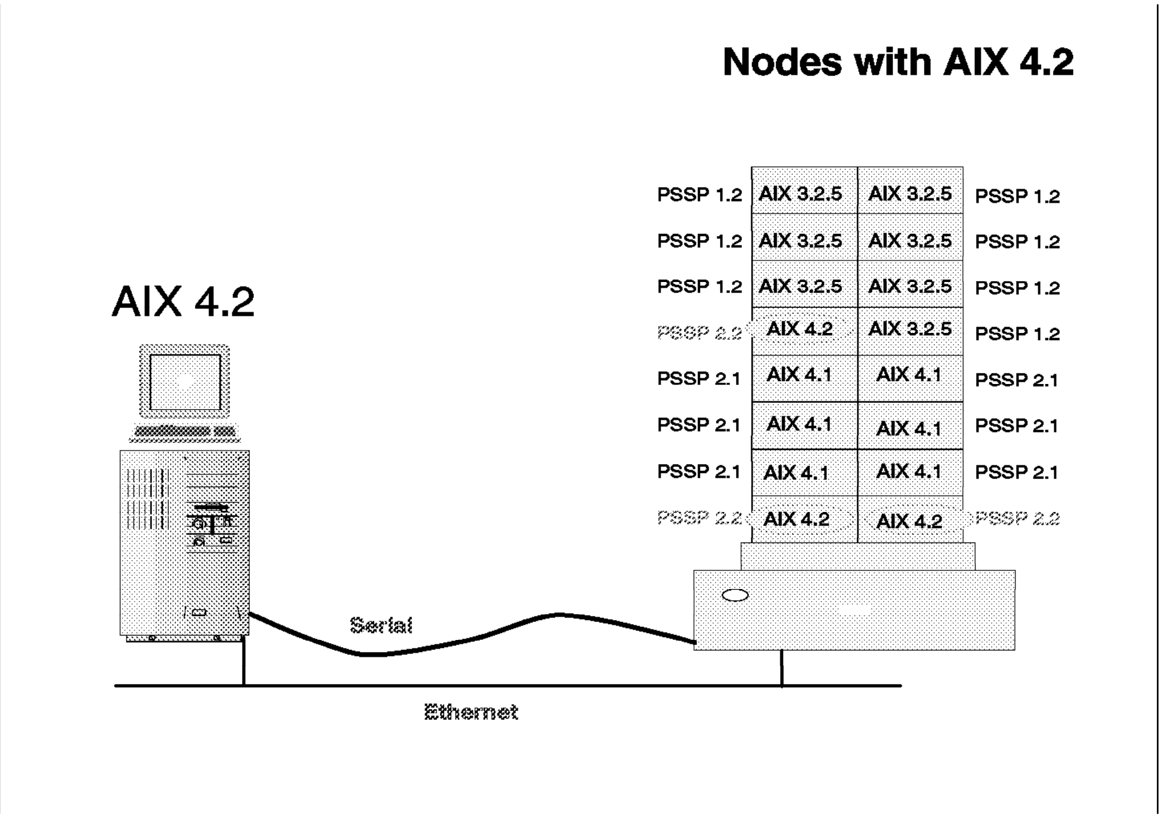
- Communication Server for AIX, Version 4
- Database Server for AIX, Version 4
- Transaction Server for AIX, Version 4
- Directory and Security Server for AIX, Version 4
- SystemView Server for AIX
- Netscape Navigator
- And many more

Following is a list of programs that work on AIX 4.2 and AIX 4.1.4:

- Performance Toolbox is a Motif-based application that contains performance management tools in a toolbox framework to help you monitor, analyze, and tune your RS/6000 SP systems for optimal local or client/server performance.
- The Communications Server for AIX enables workstations to communicate with mainframes and AS/400 hosts and other workstations through a powerful multiprotocol gateway for SNA networks.
- The Database Server for AIX is based on industry-standard SQL relational database technology. The Database Server is scalable from the LAN database client/server environment to powerful RS/6000 SP systems for businesses requiring large, highly available databases.
- The Transaction Server for AIX is based on CICS and Encina technology. The Transaction Server builds upon the services provided by the other IBM Software Servers to provide a powerful environment for the development, execution, and management of business-critical client/server applications.
- The Directory and Security Server (DSS) is based on the Open Software Foundation′s (OSF) Distributed Computing Environment (DCE). DSS provides the essential directory, time, security, and remote procedure call (RPC) services needed to support the network-centric enterprise.

- The SystemView Server for AIX provides functions that allow the LAN resources to be managed from an enterprise-wide management focal point.
- Netscape Navigator works with AIX 4.1 and AIX 4.2.
- The IBM Internet Connection Server provides a reliable foundation for businesses to build their presence on the Internet.
- And there are many more software programs that work with AIX 4.1 and AIX 4.2.

5.4 Nodes with AIX 4.2



Nodes with AIX 4.2

PSSP 1.2	AIX 3.2.5	AIX 3.2.5	PSSP 1.2
PSSP 1.2	AIX 3.2.5	AIX 3.2.5	PSSP 1.2
PSSP 1.2	AIX 3.2.5	AIX 3.2.5	PSSP 1.2
PSSP 2.2	AIX 4.2	AIX 3.2.5	PSSP 1.2
PSSP 2.1	AIX 4.1	AIX 4.1	PSSP 2.1
PSSP 2.1	AIX 4.1	AIX 4.1	PSSP 2.1
PSSP 2.1	AIX 4.1	AIX 4.1	PSSP 2.1
PSSP 2.2	AIX 4.2	AIX 4.2	PSSP 2.2

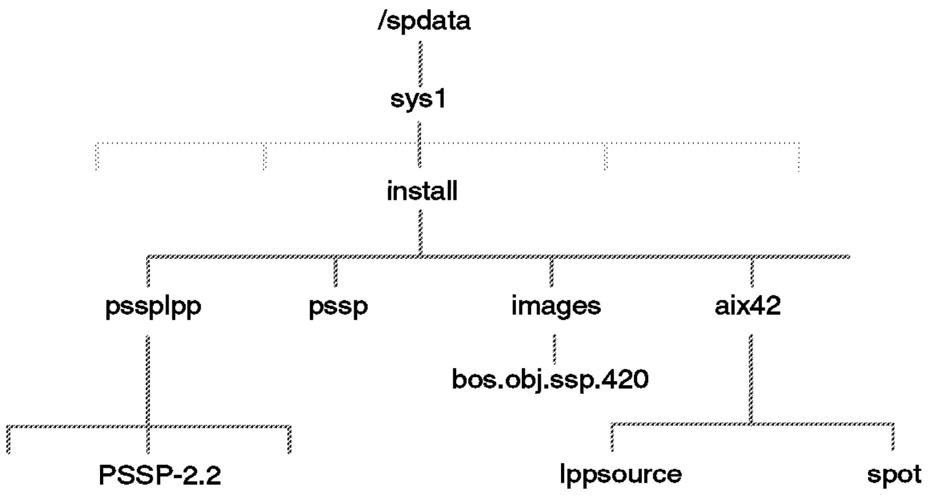
This section discusses how an RS/6000 SP node will be upgraded to AIX 4.2.

Before the RS/6000 SP will work with AIX 4.2 nodes, execute the following steps:

1. Create a system backup of your Control Workstation (CWS).
2. Uninstall all software packages that you will not need anymore (for example, exotic fonts and foreign languages that you do not use). This saves disk space and will reduce the migration time to AIX 4.2.
3. Issue the `installp -ug devices` command to remove device drivers that are not needed. This command will only remove those device drivers from AIX 4.1.4 that are not needed. The AIX 4.2 migration process will remove all device drivers in the directory /usr/lib/drivers, but not the entries in the Object Data Management (ODM) databases.
4. Upgrade the CWS with the migration option to AIX 4.2. See the *AIX Version 4.2 Installation Guide*, SC23-1924, for the detailed AIX 4.2 migration process.
5. Reinstall the PSSP *ssp.css* option.

Now you can change the SDR for the nodes and define the migrate or install bootp response.

AIX 4.2 Directories



Before you install the AIX 4.2 and PSSP 2.2 software images, you need the following directories:

- /spdata/sys1/install/pssplpp/PSSP-2.2
- /spdata/sys1/install/pssp
- /spdata/sys1/install/images
- /spdata/sys1/install/aix42/lppsource

The pssplpp directory contains the PSSP-2.2 subdirectory. The PSSP-2.2 directory contains all Parallel System Support Programs software images and PTFs.

The pssp directory contains the Network Installation Management (NIM) configuration files for the nodes.

The images directory contains the **spimg** image for AIX 4.2 and system backup images from the nodes. The spimg contains a single file containing a system backup (mksysb) image of a minimal AIX 4.2 system.

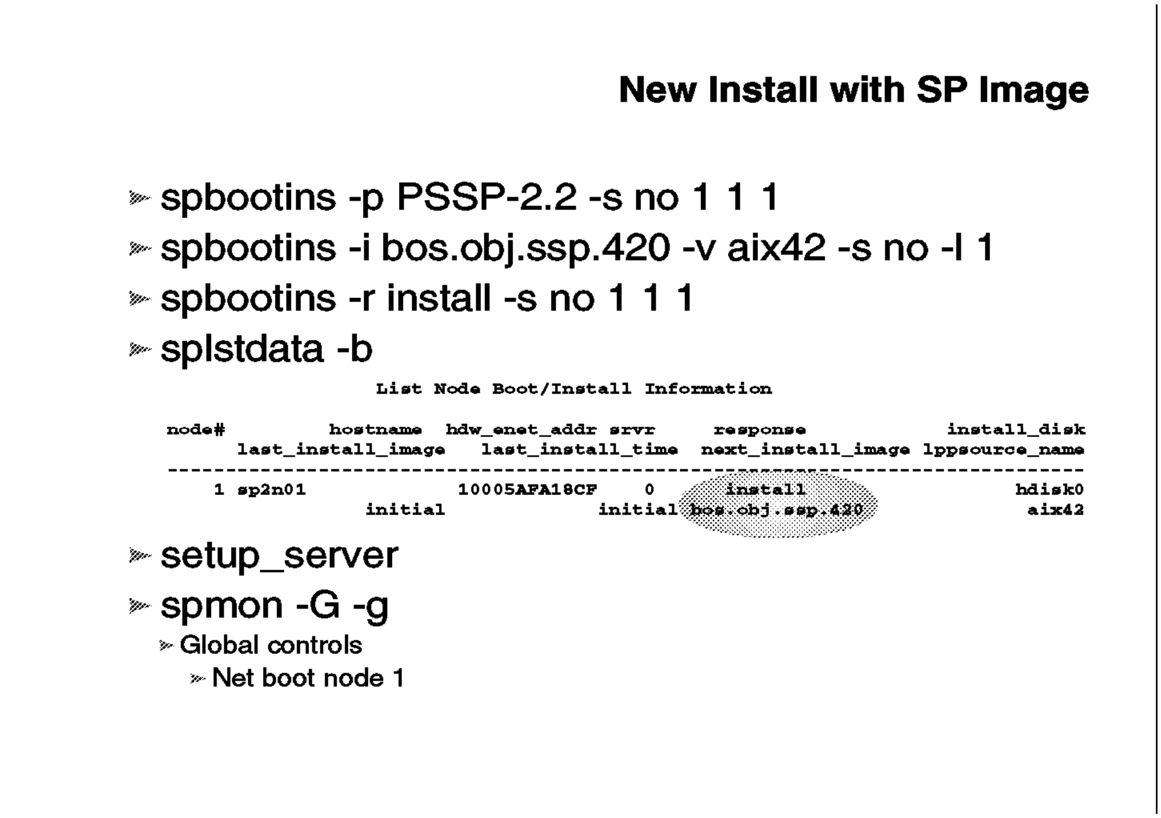
The aix42 directory contains the lppsource subdirectory. The name aix42 is a synonym for the Control Workstation LPP source directory. With PSSP 2.2 you can have different NIM filesets and NIM SPOTs on the Control Workstation. The subdirectory lppsource contains all AIX 4.2 images that are needed to install and migrate a node to AIX 4.2. The spot directory will be created by NIM.

The following table shows the disk space requirement for AIX 4.2 and PSSP 2.2.

Table 7. AIX 4.2 and PSSP 2.2 Directories

Directory	Size
/spdata/sys1/install/pssplpp/PSSP-2.2	275 Mb total
/spdata/sys1/install/pssplpp/PSSP-2.2/ptf.1	85 Mb
/spdata/sys1/install/pssp	1 Mb
/spdata/sys1/install/images	150 Mb minimum
/spdata/sys1/install/images	480 Mb average
/spdata/sys1/install/aix420	600 Mb minimum

5.5 New Install with System Backup



New Install with SP Image

- spbootins -p PSSP-2.2 -s no 1 1 1
- spbootins -i bos.obj.ssp.420 -v aix42 -s no -l 1
- spbootins -r install -s no 1 1 1
- splstdata -b

```
                    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
-------------------------------------------------------------------------------
    1 sp2n01            10005AFA18CF    0       install                 hdisk0
             initial                initial bos.obj.ssp.420              aix42
```

- setup_server
- spmon -G -g
  - Global controls
    - Net boot node 1

This section describes the steps you need to perform the install of AIX 4.2 on RS/6000 SP node 1.

1. Verify on your AIX 4.2 Control Workstation that the file *bos.obj.ssp.420* exists in the /spdata/sys1/install/images directory.
2. Verify that all AIX 4.2 images for NIM are installed in the directory /usr/spdata/sys1/install/aix42/lppsource. The SPOT directory for AIX 4.2 is /usr/spdata/sys1/install/aix42/spot and is created by NIM.
3. Define the level of PSSP for node 1 with the command:

   ```
   spbootins -p PSSP 2.2 -s no 1 1 1
   ```

4. Change the install image name and define the LPP source name:

   ```
   spbootins -i bos.obj.ssp.420 -v aix42 -s no -l 1
   ```

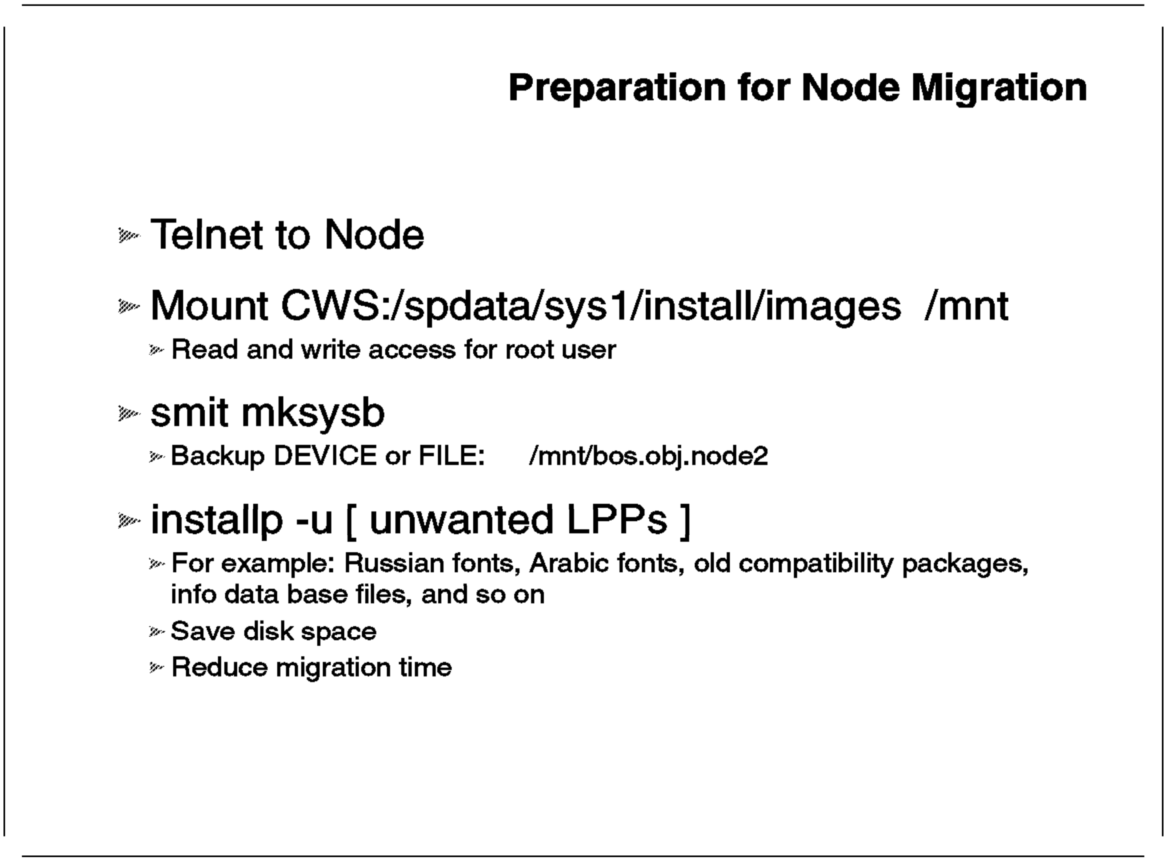
   The spimg feature of PSSP 2.2 contains the bos.obj.ssp.420 system backup image. You might need a PTF for spimg to receive the minimal AIX 4.2 image.
5. Set the response from server to bootp request to “install.”
6. Verify the SDR data with the `splstdata` command.
7. Perform `setup_server` to create the NIM definitions and to allocate resources.
8. Use Perspectives or SPmon to net boot node 1.

After 35 minutes to 40 minutes, the overwrite installation is finished and you can start working on this node.

5.6 Migration to AIX 4.2

This section describes the preparation and installation steps required to migrate an RS/6000 SP node to AIX 4.2.

5.6.1 Preparation for Node Migration

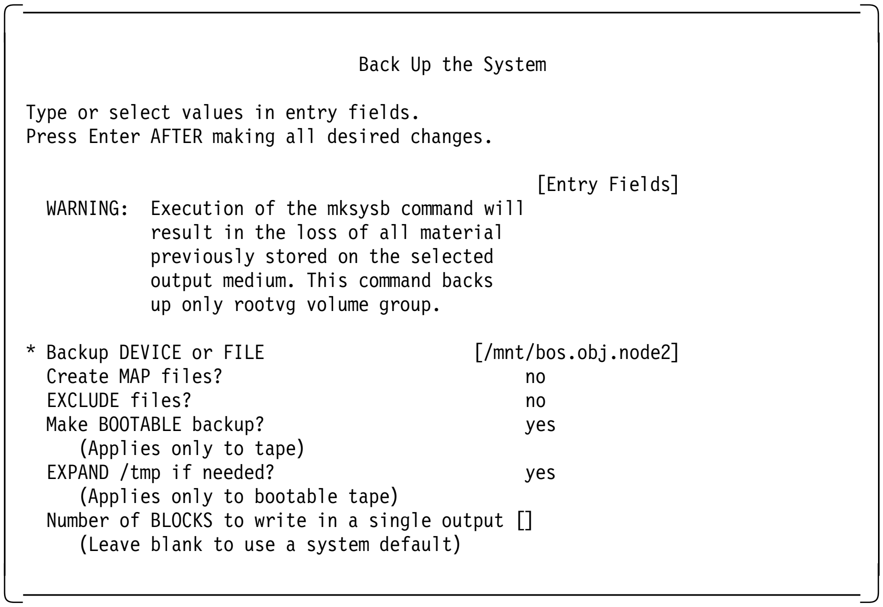


Preparation for Node Migration

- Telnet to Node
- Mount CWS:/spdata/sys1/install/images /mnt
  - Read and write access for root user
- smit mksysb
  - Backup DEVICE or FILE: /mnt/bos.obj.node2
- installp -u [unwanted LPPs]
  - For example: Russian fonts, Arabic fonts, old compatibility packages, info data base files, and so on
  - Save disk space
  - Reduce migration time

Before migrating your RS/6000 SP nodes, perform the following steps:

1. Export the /spdata/sys1/install/images directory for the node that will be migrated. Verify that the root user has read/write permissions for that directory. You can use the `smit _nfs` command to export that directory for the node, or you can modify the /etc/exports file. Verify the exported directories with the `exportfs -va` command.
2. Telnet to the node.
3. Mount the /spdata/sys1/install/images directory from the Control Workstation over the directory /mnt.
4. Use the `smit mksysb` command to perform the system backup. Set the backup name so that the backup will be performed over the network onto the Control Workstation.



```
                              Back Up the System

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

                                                  [Entry Fields]
  WARNING:  Execution of the mksysb command will
            result in the loss of all material
            previously stored on the selected
            output medium. This command backs
            up only rootvg volume group.

* Backup DEVICE or FILE                     [/mnt/bos.obj.node2]
  Create MAP files?                              no
  EXCLUDE files?                                 no
  Make BOOTABLE backup?                          yes
     (Applies only to tape)
  EXPAND /tmp if needed?                         yes
     (Applies only to bootable tape)
  Number of BLOCKS to write in a single output []
     (Leave blank to use a system default)
```

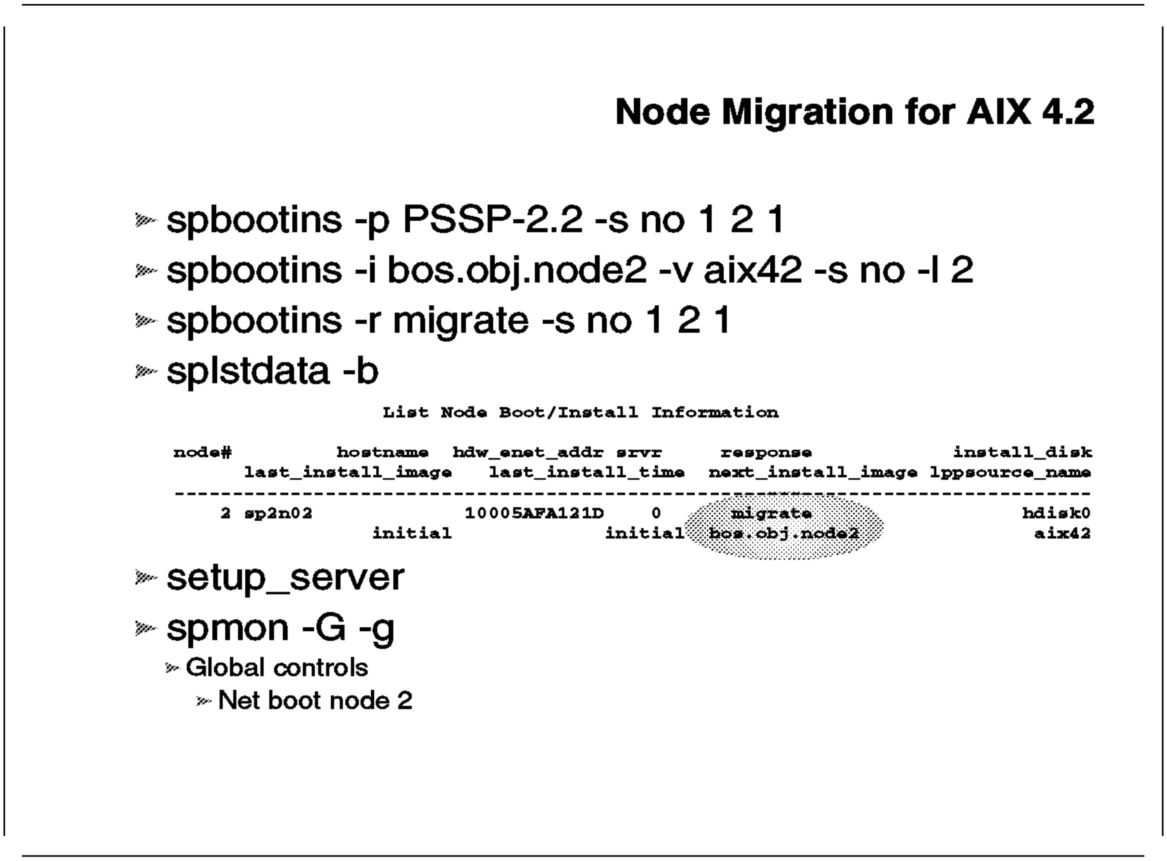
5. Before starting the migration, perform some cleanup on the node. Remove software packages that you will not need anymore. Uninstall software that is not required on your AIX 4.2 system, such as:

   - Old compatibility packages
   - Unused foreign languages
   - Exotic fonts
   - Device drivers

   The uninstall will save you disk space on the AIX 4.2 system and reduce the migration time.

6. Now the node is ready for the migration to AIX 4.2.

5.6.2 Node Migration for AIX 4.2



Node Migration for AIX 4.2

- spbootins -p PSSP-2.2 -s no 1 2 1
- spbootins -i bos.obj.node2 -v aix42 -s no -l 2
- spbootins -r migrate -s no 1 2 1
- splstdata -b

```
                    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
-------------------------------------------------------------------------------
    2 sp2n02              10005AFA121D    0       migrate                 hdisk0
                initial              initial  bos.obj.node2                aix42
```

- setup_server
- spmon -G -g
  - Global controls
    - Net boot node 2

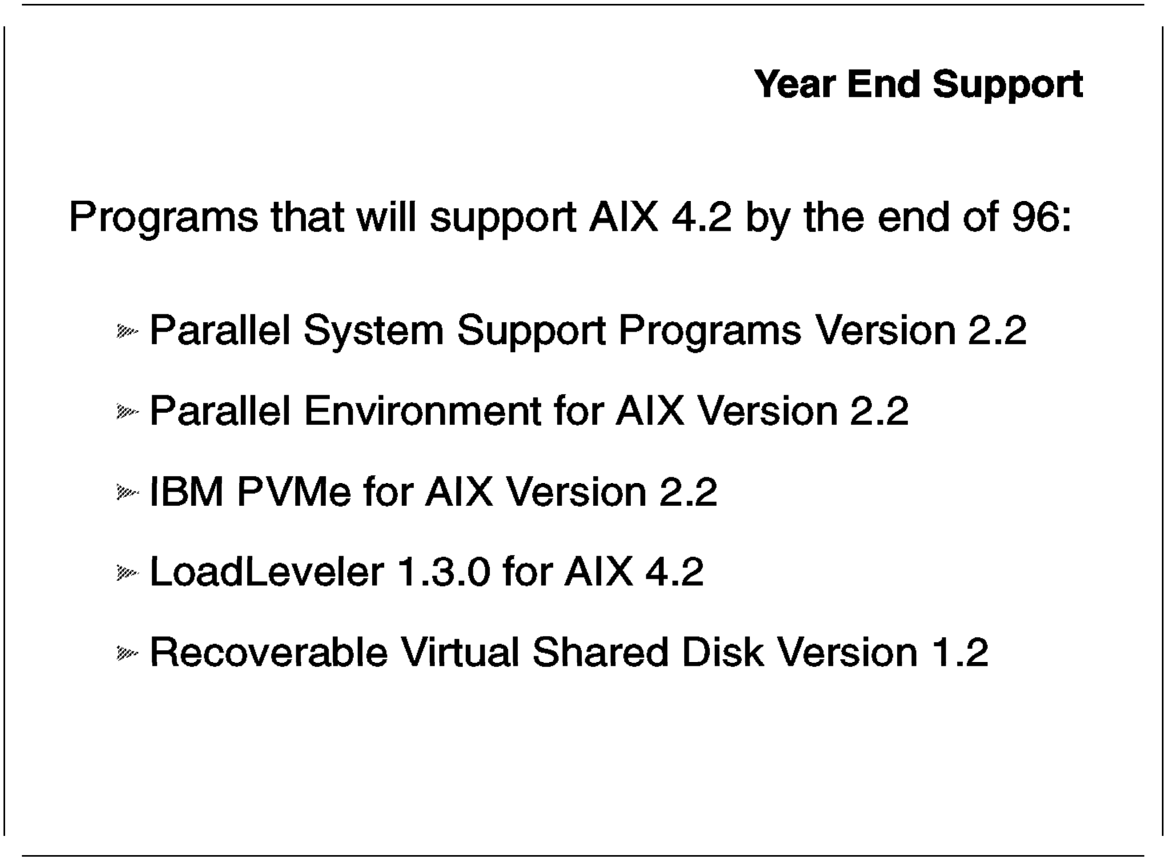
This section describes which steps are needed to migrate an RS/6000 SP node to AIX 4.2.

Following are the migration steps for node 2:

1. Verify that all AIX 4.2 images for NIM are installed in the directory /usr/spdata/sys1/install/aix42/lppsource.
2. Use the `spbootins -p PSSP-2.2` command to change the level of PSSP to 2.2.
3. Change the LPP source name with the “-v” parameter to *aix42* and specify your node system backup as network install image.
4. Specify the boot/install server′s response to the bootp request from the nodes to “migrate.”
5. Verify the SDR with the `splstdata -b` command.
6. Start the `setup_server` script to create the Network Installation Management (NIM) definitions for the node.
7. Net boot the node with SPmon or Perspectives.

After 45 minutes to 60 minutes, the installation will finish and you can restart the switch with `Estart`.

5.7 Year End Support



Year End Support

Programs that will support AIX 4.2 by the end of 96:

- Parallel System Support Programs Version 2.2
- Parallel Environment for AIX Version 2.2
- IBM PVMe for AIX Version 2.2
- LoadLeveler 1.3.0 for AIX 4.2
- Recoverable Virtual Shared Disk Version 1.2

This section describes which programs will be supported on AIX 4.2 by the end of 1996.

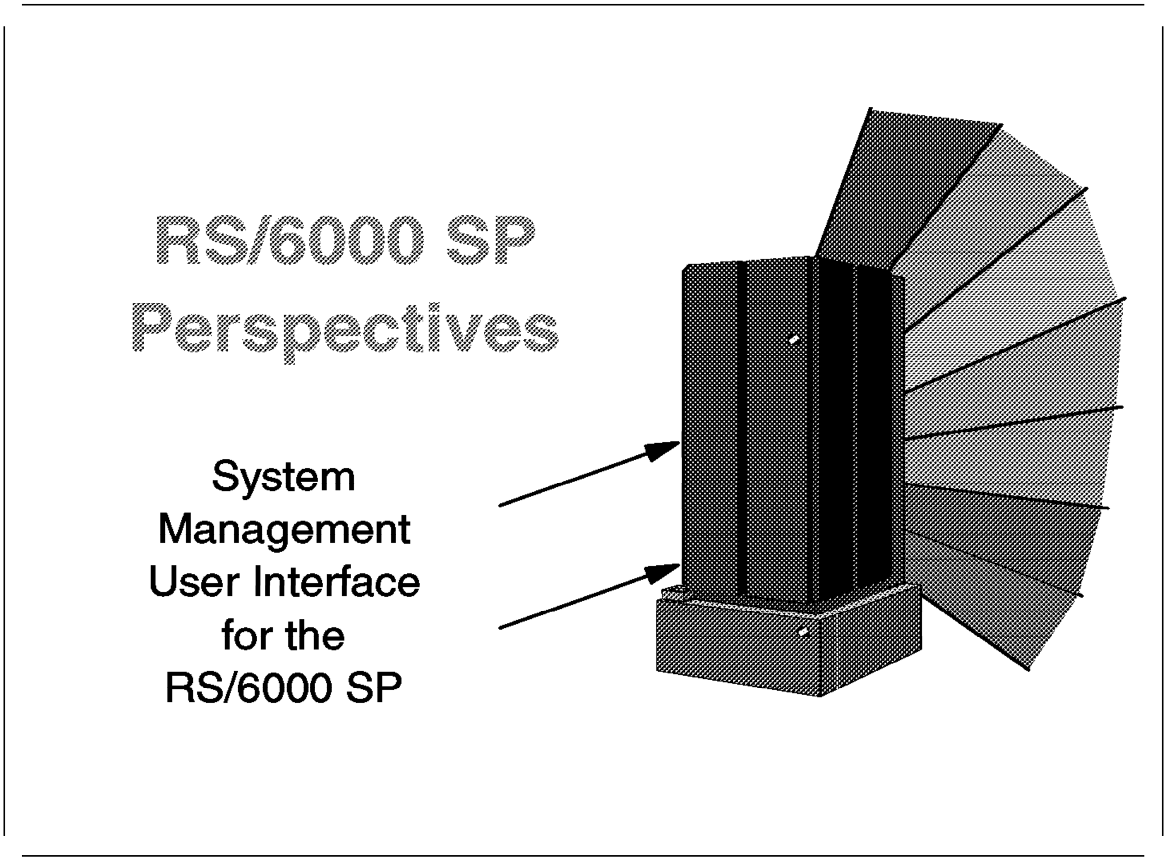
Parallel System Support Programs (PSSP) version 2.2, Recoverable Virtual Shared Disk (RVSD) version 1.2, Parallel Environment (PE) version 2.2, and PVMe version 2.2.

LoadLeveler version 1.3 will support by the end of 1996:

- IBM AIX 4.2 Operating System in binary compatibility mode
- Sun SPARCstation Systems with the Solaris Operating System
- Silicon Graphics Systems (SGI)
- Hewlett-Packard Systems (HP)

More information can be found at the IBM RS/6000 technology home page: `http://www.rs6000.ibm.com/tech`.

Chapter 6. Perspectives



RS/6000 SP Perspectives

System Management User Interface for the RS/6000 SP

RS/6000 SP system administrators are now used to the Control Workstation being the single point of control for the entire system.

System management tasks are performed on the RS/6000 SP from the Control Workstation in three different ways:

- Typing commands in shell sessions
- Navigating through the text menus of the Integrated System Management Tool (SMIT)
- Opening windows and clicking options in the RS/6000 SP graphical monitoring and control tool (SPmon)

There is no real comparison within these facilities, but usually gaining in user friendliness makes you lose functions. This is especially true at the SPmon level, as this tool covers only a small subset of system management tasks.

At the same time, conventional AIX system management tools, such as Visual System Management (VSM), constantly improve the comfort provided to the system administrator in his daily tasks.

The RS/6000 SP Perspectives initiative aims at extending this VSM concept to the entire RS/6000 SP system management, while providing a single, consistent view (or perspective) into the system.

6.1 Introduction to Perspectives

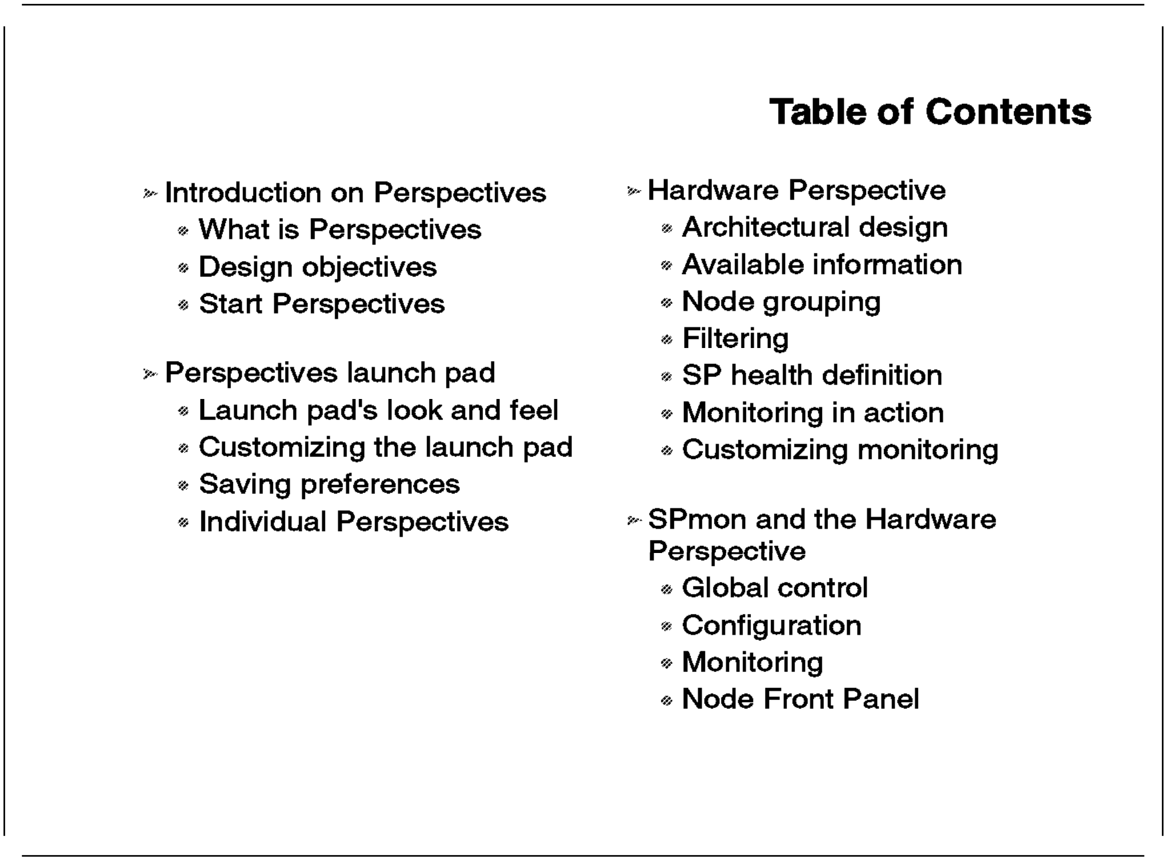


Table of Contents

- Introduction on Perspectives
  - What is Perspectives
  - Design objectives
  - Start Perspectives
- Perspectives launch pad
  - Launch pad's look and feel
  - Customizing the launch pad
  - Saving preferences
  - Individual Perspectives
- Hardware Perspective
  - Architectural design
  - Available information
  - Node grouping
  - Filtering
  - SP health definition
  - Monitoring in action
  - Customizing monitoring
- SPmon and the Hardware Perspective
  - Global control
  - Configuration
  - Monitoring
  - Node Front Panel

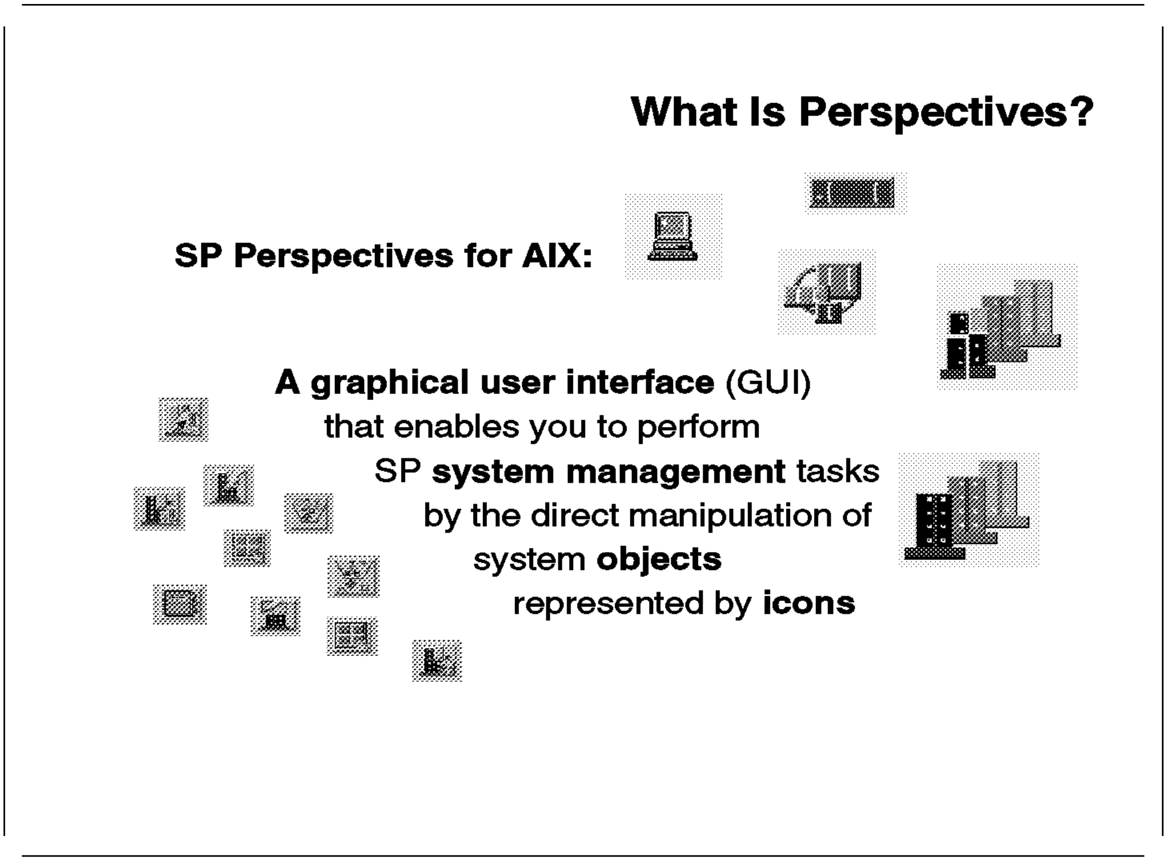
The first section details the objectives that were taken into account in designing Perspectives and are included in its current implementation.

The second section describes the look and feel of the two main elements of the Perspectives GUI: the launch pad and the individual Perspective concept. Relations between the two elements are detailed through the customization of the launch pad in order to expand the number of individual Perspectives and applications.

The third section focuses on one of the individual Perspectives: the Hardware Perspective. Architectural, graphical, and functional concepts are described. It gives us the opportunity to develop new concepts brought by PSSP Version 2 Release 2, such as node grouping.

The fourth section aims at answering user questions about the part played by SPmon within PSSP Version 2 Release 2 and its future releases. Analogies between SPmon and the Hardware Perspective are given for most of the usual RS/6000 SP monitoring and control tasks.

6.1.1 What Is Perspectives?



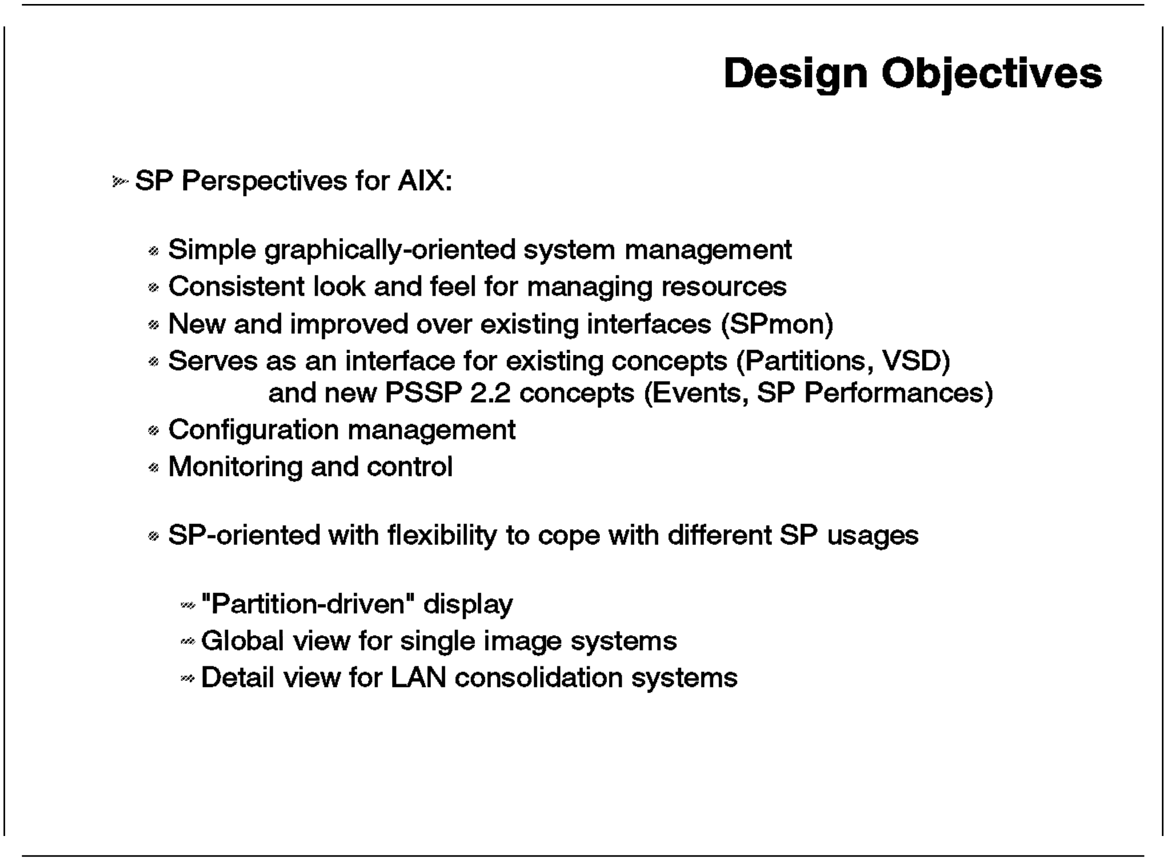
What Is Perspectives?

SP Perspectives for AIX:

A graphical user interface (GUI)
that enables you to perform
SP **system management** tasks
by the direct manipulation of
system **objects**
represented by **icons**

You simply select an RS/6000 SP system object by clicking it with a mouse, and then select an action to perform on that object from the menu bar or tool bar. Examples of visual objects are shown on the right of the foil, and examples of icons are shown on the left of the foil.

6.1.2 Design Objectives



Design Objectives

- SP Perspectives for AIX:
  - Simple graphically-oriented system management
  - Consistent look and feel for managing resources
  - New and improved over existing interfaces (SPmon)
  - Serves as an interface for existing concepts (Partitions, VSD) and new PSSP 2.2 concepts (Events, SP Performances)
  - Configuration management
  - Monitoring and control
  - SP-oriented with flexibility to cope with different SP usages
    - "Partition-driven" display
    - Global view for single image systems
    - Detail view for LAN consolidation systems

The main design objective for Perspectives is to improve the ease of using system management on the RS/6000 SP system. This leads to side objectives:

Easy access

Use of graphical interface with usual mouse controls and meaningful representations.

Consistency

Same look and feel for each individual Perspective, no matter what objects are being manipulated, and a consistent representation of the same objects in the different Perspectives.

Completeness

Application of the Perspectives concepts to any managed object on the RS/6000 SP system, reinforcing the single point of control philosophy.

Suitability

Complete coverage of the system management tasks, from the configuration to the monitoring and control tasks.

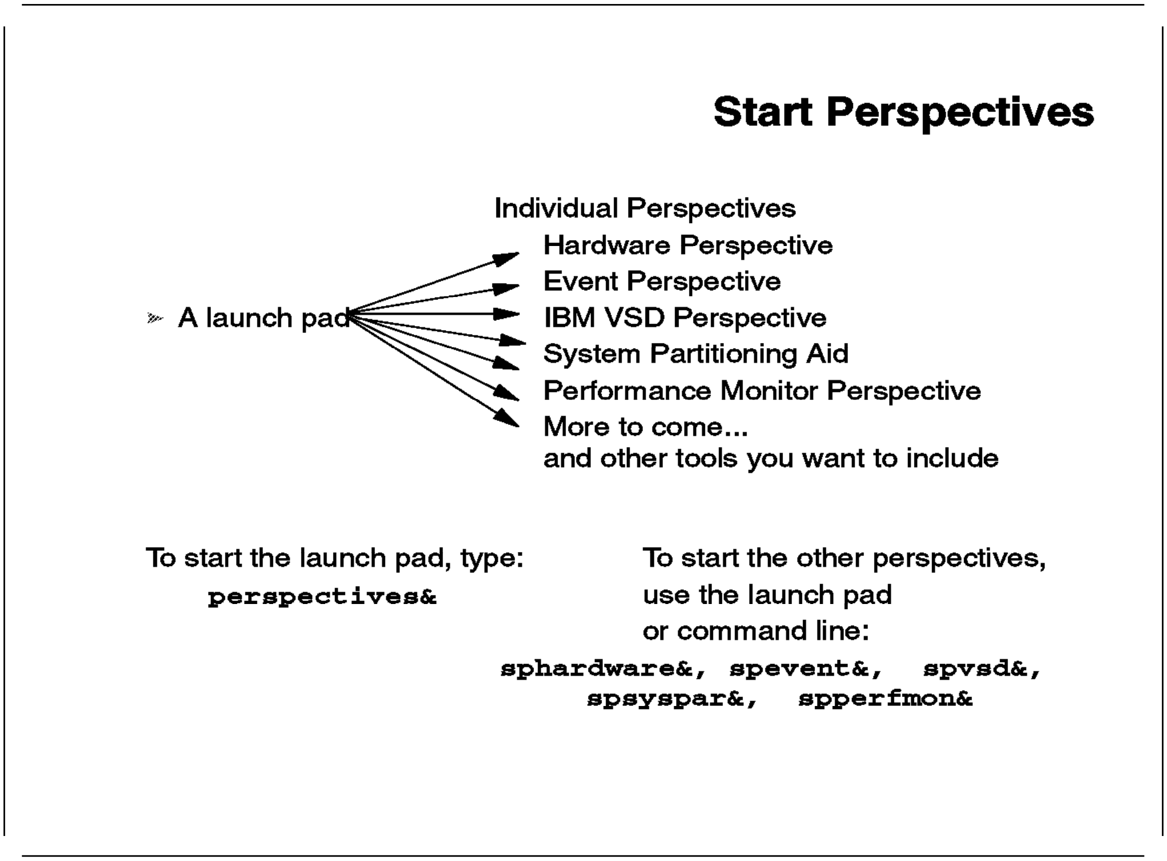
Flexibility

Conform to various utilities. Give the opportunity to massively parallel computing users to see a single image system and manage it at the global system level. Introduce mechanisms for the server consolidation users to be able to differentiate the types of servers they are using and manage them at the node level.

For these long term objectives, the first release of Perspectives delivered with PSSP Version 2 Release 2 has its emphasis on control and monitoring over configuration. Installation of the system and the application of a given partitioning layout are still to be performed using either SMIT or the command line interface. But Perspectives can now be used to perform the following system management tasks:

- Monitor and control hardware
- Create and monitor system events
- Define and manage IBM Virtual Shared Disks
- Generate and save system partition configurations
- Monitor system performances

6.1.3 Start Perspectives



Start Perspectives

Individual Perspectives

- A launch pad
  - Hardware Perspective
  - Event Perspective
  - IBM VSD Perspective
  - System Partitioning Aid
  - Performance Monitor Perspective
  - More to come...
    and other tools you want to include

To start the launch pad, type:

```
perspectives&
```

To start the other perspectives, use the launch pad or command line:

```
sphardware&,  spevent&,  spvsd&,
spsyspar&,  spperfmon&
```

Perspectives is made of two elements:

1. A launch pad tool detailed in the next section. It can be started with the `perspectives&` command. The launch pad is included in the ssp.gui fileset.
2. Individual Perspective applications, also called *perspectives*. These perspectives are included either in the ssp.gui fileset for applications concerning other basic ssp filesets (partitioning tool, hardware monitoring, event management), or in specific filesets of mechanisms they are interfacing. The VSD perspective is thus in ssp.csd.gui and the Performance Monitor perspective is in ptpe.gui. Each perspective can be started independently of the launch pad with its respective command. Some are mentioned on this foil.

Future releases are likely to bring more perspectives on uncovered aspects of system management.

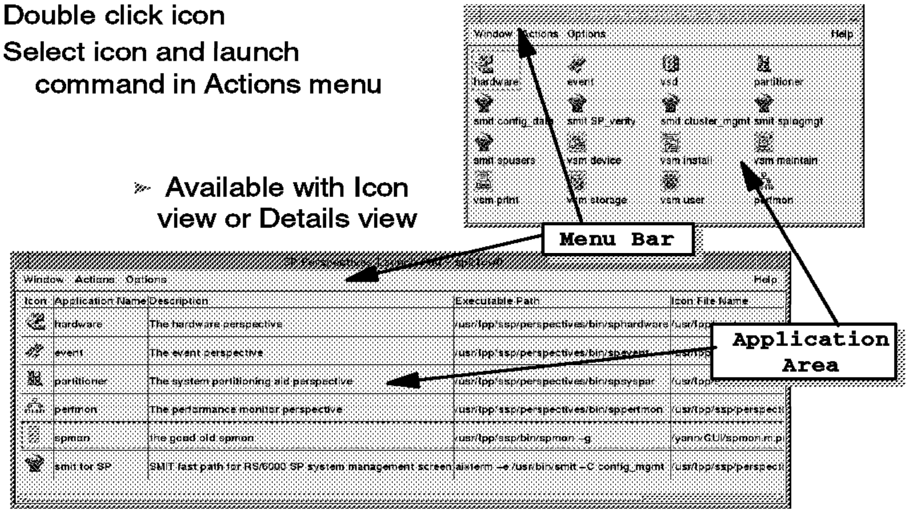
The launch pad can also be customized to include any program the administrator would like to be accessible through this centralized interface. Customization of the launch pad is described in 6.2.1, “Customizing the Launch Pad” on page 213.

6.2 Perspectives Launch Pad

Perspectives Launch Pad

- To launch an application
- Double click icon
- Select icon and launch command in Actions menu

- Available with Icon view or Details view



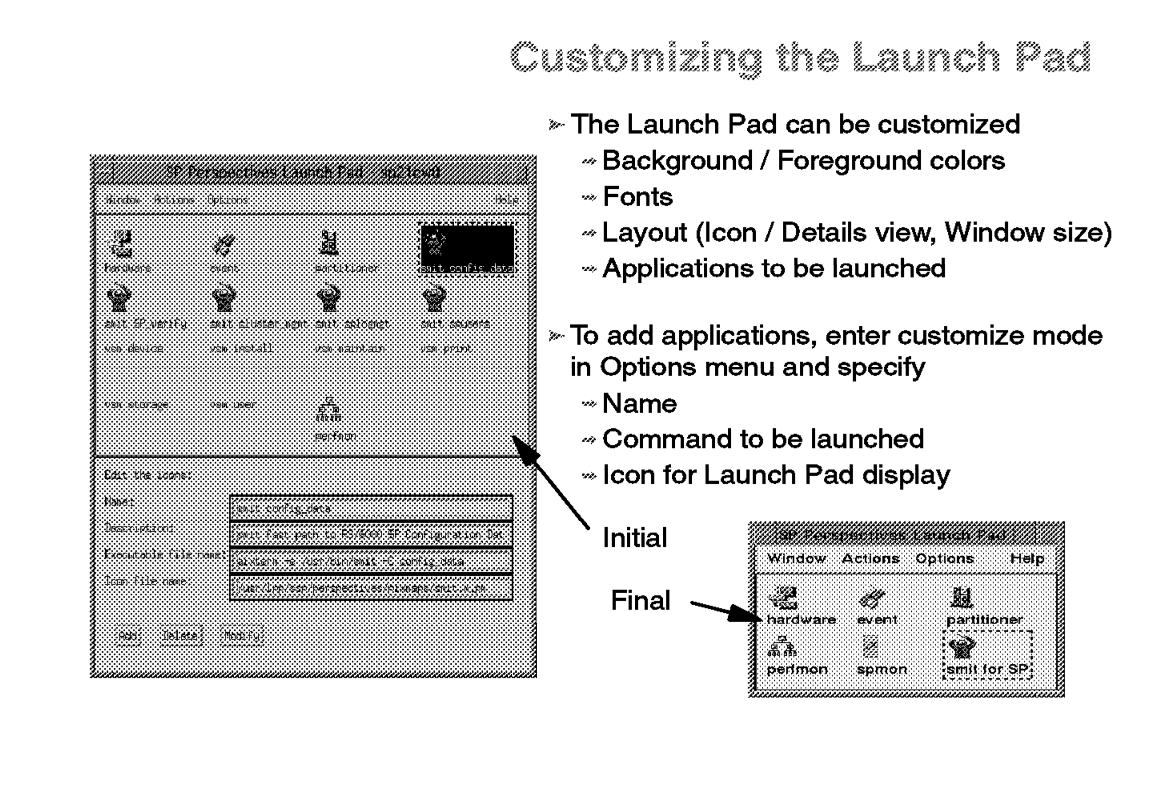
Icon	Application Name	Description	Executable Path	Icon File Name
	hardware	The hardware perspective	/usr/lpp/ssp/perspectives/bin/sphardware	/usr/lpp
	event	The event perspective	/usr/lpp/ssp/perspectives/bin/spevent	/usr/lpp
	partitioner	The system partitioning aid perspective	/usr/lpp/ssp/perspectives/bin/spsyspar	/usr/lpp/
	perfmon	The performance monitor perspective	/usr/lpp/ssp/perspectives/bin/spperfmon	/usr/lpp/ssp/perspect
	spmon	the good old spmon	/usr/lpp/ssp/bin/spmon -g	/yann/GUI/spmon.m.p
	smit for SP	SMIT fast path for RS/6000 SP system management screen	aixterm -e /usr/bin/smit -C config_mgmt	/usr/lpp/ssp/perspect

The launch pad icon view presented on this foil is the default view the user gets when launching Perspectives without any customization.

It already includes icons of any available individual perspective. Each time a new perspective fileset is installed on the system, the corresponding entry is added to the launch pad. See 6.1.3, “Start Perspectives” on page 211 for more information on filesets containing individual perspectives.

The launch pad also includes default entries to RS/6000 SP menus of the SMIT and VSM tools. SPmon is not included by default, so on the next page, we add it to the launch pad as an example of customizing Perspectives.

6.2.1 Customizing the Launch Pad



Customizing the Launch Pad

- The Launch Pad can be customized
  - Background / Foreground colors
  - Fonts
  - Layout (Icon / Details view, Window size)
  - Applications to be launched
- To add applications, enter customize mode in Options menu and specify
  - Name
  - Command to be launched
  - Icon for Launch Pad display

Edit the icons:

Name: smit config_data

Description: smit fast path to RS/6000 SP Configuration Dat

Executable file name: aixterm -e /usr/bin/smit -C config_data

Icon file name: /usr/lpp/ssp/perspectives/pixmaps/smit.m.pm

In this example, the customization of the launch pad consists of:

1. Removing unused icons
2. Adding SPmon
3. Adding LoadLeveler
4. Adding SMIT for RS/6000 SP panel

We used the following process:

1. In the Options menu of the launch pad menu bar, choose the **Customize Applications** item. This makes the launch pad window expand with editing information. By selecting an icon, the related information is displayed in this new area. Clicking **Delete** removes the current icon from the launch pad. This is how we remove the icons we do not want to keep.

2. Clicking an area where no icon is shown in the launch pad makes the editing fields become blank. This fields are now ready for receiving information about your new applications. This is what we use for adding the SPmon entry:

   - With your favorite icon editor (we use the Desktop one located in /usr/dt/bin under the name dticon), draw an icon that reminds you of SPmon. You can store it anywhere on the system, but other Perspectives icons are in /usr/lpp/ssp/perspectives/pixmaps. We save it under the spmon.m.pm name.

- In the editing area of the Perspectives launch pad, enter the following information in the corresponding fields:

  Name: SPmon

  Description: This is the icon for SPmon.

  Executable file name: /usr/lpp/ssp/bin/spmon -g

  Icon file name: /usr/lpp/ssp/perspectives/pixmaps/spmon.m.pm

- Click the **Add** button, and the new application appears in the launch pad.

3. To add the LoadLeveler Job management user interface, follow the same process with the following information. ll.m.pm should be available on your system in /usr/lpp/LoadL/nfs/samples. You have to move it into the Perspectives pixmaps directory. The xloadl executable is in /usr/lpp/LoadL/nfs/bin and should have a link in the loadl administrator directory. Within the launch pad, you can call the initial program or the link.

   Name: LoadLeveler

   Description: xloadl interface for job management with LoadLeveler

   Executable file name: /home/loadl/bin/xloadl

   Icon file name: /usr/lpp/ssp/perspectives/pixmaps/ll.m.pm

4. To add the SMIT RS/6000 SP System management panel, follow the same process with the following information:

   Name: SMIT for SP

   Description: SMIT fast path for RS/6000 SP system management

   Executable file name: aixterm -e /usr/bin/smit -C config_mgmt

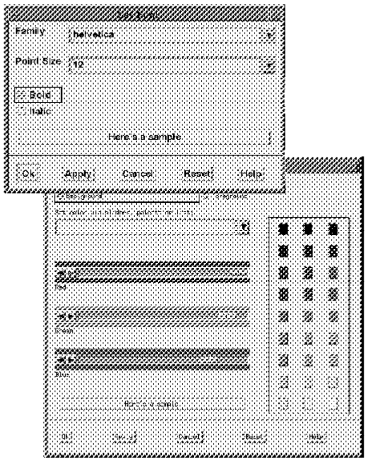
   Icon file name: /usr/lpp/ssp/perspectives/pixmaps/smit.m.pm

5. Close the editing area of the launch pad by clicking **Leave Customize Mode** in the Options menu of the menu bar.

6.2.2 Saving Preferences

Saving Preferences

- Launch pad and individual perspectives common preferences:
  - Colors, fonts
  - Layout: window size, displayed objects
- Launch pad specific preference:
  - Applications
- Preference can be saved
  and retrieved separately
- Preferences can be saved in profile files:
  - A different file for each individual perspective
  - Several possible profiles for each perspective
  - System profiles: Perspectives directory
  - User profiles: user directory
  - "Profile" used by default at application start



Family helvetica

Point Size 12

Bold

Italic

Here's a sample

Once customization has been performed on the Perspectives launch pad or an individual perspective layout, these modifications can be saved in *preferences files*. Common preferences are color, font, and layout. The launch pad has a specific preference that can be saved: the applications included in the application area.

Preferences can be saved and retrieved separately. For example, you may save only the font type for a given perspective. You could also save all preferences for a given perspective, and then retrieve only the color.

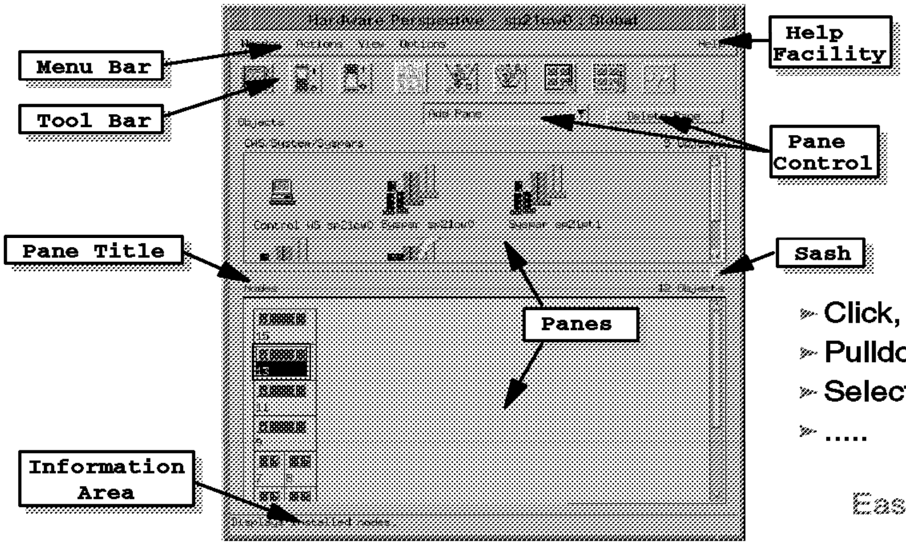
Preferences are saved for each perspective, including the launch pad, in *User* or *System* files. The User files are located within the user Home directory. The System files are located in the /usr/lpp/ssp/perspectives/profiles directory. Any number of profiles of each type can be generated. A particular system profile, named *Profile*, is used by default when launching a particular tool.

The naming convention is the concatenation of a period, the name of the executable that launches the perspective, and the profile name (*.perspective_launching_commandprofile_name*).

For example, in the case of the launch pad, the my_profile file name is .perspectivesmy_profile. In the case of the Hardware Perspectives, it is .sphardwaremy_profile.

6.2.3 Individual Perspectives User Interface

Perspectives User Interface



- Click,
- Pulldown
- Select
- .....

Easy!

This foil presents the various graphical elements of an individual perspective.

The basic entity manipulated by Perspectives is called an object. An object is an instance of a given class. Examples of classes are node, VSD node, VSD, partition, and system. The classes manipulated by Perspectives do not match the classes of the SDR, even if some concepts lead to classes in both the SDR and Perspectives.

The objects manipulated are shown in panes. Selecting a pane gives access to a specific action menu. Several classes sharing the same pane means that they have a common behavior for the administrator.

Performing an action on an object in a pane sometimes modifies the objects displayed in the other panes. In this case, the class that instance is selecting in the first pane *drives* the use of the tool. For example, this is the case of the partition and system classes in the use of the Hardware Perspective, as described in 6.3.3, “Partition Driven” on page 225.

Panes can be added or hidden from the main window using the pane control area. Space occupied by the panes in the display can be customized using the *sash*.

Manipulation of an individual perspective consists of:

1. Selecting an object or a set of objects in one pane

2. Applying an action to them through an icon of the tool bar or a menu entry in the menu bar.

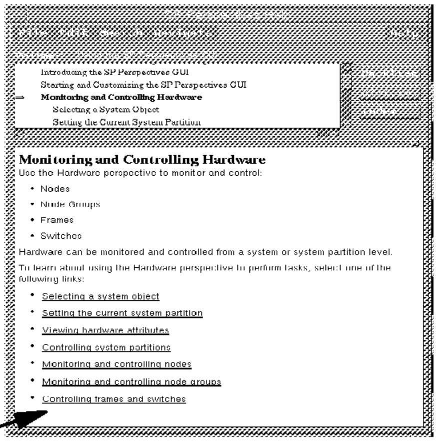
Graphical information regarding the individual perspectives is saved as part of the layout preference. It includes the following parameters:

- Global window size
- Global window position
- Displayed panes and other maskable areas (for example, information)
- For each pane:
  - Percentage of the main window occupied
  - Object labels
  - Sort means
  - Filtering
  - Specific information for the perspective (monitored conditions in the case of the hardware or VSD perspectives)

6.3 Hardware Perspective

Hardware Perspective

- The perspective that gives you access to the hardware of your SP for system management tasks
  - View configuration
  - Monitoring and Control
- Interaction with the system at various levels
  - Frame basis
  - Partition basis
  - Node basis
  - New node group basis
- Mouse control; select and run an action with a click
- Extensive online help !



Introducing the SP Perspectives GUI
Starting and Customizing the SP Perspectives GUI
⇒ **Monitoring and Controlling Hardware**
Selecting a System Object
Setting the Current System Partition

Monitoring and Controlling Hardware

Use the Hardware perspective to monitor and control:

- Nodes
- Node Groups
- Frames
- Switches

Hardware can be monitored and controlled from a system or system partition level.

To learn about using the Hardware perspective to perform tasks, select one of the following links:

- Selecting a system object
- Setting the current system partition
- Viewing hardware attributes
- Controlling system partitions
- Monitoring and controlling nodes
- Monitoring and controlling node groups
- Controlling frames and switches

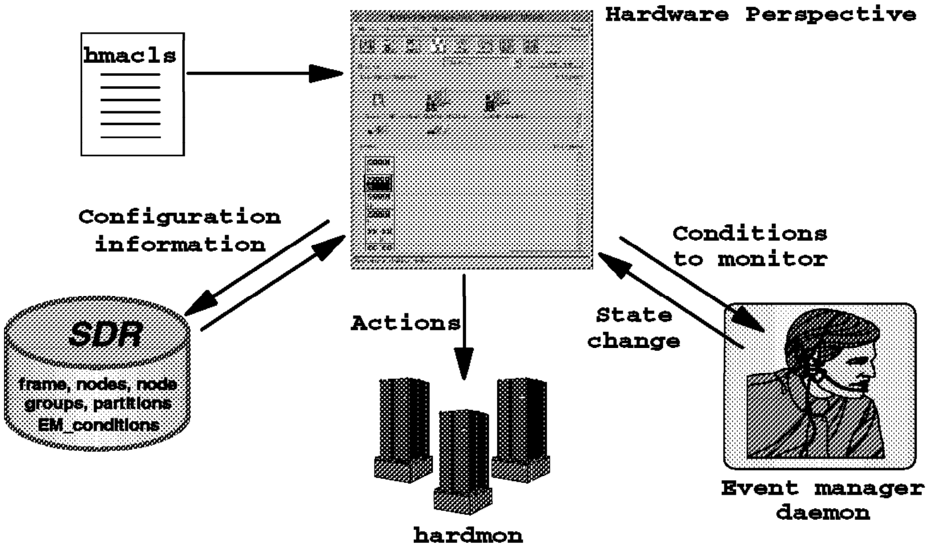
The Hardware Perspective is one of the individual perspectives provided with PSSP Version 2 Release 2. Its scope encompasses the functions of the former SPmon tool.

Thanks to SPmon, hardware monitoring and control is an area where RS/6000 SP administrators are familiar with graphical user interfaces. The Hardware Perspective improves the automation of system management tasks in different ways:

- By providing more levels of interaction with the system. To the conventional node and partition levels, this perspective adds the frame level and a new concept named node group. For the first time, the Control Workstation is also part of the monitored cluster.
- By extending the set of automated actions on RS/6000 SP resources. Running parallel commands and launching the switch are now possible from Perspectives.
- By improving the modularity of system management tasks. Individually monitoring each parameter is no longer necessary, as groups of conditions are available to define the health of the RS/6000 SP and monitor it.
- By simplifying the desktop management. All information is now available in the same window, and extensive help is provided either interactively in the information area or by request through the Help menu.

6.3.1 Architectural Design

Architectural Design



This section describes the interactions of the Hardware Perspective with the other PSSP components. Interactions are for information queries as well as action triggers. Therefore, this section discusses security and rights when interacting with the system.

6.3.1.1 Interaction with Other Components

The three components you interact with when using the Hardware Perspectives are the SDR, the hardmon daemon, and the event manager daemon. These interactions are described below:

- **The SDR**

  When the Hardware Perspective starts, it reads the SP configuration from the SDR. It constructs the display of the initial panes using this information. Each time a pane is redisplayed after being deleted, another query for information is made to the SDR. Simply activating a displayed pane by clicking its contents does not query the SDR. Consequently, for modifications made to the SDR to be taken into account by the Perspectives′ display, you have to either close and restart the tool or delete and redisplay the panes.

  For node group management, Perspectives can write to the SDR. The Hardware Perspective allows the creation of instances of object classes that are permanently stored in the SDR. Node groups are described in 6.3.5, “Node Group Creation” on page 228. Declaring an object in Perspectives automatically creates its instance in the SDR when it needs to be permanently stored.

- **The hardmon daemon**

  To perform an action on an object, Perspectives interacts with the hardmon daemon. For example, this is the case when powering on a node or turning a key. The hardmon daemon is thus simply receiving orders from the Perspective GUI. Unlike the former versions of PSSP, it does not provide information about hardware state to the GUI. This information goes through the event manager (but the hardmon daemon is still providing the event manager daemon with information concerning the hardware).

  There is an exception to that rule: the 3DigitDisplay window is still provided by the hardmon.

- **The event manager daemon**

  Perspectives is an event manager client in two ways:

  1. On a request basis, when a list of resource variables and associated values is requested by the user. This list is not constantly refreshed, but clicking one element of the list refreshes the value of that element. The query is performed at the user′s request.

  2. On a regular basis, when the **node status** tab of the notebook is displayed or when the monitoring of conditions is activated on a given object class. The session with the event manager daemon is established for the monitoring through the EMAPI at the launching of the tool. When monitoring begins, Perspectives registers for the events that are linked to the monitored condition and queries for the initial value of the condition.

  When an action is performed on an object, the Hardware Perspective is communicating with the hardmon for the action to be done. If the result is monitored, the state change comes through the event manager daemon.

6.3.1.2 Trouble Shooting

If you start a Hardware Perspective session, a session is established with the event manager daemon for future requests. When the event manager daemon is stopped, a message appears in Perspective:

```
Lost the connection to the event manager.  Monitoring will not be
available for the remainder of this Perspectives session.
```

Even if the event manager daemon is restarted, the connection is not re-established. The monitoring interface is still available, but monitoring requests fail, with resources in an unknown state.

If you start a Hardware Perspective session, and the event manager is not running, the initial session is not established between the two applications. When the event manager starts, the session is established. This means that Perspectives can recover from an initial lack of event manager but not from a failure of event manager.

If the hardmon fails, no more monitoring or actions are possible on the hardware. This does not result directly from the hardmon failure, but from the fact that the event manager cannot get the hardware information anymore. In this case, conditions in node status turn to unknown. When the hardmon starts again, there is a delay of a few seconds for event manager internal synchronization, and then the information is automatically restored in the Perspective monitoring displays.

If the sdr daemon is reinitialized during a Perspective session, Perspective loses communication with the SDR for a while. During this time, the activation of monitoring is impossible, as monitored conditions cannot be accessed in the SDR.

6.3.1.3 Authorizing Users for the Hardware Perspective

As with the former versions of PSSP, access to the hardware through the hardmon daemon is still ruled by the hmacls file. This file has been in /spdata/sys1/spmon/ since PSSP version 2.1 availability. This file authorizes root.admin automatically. If a non-root user needs to interface with the hardware, the appropriate ACLs have to be added to that file. Then the client must have a Kerberos token that has the desired hardmon authority to control the RS/6000 SP hardware.

In PSSP Version 2 Release 2, there are five commands with a specific authorization:

- `cshutdown`
- `cstartup`
- `Efence`
- `Estart`
- `Eunfence`

These commands are `sysctl` commands ruled by the /etc/sysctl.rootcmds.acl ACL file. This means that a non-root user with the appropriate ACLs in the former file can issue these commands to the nodes from the Control Workstation. To do so, the user needs a valid ticket for rcmd, as sysctl is based on Kerberos on the RS/6000 SP. The user also needs to have authority for hardmon for the frames the command is issued for.

These various authorization levels lead to the following cases when using Perspectives:

1. ***You are a non-root user with no special authorization.***

   You can launch the Hardware Perspective and you get the following message:

   ```
   Attention: You only have read access to the SDR.  Some operations
   will not be available.
   ```

   All information are displayed in the panes, but icons and menus are only available for viewing objects or running commands. If you do not have the authorization, you cannot run commands through the graphical interface. You can activate any monitoring on any object. The only information you cannot obtain is the 3DigitDisplay that is still coming directly from the hardmon.

   So, unlike SPmon, which could not even get started without having a Kerberos ticket, the Hardware Perspective can be used by any user.

2. ***You are a non-root user with the following conditions:***

   - ***You are kerberized with no instance (that means out of the admin instance).***
   - ***You have a valid Kerberos ticket.***

- ***You have ACLs in the hmacls file (**a **for the CWS and at least** v **for the frame).***
- ***You have ACLs in the sysctl.rootcmds.acl file.***

You can launch the Hardware Perspective with the SDR read access message. You have access to additional entries in the menus to reach the commands ruled by sysctl.rootcmds.acl. But these commands will fail because you are not part of the admin instance within Kerberos.

3. ***You are a non-root user with the following conditions:***

   - ***You are kerberized with the admin instance.***
   - ***You have a valid Kerberos ticket.***
   - ***You have ACLs in the hmacls file (**a **for the Control Workstation and at least** v **for the frame).***

   You can launch the Hardware Perspective with the SDR read access message. You have access to the normal menus. However, when you start the Power On interface, you will find that you do not have the entries for cstartup. It is the same situation when you start the Power Off interface: you do not have entries for cshutdown and fencing operations. This is because you do not have the right ACLs in the sysctl.rootcmds.acl file.

   You are thus in the situation where you can power off a node but cannot perform a cshutdown on it.

4. ***You are a non-root user with the following conditions:***

   - ***You are kerberized with the admin instance.***
   - ***You have a valid Kerberos ticket.***
   - ***You have ACLs in the hmacls file (**a **for the CWS and at least** v **for the frame).***
   - ***You have ACLs in the sysctl.rootcmds.acl file.***

   You can launch the Hardware Perspective with the SDR read access message. You have access to normal entries in the menus. You can do any monitoring and control your hardmon ACLs allow you, especially if you have *a* for the Control Workstation and *vsm* for the frames. The only restriction comes from the read access on the SDR, which prevents you from creating persistent objects such as node groups. The Create node group entry is not available in the Actions menu.

6.3.1.4 Remote Access to SDR

A function that has been available since the availability of PSSP version 2.1 seems to be totally ignored by administrators: the ability to manage, with certain commands, a remote SDR from another Control Workstation. It is made possible by the partitioning function.

When partitioning a system, the SDR is partitioned. Each partition is independently addressed using a different IP address. This can be the primary address for the Control Workstation network adapter (the adapter having the reliable hostname) or an alias declared on the interface of this adapter. The addressing is performed by setting the SP_NAME variable to the correct address or hostname.

If this SP_NAME variable is set to another IP address of another Control Workstation available on the network, you will be pointed to the SDR of that Control Workstation. Although you only have read access to that SDR, but you will be able to display what you want out of it.

For example, if running in PSSP version 2.1, you can execute the splstdata command on the remote SDR and see the contained information. Executing SPmon is not possible, as an immediate access to the hardmon makes Kerberos refuse the remote connection for authentication reasons. But things are different with the Hardware Perspective, which does not automatically require authentication, as described in 6.3.1.3, “Authorizing Users for the Hardware Perspective” on page 221.

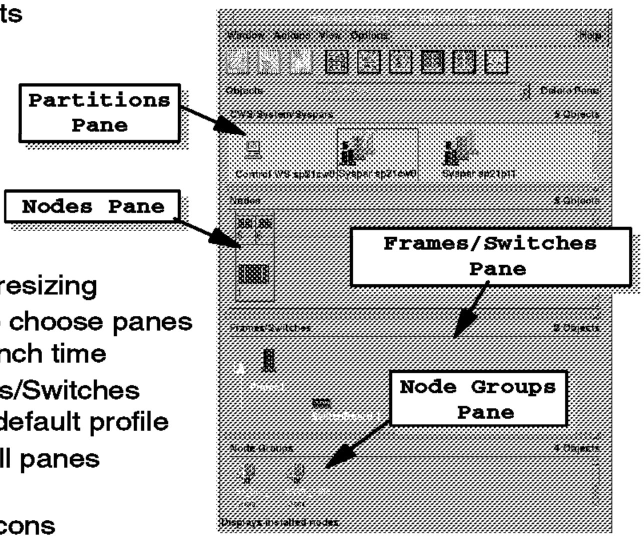
If you set the SP_NAME variable to a remote Control Workstation IP address and launch the Hardware Perspective on your own Control Workstation, the contents of the remote SDR are displayed. You can even monitor conditions of the remote Control Workstation, and thus the remote RS/6000 SP.

Finally, another solution to remotely monitor a distant system is the usual export of the DISPLAY variable. When launching a perspective on a distant system, if access to the local system is denied or the wrong DISPLAY is specified, the process will terminate without errors. To see a trace of this termination, launch the Perspective in background, for example, sphardware&. If something is missing for the execution, when hitting Enter after the process death, you will get: [1] + Done(1) sphardware& .

6.3.2 Available Information

Available Information

- All SP hardware components in one single window
- Frame, Icon, Tree views
- Partition-driven
- Four panes
  - Constant proportions on resizing
  - Use Save Preferences to choose panes to be displayed at launch time
  - Node Groups and Frames/Switches hidden at launch time in default profile
  - Control (add/delete) on all panes
- Actions through menus or icons on active pane



The Hardware Perspective gives access to objects in four different panes:

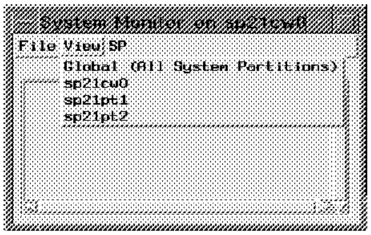
- *The Partitions Pane*, also called the CWS/System/Syspars pane in the tool, hosts three classes of objects: the Control Workstation, the global system, and each defined partition.
- *The Nodes Pane* presents the nodes included in the current partition.
- *The Node Groups Pane* presents the node groups defined in the current partition.
- *The Frames/Switches Pane* presents the frames of the system and optionally included switch boards.

Important: To perform an action on a displayed object, you have to select it. This makes the pane active and opens the entry into the corresponding Actions menu in the menu bar.

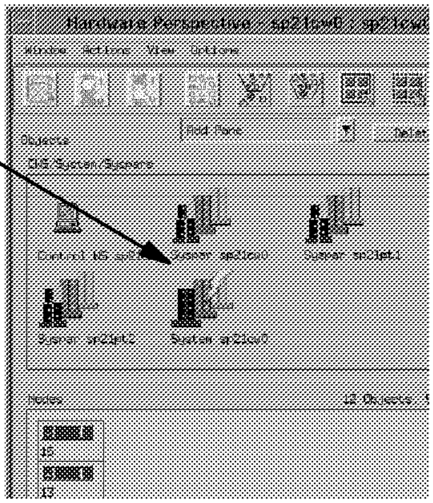
6.3.3 Partition Driven

Partition Driven (SPmon Analogy)

- In Hardware Perspective
  Selection of an object, partition, or system in the Partitions pane, and activation as current partition
- In SPmon
  Choice of a view, global or partition



- In the other windows or panes
  Visualization of objects of the partition



The partitions pane plays a special part in the tool, as it drives the other panes from the displayed object viewpoint. In the partitions pane, *the current partition* is chosen and displayed.

Important: To select the current partition, click one of the system or partition objects displayed in the partition pane and select the Set Current Partition entry of the corresponding Actions menu. A flash is displayed on the top of the icon representing the current partition.

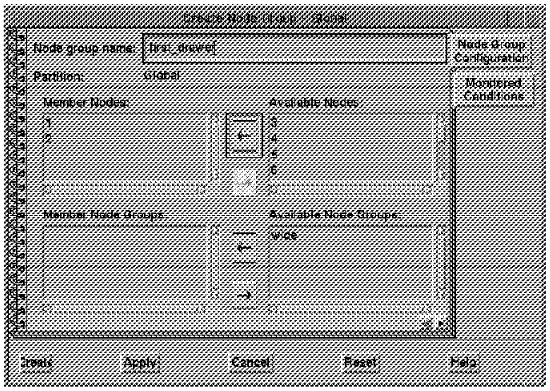
The contents of the nodes and node groups panes are modified, as only elements of the current partition are displayed. The frames/switches pane is unchanged because they are not partition-dependent.

This foil shows the analogy between the new Perspective and the View option available in most SPmon windows. This is for the same purpose of driving the other windows′ displays.

6.3.4 Node Grouping

Node Grouping

- Grouping nodes for task purposes
- Expands Working Collective concept
  - Increased flexibility: several node groups including nodes or other node groups
  - Increased scope: not only for dsh purpose



Create Node Group : Global

Node group name: first_drawer

Partition: Global

Member Nodes:
1
2

Available Nodes:
3
4
5
6

Member Node Groups:

Available Node Groups:
wide

- 2 types of node groups
  - System node groups
  - Partition node groups

The concept of the Node Group was introduced by PSSP 2.2. It allows more flexibility in addressing the nodes of the RS/6000 SP for command execution purposes.

Node grouping is thus an expansion of the Working Collective concept (WCOLL), which was available in the former versions of PSSP (WCOLL is still usable with PSSP 2.2). Node grouping provides:

- **More flexibility**

  Previously, only one working collection file could be pointed to by the WCOLL variable at any time. With node grouping, as many node groups as necessary can be defined. There can be overlap between node groups, and a node group can be defined by specifying constituent nodes or other node groups.

- **A wider scope**

  Previously, the working collective could only be used by the dsh command. Now, node groups can be used by many PSSP commands and PSSP system management tools, as described in 6.3.6, "Use of Node Groups" on page 232.

Two types of node groups can be defined:

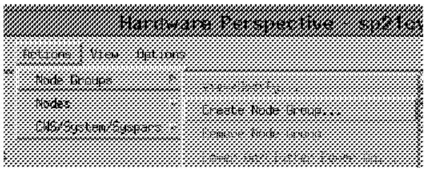
1. System node groups, which can be defined *over* partition boundaries.

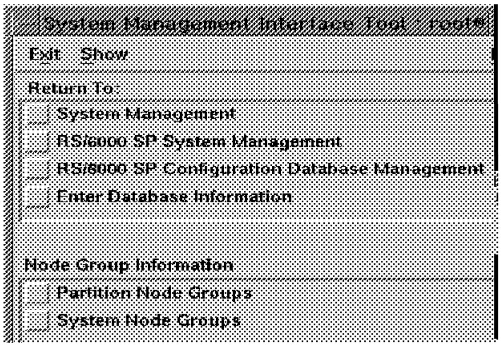
2. Partition node groups, which are defined *within* partition boundaries. A partition node group can only be made with nodes of the corresponding partition.

6.3.5 Node Group Creation

Node Group Creation

- With Hardware Perspective




System Management Interface Tool : root@
Exit
Show
Return To:
System Management
RS/6000 SP System Management
RS/6000 SP Configuration Database Management
Enter Database Information
Node Group Information
Partition Node Groups
System Node Groups

- With SMIT

- Permanently stored in SDR
  - SysNodeGroup class for system node groups
  - NodeGroup class for partition node groups

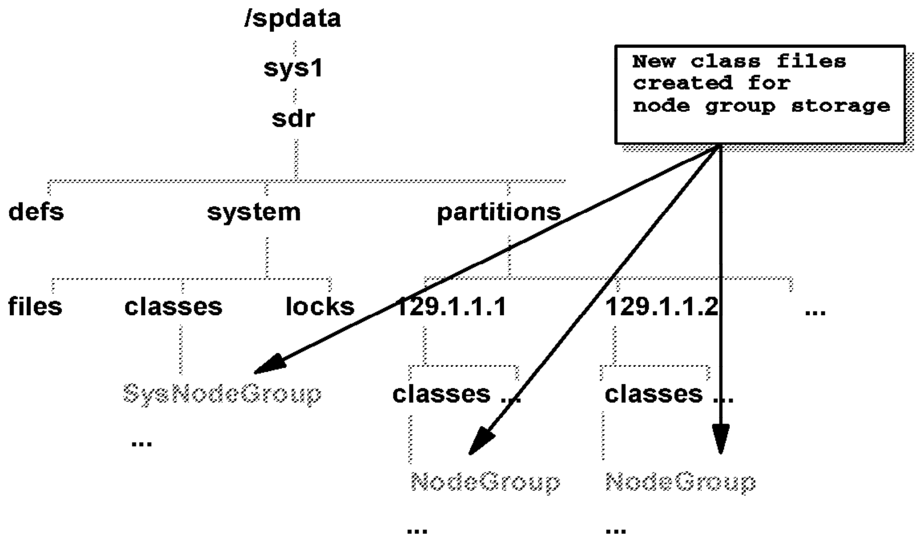
6.3.5.1 Node Group Definition

Within SMIT, system node groups and partition node groups are created by two different entries in the Node Group Information menu located under the Enter Database Information item, as shown on the foil.

Within the Hardware Perspective tool, system node groups and partition node groups are created from the same menu. To create a system node group, the current partition in the CWS/System/Syspars pane must be the System. The Node Group pane has to be displayed (it is hidden by default when launching the Hardware Perspective) and selected. The Node Group entry is then accessible under the Actions menu, and the Define Node Group option can be used.

To create a partition node group, the same actions have to be performed, but with a partition (Syspar object) as the current partition in the CWS/System/Syspars pane. The node group is then created within that partition.

SDR Partitioned Structure



/spdata

sys1

sdr

defs

system

partitions

files

classes

locks

129.1.1.1

129.1.1.2

...

SysNodeGroup

...

classes ...

classes ...

NodeGroup

...

NodeGroup

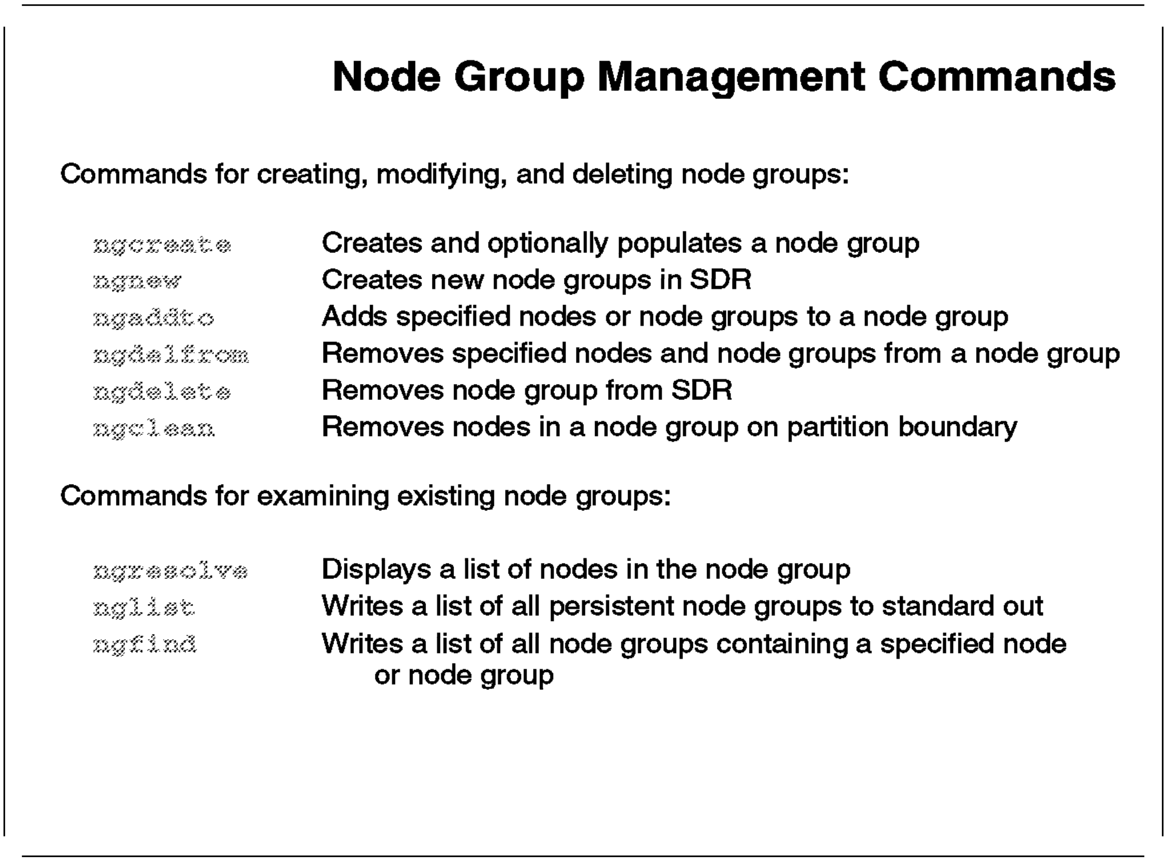
...

New class files
created for
node group storage

Both system node group and partition node group entries are stored in the SDR. System node groups are stored within the global SDR classes, in a class named SysNodeGroup. Partition node groups are stored within the partitioned classes, in a class named NodeGroup. Data of these classes are couples of values (node group name, node number, or node group name). Therefore, a node group has as many entries in the node group SDR class as it has constituent nodes or node groups.

There is a default partition node group named ALLPMAN that contains all the nodes of the RS/6000 SP. It is used as a facility for the Problem Management subsystem commands to address all the node in the partition. Deleting it will not make any Problem Management scripts fail unless you customized them and used the ALLPMAN node group.

6.3.5.2 Node Group Management



Node Group Management Commands

Commands for creating, modifying, and deleting node groups:

ngcreate	Creates and optionally populates a node group
ngnew	Creates new node groups in SDR
ngaddto	Adds specified nodes or node groups to a node group
ngdelfrom	Removes specified nodes and node groups from a node group
ngdelete	Removes node group from SDR
ngclean	Removes nodes in a node group on partition boundary

Commands for examining existing node groups:

ngresolve	Displays a list of nodes in the node group
nglist	Writes a list of all persistent node groups to standard out
ngfind	Writes a list of all node groups containing a specified node or node group

The easiest way of managing node groups is through the Hardware Perspective. A set of commands has also been added to manage the node groups. When referring to persistent storage, we mean the SDR.

ngcreate Creates and optionally populates a named node group.

ngnew Creates but does not populate new node groups in persistent storage.

ngaddto Adds nodes and node groups to the definition list of the destination node group.

ngdelfrom Deletes nodes and node groups from the definition list of the destination node group.

ngdelete Removes node groups from persistent storage.

ngclean Cleans up a node group, removing references to nodes and node groups that are not in the current system partition.

ngresolve Returns a list of hosts in the specified node group.

nglist Returns a list of all node groups in the current system partition.

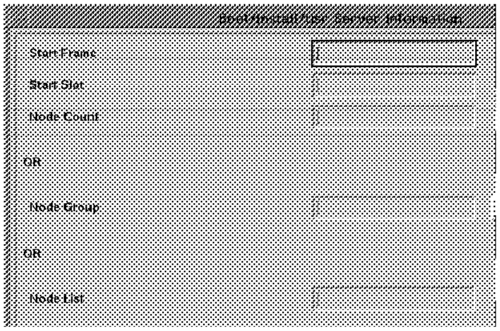
ngfind Returns a list of all node groups whose definition list contains the specified node or node group.

Only the root user can create and manage node groups. Any user can use the node groups if that user is authorized to run the command in which the node group is addressed. See 6.3.6, “Use of Node Groups” on page 232, for details on the use of node groups.

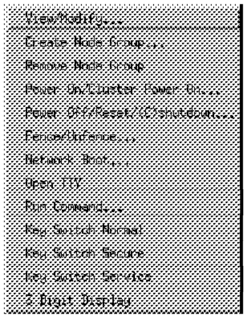
6.3.6 Use of Node Groups

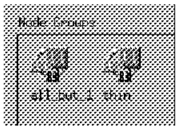
Use of Node Groups

- SMIT menus



- Specific Hardware Perspective menu and pane





- New -N option for PSSP commands:

  Partition node group

  dsh -N first_drawer "ps -eaf"

  System node group

  dsh -G -N thin_nodes "ps -eaf"

Node groups can be defined within SMIT or the Hardware Perspective menus as soon as nodes are declared in the SDR. This happens in the early stages of the installation process. Node groups, once defined, can then be used for the remaining steps of the installation process, such as the declaration of boot/install/usr server information shown on this foil. The network boot can also be achieved on a given node group. During these steps of the installation process, system node groups are used.

Once the system is installed, node group addressing can be used for various system management tasks. When used through a Hardware Perspective menu, a new pane with corresponding icons is manipulated. An entry within the Actions menu gives access to the specific menu that is represented on the foil. All actions on this menu apply to the selected node group. The exception is the 3DigitDisplay action, which displays information for the whole partition.

Apart from using the graphical tool for system management, you can use regular PSSP system management commands. These commands have been modified to address node groups. This is usually done by the *-N* option.

The following commands can address node groups:

- dsh
- Efence
- Eunfence

- hostlist
- spbootins
- splstdata

Efence and Eunfence can be directly followed by a node group name. The other commands use the -N option.

If used separately, the -N option uses the partition node group as a parameter. It must then be followed by a partition node group name from the current partition. The partition node group name is then resolved, and the command is performed on the constituent nodes.

To use a system partition name as a parameter of the -N option, this option has to be combined with the -G option. An example is given on the foil.

The following commands use the *-g* parameter to indicate a node group:

- cshutdown
- cstartup

Note: Traditionally, cshutdown and cstartup have the -N option, but it is used to address individual nodes.

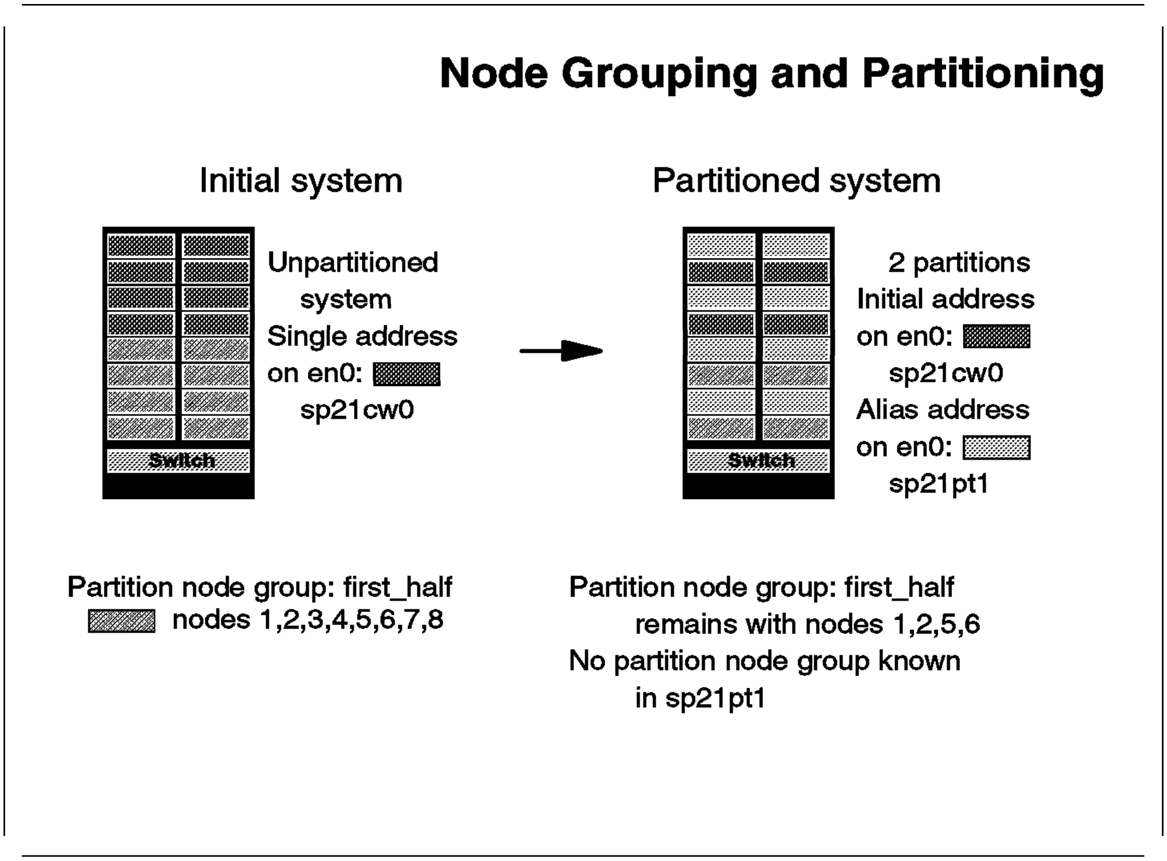
The following commands cannot directly address node groups:

- hmcmds
- hmmon
- nodecond
- psyslprt
- psyslclr
- penotify
- SDRGetObjects

The parallel commands (pcp, pdf, pfind, and so on) also cannot address node groups.

Nevertheless, most of these commands can benefit from the hostlist command to provide them with the list of nodes in the node group and thus the indirect support of node groups.

6.3.7 Node Grouping and Partitioning



Node Grouping and Partitioning

Initial system

Unpartitioned
system
Single address
on en0: ▨
sp21cw0

Partitioned system

2 partitions
Initial address
on en0: ▨
sp21cw0
Alias address
on en0: ▨
sp21pt1

Partition node group: first_half
▨ nodes 1,2,3,4,5,6,7,8

Partition node group: first_half
remains with nodes 1,2,5,6
No partition node group known
in sp21pt1

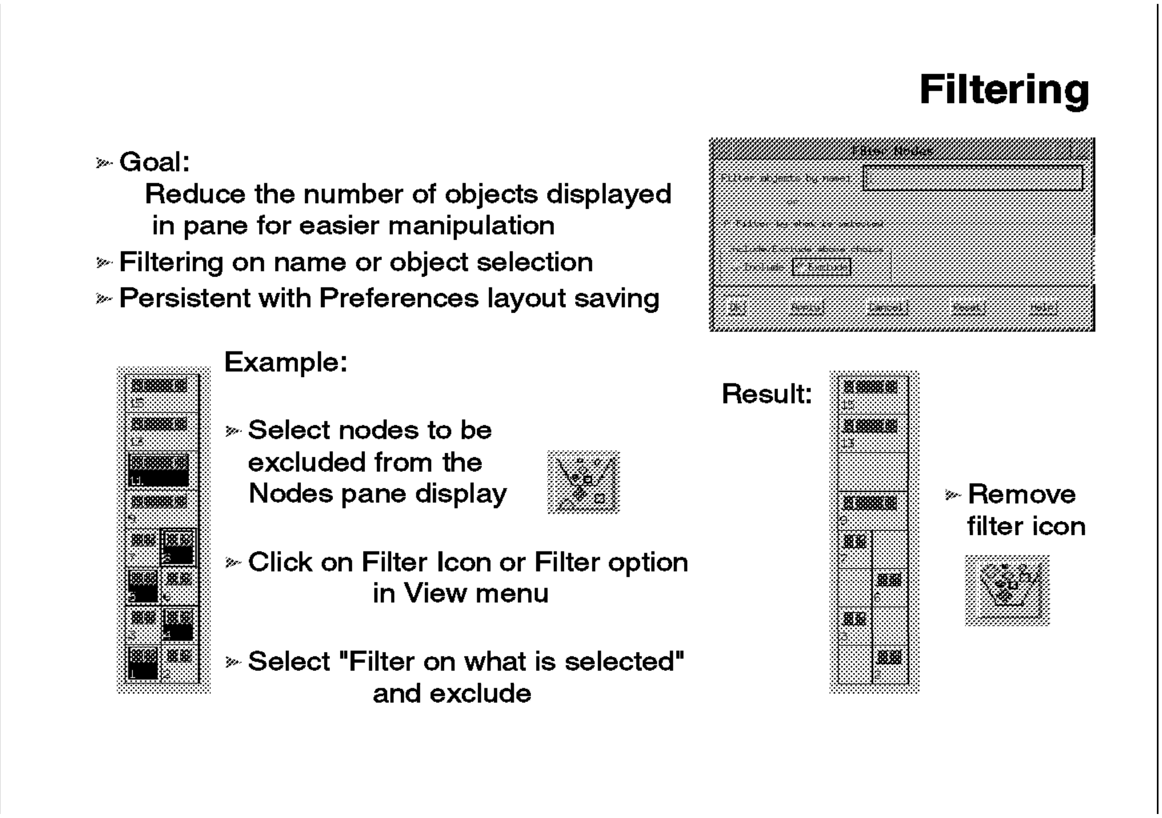
Node groups can be global to the system (known as system node groups) or local to one partition (known as partition node groups). But what happens to the partition node groups when the system is "repartitioned"? This scenario is illustrated by this foil.

On the left of the foil, the RS/6000 SP system is shown "unpartitioned." This means that it has only the default partition encompassing all the nodes. This allows the definition of a partition node group. It is called first_half because it groups the four bottom drawers of the frame. The default partition is addressed with the initial IP address on the Control Workstation network adapter; in this case, it is sp21cw0 on Ethernet.

The system is then partitioned into two partitions as represented on the right of the foil. The contents of the first_half node group are then logically split into the two partitions. When partitioning, the existing partition node groups remain in the partition in which they were initially defined, with the nodes remaining in the partition. In this example, first_half is still known in the partition addressed by sp21cw0, but it has lost half of its contents. No partition node group has been created in the partition addressed by the alias.

The partitioning process does not affect system node groups.

6.3.8 Filtering



Filtering

- Goal:
  Reduce the number of objects displayed in pane for easier manipulation
- Filtering on name or object selection
- Persistent with Preferences layout saving

Example:

- Select nodes to be excluded from the Nodes pane display
- Click on Filter Icon or Filter option in View menu
- Select "Filter on what is selected" and exclude

Result:

- Remove filter icon

Filtering is a graphical mechanism. It allows the masking of objects in a given pane. There is no trace of filters stored permanently within the PSSP infrastructure, such as the SDR. It is useful to work only on a subset of objects with the assurance that no interface mistakes (such as a wrong click or a wrong action) disturb other objects.

Filtering can be performed by excluding or including objects that you specify either by their names or by selecting them with the mouse.

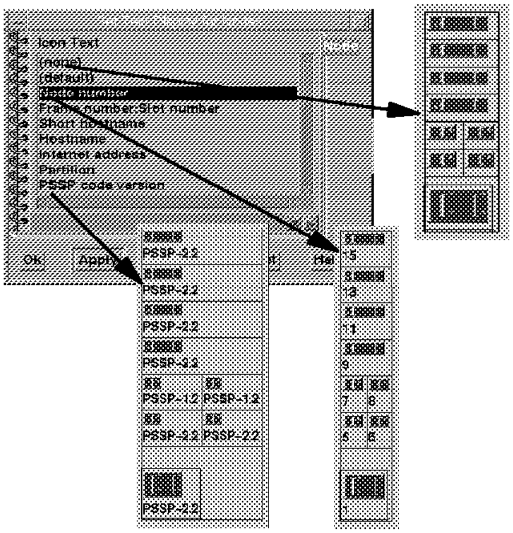
Filtering is only known at the interface level. For example, when activating a filter to mask some nodes in the nodes pane, and launching a given monitoring on the nodes, the display only shows the filtered monitored nodes, but monitoring is active for all of the nodes. When removing the filter, all nodes appear in their monitoring state. This is true because the monitoring mechanism is applied at the class level, as explained in 6.3.11.1, “Activating Monitoring” on page 239, and does not really care about selected objects. For other mechanisms that rely on selected elements, like powering off nodes, the operation will never be applied on filtered elements because they cannot be selected.

Filtering information is saved in the Preferences files. This enables an easy restitution of a filtered environment.

6.3.9 Object Label Modification

View Modification

- Goal:
  Choose the appropriate text to be displayed under the objects of a given pane
- Any Note Book information can be used
- Allow the panes to be customized according to the current system management task
- Example: PSSP code version on nodes, useful within a gradual system migration process



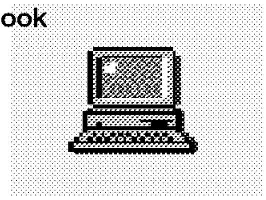
Information presented in panes is using graphical icons. There is also a text displayed below each icon. This text can be customized by selecting the **Object Label** entry in the View menu.

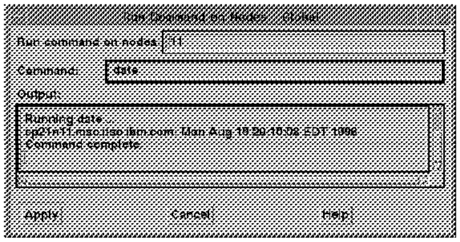
A window pops up with various choices for the displayed label. The <none> choice allows the deletion of the label. As mentioned on this foil, the choice of the PSSP code level as a label for each node gives the ability to have an accurate view of the step you reached within a gradual migration process.

6.3.10 New Interfaced Functions

New Access to Functions

- Access to the SDR information through the Note Book
- Monitor the Control Workstation
- Start Switch for a selected partition
- Extensive, flexible system monitoring
- Run command on node or node group





Run command on nodes: 11

Command: date

Output:

```
Running date ...
sp21n11.msc.itso.ibm.com: Mon Aug 18 20:10:02 EDT 1996
Command complete.
```

The Perspectives concept enlarges the scope of the system management on the RS/6000 SP. This foil summarizes new possibilities given by the Hardware Perspective. These functions, when available, were only command line interfaced in the past.

- **SDR configuration view**

  Information stored in the SDR for any object visualized in a Hardware Perspective pane can be displayed using the Note Book facility. It also applies to the Control Workstation, as described in the next paragraph.

- **Control Workstation**

  The Control Workstation is now managed as part of the cluster. SDR information can be queried through the Note Book, and special conditions can be monitored. This removes the need for an external RS/6000 system management to be used with RS/6000 SP system management tools. It is convenient to monitor special conditions on the Control Workstation, such as CPU load or disk shortage. For extensive information on monitoring with the Hardware Perspective, see 6.3.11, "SP Health Definition" on page 239.

- **Switch control**

  The switch can now be started in the current partition. The Estart command is part of the Actions menu located under the Frame/Switch sub-menu.

- **System monitoring**

  This is extensively explained in 6.3.11, “SP Health Definition” on page 239.

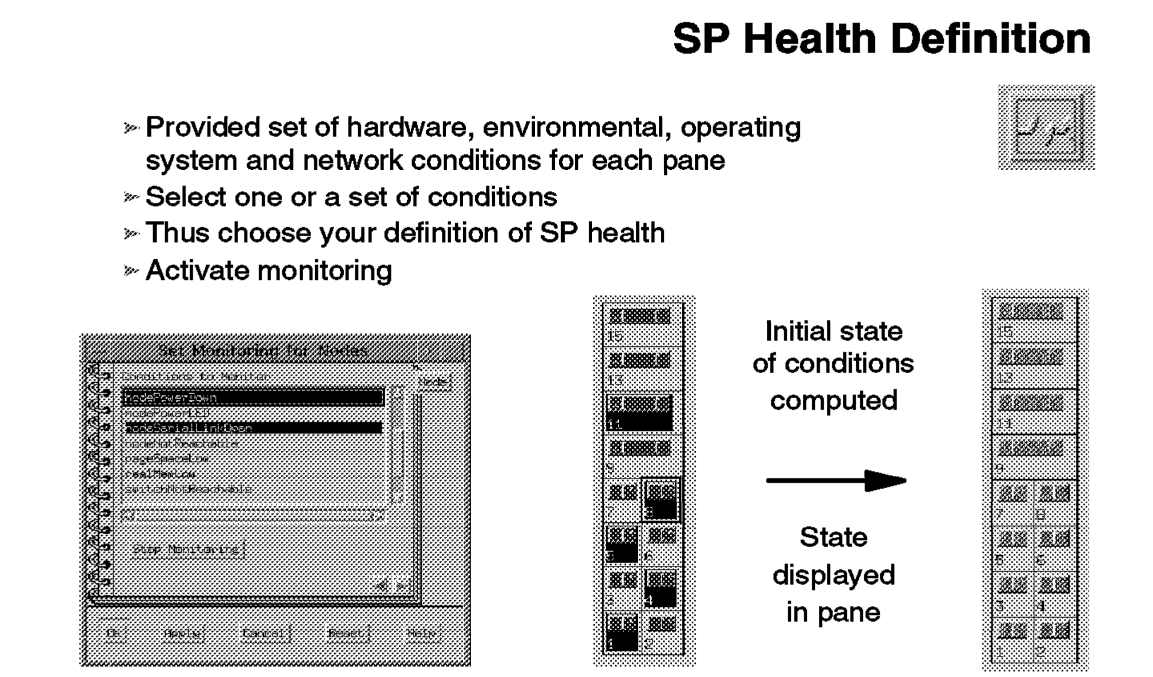
- **Command execution**

  There is a new **Run Command** entry in the nodes and node groups sub-menus that displays a graphical interface for running dsh commands on the selected nodes or node groups. The messages returned in that window are sometimes misleading. If a node is unavailable for dsh or a node group list cannot be resolved in the SDR, the following message appears:

  ```
  dsh: 5025-511 No hosts in working collectives
  ```

  This does not mean that this interface is dependant upon any WCOLL variable declaration, although the message may imply this.

6.3.11 SP Health Definition



SP Health Definition

- Provided set of hardware, environmental, operating system and network conditions for each pane
- Select one or a set of conditions
- Thus choose your definition of SP health
- Activate monitoring

Set Monitoring for Nodes

Conditions to Monitor

nodePowerDown
nodePowerLED
nodeSerialLinkOpen
nodeNotReachable
pageSpaceLow
realMemLow
switchNotReachable

Initial state of conditions computed

State displayed in pane

The monitoring task on a system like the RS/6000 SP consists of following the evolution of some crucial parameters in order to react if state changes occur. Even if there are general parameters that are likely to be addressed, there is not a general consensus on the meaning of “everything is OK.” It is heavily dependent on what “everything” is.

A solution would be to give access to as many monitoring windows as requested parameters. This is the philosophy of SPmon. But it occupies an important surface of the Control Workstation screen for very little relevant information.

The alternative proposed by the Hardware Perspective is radically different: the use of a single window. Monitoring is achieved directly within the panes of the application. As there is only one representation of each object, the function of grouping monitored parameters has been implemented.

6.3.11.1 Activating Monitoring

The monitoring of parameters is performed through conditions on the state of these parameters. By choosing one or several conditions, you can define what health means for your RS/6000 SP system. If one of these conditions is in an abnormal state, your RS/6000 SP is not healthy anymore.

Important: The implementation of this concept is made on a class basis in the current release of Perspectives. This means that all the instances of a class have the same monitored conditions at any time. Different monitoring conditions can be activated on different classes of the same pane.

Monitoring can be requested with the monitor icon or the Monitor entry in the Views menu. This pops up a window that allows you to choose potential conditions. In the case of an active pane containing objects of the same class (the nodes pane for example), there is no need for an object to be selected in the pane; the pane being active is sufficient. In the case of an active pane with several classes of objects, the window that popped up contains tabs corresponding to all the included classes.

Once conditions have been selected, the monitoring can be activated by the OK or the apply button. The initial state of the conditions is queried from the event manager and displayed in the pane by an icon modification.

6.3.11.2 Monitoring Conditions

Available monitoring conditions are permanently stored within the SDR in the EM_Condition class. These conditions are based on resource variables and instance vectors manipulated by the event manager subsystem. The instance vector concept introduced by the event manager is referred to as *resource identifier* within the Perspectives documentation. For more information on the structure of the EM_Condition, refer to 6.3.14, "Customizing Monitored Conditions" on page 243.

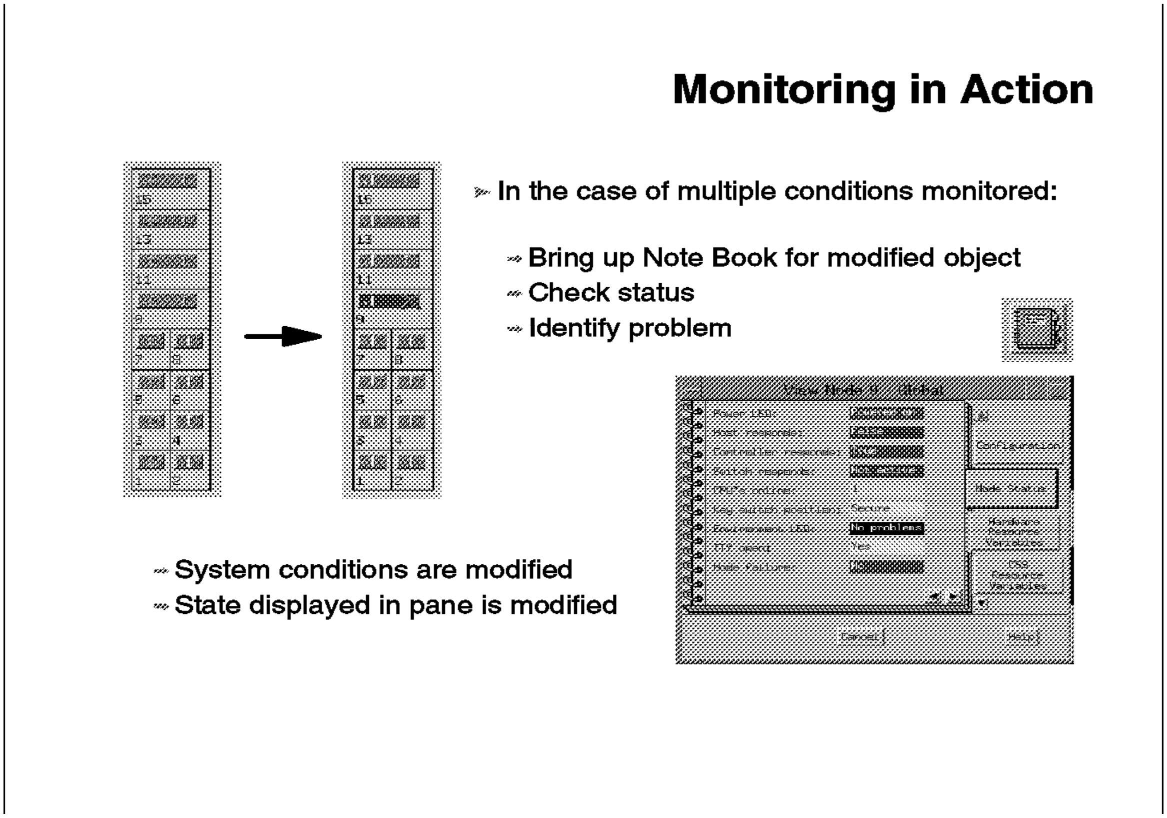
Each time monitoring is requested, before popping up the conditions window, the Hardware Perspective reads the EM_Condition contained in the SDR. All conditions are stored in the same class without consideration of applicable object class. Nevertheless, all conditions may not be applicable to all classes.

The applicability of a condition to a class is fixed by an attribute of each EM_Condition, named *unspecified*. For example, if unspecified is NodeNum, the condition will be monitored on nodes; if SwitchNum, it will be monitored on the switch; and if FrameNum, it will be monitored on the frame.

Classes are collective classes such as Frame, Partitions (Syspar), and System. They may or may not have their own monitoring conditions, but they have conditions that can be monitored on their constituents (usually nodes). Node groups have the same monitoring conditions as nodes. The type of monitored conditions corresponding to a given unspecified attribute value and the related class is hard-coded in the Hardware Perspective.

When the condition window is created, the suitable conditions are extracted from the SDR using these attributes and the class on which monitoring is requested.

6.3.12 Monitoring in Action



Monitoring in Action

- In the case of multiple conditions monitored:
  - Bring up Note Book for modified object
  - Check status
  - Identify problem

View Node 9 - Global

Power LED: Powered off
Host responds: False
Controller responds: No
Switch responds: Not active
CPU's online: 1
Key switch position: Secure
Environment LED: No problems
TTY open: Yes
Node failure: No

- System conditions are modified
- State displayed in pane is modified

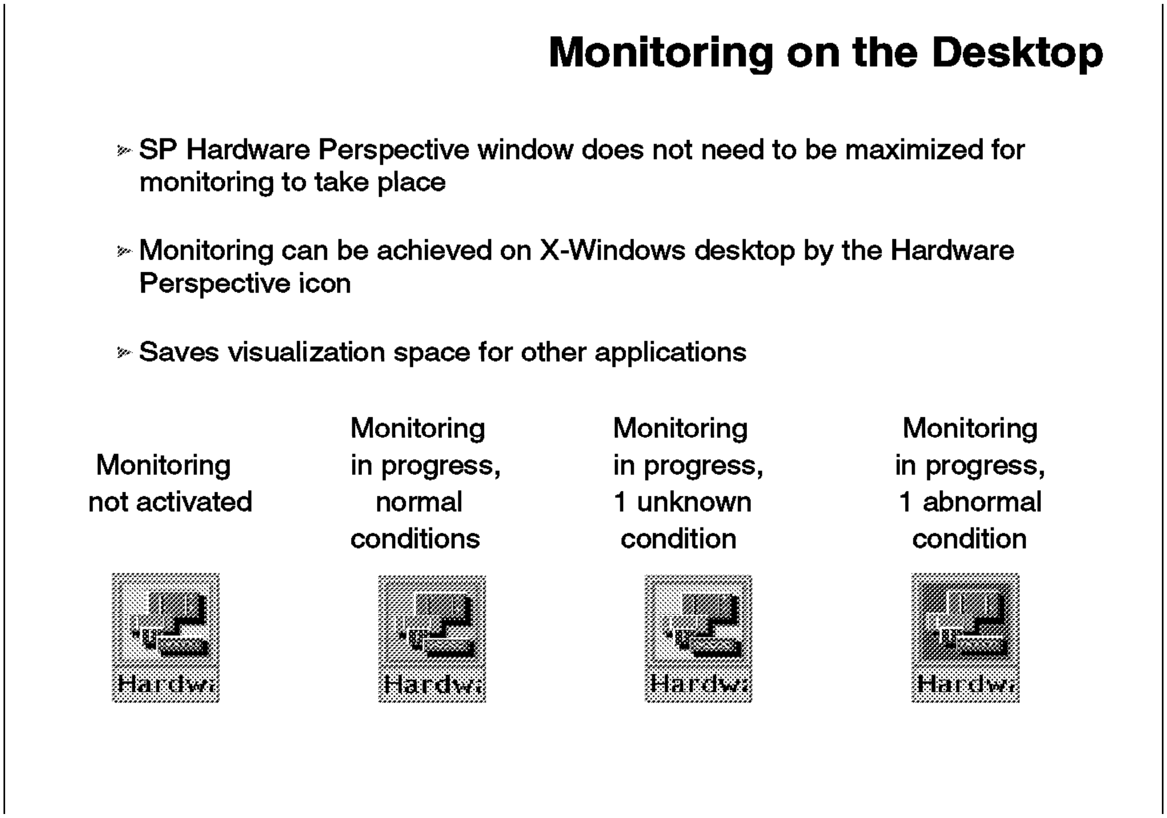
If a unique condition is monitored, the problem determination is straightforward. In the case where several conditions are simultaneously monitored on the same class, the state modification of one of them leads to a change in the state of the monitored object. The problem determination phase aims at finding which condition changed.

There are several locations where problem determination can be achieved. The common entry is the Note Book.

For trivial conditions, the **Status** tab can be useful to detect a red parameter.

For more complex conditions, the **Monitored Conditions** tab gives access to the condition in which the state has changed.

6.3.13 Monitoring on the Desktop



Monitoring on the Desktop

- SP Hardware Perspective window does not need to be maximized for monitoring to take place
- Monitoring can be achieved on X-Windows desktop by the Hardware Perspective icon
- Saves visualization space for other applications

Monitoring not activated	Monitoring in progress, normal conditions	Monitoring in progress, 1 unknown condition	Monitoring in progress, 1 abnormal condition

The display surface requirement can be minimized for monitoring. The task can be achieved by the Hardware Perspective icon itself.

Once monitoring has been started on one or several object classes within the panes, the icon of the application on the desktop will reflect monitoring status.

When monitoring is inactive, the icon normally shows the Hardware Perspective′s logo on a gray background.

The icon background is used to feature monitoring conditions:

Green Monitoring in progress, all the conditions are in the normal state.

Yellow Monitoring in progress, one condition is in an unknown state.

Red Monitoring in progress, one condition is in an abnormal state.

When an abnormal state occurs, the system administrator maximizes the application and proceeds as described in 6.3.12, “Monitoring in Action” on page 241. The use of the icon saves desktop space for other applications.

6.3.14 Customizing Monitored Conditions

Customizing Monitored Conditions

- Monitored conditions are permanently stored in SDR in class EM_Condition
- Object class on which a condition applies is determined by the "unspecified" attribute

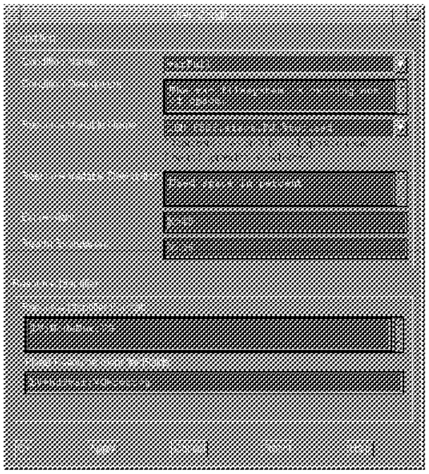
FrameNum	monitored on frame
SwitchNum	monitored on switch
NodeNum	monitored on node or node group

- To add monitored conditions:

```
spevent&
```

- Example:

```
EM_Condition varFull
IBM.PSSP.aixos.FS.%Totused
'X>90'   'X<80'
'LV=hd9var;VG=rootvg'  NodeNum
"The var filesystem is running
out of space."
```



Customizing the monitored conditions is done by modifying or adding entries in the EM_Condition class in the SDR. This class is partition-dependant. If a new condition has to be accessible to any partition of the system, it has to be added in the SDR classes of all the partitions.

The creation of monitored conditions is done through the Event Perspectives. A description of the steps of the process follows:

1. Open the Event Perspectives either by launching the application from the Perspectives launch pad or by typing `spevent&`
2. Make the events pane active by clicking it.
3. In the **Actions** menu, under **Event Definitions** choose the **Create** entry. The Create Event Definition window pops up.
4. In the **Definition** tab of this window, activate the Condition area by clicking the arrow at the right of the condition name field. This activates the Create Condition button. Click this button, and the Create Condition window appears.
5. Fill the condition fields of the Create Condition window according to information given on the EM_Condition class in 6.3.14.1, "The EM_Condition Class" on page 244. Click **OK** or **Apply** to create the condition.
6. Close the Create Condition window and the Create Event Definition window. This can be done either by creating an event related to the new condition or

simply by clicking **Cancel** in the Create Event Definition window. Despite this cancellation, and consequently the fact that no event was created, the condition you created in the Create Condition window is permanently stored in the EM_Condition class of the SDR.

To view a predefined condition, follow the same process, but instead of the Create Condition button, click the **View Condition** button located on the left. A window displaying the values of the various EM_Condition attributes for the condition you selected is displayed. It is not possible to modify the displayed information.

There is no graphical interface or automatic way of modifying or deleting instances of the EM_Condition class. Despite of the fact that these commands are usually not recommended for a direct use, modification and deletion have to be made using the PSSP commands for SDR classes modification: `SDRDeleteObjects` and `SDRChangeAttrValues`.

Note: You should archive the SDR before using these commands, or at least save the EM_Condition file located under the partition classes directory.

6.3.14.1 The EM_Condition Class

The EM_Condition class has the following attributes:

name Name that appears in the monitored conditions pop-up list

variable Resource variable on which the monitoring applies (refers to variables of the EM_Resource_Variable class)

predicate Expression of the condition that will be true if the monitored event occurs

rearm Initial value of the condition expression when activating monitoring

specified Values of elements of the instance vector that allow unique identification of the instance of the resource that has to be monitored

unspecified Element of the instance vector that is not specified, and as such attaches to a given object class in Perspectives on which the monitoring condition will be activated

description Textual description of the condition for help purposes

When the SDR is created, prepared conditions are placed in the EM_Condition class by the loadeventconditions Perl script. Initial conditions are as follow:

```
# SDRGetObjects EM_Condition name unspecified description

name unspecified description
frameControllerNotResponding FrameNum
    ″The frame controller is not responding.″
framePowerOff                NodeNum
    ″The power to the frame has been turned off.″
hostResponds                 NodeNum
    ″The node is not responding.″
keyNotNormal                 NodeNum
    ″Key mode switch on a node was switched out of the Normal position.″
nodeEnvProblem               NodeNum
```

```
    ″The environment indicator LED on the node is illuminated. A
hardware problem was detected.″
nodePowerDown               NodeNum
    ″The power to the node is off.″
nodePowerLED                NodeNum
    ″Node power is off when powerLED is not 1.″
nodeSerialLinkOpen          NodeNum
    ″The serial link to the node (TTY) is open.″
nodeNotReachable            NodeNum
    ″Group services has found no way to communicate with the node. The
node is presumed to be down.″
pageSpaceLow                NodeNum
    ″The paging space utilized on the node exceeds 85 percent.″
programRunning              NodeNum;ProgName;UserName
    ″The specified program is being run on the specified node by the
specified user.″
realMemLow                  NodeNum
    ″The real memory on the system is over 85 percent utilized.″
switchPowerLED              SwitchNum
    ″Switch power is off when powerLED is not 1.″
switchNotReachable          NodeNum
    ″The switch adapter on the node is not responding to ip or the node
is isolated.″
switchResponds              NodeNum
    ″The switch adapter on the node is not responding or the node is
isolated.″
tmpFull                     NodeNum
    ″The file system for LV=hd3 and VG=rootvg is running out of space.″
```

The programRunning condition is provided but cannot be used as such, because the unspecified attribute does not have a unique value. The consequence is that the condition will not appear in the list of possible conditions for a node. The programRunning condition is for further releases, but it can also be an inspiration source as described in the next paragraph regarding mksysb execution on a node.

6.3.14.2 Example of Possible Customization

This section describes examples of possible customization.

/var space on Control Workstation. The var filesystem on the Control Workstation receives the log files for the SP daemons. If something goes wrong with a daemon (the hardmon cannot start, for example), the var/adm/SPlogs corresponding subdirectory is filled with error messages. Perspectives-monitored conditions give you a means of monitoring the /var filesystem by entering the following information into the Create Condition window:

Condition Name: varFull
Condition Description: The var filesystem is running out of space
Resource Variable Name: IBM.PSSP.aixos.FS.%totused
Resource Variable Description: Used space in percent
Expression: X>90
Rearm Expression: X<80
Resource Identifier Format: LV;NodeNum;VG
Fixed Resource Identifier Fields: LV=hd9var;VG=rootvg

Checking the EM_Condition class will give:

```
# SDRGetObjects EM_Condition name==varFull

varFull      IBM.PSSP.aixos.FS.%totused X>90        X<80
LV=hd9var;VG=rootvg NodeNum      ″The var filesystem is running out of
space.″
```

To activate the condition, click the **Control Workstation** icon in the CWS/System/Syspar pane, and then click the **monitor** icon. This will bring up the list of potential conditions for the classes displayed in the pane. Choose the **CWS** tab and look at the list. The varFull condition should appear at the bottom. Clicking the monitor icon or choosing the Monitor option in the View menu lead to the reload of the SDR information. This makes your newly created condition appear.

We specified “unspecified” to be NodeNum, because conditions listed to be monitored on the Control Workstation are those that are accessible for the nodes (the Control Workstation is node 0). Some may not be suitable or may be meaningless in the case of the Control Workstation.

mksysb execution on a node. We are using a variant of the programRunning condition in EM_Condition:

Condition Name: mksysbRunning
Condition Description: Mksysb running on node
Resource Variable Name: IBM.PSSP.Prog.pcount
Resource Variable Description: Count and list of processes running a program for a user
Expression: X@0>0
Rearm Expression: X@0==0
Resource Identifier Format: ProgName;NodeNum;UserName
Fixed Resource Identifier Fields: ProgName=/usr/bin/savevg;UserName=root

When activated, this condition will lead to a change in the monitored status of the node each time a savevg command is activated.

This can be applied to any program for any user, assuming the right program name is given in the condition. Use ps -eaf to verify the name of a given program process when executing.

6.3.14.3 Example of Impossible Condition Customization

This section gives examples of limitations in monitoring conditions.

Setup_server monitoring. It would be useful to monitor the execution of the setup_server script on the boot/install servers.

If the boot/install server nodes are not fully customized to the installation task, problems can occur at installation of their client nodes. The customization of boot/install server nodes is done by the setup_server program that executes on the nodes either upon request or at the first boot after boot/install server installation. During this cutomization, the setup_server program performs the transfer of the images for the client nodes to the boot/install server, the installation of NIM filesets if requested, and the creation of boot files for the client nodes. It can be quite a long process and it executes in background. That means that the node can be accessed for login or running other applications.

A way of seeing that the node is still not available for use as boot/install server is that the host_responds turns green only at the end of setup_server. If for any reason, host_responds is not reliable on your system, you have no other means of knowing the progress of setup_server, except to login on the node and check the processes by running the ps command.

It would be nice to be able to monitor the execution of setup_server from the Control Workstation with the Hardware Perspective′s monitoring function. The following command may seem as if it will work:

Condition Name: setupserverRunning
Condition Description: setup_server running on node
Resource Variable Name: IBM.PSSP.Prog.pcount
Resource Variable Description: Count and list of processes running a program for a user
Expression: X@0>0
Rearm Expression: X@0==0
Resource Identifier Format: ProgName;NodeNum;UserName
Fixed Resource Identifier Fields:
ProgName=/usr/lpp/ssp/bin/setup_server;UserName=root

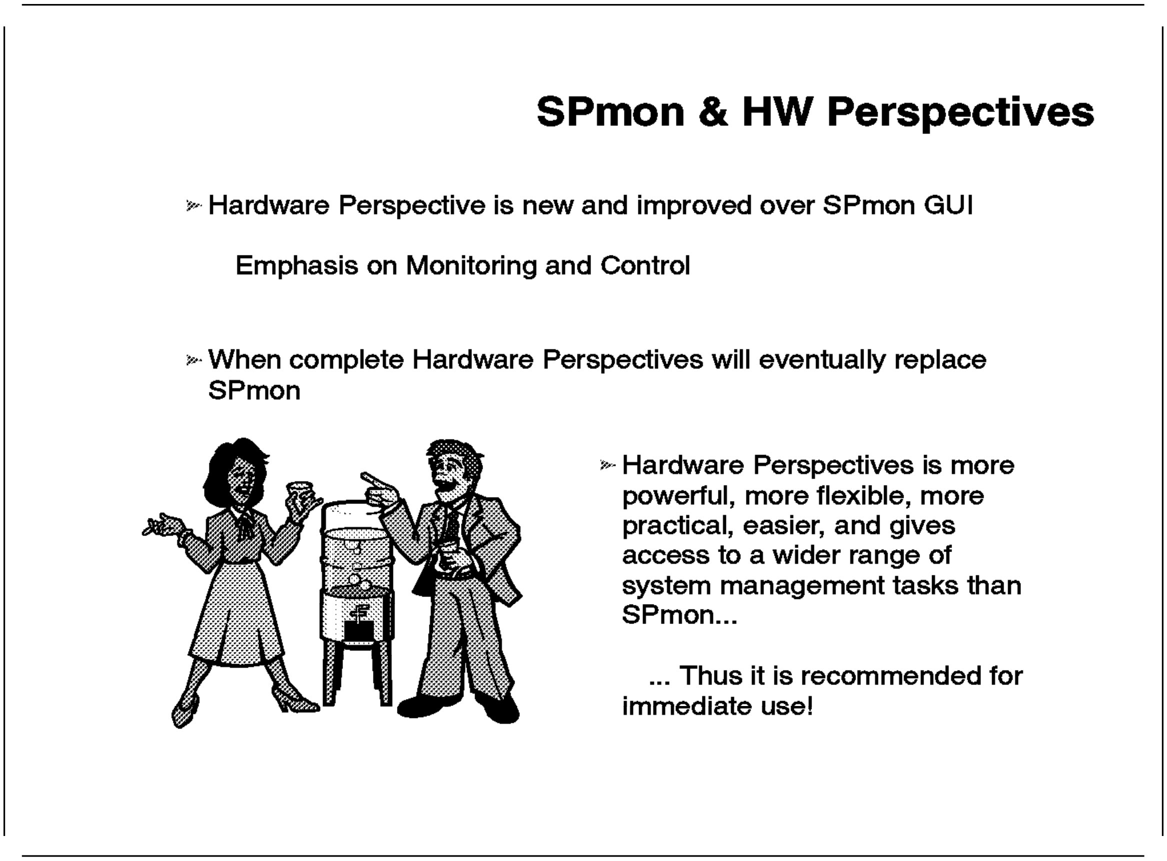
Unfortunately, setup_server is a Perl script, and as such runs with *perl* as program name. The replacement of ProgName=setup_server by ProgName=perl does not fix the problem because other Perl scripts may run on the node (like the pmanrmd daemon that is also a Perl script and that runs permanently on the node if the problem management subsystem is used).

This impossibility in individually monitoring setup_server also applies on other Perl scripts, like supper.

WORM execution monitoring. Another interesting monitoring would be the WORM daemon, which is supposed to run on the nodes prior to switch launching. But this is affected by another limitation of monitoring through EM_Conditions: the limitation to 32 characters for the length of the parameters. In the case of the WORM we would have ProgName=/usr/lpp/ssp/css/fault_service_Worm_RTG_SP , which is 42 characters long.

6.4 SPmon and Hardware Perspectives

This section presents a comparison of different functions that can be executed by using SPmon or Hardware Perspective.



SPmon & HW Perspectives

- Hardware Perspective is new and improved over SPmon GUI

  Emphasis on Monitoring and Control

- When complete Hardware Perspectives will eventually replace SPmon

- Hardware Perspectives is more powerful, more flexible, more practical, easier, and gives access to a wider range of system management tasks than SPmon...

  ... Thus it is recommended for immediate use!

This foil aims at answering questions about the current coexistence of SPmon and the Hardware Perspective as two means of monitoring the RS/6000 SP system.

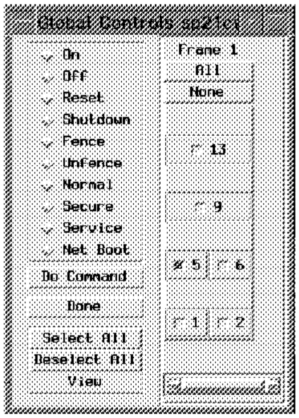
The conclusion is that no effort or investment should be put to the SPmon customization stream, as the Hardware Perspective offers a better solution.

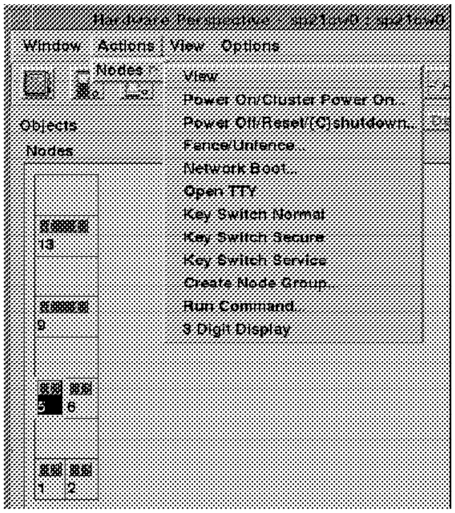
The following foils explain how to use Perspectives to realize common monitoring tasks performed with SPmon.

6.4.1 Global Control

Global Control

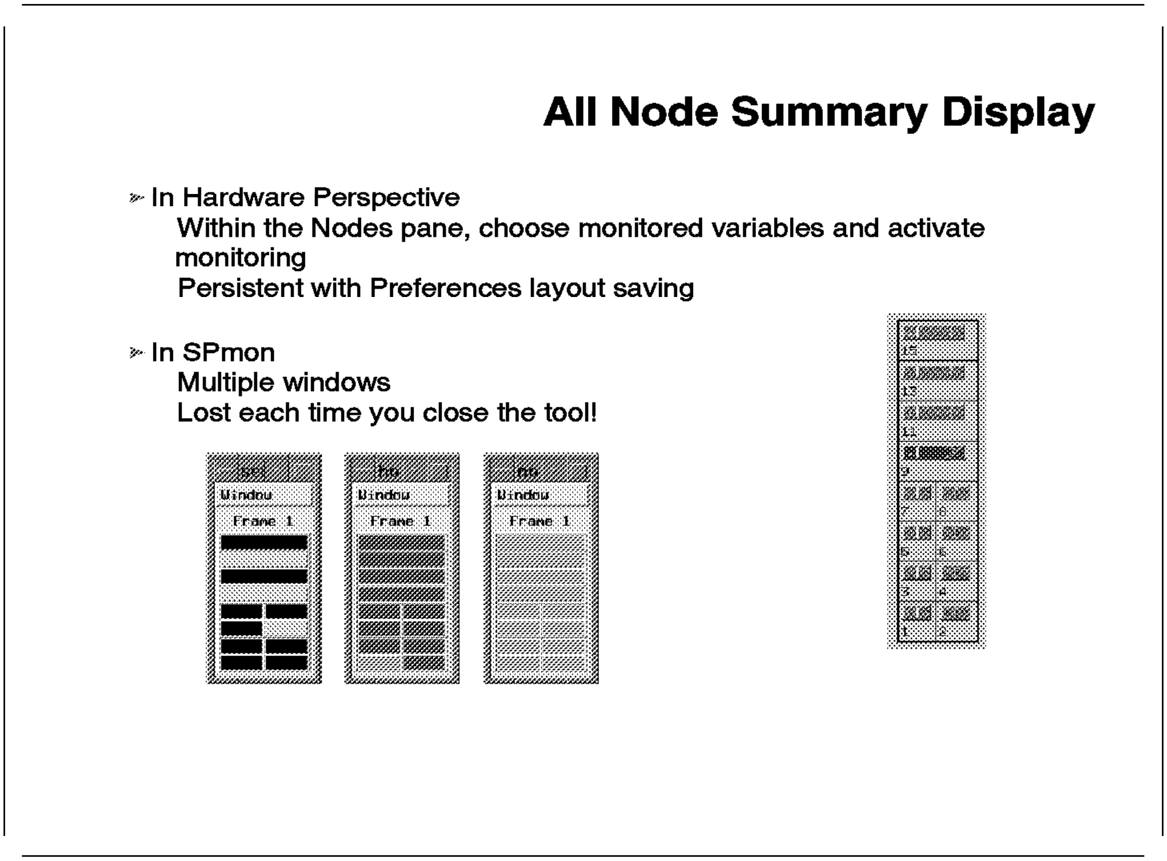
- In Hardware Perspective
  Within Nodes pane, extensive actions menu and icons
- In SPmon
  Unique window with limited choices





Within the Hardware Perspective, the control is performed by selecting nodes and activating an option of the Action menu. This menu has more entries than the SPmon control panel, and the display of the nodes in Perspectives can be more easily modified by filtering.

6.4.2 All Node Summary Display



All Node Summary Display

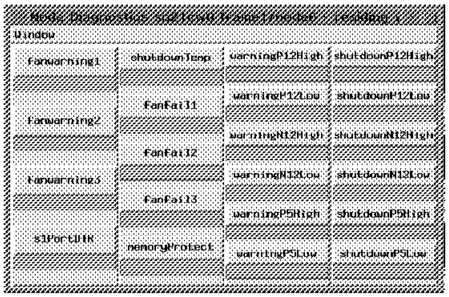
- In Hardware Perspective
  - Within the Nodes pane, choose monitored variables and activate monitoring
  - Persistent with Preferences layout saving
- In SPmon
  - Multiple windows
  - Lost each time you close the tool!

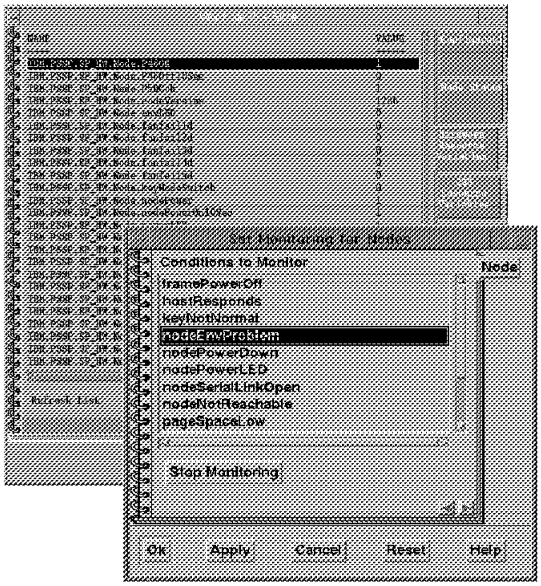
As extensively described in 6.3.11, “SP Health Definition” on page 239, monitoring is now performed in the main window, avoiding the opening of multiple windows. Layout and monitoring information can be stored and retrieved with the Preferences saving mechanism described in 6.2.2, “Saving Preferences” on page 215.

6.4.3 Environmental Variables

Environmental Variable

- In Hardware Perspective
  Unique condition, nodeEnvProblem for all the environmental variables, details if necessary
- In SPmon
  Crowded window




NAME
VALUE
IBM.PSSP.SP_HW.Node.P4O6
1
IBM.PSSP.SP_HW.Node.P4OfIIUSec
0
IBM.PSSP.SP_HW.Node.P5DCok
1
IBM.PSSP.SP_HW.Node.codeVersion
1286
IBM.PSSP.SP_HW.Node.envLED
5
IBM.PSSP.SP_HW.Node.fanfail1d
0
IBM.PSSP.SP_HW.Node.fanfail2d
0
IBM.PSSP.SP_HW.Node.fanfail3d
0
IBM.PSSP.SP_HW.Node.fanfail4d
0
IBM.PSSP.SP_HW.Node.fanfail5d
0
IBM.PSSP.SP_HW.Node.keyModeSwitch
0
IBM.PSSP.SP_HW.Node.nodePower
1
IBM.PSSP.SP_HW.Node.nodePowerOn10Sec
1
Refresh List
Set Monitoring for Nodes
Conditions to Monitor
Node
framePowerOff
hostResponds
keyNotNormal
nodeEnvProblem
nodePowerDown
nodePowerLED
nodeSerialLinkOpen
nodeNotReachable
pageSpaceLow
Stop Monitoring
Ok
Apply
Cancel
Reset
Help

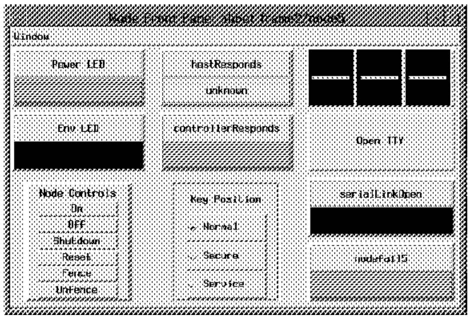
A finer grain is provided for monitoring environmental variables. There are now two levels of monitoring. A condition named *nodeEnvProblem* allows the global monitoring of hardware conditions. In case of failure of that condition, the abnormal value can be detected by browsing the Hardware Resource Variable list and associated values.

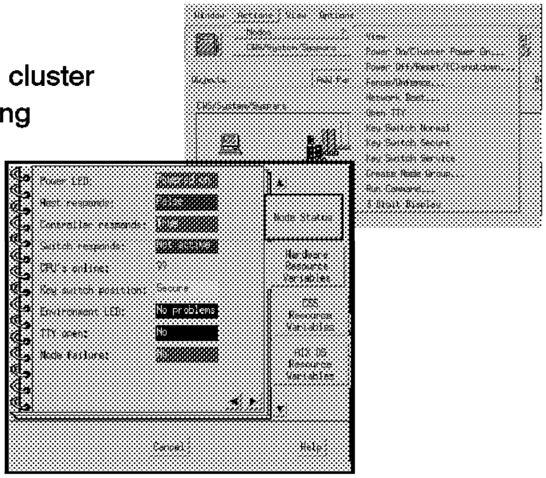
6.4.4 Node Front Panel

Node Front Panel

- In Hardware Perspective,
  read-only Note status Tab in Note Book and Node Action menu

- Node front panel display in SPmon
  - Feeling, for operators, of a single physical interface to a node in the cluster
  - Useful for manual node conditioning





Power LED: Powered on
Host responds: False
Controller responds: True
Switch responds: Not active
CPU's online: ??
Key switch position: Secure
Environment LED: No problems
TTY open: No
Node failure: ??

The Hardware Perspective Note Book is a facility that allows you to access information for a given object:

- Configuration information extracted from the SDR
- Status information for monitored parameters coming from the event manager daemon
- Monitored conditions that are either static (extracted from the SDR) or dynamic (refreshed by the event manager)
- Detailed information on customized monitor condition such as predicate states

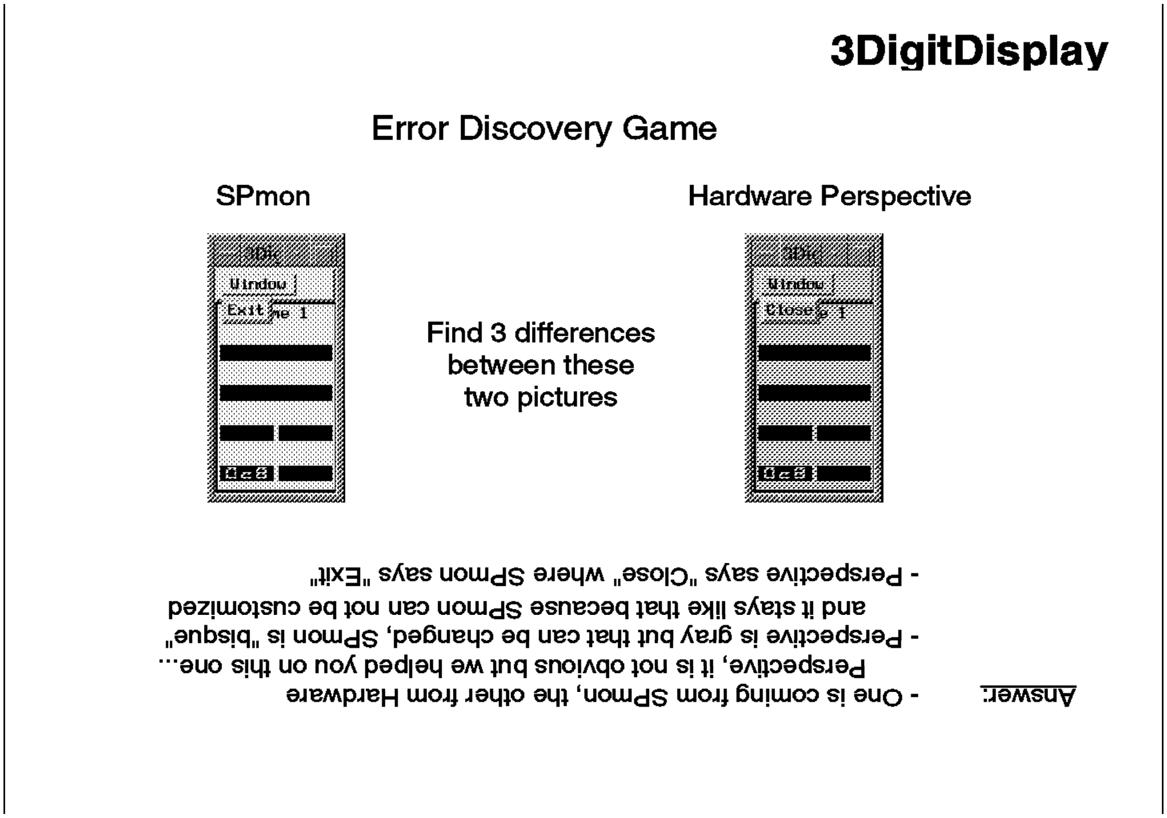
This allows the Note Book to be used as an information vector as well as a problem determination tool for monitored conditions or a read-only front panel to display the status of an object.

This applies to a node, thus converting the Node Status tab of the Note Book to a read-only Node Front Panel.

The reason for the success of the Node Front Panel among operators is that it isolates the control of a given node from the rest of the cluster. This avoids potential manipulation mistakes that may affect the other nodes.

Currently within Perspectives, actions have to be launched on the node through the Action menu, which leads to a few more manipulations during manual node conditioning.

6.4.5 3DigitDisplay



3DigitDisplay

Error Discovery Game

SPmon

Hardware Perspective

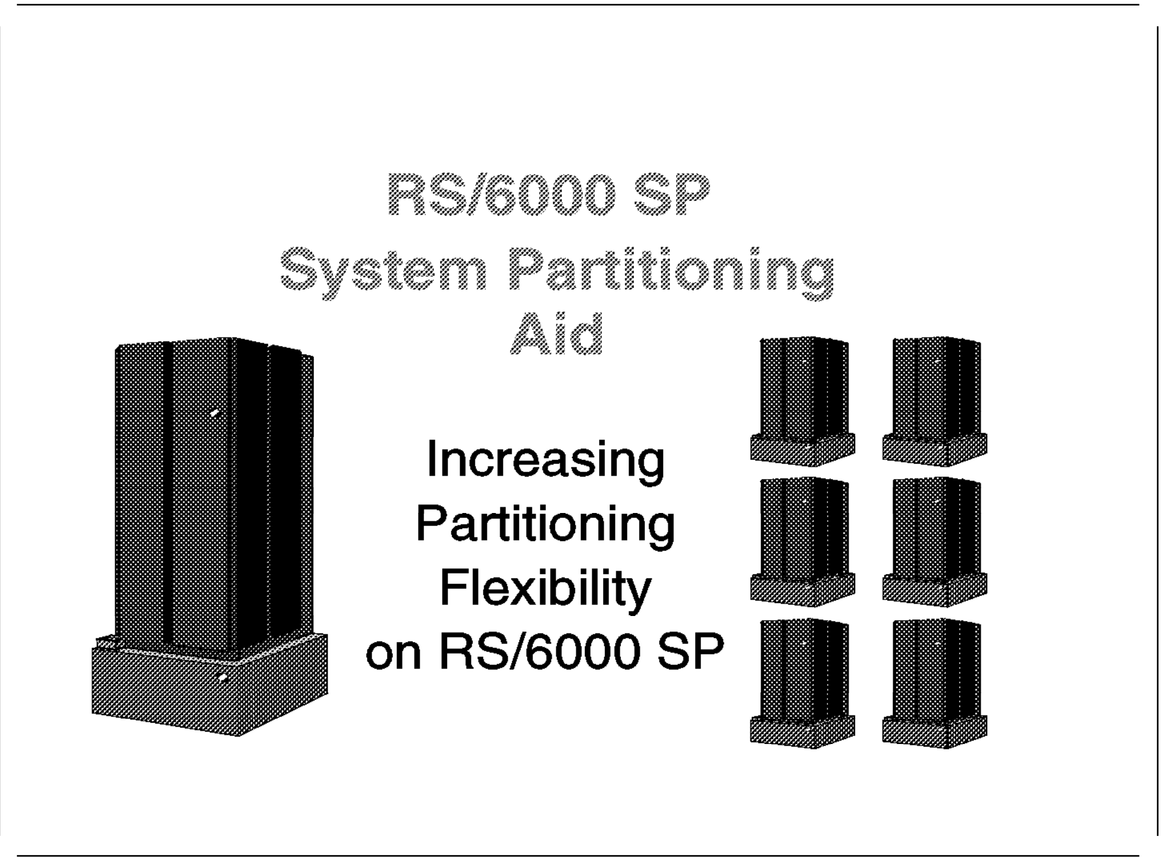
Find 3 differences between these two pictures

Answer:

- One is coming from SPmon, the other from Hardware Perspective, it is not obvious but we helped you on this one...
- Perspective is gray but that can be changed, SPmon is "bisque" and it stays like that because SPmon can not be customized
- Perspective says "Close" where SPmon says "Exit"

This simple chart highlights the fact that the 3DigitDisplay window has not changed. It is also the only information that Perspectives still gets directly from the hardmon, without querying the event manager daemon.

Chapter 7. System Partitioning Aid



RS/6000 SP
System Partitioning
Aid

Increasing
Partitioning
Flexibility
on RS/6000 SP

Since partitioning was introduced on the RS/6000 SP with PSSP 2.1, it has been the subject of many workshops, forums, and contacts with customers. These discussions were sometimes a result of a misunderstanding about partitioning, which was supposed to improve system management and application availability. This misunderstanding was induced by two main factors:

1. The System Partitioning mechanism was not explained correctly in the following areas:
   - What was it designed for?
   - In which cases is it valuable?
   - How is it implemented?
2. The implementation of System Partitioning enlarged the scope of flexible system management, compared to the other RS/6000 SP system management mechanisms available at that time, in such a way that customers used partitioning for auxiliary and marginal reasons. But partitioning comes as a whole, so while these customers found what they were looking for, other mechanisms within partitioning were seen as constraints on other aspects of system management.

This is one of the many areas where PSSP V2.2 has made significant improvements. In this presentation, we will see that PSSP Version 2 Release 2 has elements to relieve this frustration. One of these is the *System Partitioning Aid*, which is introduced in the next section.

7.1 System Partitioning Aid Overview

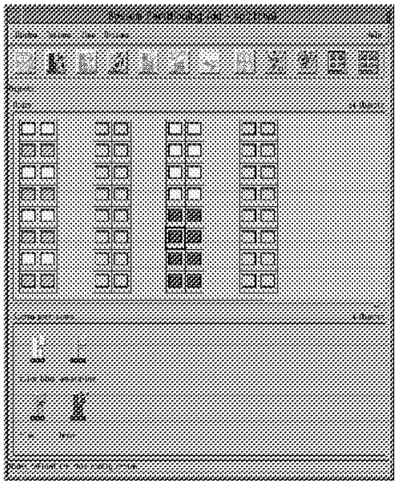
System Partitioning Aid Overview

Partitioning aid

- Help user to cope with real partitioning constraints
- Validate user requirements
- Generate topology files
- Provide performance information
- Graphical User Interface
- Free switchless system of switch partitioning constraints

Planning aid

- Work on current SDR on CWS
  or
- Work on regular RS/6000 to plan system installation



The System Partitioning Aid was developed to increase flexibility in partitioning the RS/6000 SP system.

With PSSP 2.1, the user had a set of predefined configurations that were both switch-oriented and limited in number, for big configurations with a lot of nodes. The rationale was that the number of possible partitioning configurations was huge and exponentially related to the number of nodes. The odds that a given configuration would be used by a customer made it more economically feasible to design tailor-made topology files on request (through an RPQ), than to generate all possible topology files.

However, relying on generating RPQ was not a good solution. The System Partitioning Aid was thus designed as an interface between your requirements and the topology files structure.

The System Partitioning Aid has two main utilities:

- **Partitioning Aid**

  The System Partitioning Aid can be seen as an expert system that includes rules on switch network inter-connection and related switch performances. The user uses a graphical interface to express how the system should be partitioned. The number of partitions and the corresponding node assignment are specified by mouse clicks. On request, the tool checks that the requirements are compatible with the switch fabric and, if that is the case, the tool then generates the files to support system partitioning. Application of partitioning using the generated files is still performed as before (refer to *RS/6000 SP Administration Guide,* GC23-3897).

Some configurations may be allowed even if they do not provide the best performances in communications over the switch. The user can find details about performance in a separate file generated with the topology files for the partitioning layout that is specified. This performance file is also available from GUI screens.

The System Partitioning Aid also allows the relaxing of switch constraints. This allows switchless systems to benefit from some of the independence that partitioning brings (such as independence from OS levels or providing separate views in management tools).

- **Planning Aid**

  The System Partitioning Aid can be installed and used on the Control Workstation of the RS/6000 SP. This frees the administrator from working on the configuration of the current RS/6000 SP, even though it is still an alternative to do so. From the system partitioning viewpoint, any RS/6000 SP hypothetical but realistic configuration can be loaded to help in planning system configuration.

  The tool can even be installed on a non-Control Workstation RS/6000 to investigate partitioning means, or to perform off-line system partitioning topology files design.

The System Partitioning Aid works on a partitioning layout for up to 128 node systems. It is thus applicable for first stage switch and second stage switch systems.

Topology files generated by the System Partitioning Aid can be copied on a RS/6000 SP Control Workstation running PSSP 2.1 and the partitioning layout can be applied.

7.2 Introduction: System Partitioning

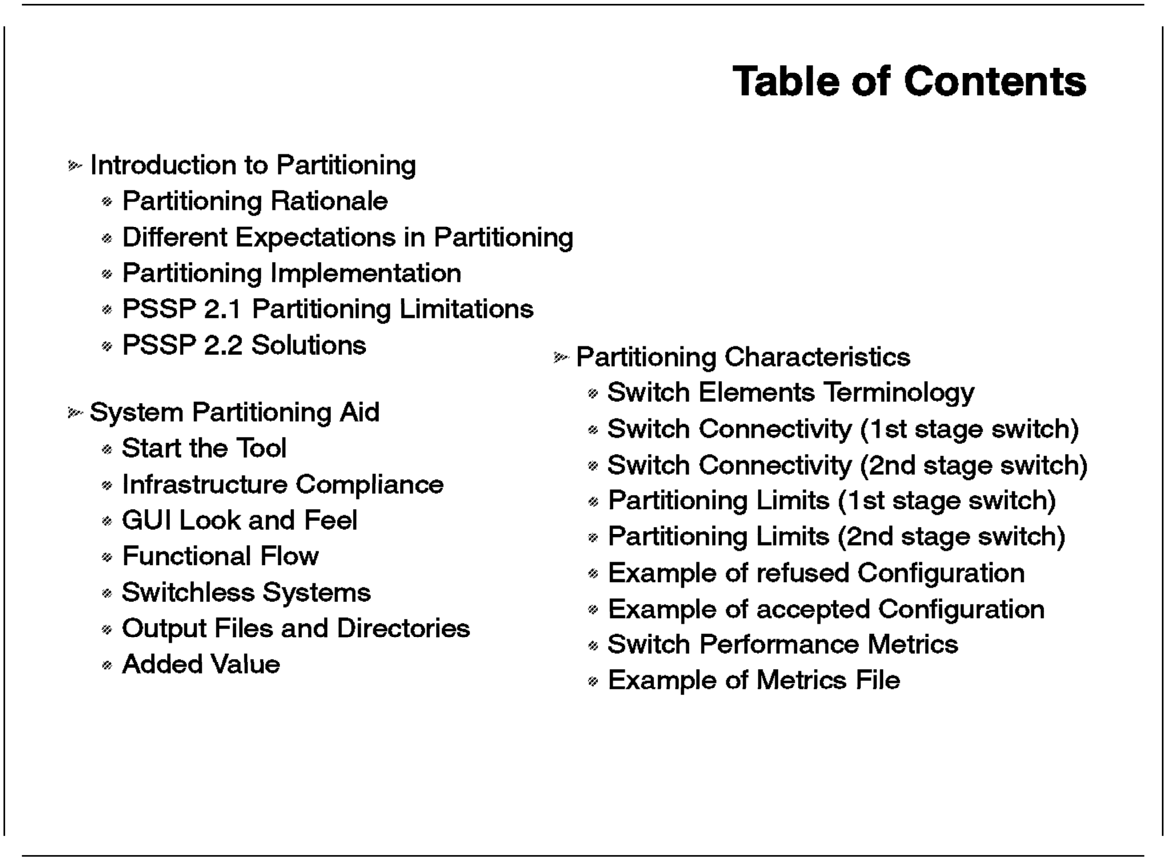


Table of Contents

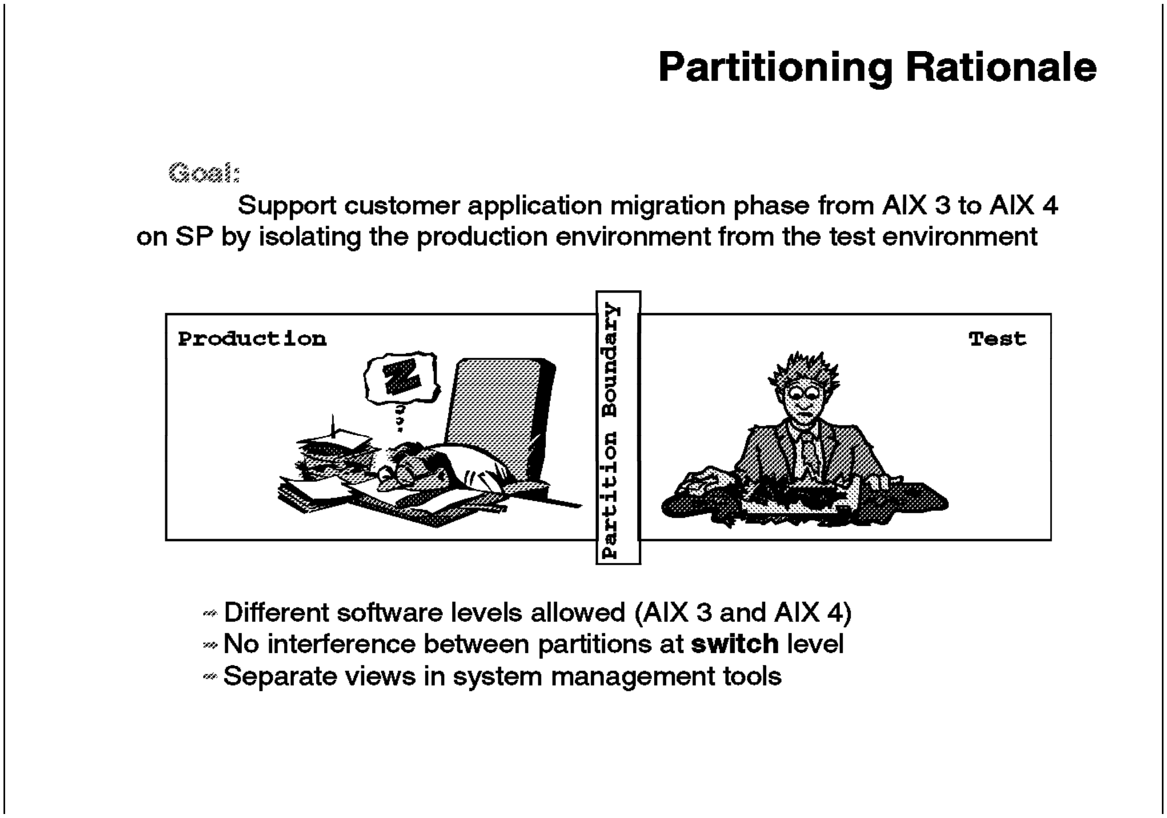
- Introduction to Partitioning
  - Partitioning Rationale
  - Different Expectations in Partitioning
  - Partitioning Implementation
  - PSSP 2.1 Partitioning Limitations
  - PSSP 2.2 Solutions
- System Partitioning Aid
  - Start the Tool
  - Infrastructure Compliance
  - GUI Look and Feel
  - Functional Flow
  - Switchless Systems
  - Output Files and Directories
  - Added Value
- Partitioning Characteristics
  - Switch Elements Terminology
  - Switch Connectivity (1st stage switch)
  - Switch Connectivity (2nd stage switch)
  - Partitioning Limits (1st stage switch)
  - Partitioning Limits (2nd stage switch)
  - Example of refused Configuration
  - Example of accepted Configuration
  - Switch Performance Metrics
  - Example of Metrics File

This introduction positions the System Partitioning Aid, and system partitioning itself, within the wide scope of system management facilities. To understand what PSSP Version 2 Release 2 brings to that area, we cover the following aspects:

- Discussion of the rationale of partitioning in the PSSP 2.1 release
- User′s view of partitioning
- How the implementation of partitioning modified the RS/6000 SP
- Highlighting of PSSP 2.1 limitations in system partitioning and other system management areas
- Positioning of major PSSP Version 2 Release 2 enhancements in system management areas

The sections after this introduction relate to the System Partitioning Aid.

7.2.1 Partitioning Rationale



Partitioning Rationale

Goal:
Support customer application migration phase from AIX 3 to AIX 4 on SP by isolating the production environment from the test environment

- Different software levels allowed (AIX 3 and AIX 4)
- No interference between partitions at **switch** level
- Separate views in system management tools

System Partitioning was introduced on the RS/6000 SP with PSSP 2.1. It was a major PSSP release, because it opened the RS/6000 SP to AIX Version 4 support.

At that time, a large number of RS/6000 SP systems were already installed. They were running parallel applications which took advantage of the High Speed Network (also called Switch), with AIX Version 3 and PSSP 1.2.

To be able to move towards AIX 4, function was required to support several operating systems on the same RS/6000 SP system. It was unrealistic to force users to buy a second RS/6000 SP simply for test purposes. Since the new environment was to be set for test purposes, another function was necessary to prevent disruptive interactions between production and test applications. The main area where such interactions are likely to be disruptive is the switch network.

Hosts that are communicating over a switch′s fabric do not have total freedom to join and leave the communication cluster. In other words, switch adapter configuration is not sufficient for node availability on the switch network, as it is in most of the well-known Local Area Networks. Instead, the switch has to be started. This is usually done with the Estart command.

Running Estart on a cluster causes tremendous disturbances in communications. Although the IP protocol can take advantage of its multi-layer structure to recover from such disturbances, this is not the case of the CSS user space protocol. It is the price paid for being fast.

Apart from switch access disruption, routing is also an area where the unavailability of nodes can cause problems.

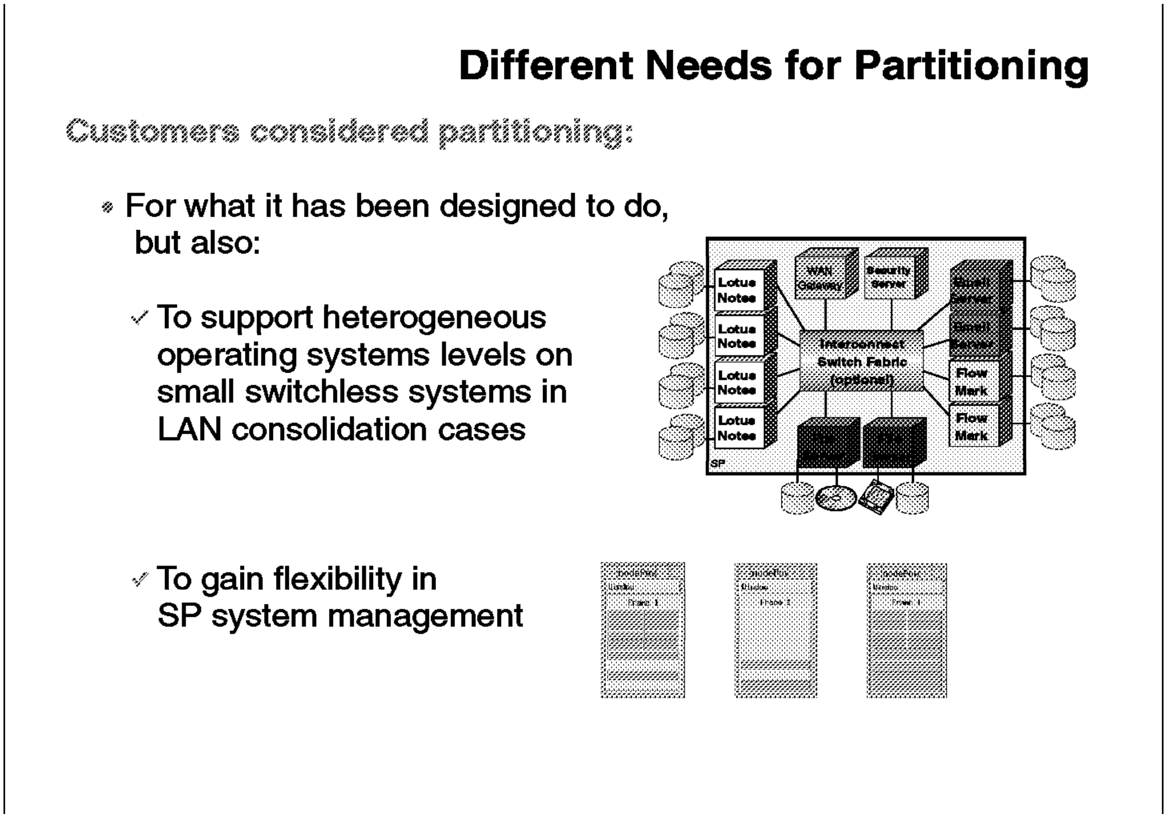
For these reasons, *switch partitioning* was introduced. It also happened that coexistence of operating systems on the RS/6000 SP was implemented, linked to switch partitioning. This fulfilled requirements for both operating system diversity, and non-disruptive interactions. Compliance of system management with partitioning will be covered in detail in section 7.2.2, “Different Expectations in System Partitioning” on page 262.

By using partitioning, support is now possible for:

- A production environment, built on AIX Version 3 applications, running in a first partition
- A test environment, testing portage on AIX Version 4, running in a second partition.

The administrator can start the switch in the test partition without disrupting the communication in the production partition. Separate views in system management tools (SPmon) and command line interface (SP_NAME variable) also prevent unattended disturbances between environments.

7.2.2 Different Expectations in System Partitioning



Different Needs for Partitioning

Customers considered partitioning:

- For what it has been designed to do, but also:
  - ✓ To support heterogeneous operating systems levels on small switchless systems in LAN consolidation cases
  - ✓ To gain flexibility in SP system management

System partitioning rationale, as described in 7.2.1, “Partitioning Rationale” on page 260, was often well understood, and it continues to bring benefits to the migration process to AIX Version 4 on the RS/6000 SP. Nevertheless, it was also often considered for other reasons:

- **Multi-OS Environments:**

  We have already seen that the partitioning requirement was linked to the operating system coexistence requirement. At the time partitioning was implemented (which allowed the RS/6000 SP to benefit from multi-OS levels), it was the only way to install different versions of AIX on different nodes.

  This is the reason why partitioning was sometimes considered as the only possible mechanism for server consolidation environments. In this environment, serial applications were usually running on separate nodes, minimizing the need for a single operating system. Moreover, different levels of operating systems were often needed.

- **Flexibility in Graphical System Management:**

  The implementation of partitioning brought tremendous changes to the RS/6000 SP. This is further described in 7.2.3, “Partitioning Implementation” on page 264. One mandatory change was the ability to interact independently with each system partition from an interface viewpoint. This led to the introduction of views within the SPmon tool, letting the administrator experience more flexibility in system management.

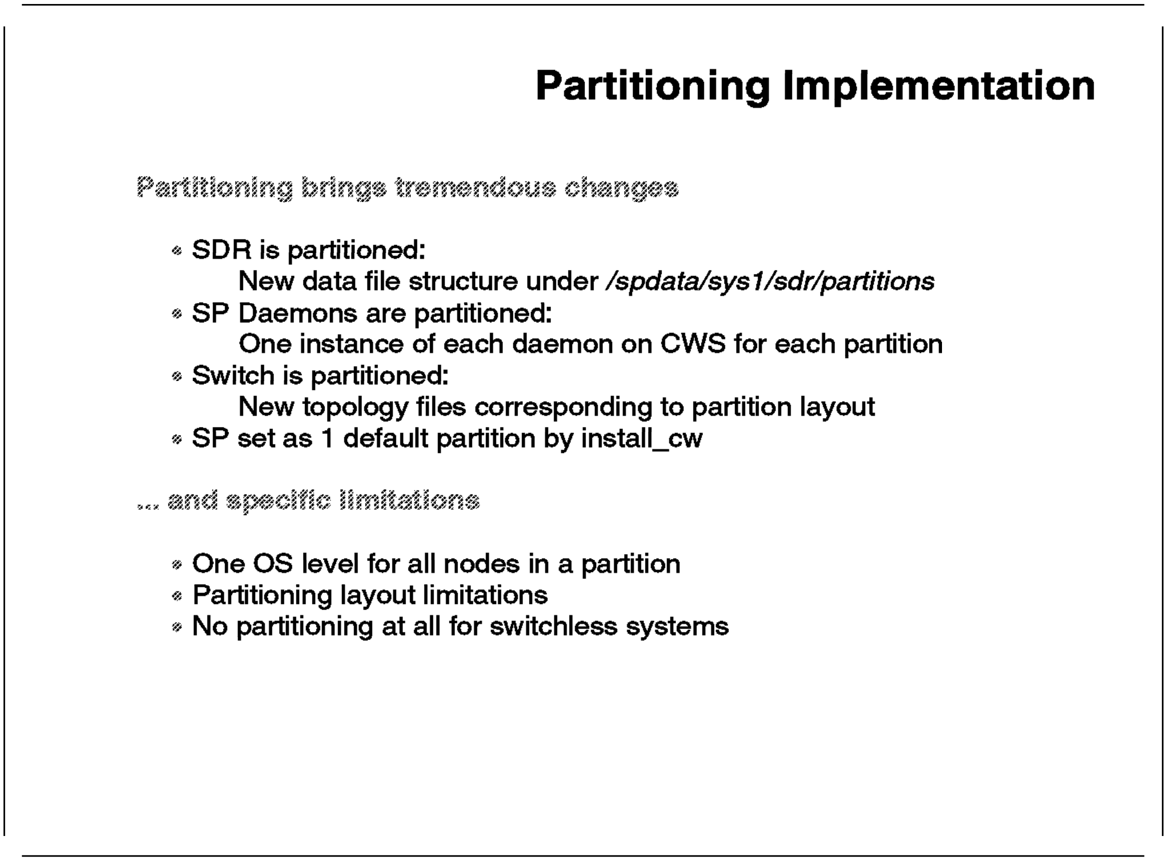
- **Node Addressing Facility:**

  The graphical interface partitioning with separate views was not the only way system management was partitioned. “Command line partitioning” was also introduced through the SP_NAME variable, allowing every PSSP command to only apply on nodes included in the partition addressed by the variable content. This gives more flexibility than the unique WCOLL variable supported by the unique dsh command.

Even if the two last arguments rarely led to partitioning implementation, they gave the user the feeling that system management flexibility could be improved in some ways.

It was different in the case of the multiple operating systems requirement, in which partitioning sometimes led to user frustration.

7.2.3 Partitioning Implementation



Partitioning Implementation

Partitioning brings tremendous changes

- SDR is partitioned:
  New data file structure under */spdata/sys1/sdr/partitions*
- SP Daemons are partitioned:
  One instance of each daemon on CWS for each partition
- Switch is partitioned:
  New topology files corresponding to partition layout
- SP set as 1 default partition by install_cw

... and specific limitations

- One OS level for all nodes in a partition
- Partitioning layout limitations
- No partitioning at all for switchless systems

The implementation of partitioning made PSSP 2.1 RS/6000 SP systems different in some areas:

- The System Data Repository was split into pieces, as the data included was separated between global data regarding the entire system, and partition data concerning nodes in a given partition. A lock mechanism was also implemented to access the global data from various partitions.
- Each partition was addressed by a specific set of PSSP daemons running on the Control Workstation. These daemons talked to the nodes through specific aliases on the Control Workstation network interface. These aliases were defined on the same network as the initial hostname.
- New topology files were necessary to provide different logical views to the communication subsystem, on the physical links available in the switch fabric. The dedicated links are different for each partitioning layout, leading to a large number of necessary topology files. The set of generated files for a given partitioning configuration is thus called the *partition layout*. This term tends to replace the unique *topology* term used so far.
- Even if partitioning was not explicitly used on the RS/6000 SP system (that is, even if multiple partitions were not used), by default a single global partition is still created within the `install_cw` script during the early stages of system installation. The default partition encompassed all nodes, and its definition was necessary for further system management.

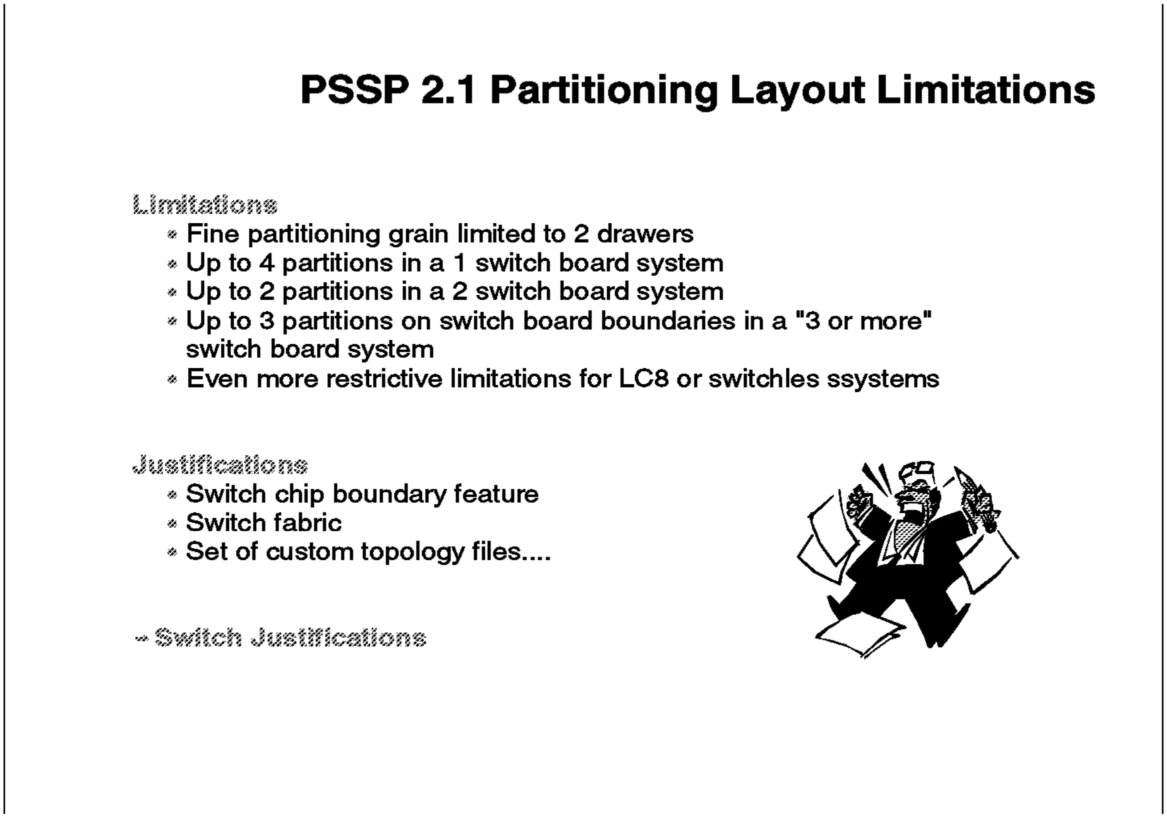
The consequences of partitioning range from the need for space in /spdata for topology files, to multiple daemons taking memory and CPU resources on the Control Workstation, or the necessity to have additional IP addresses. These are fairly acceptable when partitioning is a must, because they are necessary to its full implementation. They are considered to be constraints only by users wanting to benefit from the side “comfort” improvement brought by partitioning.

The implementation consequences previously mentioned could have been seen as minor disadvantages by most users. More significant are the limitations that came with partitioning in PSSP 2.1:

- There was still only one AIX level allowed for all the nodes in a given partition.
- The number of available layouts for system partitioning was limited in many ways, as described in 7.2.4, “PSSP 2.1 Partitioning Limitations” on page 266. When combined with the former limitation, the user had little choice in the location of nodes for a given operating system level.
- Finally, the most dissuasive limitation is that partitioning was implemented close to the RS/6000 SP hardware, especially the RS/6000 SP switch. Consequently, switchless systems could be partitioned the same way as a corresponding switched system of the same size. The Low Cost 8 switch in its HiPS version cannot be partitioned, but it can accept two partitions in its SPS version. The short frame (Low boys) systems without switches follow the same constraint as SP-8 switched systems. This tends to be the case with most small systems used within the server consolidation market, in which the coexistence of multiple operating systems within an RS/6000 SP system was the strongest. This is the area where frustration was the highest.

Section 7.2.5, “PSSP 2.2 Solutions for System Management Flexibility” on page 268 describes the PSSP Version 2 Release 2 solutions to these limitations and constraints.

7.2.4 PSSP 2.1 Partitioning Limitations



PSSP 2.1 Partitioning Layout Limitations

Limitations

- Fine partitioning grain limited to 2 drawers
- Up to 4 partitions in a 1 switch board system
- Up to 2 partitions in a 2 switch board system
- Up to 3 partitions on switch board boundaries in a "3 or more" switch board system
- Even more restrictive limitations for LC8 or switchles ssystems

Justifications

- Switch chip boundary feature
- Switch fabric
- Set of custom topology files....

→ Switch Justifications

We have seen that partitioning is related to the switch network, and this is also true when choosing a layout for setting system partitions. Within PSSP 2.1, the administrator had some limitations in the way partition layout could be set. Some of these limitations are hardware-dependent and are thus justified, because partitioning was to be implemented with limited changes to the hardware. But other limitations have less justification, as follows:

- The switch chip boundary limitation is mandatory and is likely to be in the partitioning path for some time. Elements on switch components and connectivity are given in 7.4.1, “Switch Elements Terminology” on page 288 and 7.4.2, “Switch Connectivity Characteristics” on page 290.

  The consequence of this limitation affects partitioning granularity. Because each switch chip is related to two fixed drawers within an RS/6000 SP frame, nodes belonging to these drawers will never be in separate partitions. Partitioning can only be achieved by building groups of these drawers and corresponding nodes. This leads to the maximum number of four partitions in a single frame RS/6000 SP system.

- The switch chips involved in the former limitation in partitioning granularity were the chips connected to the switch adapters in the nodes. There are other more subtle limitations, described in 7.4.2.2, “Second Stage System” on page 292, that come from other elements of the switch fabric, especially those allowing inter-frame connectivity on the switch. These limitations are also likely to be persistent.

- The previously-mentioned limitations were related to hardware. Other limitations were related to the material provided within the PSSP files: the topology files. Four factors determined the makeup of the topology file set provided:

  1. The number of possible combinations was too high to allow the writing of every topology file for every layout. Thus the size of the set has to be limited.

  2. Partitioning implementation by customers that followed the rationale of supporting one production environment and one test environment. Thus the greater need was in a two-partition layout.

  3. The installed base of RS/6000 SP was showing a peak number for single or two-frame systems, which brought the focus onto these systems.

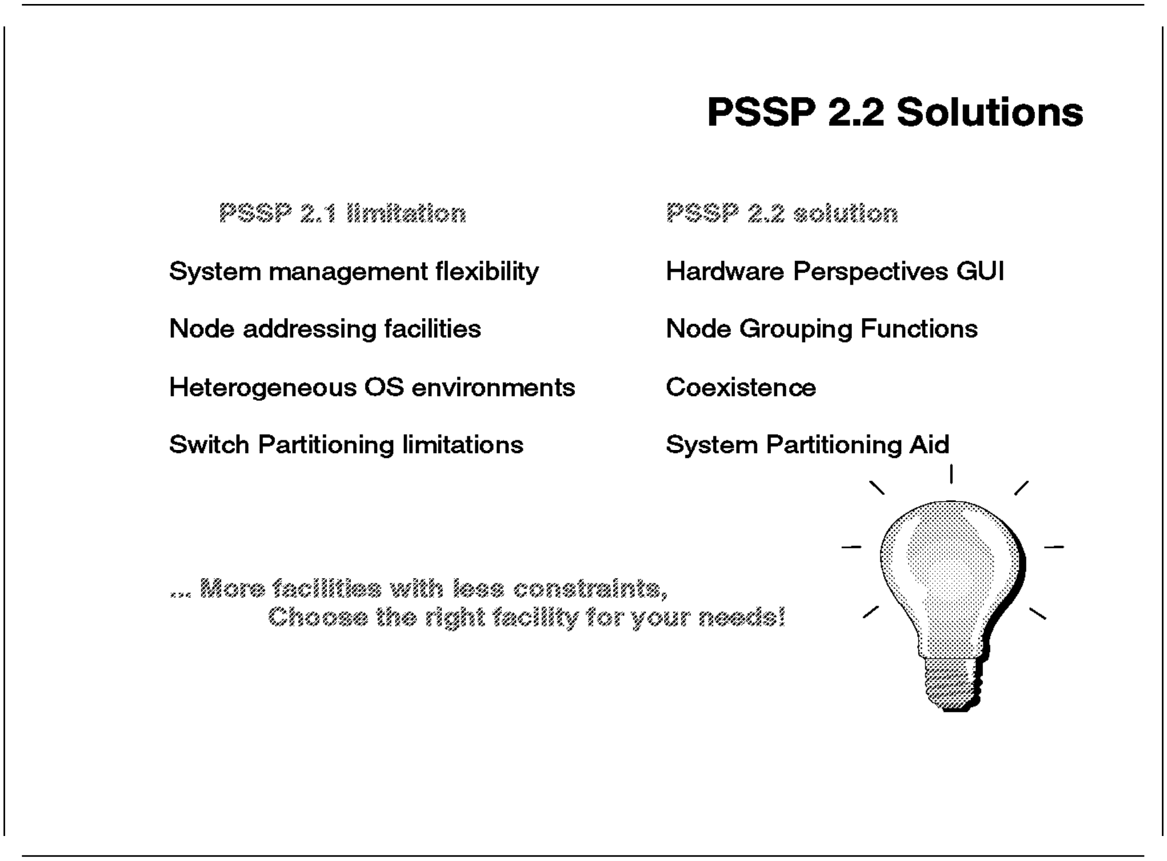
  4. Users of huge systems may not need fine grain partitioning.

  These factors led to the delivery of a set of system partition layouts which allowed any combination of chip boundary using two partitions for two-frame systems, and any combination of frame boundary using three partitions for systems with more than two frames.

  You could still obtain a specific layout for unsupported configurations by special request (through an RPQ), for no extra charge.

- These limitations were mostly switch limitations, and they were even greater on LC_8 switches, since partitioning used to be impossible on these. More significant was the fact that even without having a switch, you were still subject to the same limitations.

7.2.5 PSSP 2.2 Solutions for System Management Flexibility



PSSP 2.2 Solutions

PSSP 2.1 limitation	PSSP 2.2 solution
System management flexibility	Hardware Perspectives GUI
Node addressing facilities	Node Grouping Functions
Heterogeneous OS environments	Coexistence
Switch Partitioning limitations	System Partitioning Aid

... More facilities with less constraints,
Choose the right facility for your needs!

PSSP Version 2 Release 2 provides solutions to satisfy the needs of users who had considered partitioning as an alternative for all the requirements mentioned in 7.2.2, "Different Expectations in System Partitioning" on page 262:

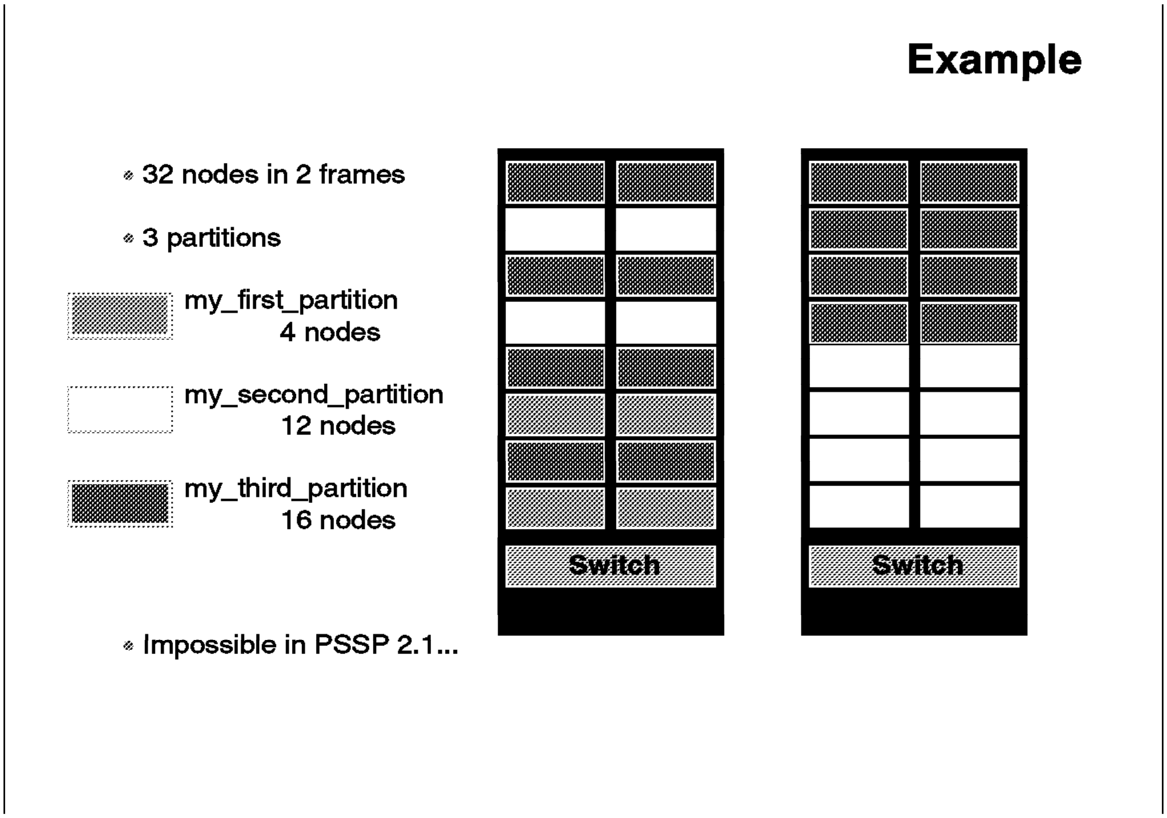
- System management flexibility is increased when the Perspectives GUI interacts with most of the other PSSP facilities. See Chapter 7, "System Partitioning Aid" on page 255 for information on the Perspectives GUI.
- Node addressing is facilitated by the introduction of node grouping functions. These functions allow the manipulation of multiple nodes working collectively, and expand their scope beyond the simple dsh command. See 6.3.4, "Node Grouping" on page 226 for information on node grouping.
- The coexistence of different levels of the AIX operating system is now supported within one system partition, introducing a new degree of freedom in the OS allocation to nodes.
- In cases where partitioning is a requirement, some of its previous limitations (such as those related to the limited number of layouts), have been eliminated. There is no longer a limitation in partitioning switchless systems. In addition, by using the System Partitioning Aid, users can express their requirements in a user-friendly graphical way and have them validated by the system.

  The following sections are devoted to the functions and related concepts of the System Partitioning Aid.

With the set of available resources splitting mechanisms, each user should be able to find and select the most appropriate one. Moreover, the mechanisms themselves have increased flexibility, which improves their suitability and interoperability.

7.3 System Partitioning Aid Functions

This section introduces the System Partitioning Aid functions. To illustrate some of the aspects presented, we use an example of a given partitioning requirement. This example is presented in the following figure:



Example

- 32 nodes in 2 frames
- 3 partitions

▨ my_first_partition 4 nodes

□ my_second_partition 12 nodes

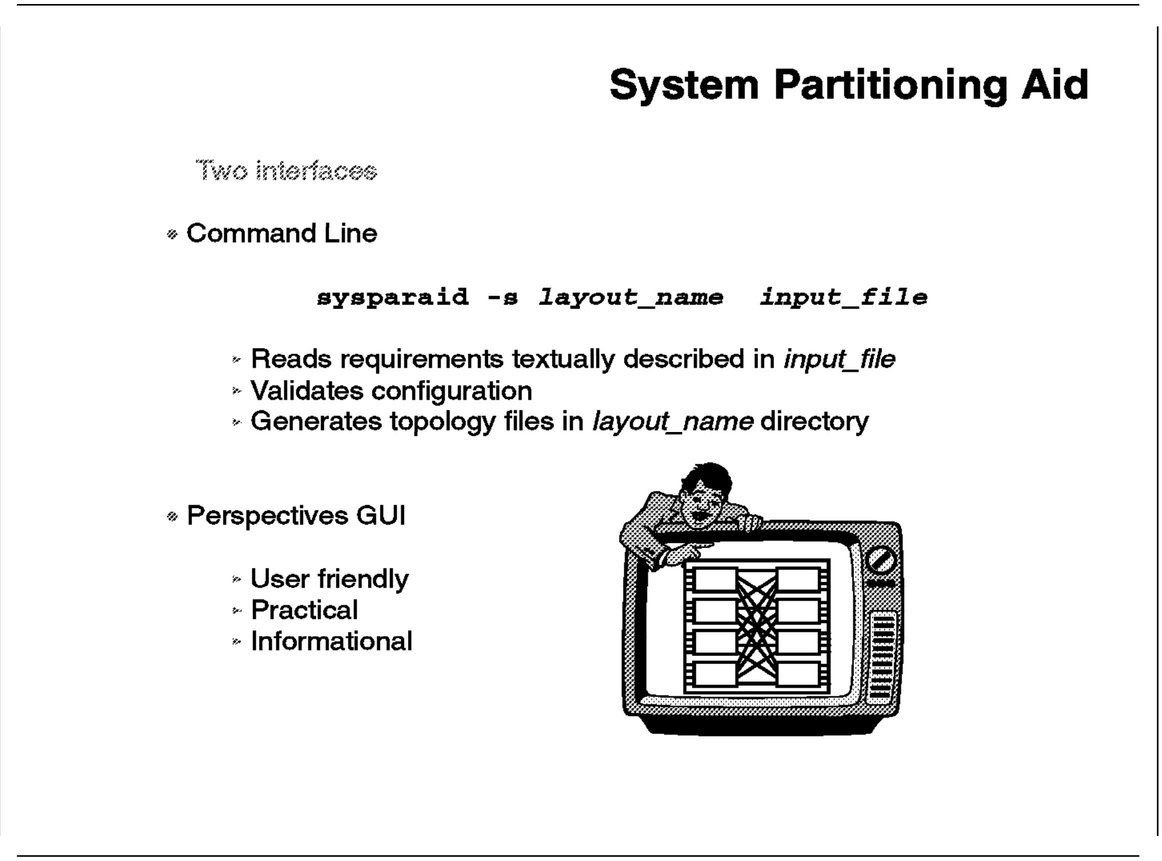
■ my_third_partition 16 nodes

- Impossible in PSSP 2.1...

The partitioning of a two-frame RS/6000 SP system in three different partitions is a requirement that could not be satisfied by PSSP 2.1. Topology files for two-switch systems were only provided to support two partitions.

In 7.4, “Switch Partitioning Additional Information” on page 288, we present additional information on partitioning to understand issues and concepts manipulated by the System Partitioning Aid.

7.3.1 System Partitioning Aid



System Partitioning Aid

Two interfaces

- Command Line

```
sysparaid -s layout_name  input_file
```

  - Reads requirements textually described in *input_file*
  - Validates configuration
  - Generates topology files in *layout_name* directory

- Perspectives GUI
  - User friendly
  - Practical
  - Informational

The System Partitioning Aid aims at simplifying the RS/6000 SP partitioning. It is an interface between the user requirements in terms of partitioning layout, and the topology files. The latter are mandatory to access the switch fabric for application communication.

The System Partitioning Aid functions can be invoked from the command line or through a graphical user interface built with the Perspectives look and feel introduced for system management tools by PSSP 2.2.

7.3.1.1 Command Line Access

The commands of the System Partitioning Aid are included in the ssp.top fileset. This fileset has ssp.basic as prerequisite in the installp mechanism. The fileset ssp.basic is actually needed to use the System Partitioning Aid, because it holds the basic topology files for a system without partitions. These topology files are used by the System Partitioning Aid as input to generate the partitioned topology files.

The following table summarizes the space needed for the files when installing and using the command version of the System Partitioning Aid:

Table 8. System Partitioning Aid Command Line Access. Space needed to install filesets containing command line interface for System Partitioning Aid.

Filesystem	Space Needed	Fileset
/	60 Kb	ssp.basic
/usr	21 Mb	ssp.basic
	4 Mb	ssp.top
/var	30 Kb	ssp.basic
/spadata	700 Kb	ssp.basic
	20 Mb	ssp.top

It is advised to forecast available space in /spdata to store the generated topology and information files.

The installation of ssp.top creates the sysparaid command in /usr/lpp/ssp/bin. To use this command, you have to build a text file describing the partition configuration you want to use. You can copy and modify the inpfile.template located in /spdata/sys1/syspar_configs/bin.

In the example supporting this chapter, the input file would be:

```
/*****************************************************************************/
Number of Nodes in System: 32
Number of Frames in System: 2
Frame Type: tall
Switch Type: SP
Number of Switches in Node Frames: 2
Number of Switches in Switch Only Frames: 0
Number of System Partitions: 3
Node Numbering Scheme: switch_port_number
System Partition Name: my_first_partition
Number of Nodes in System Partition: 4
/* List of nodes in system partition */
0
1
4
5
System Partition Name: my_second_partition
Number of Nodes in System Partition: 12
/* List of nodes in system partition */
8
9
12
13
16 - 23
System Partition Name: my_third_partition
Number of Nodes in System Partition: 16
/* List of nodes in system partition */
remaining_nodes
```

Nodes can be specified by node_number or switch_port_number. The term *switch_port_number* actually relates to the term *switch_node_number* commonly used before in the RS/6000 SP documentation. The term switch_port_number will progressively replace switch_node_number within all PSSP documentation. While both addressing modes (node_number and switch_port_number) are permitted for partitioning a system defined by an SDR, switch_port_number is the only allowed choice when running the tool without an SDR. Only one addressing mode is allowed in the same input file.

The sysparaid command is then called by:

```
# sysparaid -s my_layout my_input_file
```

If the configuration described in my_input_file is valid, the command generates the corresponding topology files under layout.my_layout. Note that my_layout can be a generic name that is the entry point created for the required layout within the /spdata/sys1/syspar_config directory structure. It can also be a fully-qualified path name under which the layout will then be created. For details on what files are generated see 7.3.7, “System Partitioning Aid Output Files” on page 284. If the configuration or the format of my_input_file is not valid, the command ends with the corresponding error message.

For more information on the sysparaid command refer to *RS/6000 SP Command and Technical Reference,* GC23-3900.

7.3.1.2 Perspectives GUI Access

The files for the System Partitioning Aid graphical interface are included in the ssp.top.gui fileset included in ssp.top. It also requires ssp.gui, which has ssp.clients as a prerequisite in the installp mechanism. The ssp.client is needed because some libraries used by the graphical tool are in that fileset.

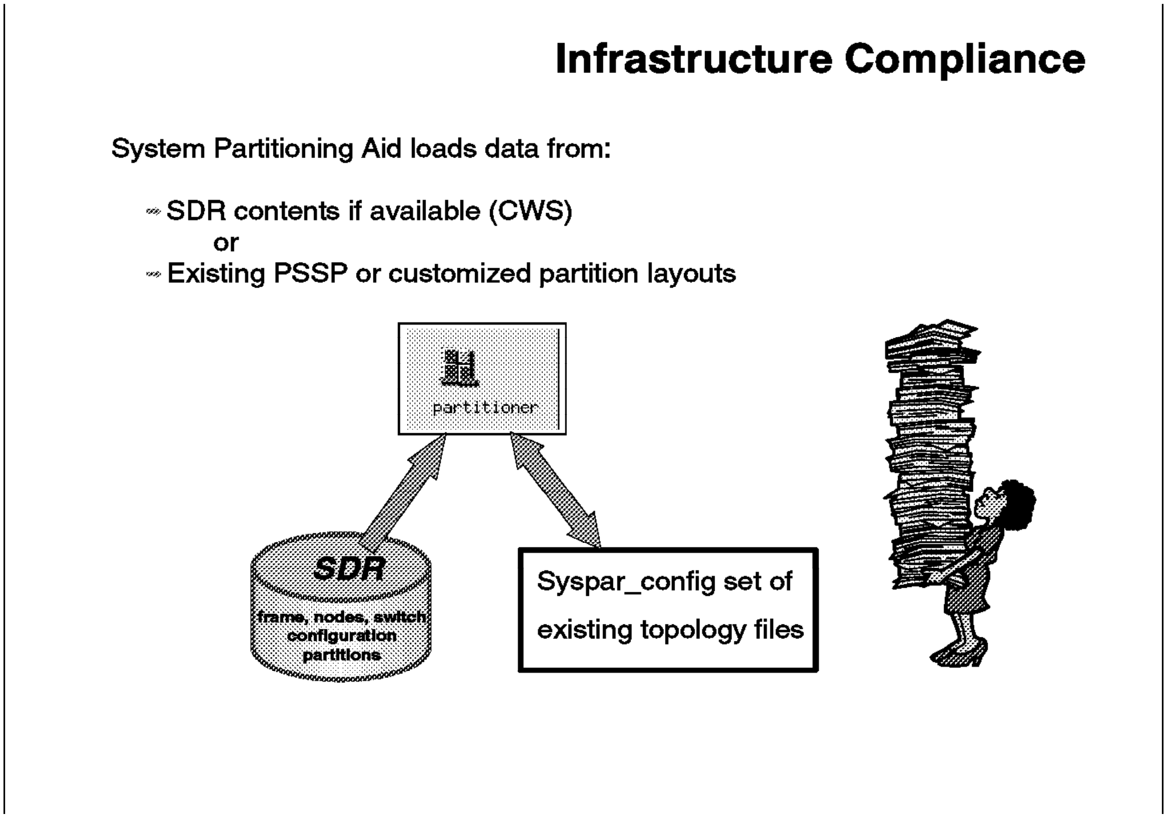
To install and use the graphical version of the System Partitioning Aid, the additional space needed for the files are summarized in the following table. The ssp.top and ssp.basic filesets still have to be installed as described in 7.3.1.1, “Command Line Access” on page 271.

Table 9. System Partitioning Aid GUI Access. Space needed to install filesets containing graphical interface for System Partitioning Aid.

Filesystem	**Space Needed**	**Fileset**
/	5 Kb	ssp.gui
	30 Kb	ssp.clients
/usr	26 Mb	ssp.gui
	12 Mb	ssp.clients
/spadata	6 Kb	ssp.clients

The System Partitioning Aid graphical tool can be launched from the Perspectives launch pad or by typing the spsyspar command on the command line. It gives a user-friendly graphical interface to express the partitioning requirements and visualize the generated files. It can be used to generate a given partitioning layout and corresponding topology files, as well as to investigate the partitioning mechanism and layout possibilities.

7.3.2 System Partitioning Aid Infrastructure



Infrastructure Compliance

System Partitioning Aid loads data from:

- SDR contents if available (CWS)

  or

- Existing PSSP or customized partition layouts

partitioner
SDR
frame, nodes, switch configuration partitions
Syspar_config set of existing topology files

The System Partitioning Aid can be used on an RS/6000 SP Control Workstation or on a normal RS/6000. On a normal RS/6000, because no PSSP is present, the filesets for the required interface, listed in 7.3.1, "System Partitioning Aid" on page 271, must be installed. However, if used on the Control Workstation, the System Partitioning Aid does not necessarily work on the included SDR. Thus the System Partitioning Aid can be used on the Control Workstation after PSSP installation, before the entry of any RS/6000 SP configuration in the SDR.

When launched, the System Partitioning Aid checks the availability of an RS/6000 SP configuration in an active SDR. If successful, the SDR configuration is displayed on the panes of the tool. If not, an informational message is prompted and the tool starts with blank panes.

Loading SDR configurations gives administrators the ability to work on their own systems with the corresponding node types (thin, wide, or high). The System Partitioning Aid loads data from the SDR, but the results of partition definition are not applied in the SDR. The System Partitioning Aid stores files in the /spdata/sys1/syspar_config directory structure. The application of partitioning from these files is still following the usual process available through SMIT.

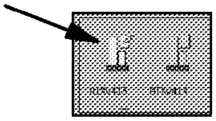
At any time, with or without SDR, the user can load any available partitioning layout from the syspar_configs structure. This layout replaces the one formerly displayed in the tool. This layout can be a PSSP predefined layout or a layout generated with the System Partitioning Aid. For more information on syspar_configs directory structure and generated files, see 7.3.6, “System Partitioning Aid Output Directories” on page 282.

Installation of the System Partitioning Aid on a regular RS/6000 can be used to forecast further partitioning disconnected from the Control Workstation. It can also act as an education tool on partitioning, or as a planning tool to help in the definition of a system configuration.

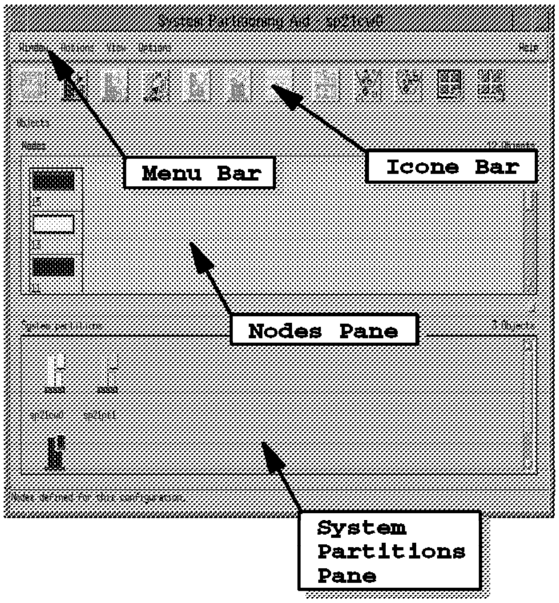
7.3.3 Graphical User Interface Use

GUI's Look and Feel

- Perspective's look and feel
- Use of the Partitions Pane
  - To create system partitions
  - To select a partition and query information
  - To set the active partition



- Use of the Nodes Pane
  - To select nodes and query information
  - To select nodes for partition assignment
- Use of both for general actions



The main window of the System Partitioning Aid follows the specifications of every Perspectives tools delivered with PSSP Version 2 Release 2. General information on Perspective GUI elements is given in Chapter 6, “Perspectives” on page 205. The window has the following elements from the bottom to the top:

- An optional help message area that displays short messages about the purpose of the object that the mouse is pointing to.
- A set of panes where elements, on which actions will be performed, are represented. The elements manipulated by the System Partitioning Aid are nodes and partitions. The two corresponding panes are displayed throughout the use of the tool and cannot be hidden.
- A row of icons that are shaded or active, depending on the selected pane or selected objects in the pane. These icons are action icons that launch System Partitioning Aid tools on the selected objects, or general purpose icons (such as select or sort) that are available in all the Perspectives tools.
- A menu bar made of the four conventional menu entries:
  - *Window*, to manage the window itself. It only contains the exit of the System Partitioning Aid.
  - *Actions*, to launch actions on items selected in the active pane. Each pane has a sub-menu containing both entries for pane-specific actions or general actions. General actions are located in all the sub-menus, but

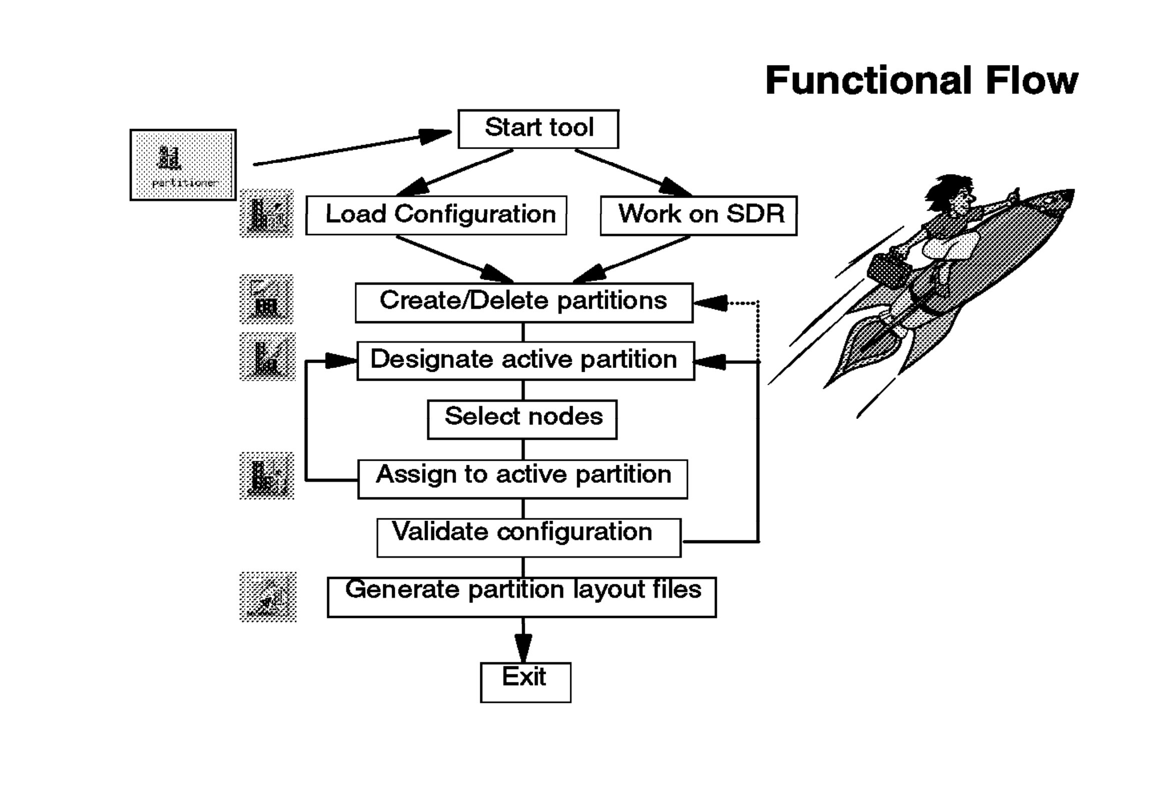
entries of a given sub-menu are usually only available when the corresponding pane is active. Otherwise the entries are shaded.

– *View*, to manage the way items are displayed in the panes and to access information related to these items.
– *Options*, to change and save the GUI elements of the main window, such as fonts, colors, and layout.

In the case of the System Partitioning Aid, samples of general actions are the validation or the generation of files for the displayed partitioning configuration. Samples of item-dedicated actions are the affectation of nodes to a partition or a query of information for a partition.

The assignment of nodes to a partition is using the concept of active partition. Only one partition is designated as active at a given moment. To assign a node to a partition, the partition has to be designated as active first. The active partition in the partitions pane has a flash on the top of the displayed item. Once the partition is active, the nodes have to be selected from the nodes pane and assigned to the active partition.

7.3.4 Functional Flow



Functional Flow

The functional flow of the command line interface of the System Partitioning Aid is fairly simple and has been explained in section 7.3.1, “System Partitioning Aid” on page 271.

This section focuses on the functional flow of the graphical tool. It is straightforward and most actions can simply be activated by icons shown on the left of the action on the foil.

Once the tool is launched, the SDR configuration, if available, is displayed. If it is not available, or it is not the purpose of the System Partitioning Aid′s current session, the user can load a configuration from the syspar_configs directory tree.

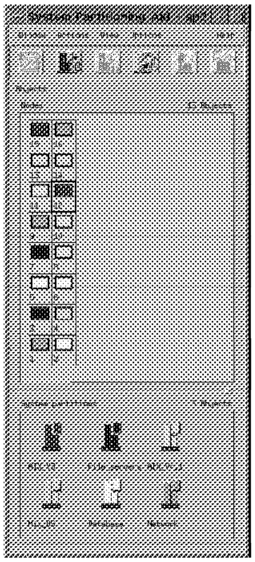
The initial configuration has a given number of system partitions. The definition of the partitioning can consist of changing the partition assignments of nodes or of creating new system partitions and assigning nodes to them.

The initial task is to define the number of needed system partitions in the partitions pane by adding or deleting partitions. Then node assignments can be performed. This is done by selecting the partition to which new nodes will be assigned and by designating that system partition as active. Only one partition is active at a given time. The node assignment task is independent from the partition choice because nodes are always assigned to the active partition. Once the destination partition is designated as activate, nodes are selected in the nodes pane and assigned to that partition. This process of changing the active partition and selecting and assigning nodes takes place as long as node reassignments are necessary.

Once the required configuration has been described, it must be validated. If it is refused by validation, the user must identify and understand the reason for the refusal, by using the message returned by the validation tool. The user must then repeat the former node reassignment process. If it is validated, the configuration can then be stored. When all tasks are done, the tool can be closed.

7.3.5 System Partitioning on Switchless Systems

Switchless Systems



- No more switch constraints in node to partition assignments
- Generation of description and empty topology file
- No switch performance or switch chip data
- If applied on a system with a switch, there is no more access to the switch in either CSS or IP mode.... no more topology file
- Useful for:
  - Real switchless systems in LAN consolidation
  - Total freedom in node OS levels assignment
  - More than 2 operating systems in the cluster

An option of the System Partitioning Aid allows the user to deactivate the switch constraints applied on system partitioning. This is done for the current manipulated configuration. If another configuration is loaded from syspar_configs, assuming it is not a previously saved switchless layout, the switch constraints will be reactivated.

When active, the switch constraints are visible in three ways:

- During node-to-partition assignment, each time a given node is selected to be assigned to a given partition, all the nodes belonging to the same switch chip are also reassigned to the same partition.
- Validation of the configuration is performed against switch constraints.
- When saving the configuration, switch topology files, information on switch chip layout, and switch performances files are created.

Deactivating the switch constraints causes to the following results:

- During node-to-partition assignment, an individual node can be selected and assigned to a given partition without any consequences for other nodes.
- No real validation is performed, even if the option is still active.
- When saving the configuration, empty switch topology files are created, and information on switch chip layout and switch performances files is not created. Only information on node-to-partition layout is created.

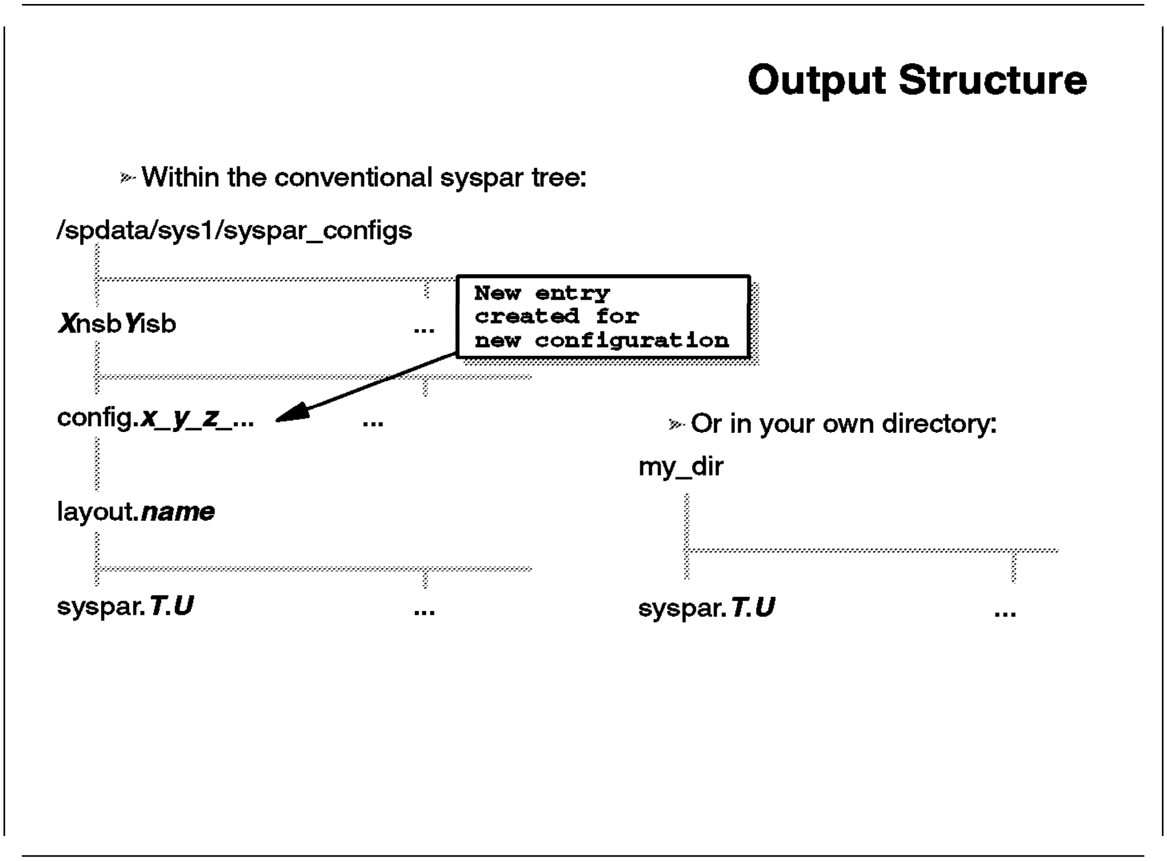
The generated partitioning layout can be applied to the system using the same process as a conventional switch layout. If applied on a system including a switch, the access to the switch network will not be possible because empty topology files are provided. The switch is unreachable in either IP or CSS mode, which makes useless the switchless partitioning of a system with a switch. The application of a switchless partition layout on an RS/6000 SP system is transparent to the spapply_config script used on that occasion. The SDR is partitioned and the daemons are recreated in the same way. Switch operations (primary node redefinition, topology files annotation) are performed at the end of the script. If a switch is detected but a switchless configuration is applied, the Eannotator command will return an error:

```
Eannotator: 0028-142 Topology file specified not valid for current
partition
```

which is normal, because the topology file is empty. The switch will thus not be configured in the SDR and will not be usable.

Switchless partitioning is useful for non-switched systems that may benefit from partitioning goodies, such as SPmon separate views or SP_NAME variable on the command line. It is even mandatory to support more than two levels of AIX and PSSP in the RS/6000 SP cluster, as only two are currently supported within a given partition by the coexistence mechanism of PSSP 2.2. Its greater use is thus in the relaxing of switch chip boundaries for assigning operating systems levels.

7.3.6 System Partitioning Aid Output Directories



Output Structure

➢ Within the conventional syspar tree:

/spdata/sys1/syspar_configs

*X*nsb*Y*isb ...

New entry created for new configuration

config.*x_y_z_*... ...

layout.*name*

syspar.*T.U* ...

➢ Or in your own directory:

my_dir

syspar.*T.U* ...

When saving the files for a valid configuration built with the System Partitioning Aid, you have two choices:

- You can specify a single name for the layout. The system then studies the required configuration and builds a configuration directory name based on the number of partitions and related nodes. For example, it gives config.x_y_z a layout with 3 partitions containing x, y, and z nodes, respectively. This directory is created within the syspar_config tree in /spdata/sys1 under the generic topology directory for the system configuration. These generic topology directories are brought by the PSSP, and they have a naming convention following the basic topology files naming convention (XnsbYisb). Under the configuration directory, the tool creates another directory named *layout.name*, where *name* is the single name you provided in this alternative.
- You can specify the full pathname of a directory. The system then extracts the basename of the path and checks the directory. If the directory is present, an error message is displayed. Otherwise a directory entry is created. This entry is the starting directory of the files created by the System Partitioning Aid, and its contents are described in 7.3.7, “System Partitioning Aid Output Files” on page 284.

The location where the tool stores the files in /spdata/sys1/syspar_configs is important. If you choose the second alternative and store the files in a different location, and you want to retrieve the files in a later session of the System

Partitioning Aid, you will have to manually move them into the /spdata/sys1/syspar_configs tree. This is the only one browsed by the System Partitioning Aid layout retrieval tool. Only layouts under /spdata/sys1/syspar_configs can be applied to partition a system.

Saving files out of the syspar_configs directory tree may be useful because a forced reinstallation of the PSSP will lead to a refresh of the syspar_configs directory tree. The new files created by the System Partitioning Aid will then be lost.

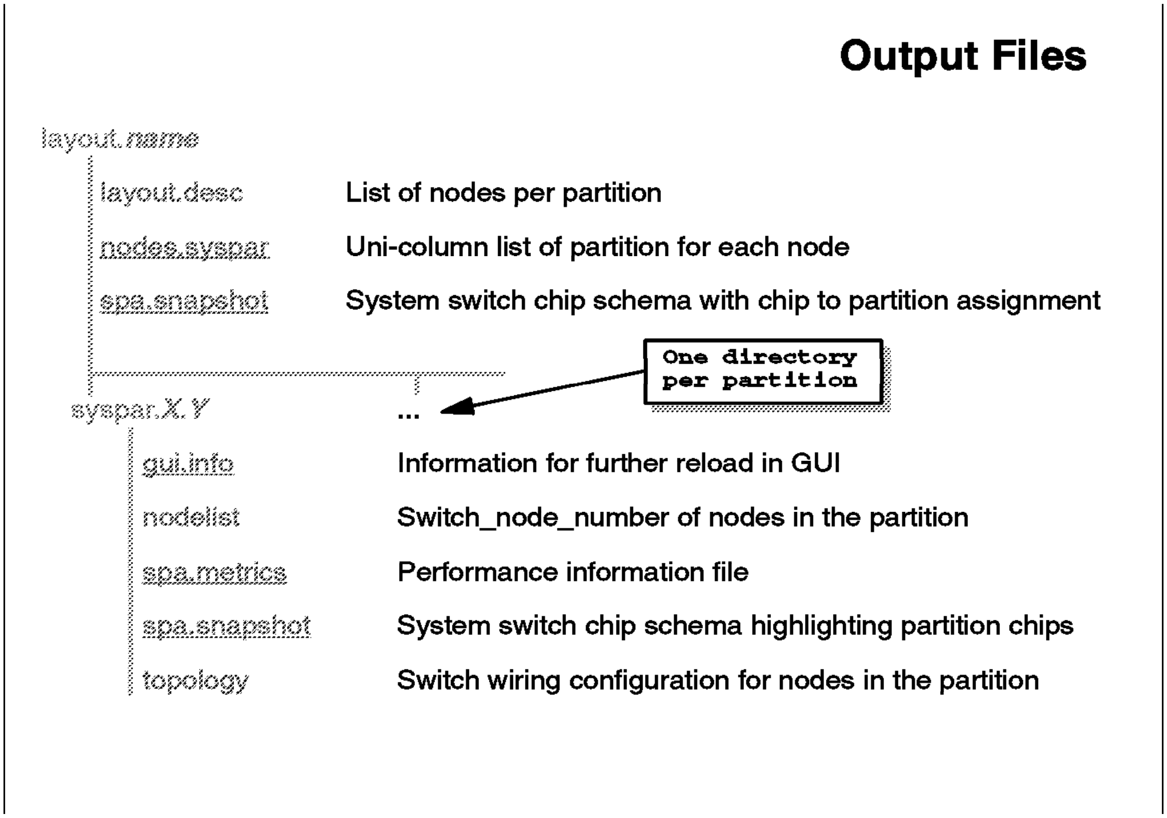
In our initial example, the files generated by the command

```
# sysparaid -s my_layout my_input_file
```

are placed under the directory

```
/spdata/sys1/syspar_configs/2nsb0isb/config.4_12_16/layout.my_layout
```

7.3.7 System Partitioning Aid Output Files



Output Files

layout.*name*

layout.desc	List of nodes per partition
nodes.syspar	Uni-column list of partition for each node
spa.snapshot	System switch chip schema with chip to partition assignment

syspar.*X.Y* ...

One directory per partition

gui.info	Information for further reload in GUI
nodelist	Switch_node_number of nodes in the partition
spa.metrics	Performance information file
spa.snapshot	System switch chip schema highlighting partition chips
topology	Switch wiring configuration for nodes in the partition

Files created when a partitioning layout is saved by the System Partitioning Aid are layout-specific files or partition-specific files.

The layout directory is always named layout.user_specified_name, as seen in 7.3.6, "System Partitioning Aid Output Directories" on page 282. This directory contains layout-specific files. It also contains one directory for each partition named syspar.X.Y, where X is the partition number and Y is the partition name specified in the System Partitioning Aid when the partition was created. The partition-specific files are created in these partition directories.

New files introduced by the System Partitioning Aid are underlined on the foil. They are:

- **spa.snapshot**

  Snapshot files, spa.snapshot, present a global view of the entire system from the switch chips′ viewpoint. The layout snapshot gives the partition number for each switch chip of the system. Each partition snapshot is presented from the partition viewpoint, which means that only chips belonging to the partition are represented.

- **spa.metrics**

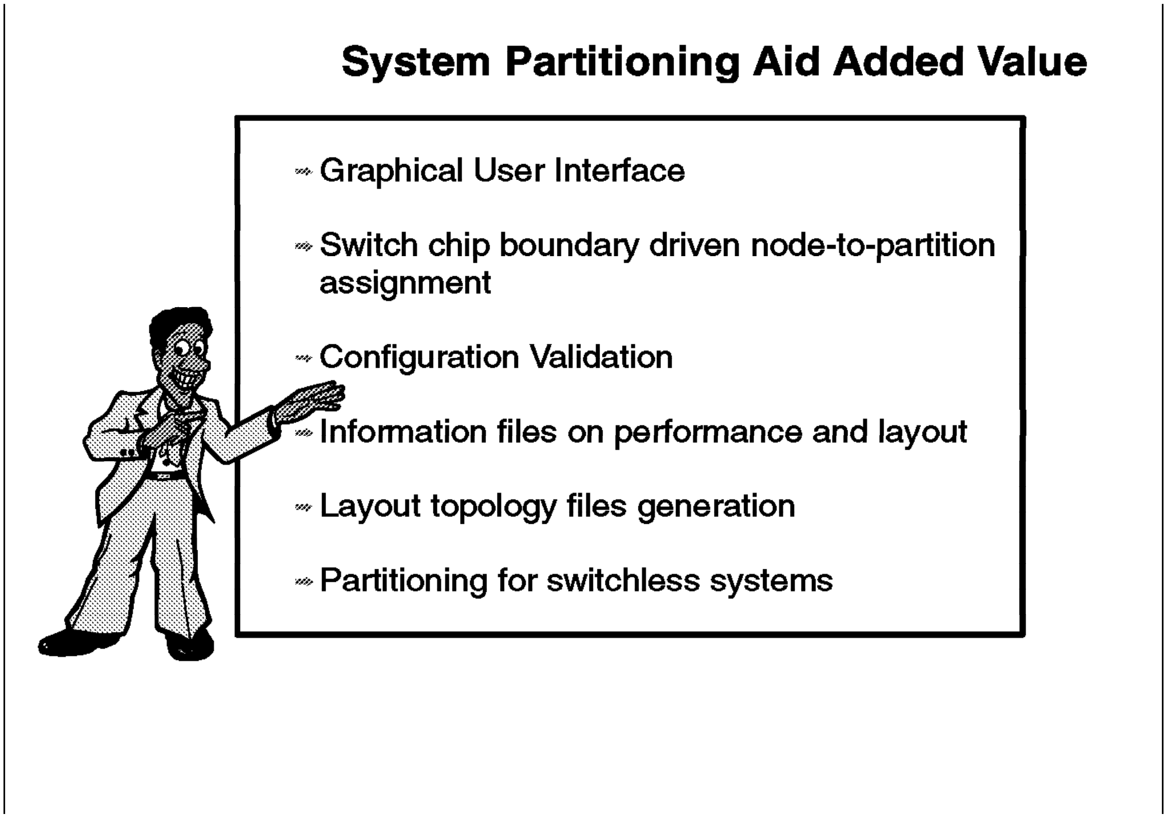
  The partition-specific performance file, spa.metrics, indicates the level of performance over the switch that can be expected after partitioning. Figures are usually expressed in percentages of the corresponding information on the same non-partitioned system.

- **gui.info** and **nodes.syspar**

  Graphical partition-specific information, which is used by the Perspectives interface itself to display further retrieving of the layout. gui.info is for graphical layout and nodes.syspar is for the System Partitioning Aid to detect that the required layout is already available within the syspar_configs directory tree.

Details on snapshot files are given in 7.4.3, “Switch Partitioning Limitations” on page 294, and details on performance files are given in 7.4.8, “Example of Metrics File” on page 306.

7.3.8 System Partitioning Aid Added Value



System Partitioning Aid Added Value

- Graphical User Interface
- Switch chip boundary driven node-to-partition assignment
- Configuration Validation
- Information files on performance and layout
- Layout topology files generation
- Partitioning for switchless systems

This section presents added value points of the System Partitioning Aid.

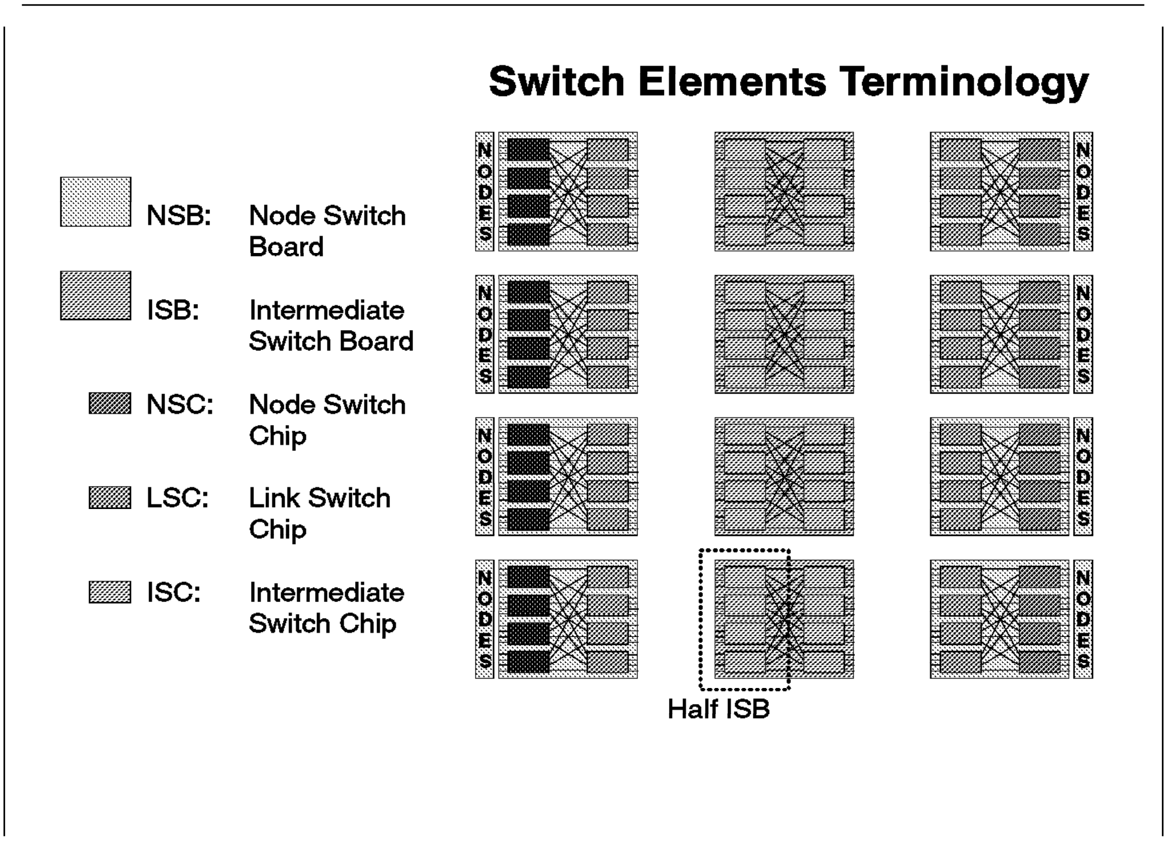
- **Graphical User Interface:** The Perspectives GUI allows a clear view of system configuration and an easy node-to-partition assignment process. It prevents struggling with node_numbers or switch_port_numbers.
- **Switch Chip Boundary Driven Assignment:** The chip boundary limitation is the basic limitation for the partitioning. Taking it into account when assigning nodes helps prevent failure in the validation process.
- **Configuration Validation:** Messages provided in cases of failure help the user to understand the issues of a failed requirement. Validation can be rather complicated when manually performed for huge systems.
- **Information on Performance and Layout:** What you gain in flexibility when partitioning the switch, you may lose in performance. Performance information helps you figure out what a given partition layout will give as switch throughput. The layout information indicates the way switch chips are assigned to various partitions. These combined forms of information are necessary while planning a system partitioning configuration.
- **Layout Topology Files Generation:** The limitation of fixed set of layouts was the most restrictive constraint on partitioning in the past. With the System Partitioning Aid, any valid layout at the switch level can now be applied on the system.

- **Partitioning for Switchless Systems:** System Partitioning Aid opens partitioning to the wide scope of LAN consolidation systems.

7.4 Switch Partitioning Additional Information

The System Partitioning Aid opens new horizons to partitioning on the RS/6000 SP by enlarging the scope of possible layouts and facilitating the expression of requirements. This does not mean that it relieves every switch constraint in partitioning. Sooner or later, when using the System Partitioning Aid, you will face these limitations. To understand how you can integrate them and reach an acceptable layout, and even to understand the messages the validation tool will give you in case of failure, more information about what is behind the scenes is needed. With this in mind, this section presents topology elements of the switch-to-position partitioning limitations. It also gives more information on performance issues related to partitioning.

7.4.1 Switch Elements Terminology



Switch Elements Terminology

- ▨ NSB: Node Switch Board
- ▨ ISB: Intermediate Switch Board
- ▨ NSC: Node Switch Chip
- ▨ LSC: Link Switch Chip
- ▨ ISC: Intermediate Switch Chip

This figure presents all the switch elements involved in building a 128-way switched RS/6000 SP system. A few terms describe the elements of this structure:

- **Switch Fabric**: A set of switching elements or switch chips interconnected by communication links within a given RS/6000 SP system.
- **Switch Board**: A basic unit of a switch fabric. It contains 8 switching elements. Depending on the configuration of the system, a certain number

of switch boards are linked together to form a switch fabric. There are two types of switch board: the node switch board and the intermediate switch board.

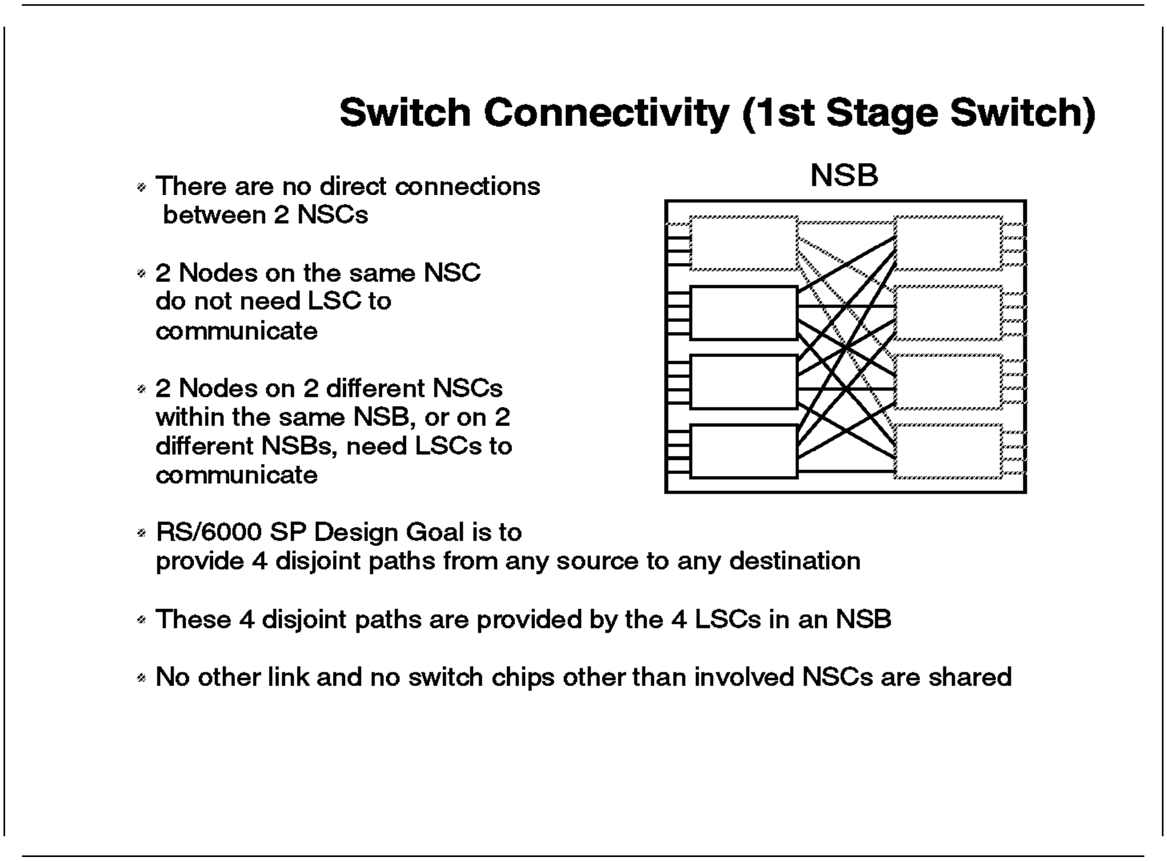
- **Node Switch Board (NSB)**: A switch board that has nodes connected to it. Up to four switch chips on any side of an NSB can have nodes attached to them. The NSB is located at the bottom of each frame containing nodes, in the switch drawer (except for expansion frames).
- **Intermediate Switch Board (ISB)**: A switch board with no nodes attached to it. ISBs form intermediate stages in the interconnection of large systems, when more than five NSBs are needed (it can be ordered as an option for 5 NSB systems).
- **Node Switch Chip (NSC)**: A switch chip, on an NSB, connected to nodes. An NSC can have up to four nodes connected to it.
- **Link Switch Chip (LSC)**: A switch chip, on an NSB, that links only to other switch chips.
- **Intermediate Switch Chip (ISC)**: A switch chip on an ISB. Each ISB has 8 ISCs.
- **Half-ISB**: A set of four chips of an ISB that are not directly connected with each other.

On the two next figures, we explain how first stage and second stage switch systems are built in terms of switch elements interconnection.

7.4.2 Switch Connectivity Characteristics

This section discusses how first stage and second stage switch systems are built in terms of switch elements interconnection.

7.4.2.1 First Stage Switch



Switch Connectivity (1st Stage Switch)

- There are no direct connections between 2 NSCs
- 2 Nodes on the same NSC do not need LSC to communicate
- 2 Nodes on 2 different NSCs within the same NSB, or on 2 different NSBs, need LSCs to communicate
- RS/6000 SP Design Goal is to provide 4 disjoint paths from any source to any destination
- These 4 disjoint paths are provided by the 4 LSCs in an NSB
- No other link and no switch chips other than involved NSCs are shared

The first stage switch architecture has been designed for systems with up to 80 thin nodes. As 80 nodes can also be placed in systems with second stage switch, the best way to express this limitation is "systems up to five NSBs."

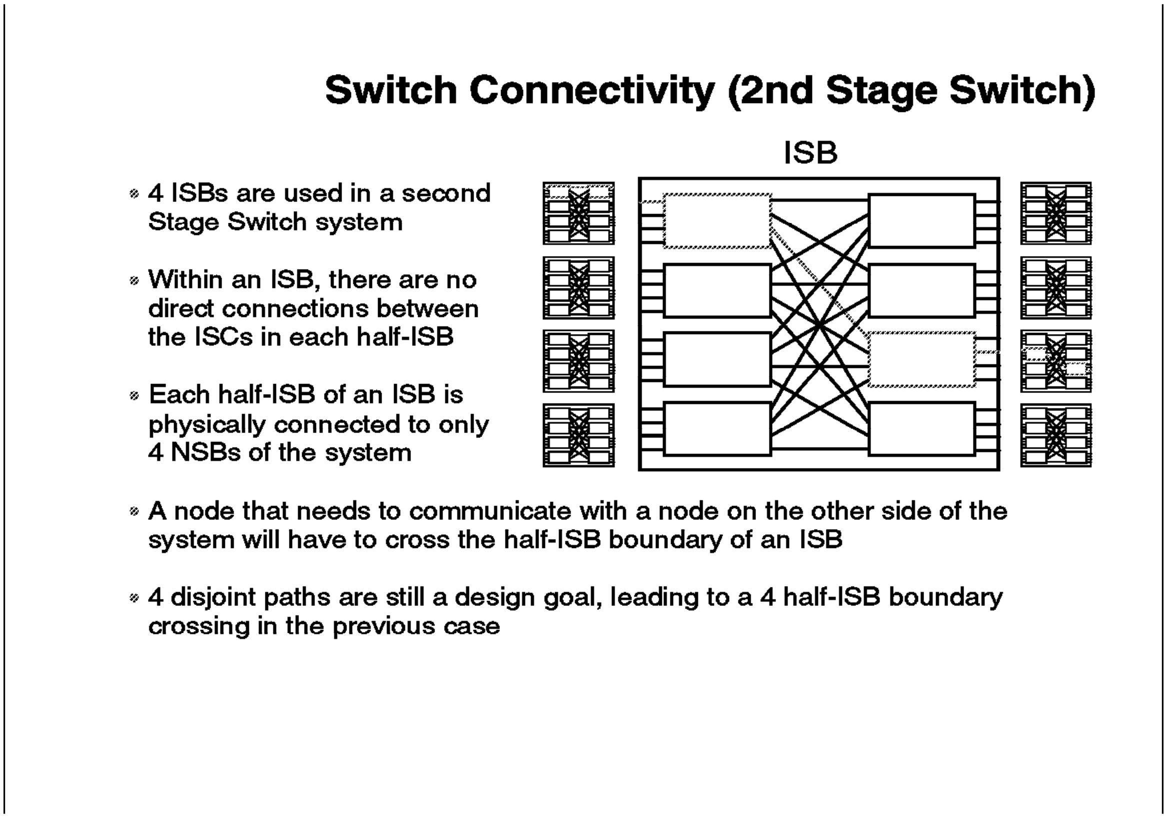
Thus, for these systems, switch interconnection is achieved by connecting together NSBs of each frame. This means that LSCs will be directly connected through switch cables from one NSB in one RS/6000 SP frame to another NSB in another frame. Each NSB has four ISCs, allowing each four connections. For performances and availability reasons, at least four cables need to connect an NSB to another NSB. Once an NSB is connected to four other NSBs by four cables, its 16 available LSC connections are used. This is the reason why each NSB can only be connected to four other NSBs, leading to the limit of five NSBs in a first stage switch system.

For these systems, LSCs that are assuring the communication between frames are of great importance. They are the bottleneck, as explained in 7.4.3, "Switch Partitioning Limitations" on page 294.

LSCs are used when two nodes on two different NSCs want to communicate. There is no direct link between NSCs, so at least one LSC is requested. The design of the switch communication leads to the need of at least four disjoint paths to assure communication between nodes. This is not a technical limitation, and communication can still occur with less than four paths. However, four paths ensure maximum performance and availability.

The four paths starting from one node to another node all encompass the NSC to which the node is linked. This is the only resource shared by the paths, apart from the corresponding NSC on the other side. In output of the NSC, the four paths are going through the four LSCs in the NSB as shown in the picture.

7.4.2.2 Second Stage System



Switch Connectivity (2nd Stage Switch)

ISB

- 4 ISBs are used in a second Stage Switch system
- Within an ISB, there are no direct connections between the ISCs in each half-ISB
- Each half-ISB of an ISB is physically connected to only 4 NSBs of the system
- A node that needs to communicate with a node on the other side of the system will have to cross the half-ISB boundary of an ISB
- 4 disjoint paths are still a design goal, leading to a 4 half-ISB boundary crossing in the previous case

Second stage switch systems are systems with too many nodes to allow direct interconnection between NSBs included in RS/6000 SP frames, as described in 7.4.2.1, "First Stage Switch" on page 290. A new element is then introduced: the Intermediate Switch Frame. It is an RS/6000 SP tall frame containing only four ISBs. The four ISBs are connected to the NSBs. A 128-way system is represented on the foil in 7.4.1, "Switch Elements Terminology" on page 288. Up to 8 NSBs can be connected, with the system being divided into two groups of four NSBs on each side of the Intermediate Switch Frame. There are no more direct connections between NSBs through LSCs. Each ISB of the Intermediate Switch Frame is linked to the four NSBs on each side of the system by four connections. This occupies 2 x 4 x 4 = 32 connections on the ISB, which is the maximum available. Seen from the NSB side, each NSB is linked to the four ISBs of the Intermediate Switch Frame by four connections.

For the purpose of our demonstration, we have only represented one ISB on this foil. But you have to keep in mind that there are in fact four ISBs. This foil highlights that the concern about connection within an ISB is similar to the one we have with an NSB in 7.4.2, "Switch Connectivity Characteristics" on page 290.

There are different communication means:

- When one node wants to communicate with another node hooked to the same NSC, even in the case of a second stage system, only the single NSC is necessary. We are in the same situation with a first stage system.

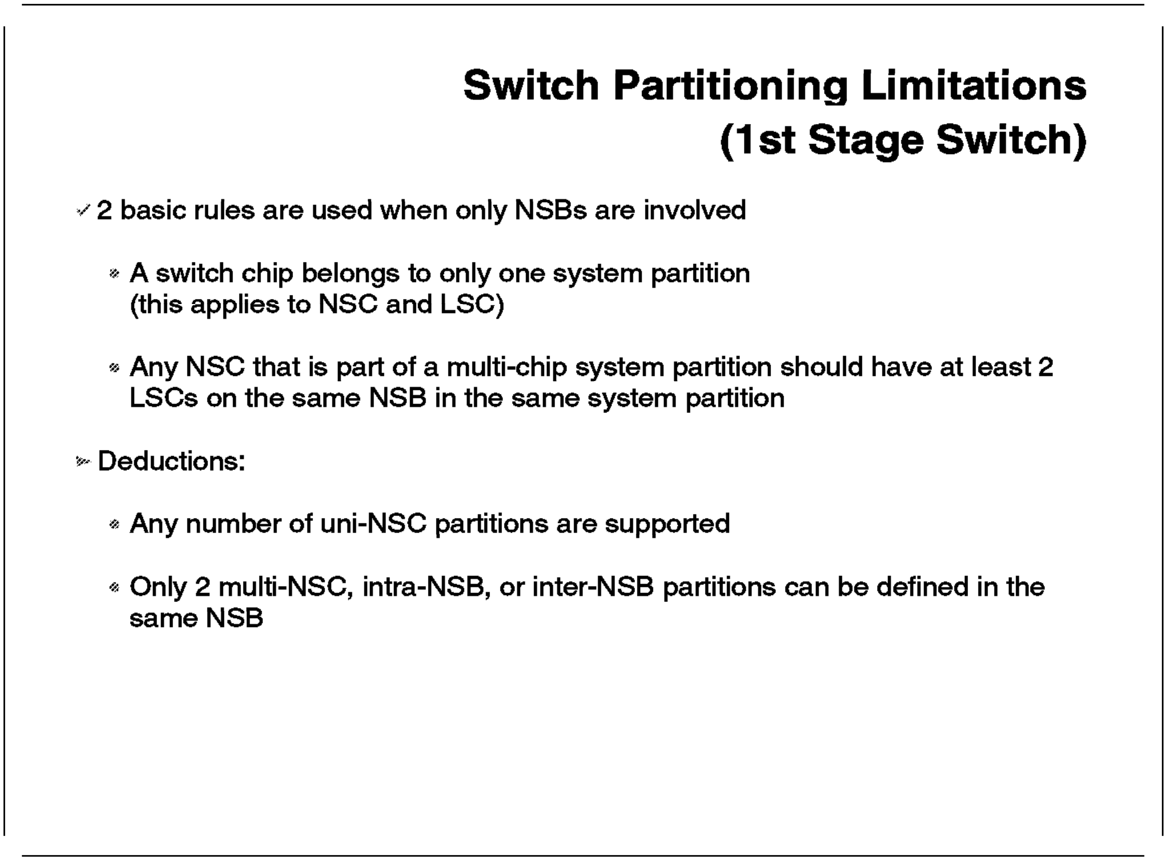
- When one node wants to communicate with another node hooked to another NSC in the same NSB, only the single NSB is necessary. There is no need to use an ISB.
- When one node wants to communicate with another node hooked to an NSC in one of the other three NSBs on the same side of the system, then four ISCs are used. This is to ensure the four necessary disjoint paths. This can be achieved by using one ISC in each ISB.
- The most complex case is when a node in one of the four frames in one side of the system wants to communicate with one node in one of the four frames on the other side of the system, it has to cross what we call the half-ISB boundary. The communication then uses two ISCs of the same ISB, one on each side of the half-ISB boundary. To ensure four disjoint paths between the two nodes in that case, eight ISCs, two in each of the four ISBs, will be needed.

More information on switch connections and performances can be found in *IBM System Journal Vol. 34, No 2, 1995,* G321-0120.

7.4.3 Switch Partitioning Limitations

Starting from the connectivity elements described in the two previous sections, we will then partition the RS/6000 SP system. Partitioning implies splitting apart the resources used, in particular the switch chips. There will be no interaction between partitions at that level. That means that the switch chips will be partitioned. *A switch chip (NSC, LSC or ISC) belongs to one and only one partition.* To satisfy that requirement and keep an acceptable number of possible partitions, the constraint of four disjoint paths between two communicating nodes has been relaxed. Now two disjoint paths are required to cope with basic availability concerns. This may affect performance, as described in 7.4.7, “Switch Performance Metrics” on page 304. The three following sections summarize the limitations brought by Switch Chip partitioning.

7.4.3.1 First Stage Switch



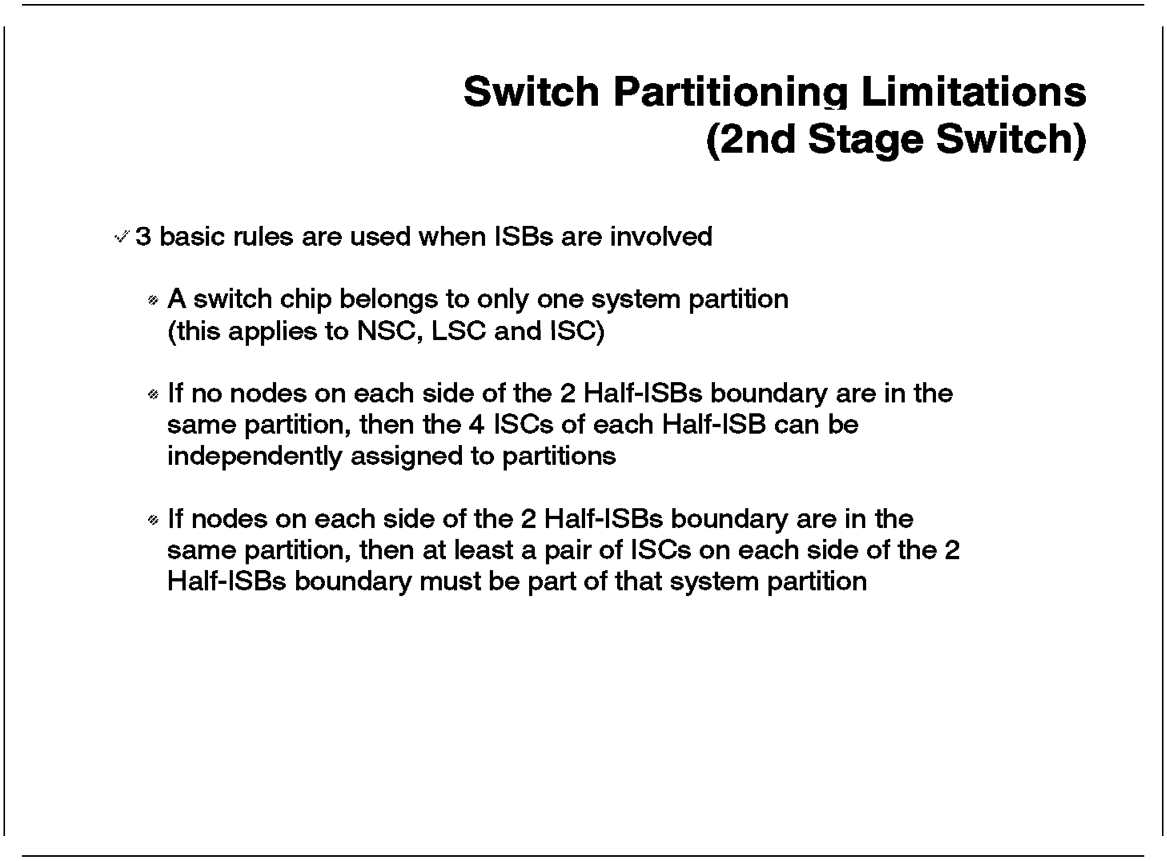
Switch Partitioning Limitations
(1st Stage Switch)

- ✓ **2 basic rules are used when only NSBs are involved**
  - **A switch chip belongs to only one system partition (this applies to NSC and LSC)**
  - **Any NSC that is part of a multi-chip system partition should have at least 2 LSCs on the same NSB in the same system partition**
- ➢ **Deductions:**
  - **Any number of uni-NSC partitions are supported**
  - **Only 2 multi-NSC, intra-NSB, or inter-NSB partitions can be defined in the same NSB**

Following the communication rules expressed in 7.4.2, “Switch Connectivity Characteristics” on page 290, when a node hooked to one NSC wants to communicate with a node hooked to a different NSC, LSCs are involved. With the number of mandatory disjoint paths fixed to two, two LSC are required. Being in the same partition, the two nodes will lead the two LSCs to be assigned to the partition. That means that two NSCs in the same partition will book two LSCs in each NSB involved. Each NSB has only four LSCs, allowing the booking by only two inter-NSB or intra-NSB partitions.

See 7.4.5, “Example of Rejected Configuration” on page 299 for an example of refused layout due to that limitation.

7.4.4 Second Stage Switch



Switch Partitioning Limitations (2nd Stage Switch)

✓ 3 basic rules are used when ISBs are involved

- A switch chip belongs to only one system partition (this applies to NSC, LSC and ISC)
- If no nodes on each side of the 2 Half-ISBs boundary are in the same partition, then the 4 ISCs of each Half-ISB can be independently assigned to partitions
- If nodes on each side of the 2 Half-ISBs boundary are in the same partition, then at least a pair of ISCs on each side of the 2 Half-ISBs boundary must be part of that system partition

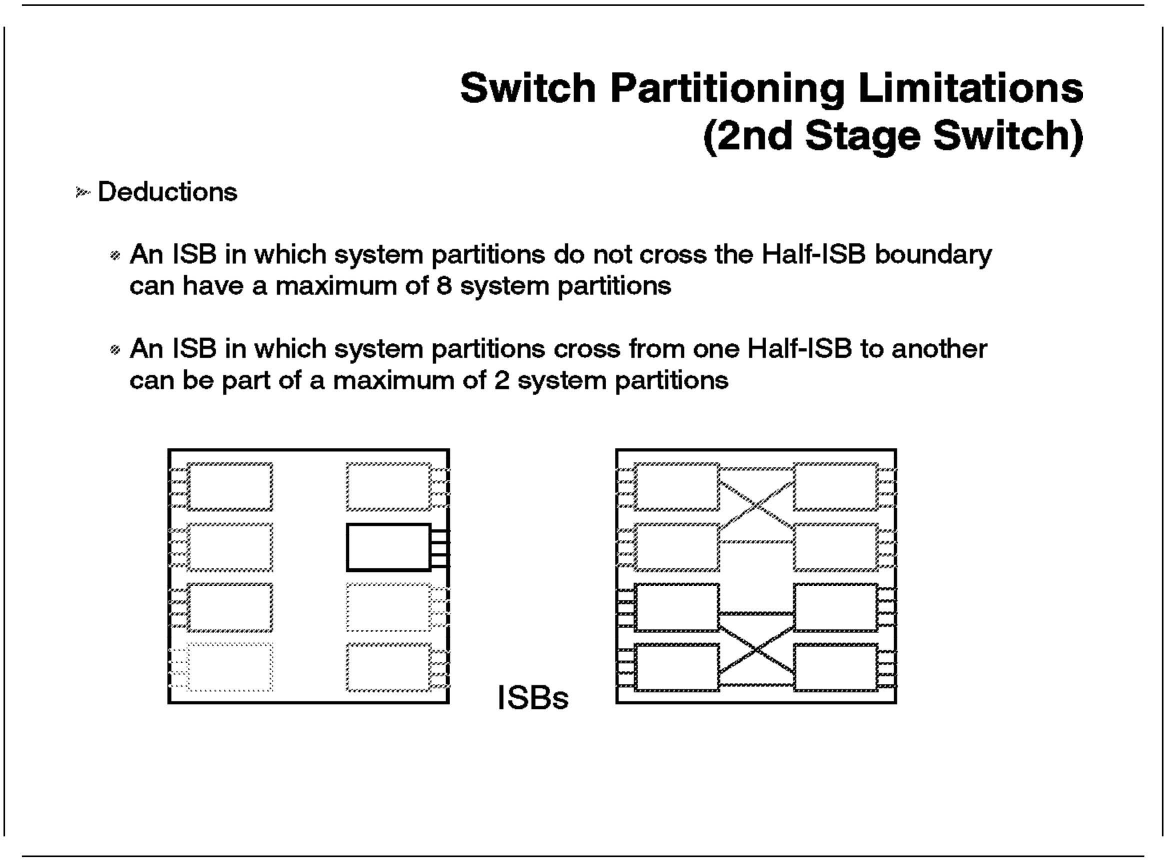
When partitioning for second stage switch systems, you have to visualize a system containing up to eight NSBs related to four ISBs. See 7.4.1, "Switch Elements Terminology" on page 288 for a representation of such a system.

The constrained resource in first stage switch systems is the LSC. In second stage switch systems, the constrained resource is the ISC. This is reinforced when nodes on the two sides of the ISB are in the same partition.

The first rule is a general rule that expands from first stage switch systems to second stage switch systems.

The second rule expresses the fact that if no partition encompasses the two sides of an ISB, then data transferred through the ISB are only managed by a single ISC. ISCs of the ISB are then independent. To take into account the availability requirement (at least two disjoint paths between any two nodes), another ISC from the same ISB or another ISB will be booked in the same partition in case the first one fails. The rule simply expresses the fact that at the single ISB level, there are no constraints for the two ISCs to be within the same half-ISB.

The third rule also takes into account the availability factor as if a partition is crossing the half-ISB boundary, using two ISCs on each side of the half-ISB. If one of these two ISCs fails, another spare ISC is needed. This is true from each side of the half-ISB boundary. This leads to the four ISCs booking (a pair on each side of the half-ISB boundary).



Switch Partitioning Limitations (2nd Stage Switch)

- Deductions
  - An ISB in which system partitions do not cross the Half-ISB boundary can have a maximum of 8 system partitions
  - An ISB in which system partitions cross from one Half-ISB to another can be part of a maximum of 2 system partitions

The consequences of the rules expressed on the previous figure are summarized here at a single ISB level.

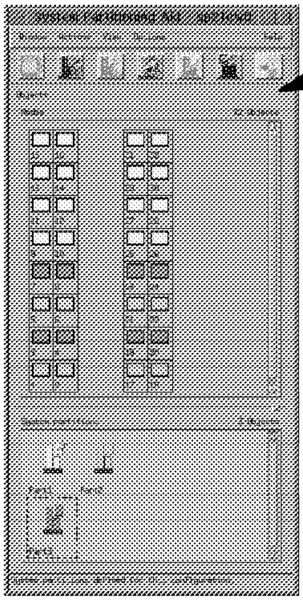
The first deduction is illustrated by the ISB on the left of the figure. It comes from the second rule of the previous figure. At the entire Intermediate Switch Frame level, another ISC will be needed in another ISB for each of the eight mentioned partitions. A partition cannot have only one ISC, as it would be a single point of failure. Even if each ISB can have eight independent partitions, the four ISBs together cannot have 32 partitions, but only a maximum of 16 partitions.

But this maximum of 16 partitions reaching the ISBs is not realistic at the NSB level. To reach the ISB, the partitions have to be inter-NSB partitions. This means that we would need to have 16 inter-NSB partitions in the system. A two-frame system can only have two inter-NSB partitions, as seen in the example in 7.4.5, “Example of Rejected Configuration” on page 299. This is due to LSC shortage. This leads to an average of one inter-NSB partition per frame. Thus eight frames will lead to eight inter-NSB partitions. So within the ISBs, an average of four ISCs will be assigned per partition.

The second deduction is illustrated by the ISB at the right of the figure. It is a consequence of the third rule expressed in the previous figure.

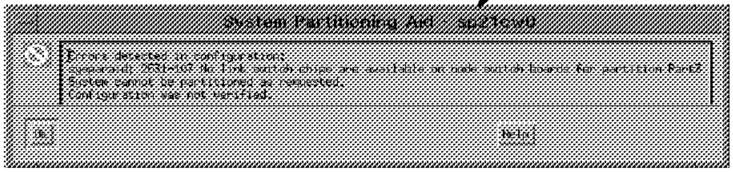
7.4.5 Example of Rejected Configuration

Example of Rejected Configuration


System Partitioning Aid - sp21cw0
Window Actions View Options Help
Objects
Nodes 32 Objects
15 16 31 32
13 14 29 30
11 12 27 28
9 10 25 26
7 8 23 24
5 6 21 22
3 4 19 20
1 2 17 18
System partitions 3 Objects
Part1 Part2
Part3
System partitions defined for this configuration.

Your requirements

SPA verdict



```
Errors detected in configuration:
syspar_aid: 2531-107 No link switch chips are available on node switch boards for partition Part2.
System cannot be partitioned as requested.
Configuration was not verified.
```

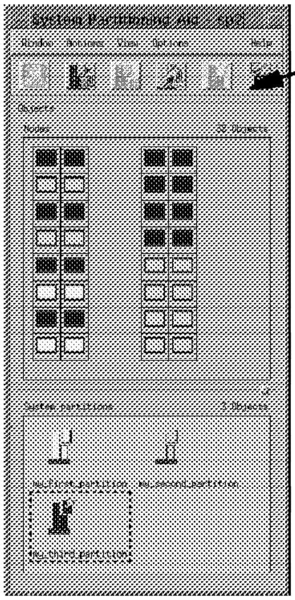
- Part1 and Part2 are 2 inter-NSB system partitions and as such require each 2 LSCs in the NSB for inter-NSB communication
- Part3 does not have any LSC left to work with !

This case is based on the limitation seen in 7.4.3.1, “First Stage Switch” on page 294. Two partitions are already each booking two LSCs from each NSB. This means that the third partition cannot have any LSCs to communicate between NSBs.

If the two first partitions cannot be changed, then the solution is to use the four remaining nodes in each NSB as a separate partitions. That means splitting Part3 into two intra-NSC partitions.

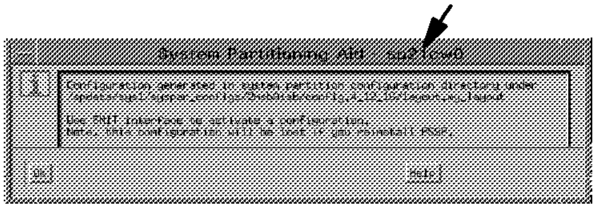
7.4.6 Example of Accepted Configuration

Example of Accepted Configuration


System Partitioning Aid - sp2
Window Actions View Options Help
Objects
Nodes 32 Objects
System partitions 3 Objects
my_first_partition
my_second_partition
my_third_partition

Our requirements

SPA verdict



Configuration generated in system partition configuration directory under
/spdata/sys1/syspar_configs/2nsb0isb/config.4_12_16/layout.my_layout

Use SMIT interface to activate a configuration.
Note, this configuration will be lost if you reinstall PSSP.

- my_first_partition is a single chip partition within the NSB of the first frame
- my_second_partition and my_third_partition are thus the only inter-NSC, and even inter-NSB, partitions in the system
- The partition layout is valid and saved

In the example that we already used in 7.3.1, "System Partitioning Aid" on page 271, we knew it was a valid layout because the sysparaid command already succeeded in generating the files.

```
#cd /spdata/sys1/syspar_configs/2nsb0isb/config.4_12_16/layout.my_layout

#ls -R

layout.desc
nodes.syspar
spa.snapshot
syspar.1.my_first_partition
syspar.2.my_second_partition
syspar.3.my_third_partition
./syspar.1.my_first_partition:
gui.info
nodelist
spa.metrics
spa.snapshot
topology

./syspar.2.my_second_partition:
gui.info
nodelist
spa.metrics
spa.snapshot
topology

./syspar.3.my_third_partition:
gui.info
nodelist
spa.metrics
spa.snapshot
topology
```

Under the layout directory, we have the layout-specific files such as the snapshot file. It shows the switch chip assignment to partitions.

```
# cat spa.snapshot

         System: 32 nodes, 2 NSBs, 0 ISBs
         Number of System Partitions: 3

         Partition Name          Partition Number        Partition Status
         --------------          ----------------        ----------------
         my_first_partition             1                Complete
         my_second_partition            2                Complete
         my_third_partition             3                Complete


         System Partition Number of all Switch Chips in the system:
         ----------------------------------------------------------

         Note: Switch chips under labels N are linked to nodes as well as
               to other switch chips, and those under labels L and I are linked
               only to other switch chips.

                   NSB 1
                 N       L
                 -       -
                 2       2

                 1       2

                 3       3

                 3       3

                   NSB 2
                 N       L
                 -       -
                 3       2

                 2       2

                 2       3

                 3       3
```

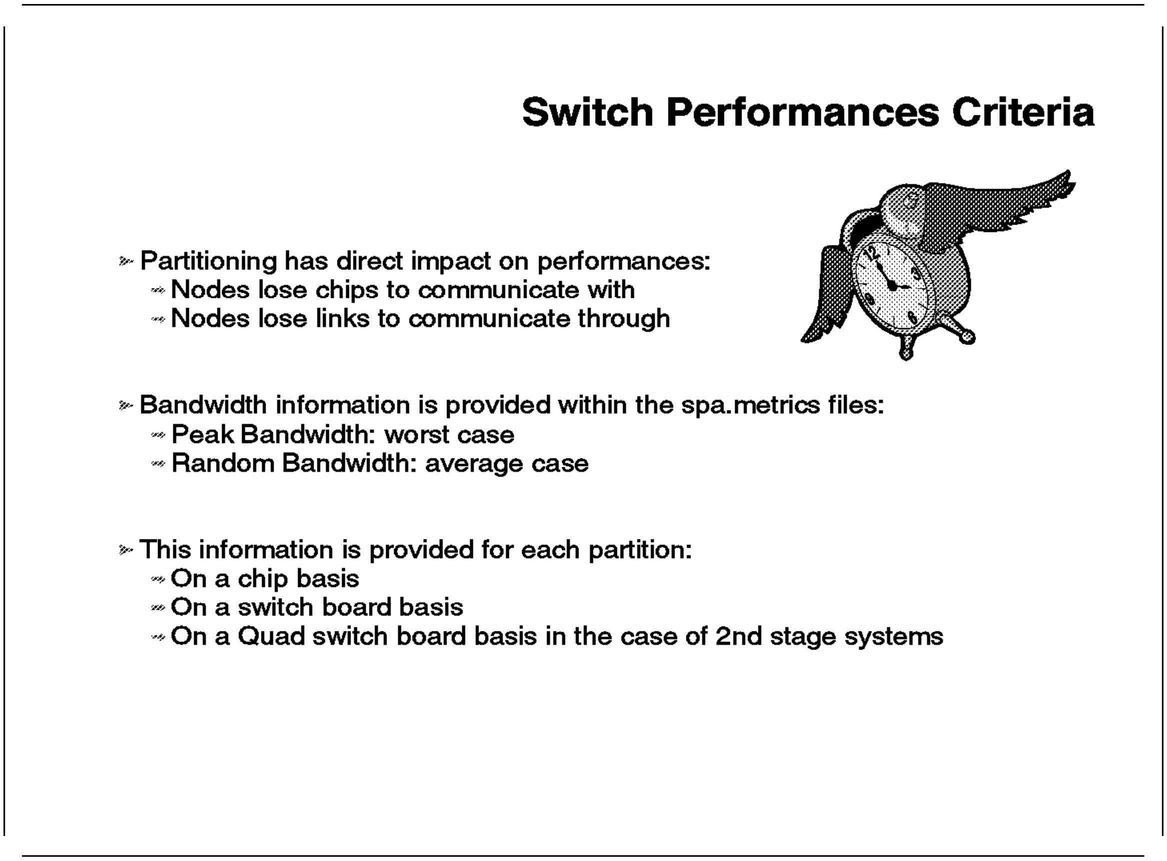
Under each partition directory, we have the partition-specific files such as the snapshot file, the graphical information file, and the topology file. Here are the examples for the third partition named my_third_partition:

```
# cat syspar.3.my_third_partition/spa.snapshot

 Partition Name:  my_third_partition
 In the following Chip Allocation Diagram :
 X denotes a switch chip in the current system partition and
 - denotes a switch chip that does not belong to the current partition.


                   NSB 1

                 -       -

                 -       -

                 X       X

                 X       X

                   NSB 2

                 X       -

                 -       -

                 -       X

                 X       X



# cat syspar.3.my_third_partition/gui.info

Number of Frames in System: 2
Partition Gui: my_third_partition
Partition Name: my_third_partition
Partition Color: 0 0 255

Nodes in Partition :
2
3
6
7
10
11
14
15
24
25
26
27
28
29
30
31

# cat syspar.3.my_third_partition/topology

#pragma comment (copyright, ”@(#) expected.top.2nsb.0isb.0 1.1 1   7/11/94 14:37:51 \0” )
#
# FUNCTIONS:  This file describes the wiring configuration for the High
#             Performance Switch.  It is used during switch initialization
#             (”Estart” command).  It should not be changed unless the
#             node-to-switch or switch-to-switch cabling differs from the
#             prescribed pattern.
#
# FORMAT:  A node-to-switch connection looks like:  s 36 3   tb2 15 0
#                                                   | || |    |   |
#                                          Switch...| || |    |   |
#                                     in switch 3.....|| |    |   |
#                                          chip 6......| |    |   |
#                                          port 3........|    |   |
#                 is connected to the TB0 or TB2 adapter......|   |
#                          in switch node number .................|
#
#      Switch-to switch connections just use the first four components twice.
```

```
#
# ORIGINS: 27
#
#CPRY
# 5765-529 (C) Copyright IBM Corporation 1993,1994,1995
# Licensed Materials - Property of IBM
# All rights reserved.
# US Government Users Restricted Rights -
# Use, duplication or disclosure restricted by
# GSA ADP Schedule Contract with IBM Corp.
#CPRY
#
####################################################################
#
#   Initial version - 3 Feb 94
#
####################################################################
#
format 1
32 28
# Node connections in frame L01 to switch 1 in L01
s 16 0  tb0 2 0     L01-S00-BH-J20 to L01-N3
s 16 1  tb0 3 0     L01-S00-BH-J22 to L01-N4
s 16 2  tb0 6 0     L01-S00-BH-J24 to L01-N7
s 16 3  tb0 7 0     L01-S00-BH-J26 to L01-N8
s 17 0  tb0 10 0    L01-S00-BH-J28 to L01-N11
s 17 1  tb0 11 0    L01-S00-BH-J30 to L01-N12
s 17 2  tb0 14 0    L01-S00-BH-J32 to L01-N15
s 17 3  tb0 15 0    L01-S00-BH-J34 to L01-N16
# On board connections between switch chips on switch 1 in Frame L01
s 16 5    s 11 6    L01-S00-SC
s 16 4    s 10 6    L01-S00-SC
s 17 5    s 11 7    L01-S00-SC
s 17 4    s 10 7    L01-S00-SC
# Node connections in frame L02 to switch 2 in L02
s 24 3  tb0 24 0     L02-S00-BH-J10 to L02-N9
s 24 2  tb0 25 0     L02-S00-BH-J8  to L02-N10
s 27 0  tb0 26 0     L02-S00-BH-J28 to L02-N11
s 27 1  tb0 27 0     L02-S00-BH-J30 to L02-N12
s 24 1  tb0 28 0     L02-S00-BH-J6  to L02-N13
s 24 0  tb0 29 0     L02-S00-BH-J4  to L02-N14
s 27 2  tb0 30 0     L02-S00-BH-J32 to L02-N15
s 27 3  tb0 31 0     L02-S00-BH-J34 to L02-N16
# On board connections between switch chips on switch 2 in Frame L02
s 24 5    s 21 4    L02-S00-SC
s 24 4    s 20 4    L02-S00-SC
s 27 5    s 21 7    L02-S00-SC
s 27 4    s 20 7    L02-S00-SC
# switch 1 to switch 2
s 11 3     s 21 3  L01-S00-BH-J19 to L02-S00-BH-J19
s 11 2     s 21 2  L01-S00-BH-J21 to L02-S00-BH-J21
s 11 1     s 21 1  L01-S00-BH-J23 to L02-S00-BH-J23
s 11 0     s 21 0  L01-S00-BH-J25 to L02-S00-BH-J25
s 10 3     s 20 3  L01-S00-BH-J27 to L02-S00-BH-J27
s 10 2     s 20 2  L01-S00-BH-J29 to L02-S00-BH-J29
s 10 1     s 20 1  L01-S00-BH-J31 to L02-S00-BH-J31
s 10 0     s 20 0  L01-S00-BH-J33 to L02-S00-BH-J33
```

The performance metrics files are described in 7.4.8, “Example of Metrics File” on page 306.

7.4.7 Switch Performance Metrics



Switch Performances Criteria

- Partitioning has direct impact on performances:
  - Nodes lose chips to communicate with
  - Nodes lose links to communicate through
- Bandwidth information is provided within the spa.metrics files:
  - Peak Bandwidth: worst case
  - Random Bandwidth: average case
- This information is provided for each partition:
  - On a chip basis
  - On a switch board basis
  - On a Quad switch board basis in the case of 2nd stage systems

The allocation of LSCs and ISCs to partitions leads to the decrease of available links for nodes to communicate through. Each partition now has a fixed number of network links. This has an impact on performance.

The spa.metrics files mentioned in 7.3.7, “System Partitioning Aid Output Files” on page 284, contain performance information for the partition they belong to.

Performance is expressed in terms of bandwidth for communication from a chip, an entire switch board, or an entire Quad switch board (the four frames on the same side of the Intermediate Switch Frame), when applicable. This information is not coming from benchmark results but from basic topology facts, deducted from the partitioning of the switch fabric, under the assumption that communication takes place with a random pattern.

Two types of bandwidth figures are provided. They are expressed in a percentage of the corresponding bandwidth available in an unpartitioned system.

- **Peak Bandwidth**: worst case

  This is the bandwidth for a node communicating with only nodes outside of the chip, board, or Quad board. The more inter-chip links lost due to partitioning, the lower this value will be.

- **Random Bandwidth**: average case

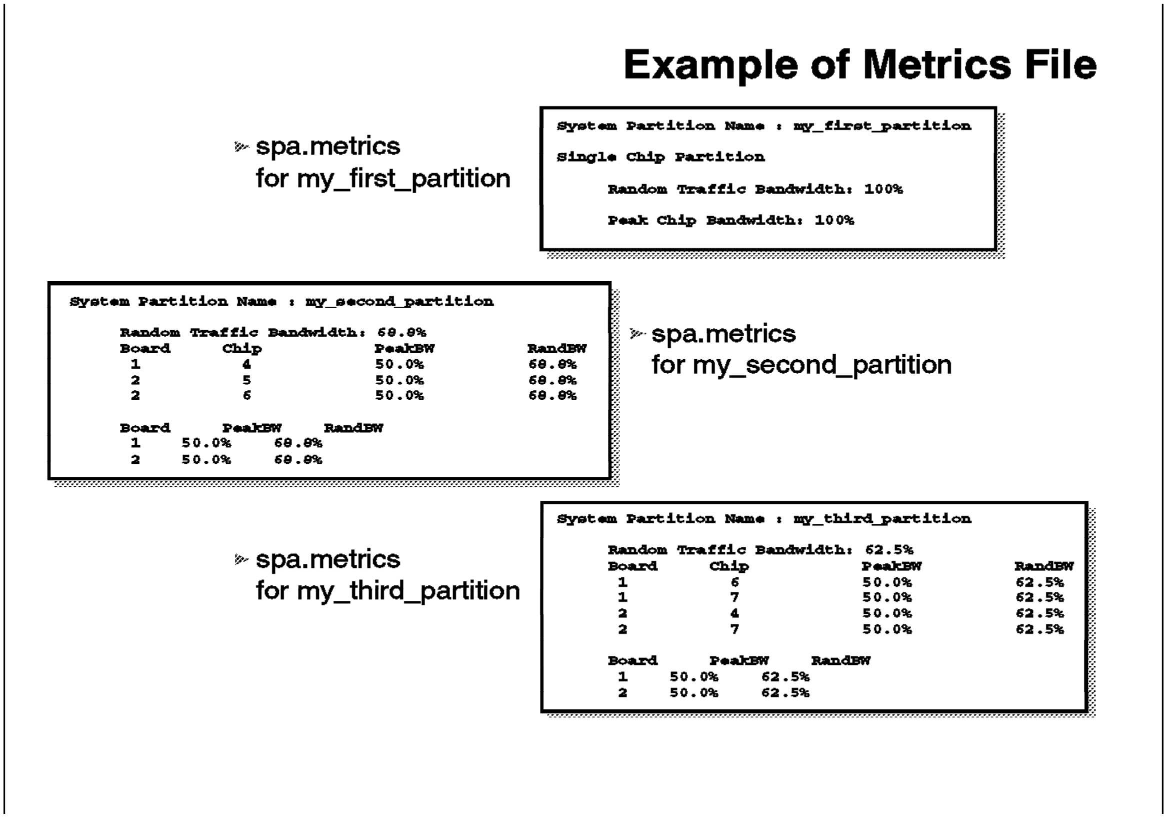
> This is the bandwidth for a node communicating uniformly with all the other nodes in the partition. Hence, the nodes are not only communicating with nodes outside of the chip, board, or Quad board, but also with internal nodes. With these internal nodes, the communication has less penalty. This leads to an average bandwidth that is better than the peak bandwidth and also closer to the reality of the partition.

This information may be used to choose a different layout that, in some cases, can lead to less performance degradation.

The decrease in the number of links and LSCs available for each NSC leads to communication performance loss when partitioning with four nodes connected to each NSC. If fewer than four nodes are connected to an NSC, the communication performance rate can still be 100% even if the system is partitioned. That means that a sufficient number of links and LSCs are still available. This is the case in the use of Wide or High Nodes without expansion frames, which leaves some of the switch_port_numbers unoccupied.

These hardware-related performance considerations can also be changed by software algorithms. Some job distribution algorithms favor neighbor communication (within the same NSC) from remote communication (inter-NSC).

7.4.8 Example of Metrics File



Example of Metrics File

- spa.metrics for my_first_partition

```
System Partition Name : my_first_partition
Single Chip Partition
      Random Traffic Bandwidth: 100%
      Peak Chip Bandwidth: 100%
```

- spa.metrics for my_second_partition

```
System Partition Name : my_second_partition

      Random Traffic Bandwidth: 68.8%
      Board       Chip              PeakBW          RandBW
       1           4                50.0%           68.8%
       2           5                50.0%           68.8%
       2           6                50.0%           68.8%

      Board      PeakBW     RandBW
       1    50.0%     68.8%
       2    50.0%     68.8%
```

- spa.metrics for my_third_partition

```
System Partition Name : my_third_partition

      Random Traffic Bandwidth: 62.5%
      Board       Chip              PeakBW          RandBW
       1           6                50.0%           62.5%
       1           7                50.0%           62.5%
       2           4                50.0%           62.5%
       2           7                50.0%           62.5%

      Board      PeakBW     RandBW
       1    50.0%     62.5%
       2    50.0%     62.5%
```

In this example, each of the three partitions has a metrics file:

```
# cat syspar.1.my_first_partition/spa.metrics

 System Partition Name : my_first_partition

 Single Chip Partition

         Random Traffic Bandwidth: 100%

         Peak Chip Bandwidth: 100%
```

The first partition is a single chip partition. As such, the bandwidth is not affected by the partitioning process:

```
# cat syspar.3.my_second_partition/spa.metrics

 System Partition Name : my_second_partition

         Random Traffic Bandwidth: 68.8%
         Board   Chip            PeakBW            RandBW
          1       4              50.0%             68.8%
          2       5              50.0%             68.8%
          2       6              50.0%             68.8%
```

```
        Board   PeakBW          RandBW
         1      50.0%           68.8%
         2      50.0%           68.8%
```

The second partition has 12 nodes, four on a single chip on the first board and eight on two chips of the second board. When a node on one of the chips wants to communicate with another node on another chip, it finds two paths to two LSCs instead of four paths to four LSCs in an unpartitioned system. This is the reason that bandwidth is 50%.

The random bandwidth has to take into account the fact that a node on a given chip is communicating with 11 other nodes, three on the same chip and eight on different chips. It has a performance penalty because of the decrease in the number of available links.

```
# cat syspar.3.my_third_partition/spa.metrics

 System Partition Name : my_third_partition

        Random Traffic Bandwidth: 62.5%
        Board   Chip            PeakBW          RandBW
         1        6             50.0%           62.5%
         1        7             50.0%           62.5%
         2        4             50.0%           62.5%
         2        7             50.0%           62.5%

        Board   PeakBW          RandBW
         1      50.0%           62.5%
         2      50.0%           62.5%
```

The peak bandwidth is similar to the second partition, because any node, when communicating with another node on a different chip, has only half of the normal number of communication links.

The random bandwidth takes into account that a node has performance penalty when communicating with 12 out of the other 15 nodes. This is a higher percentage than the 8 out of the other 11 nodes we had for the second partition. This is the reason that the random bandwidth is lower for the third partition.

Appendix A. 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 2 Release 2 projects, including the education of RS/6000 SP professionals responsible for installing, configuring, and administering PSSP Version 2 Release 2. 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 2 Release 2 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 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 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.

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:

AIX
AIXwindows
HACMP/6000
LoadLeveler
POWER Architecture
POWERparallel
POWER2 Architecture
Scalable POWERparallel Systems
SP2
AIX/6000
DB2/6000
IBM
NetView
Power PC 604
PowerPC 604
RS/6000
SP
9076 SP2

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

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

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

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

Java and HotJava are trademarks of Sun Microsystems, Inc.

Other trademarks are trademarks of their respective companies.

Appendix B. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

B.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see "How To Get ITSO Redbooks" on page 313.

- *PSSP Version 2 Technical Presentation*, SG24-4542
- *RS/6000 SMP Servers Architecture*, SG24-2583
- *RS/6000 SP High Availability Infrastructure*, SG24-4838

B.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
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RISC System/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RISC System/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
Application Development Redbooks Collection	SBOF-7290	SK2T-8037
Personal Systems Redbooks Collection	SBOF-7250	SK2T-8042

B.3 Other Publications

These publications are also relevant as further information sources:

- *PSSP Installation and Migration Guide*, GC23-3898
- *PSSP Diagnosis and Messages Guide*, GC23-3899
- *PSSP Command and Technical Reference*, GC23-3900
- *Group Services Programming Guide and Reference*, GC28-1675 (available by year end 1996)
- *PSSP System Planning Guide*, GC23-3902

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 URL 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:

- **PUBORDER —** to order hardcopies in United States
- **GOPHER link to the Internet** - type GOPHER.WTSCPOK.ITSO.IBM.COM
- **Tools disks**

  To get LIST3820s of redbooks, type one of the following commands:

  ```
  TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
  TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
  ```

  To get lists of redbooks:

  ```
  TOOLS SENDTO WTSCPOK TOOLS REDBOOKS GET REDBOOKS CATALOG
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET LISTSERV PACKAGE
  ```

  To register for information on workshops, residencies, and redbooks:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1996
  ```

  For a list of product area specialists in the ITSO:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ORGCARD PACKAGE
  ```

- **Redbooks Home Page on the World Wide Web**

  http://w3.itso.ibm.com/redbooks

- **IBM Direct Publications Catalog on the World Wide Web**

  http://www.elink.ibmlink.ibm.com/pbl/pbl

  IBM employees may obtain LIST3820s of redbooks from this page.

- **REDBOOKS category on INEWS**
- **Online** — send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL
- **Internet Listserver**

  With an Internet E-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an E-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank). A category form and detailed instructions will be sent to you.

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** (Do not send credit card information over the Internet) — send orders to:

	IBMMAIL	**Internet**
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	1-403-267-4455
Outside North America	(+45) 48 14 2207 (long distance charge)

- **1-800-IBM-4FAX (United States)** or **(+1) 415 855 43 29 (Outside USA)** — ask for:

  Index # 4421 Abstracts of new redbooks
  Index # 4422 IBM redbooks
  Index # 4420 Redbooks for last six months

- **Direct Services** - send note to softwareshop@vnet.ibm.com

- **On the World Wide Web**

Redbooks Home Page	http://www.redbooks.ibm.com
IBM Direct Publications Catalog	http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Internet Listserver**

  With an Internet E-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an E-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank).

IBM Redbook Order Form

Please send me the following:

Title	**Order Number**	**Quantity**

- **Please put me on the mailing list for updated versions of the IBM Redbook Catalog.**

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.

DO NOT SEND CREDIT CARD INFORMATION OVER THE INTERNET.

List of Abbreviations

ACL	Access Control List
AIX	Advanced Interactive Executive
AMG	Adapter Membership Group
ANS	Abstract Notation Syntax
APA	all points addressable
API	Application Programming Interface
BIS	Boot-Install Server
BSD	Berkeley Software Distribution
BUMP	Bring-Up Microprocessor
CP	Crown Prince
CPU	Central Processing Unit
CSS	Communication Subsystem
CWS	Control Workstation
EM	Event Management
EMAPI	Event Management Application Programming Interface
EMCDB	Event Management Configuration Database
EMD	Event Manager Daemon
EPROM	Erasable Programmable Read Only Memory
FIFO	First In - First Out
Gb	Gigabytes
GL	Group Leader
GS	Group Services
GSAPI	Group Services Application Programming Interface
hb	heart beat
HPS	High Performance Switch
hrd	host respond daemon
HSD	Hashed Shared Disk
IBM	International Business Machines Corporation
IP	Internet Protocol
ISB	Intermediate Switch Board
ISC	Intermediate Switch Chip
ITSO	International Technical Support Organization
JFS	Journal File System
LAN	Local Area Network
LCD	Liquid Crystal Display
LED	Light Emitter Diode
LRU	Least Recently Used
LSC	Link Switch Chip
LVM	Logical Volume Manager
Mb	Megabytes
MIB	Management Information Base
MPI	Message Passing Interface
MPL	Message Passing Library
MPP	Massively Parallel Processors
NIM	Network Installation Manager
NSB	Node Switch Board
NSC	Node Switch Chip
OID	Object ID
ODM	Object Data Manager
PE	Parallel Environment
PID	Process ID
PROFS	Professional Office System
PSSP	Parallel System Support Program
PTC	Prepare to Commit
PTPE	Performance Toolbox Parallel Extensions
PTX/6000	Performance Toolbox/6000
RAM	Random Access Memory
RCP	Remote Copy Protocol
RM	Resource Monitor
RMAPI	Resource Monitor Application Programming Interface
RPQ	Request for Product Quotation
RSI	Remote Statistics Interface
RVSD	Recoverable Virtual Shared Disk
SBS	Structured Byte String
SDR	System Data Repository
SMP	Symmetric Multiprocessors
SNMP	System Network Management Protocol
SPDM	SP Data Manager

SPMI	System Performance Measurement Interface
SRC	System Resource Controller
SSI	Single System Image
TS	Topology Services
TCP/IP	Transmission Control Protocol / Internet Protocol
UDP	User Datagram Protocol
VSD	Virtual Shared Disk
VSM	Visual System Management

Index

Special Characters

/etc/bootptab.info 152
  sphrdwrad 152
/etc/exports 179, 200
/etc/nologin 174
/etc/rc.net 146
/etc/rc.sp 159
/etc/sysctl.rootcmds.acl file 221
/usr/lib/drivers 187
/usr/lib/methods 187
/usr/lib/microcode 187
./bosinst.data 143
./setup 143
./signature 143
./sp_bundle 143
.rhosts 149
~ root/.rhosts 149

Numerics

325.manual.cust script 159

A

abbreviations 317
accounting support 132
acronyms 317
AIX 4 No Prompt Install 161
AIX 4.2 191
AIX 4.2 support 189
AIX 4.2 year end support 203
AIX System Backup & Recovery/6000 174
allnimres 101, 102
AMD 132
arp 152
asymmetric shared memory systems 8

B

battery backup 29, 47
benefits of coexistance 123
bibliography 311
boot-install server 86
Boot/Install process 104
bootp_response 101
bos.rte.mp 41
bos.rte.mp.usr 97
bosinst_data_migrate 177
BSD automount 132
BUMP 21, 28, 29, 30, 42, 43, 44, 45

C

cache coherency 11, 18, 19
call home 29, 47
Client Input Output Sockets 136
cluster power control 47
coexistance versus partitioning 124
compatible programs 193
context switching 2
create AIX 3.2.5 partition 154, 164
create CWS system backup 174
create_krb_files 95
cross bar switch 17, 18, 25
cw_name 159
CWS disk requirements 103
CWS is installed 157
CWS migration steps 141
CWS PSSP migration 2.1 to 2.2 171

D

DB2 23
debugging NIM 108
delnimast -l 0 173
delnimclient 92
delnimmast 91
devices.rs6ksmp.base 41
devices.rs6ksmp.base.usr 97
diagnostic flag 43
documentation 133
dsh 160, 184
dsh command running with Perspectives 238

E

enter node information into SDR 150, 163
Eprimary 158
Estart 158, 202
Examples
  customizing Perspectives launch pad 213
  node groups and partitioning 234
  partitioning performance metrics files 306
  partitioning with System Partitioning Aid 270, 300
  rejected partition configuration 299
expect 186
exportfs -va 179, 200

F

fast AIX 4.1 install 143
file collections 132
filtering 235
flow of control 13, 15
four-frame configuration 38

FPING 158
frame number 32, 34
frame supervisor 28, 29, 32, 109, 111

G

gui.info file 285

H

half-ISB 289
Hardware Perspective
  3DigitDisplay 254
  all nodes summary display 250
  architectural design 219
  current partition 225
  customizing monitoring 243
  dsh interface 238
  environmental variables 251
  event manager interaction 220
  filtering 235
  global control 249
  hardmon interaction 220
  manageable objects 224
  monitoring 239
  node grouping 226
  panes 224
  partition driven 225
  remote SDR access 222
  SDR interaction 219
  security 221
  SPmon coexistence 248
Hewlett-Packard Systems 203
High Availability Subsystem 130
High Nodes
  cross bar switch 18
  four-frame configuration 38
  high node 1, 31, 35, 36, 37, 38, 39, 41, 46, 49, 52
  IBM SystemGuard processor 20
  installation process 41
  multiprocessing concepts 2
  operator panel 30
  PowerPC 604 High Nodes 1, 22
  PSSP resources 49
  shared DASD systems 6
  shared everything systems 8
  shared nothing system (MPP) 4
  single frame configuration 35
  SMP architecture 10
  SMP as a commercial server 12
  SMP concepts 2
  SMP programming model 13
  SP PowerPC 604 High Node 25
  supported configurations 31
  system monitoring and control 46
  threads programming considerations 15
  three-frame configuration 37
  two-frame configuration 36
High Nodes installation process 41
High Performance Switch 131
hmacls file 221
how does PSSP coexistence work? 134
how to use FPING and repair Kerberos 158
HP 203
HR_FPING 158
http://www.rs6000.ibm.com/tech 136

I

IBM Internet Connection Secure Server 192
  CICS 192
  DB2 192
  Internet 192
  national language support 192
IBM SMP Design 17
IBM SystemGuard processor 20
IBM Virtual Shared Disk 135
Informix 23
install_cw 149, 175
Installation 77, 79
Installation Design 75
installp -ug ssp 183
integration 76
Intermediate Switch Board (ISB) 289
Intermediate Switch Chip (ISC) 289
Internet 52

K

kernel thread 14
kernel-only thread 14
kinit 175
krb-srvtab 159
krb.conf 159
krb.realms 159

L

L1 cache 10
L2 cache 10
Launch Pad
  customizing 213
  views 212
Link Switch Chip (LSC) 289
LoadLeveler 135
login control 132
lppsource 77
lslpp -l ″ssp.*″ 182
lsnim 96
lssrc -a 171

M

man pages 133
Micro Channel 26
migrate 77

migrate 3.2.5 node to AIX 4.1.4 181
migrate 4.1.4 node to PSSP 2.2 184
migrate the CWS from AIX 3.2.5 to AIX 4.1.4 142
migrating the CWS from PSSP 2.1 to PSSP 2.2 167
migration considerations 137
migration from PSSP 1.2 to PSSP 2.1 161
migration from PSSP 1.2 to PSSP 2.2 143
migration to AIX 4.2 200
mkconfig 99
mkinstall 100
mknimclient 98
mknimint 96
mknimmast 93
mknimres 97
mksysb 174
Monitoring
  activating 239
  with icon 242
Monitoring conditions
  customizing 243
  examples 245
  SDR class 240
monitoring the SP 130
mpcfg 44
MPP 1, 6, 24, 50, 53
multiprocessing concepts 2

N

Netscape 191
Network Time Protocol (NTP) 132
new install with system backup 198
New Installation Methodology 73
NIM 80, 81, 82, 84
NIM configuration data files 128
NIM filesets 80
NIM Resource Objects
  boot 84
  lppsource 84
  migrate 85
  mksysb 84
  nim_script 85
  noprompt 84
  prompt 84
  psspscript 85
  SPOT 84
Node group
  -N option 232
  concept 226
  creation 228
  management commands 230
  partition node group 226
  partitioning compliance 234
  SDR classes 229
  system node group 226
  use by PSSP commands 232
node migration 177
node migration for AIX 4.2 202
node number 28, 32, 33, 34
node supervisor 28, 29, 42, 43, 48
Node Switch Board (NSB) 289
Node Switch Chip (NSC) 289
nodecond2 186
nodegroup 77
nodes with AIX 4.2 195
nodes.syspar file 285

O

object label 236
operator panel 21, 28, 29, 30, 46, 47
Oracle 23

P

pane 216
Parallel Environment 203
Parallel Environment for AIX 136
Parallel ESSL for AIX 136
Parallel I/O File System 136
Partition
  active partition in System Partitioning Aid 276
  current partition in Perspectives 225
partition layout 133
partition node group 226
Partitioning
  first stage switch limitations 294
  metrics files 306
  partitioning and node grouping 234
  performance criteria 304
  PSSP 2.1 limitations 266
  rationale 260
  rules 294, 296
  second stage switch limitations 298
  switchless systems 280
PCI based RS/6000 175
pcp -a 185
peak bandwidth 304
perfagent.server 130
Performance Toolbox for AIX (5765-654) 130
Performances
  with partitioning 304
Perl 130
Perl distribution package 130
persistent 78
Perspectives 132
  customizing the launch pad 213
  design objectives 209
  individual perspectives 211
  launch pad 212
  launching 211
  look and feel 216
  preferences 215
Perspectives for PTPE 132
planning and preparation for coexistance 126
planning and preparation for migration 140

PowerPc 604 25
PowerPC 604 High Nodes 1, 22
Preferences
  customizing 215
  layout preference 217
preparation for node migration 179, 200
prepare the CWS 146, 161
preservation installation for PSSP 2.2 186
print support 132
process 13, 14, 15
processor_type 42
PSSP 2.2 and High Nodes 39
PSSP 2.2 and PSSP 2.1 options 169
PSSP 2.2 compared to PSSP 2.1 130
PSSP resources and the High Nodes 49
PSSP Version 2 Release 2
PSSP-1.2 147
pssp_script 106, 183
pssp.installp 148
  smit install_latest 148
public domain source code 133
PVMe 203
PVMe for AIX 136

R
random bandwidth 304
rcmdtgt 159
reasons for migration 139
Recoverable Virtual Shared Disk 135
recovering a BIS 107
reentrant 15
remote SDR access 222
Resource Manager 133
Resource Server 96
restore Kerberos and SDR information 148
restore system backup on CWS 175
RVSD 203

S
s1term 30, 43
s1term -w 187
SAP 23
sash 216
sbb 45
SDR_dest_info 159
SDRArchive 154, 174
server consolidation 22, 23
server_name 159
services_config 87, 155
setting up install environment 86
setup_authent 149
setup_server 75, 87, 159, 181
setup_server and wrappers 87
SGI 203
shared DASD systems 6
shared everything systems 8
shared nothing systems (MPP) 4
short frames 27
Silicon Graphics Systems 203
single frame configuration 35
slot number 32, 33
smit _nfs 200
smit mksysb 200
smit server_dialog 79, 186
SMP 1, 2, 8, 10, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 26, 39, 41, 42, 43, 45, 46, 47, 50, 52, 77
SMP architecture 10
SMP as a commercial server 12
SMP programming model 13
snooping 19
software coexistance 119
Solaris operating system 203
SP authentication server 130
SP health definition 239
SP problem management 131
  SNMP 131
  TME 10 Netview 131
SP Switch 131
SP user commands 130
SP_NAME used for remote access 222
spa.metrics file 284
spa.snapshot file 284
spadaptrs 153
spapply_config 155
spbootins 153, 159, 186, 198
  smit server_dialog 186
spbootins -install 179
spbootins -p PSSP-1.2 154
spbootins -p PSSP-2.1 173
spbootins -p PSSP-2.2 181, 202
spbootins -r migrate 177, 202
spcustomize_syspar 155
spethernt 151
spframe -r yes 1 1 /dev/tty0 150
sphostname 153
sphrdwrad 152
  /etc/bootptab.info 152
spimg 198
splst_versions -G -t 154
splstdata -b 181, 198, 202
spmon 175
spmon -d 150
spmon -G -g 198
SPmon and the Hardware Perspective 248
spnet_en0 98
SPOT 78
sprestore_config 175
spsitenv cw_lppsource_name=aix414 150
ssp.authent 130
ssp.basic 130
ssp.clients 130
ssp.css 131
ssp.docs 133

ssp.gui 132
ssp.gui.perfmon 132
  Perspectives for PTPE 132
ssp.ha 130
ssp.jm 133
ssp.perlpkg 130
ssp.pman 131
ssp.public 133
  AMD 133
  Expect 133
  NTP 133
  Perl 133
  Perl 5 133
  SUP 133
  Tcl 133
  TclX 133
  TK-X11 133
ssp.st 133
ssp.sysctl 131
  pdf 131
ssp.sysman 132
  lppdiff 132
ssp.top 133
ssp.usr.2.2.0.0 148
stand-alone 82
stopsrc 171
Sun SPARCstation Systems 203
supervisor 115
supervisor card 28, 29, 48
supported configurations 31
swapping 2
Switch
  fabric 288
  first stage connectivity 290
  second stage connectivity 292
  terminology 288
switch API 133
switch chip 32, 34
switch device driver 131
switch node number 32, 33, 34
switch port number 32, 35
switch table API 133
switchless systems partitioning 280
Sybase 23
symmetric shared memory systems 8
sysback 174
Sysctl 131
syspar_ctrl -A -G 155
syspar_ctrl -c hb 155
syspar_ctrl -c hr 155
sysparaid command 271
system backup 174
System Data Repository
  EM_Condition class 240, 244
  NodeGroup class 229
  SysNodeGroup class 229
System Management Tools 132
System Monitor GUI 132
system monitoring and control 46
system node group 226
System Partitioning Aid 133
  command line access 271
  infrastructure compliance 274
  installing command line interface 271
  output directories 282
  output files 284
  overview 257
  space needed for System Partitioning Aid command line interface use 271
  use on a non CWS RS/6000 274
System Partitioning Aid GUI
  functional flow 278
  installing 273
  look and feel 276
  panes 276
  PSSP 2.2 system management solutions 268
  space needed 273
SystemGuard 17, 20, 21, 28, 29, 47, 48

T
tftp 77
the supervisor interface 112
thread 8, 10, 13, 14, 15, 18, 50
thread-safe 15, 16, 50
threads programming considerations 15
three frame configuration 37
Ticket granting ticket 159
  rcmdtgt 159
two-frame configuration 36

U
unallnimres 102
unsupported environment 135
user authentication commands 130
user space 50, 51
user thread 14, 15

V
version 77
versions that support coexistance 122

W
Web 191
what is a High Node? 28
why use AIX 4.2 191
wrapper 76
wrappers 87, 89
  allnimres 101, 102
  bootp_response 101
  bos.rte.mp.usr 97
  create_krb_files 95
  delnimclient 92

wrappers *(continued)*
- delnimmast 91, 93
- devices.rs6ksmp.base.usr 97
- lsnim 96
- mkconfig 99
- mkinstall 100
- mknimclient 98
- mknimint 96
- mknimmast 93
- mknimres 97
- spnet_en0 98
- unallnimres 102

Printed in U.S.A.

SG24-4868-00