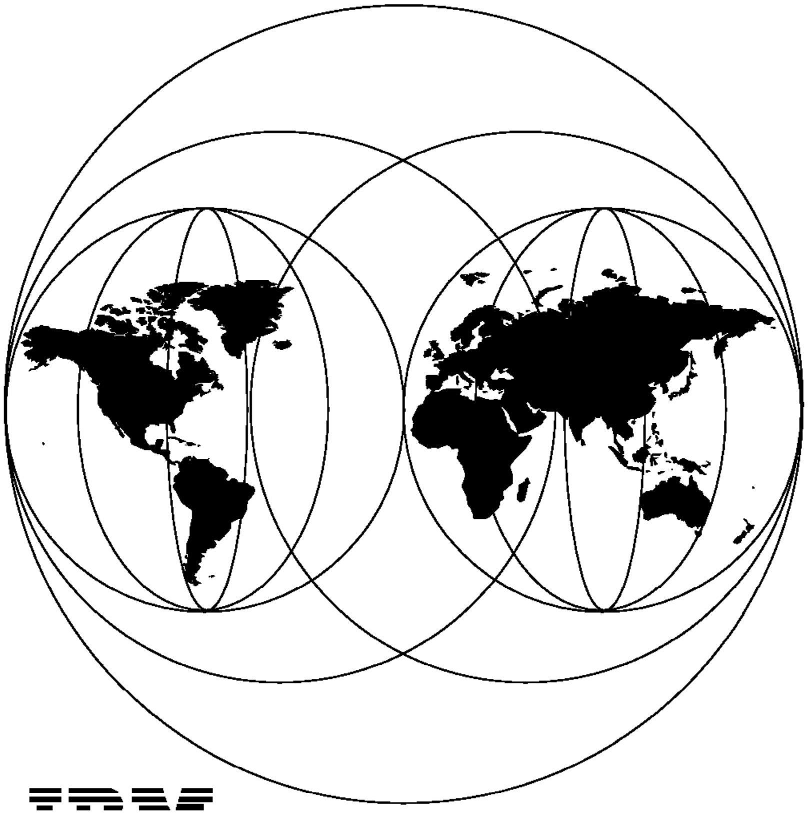
International Technical Support Organization

SG24-4542-00

RS/6000 Scalable POWERparallel Systems: PSSP Version 2 Technical Presentation

December 1995



IBM

International Technical Support Organization
Poughkeepsie Center

International Technical Support Organization

SG24-4542-00

RS/6000 Scalable POWERparallel Systems: PSSP Version 2 Technical Presentation

December 1995

Take Note!

Before using this information and the product it supports, be sure to read the general information under "Special Notices" on page ix.

First Edition (December 1995)

This edition applies to Version 2 Release 1 of Parallel System Support Programs for AIX for use with the IBM AIX/6000 Version 4 release 1.3

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

An ITSO Technical Bulletin Evaluation Form for reader′s feedback appears facing Chapter 1. If the form has been removed, comments may be addressed to:

IBM Corporation, International Technical Support Organization
Dept. 541 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.

Abstract

This document is a technical presentation in support of PSSP V2. It includes mid-size foils and related text and notes. It is intended to be used by IBM SEs and professionals who have to provide technical presentations for system programmers and administrators in charge of RS/6000 SP centers who have to install PSSP V2.

This presentation focuses on the following topics:

- PSSP V2 overview, new functions and improvements
- PSSP V2 partitioning
- High-availability control workstation (HACWS)
- Kerberos functions

Some knowledge of AIX Version 4, PSSP V2, and RS/6000 SP is assumed.

(255 pages)

Contents

Special Notices

This publication is intended to help IBM SEs and specialists who are involved in Parallel System Support Programs (PSSP) Version 2 projects including education of system programmers and administrators in charge of installing and administering PSSP V2 in RS/6000 SP computing centers. The information in this publication is not intended as the specification of any programming interfaces that are provided by PSSP V2. See the PUBLICATIONS section of the IBM Programming Announcement for PSSP V2 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.

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.

AIX
AIX/6000
AIXwindows
IBM
LoadLeveler
OS/2
POWERparallel
RISC System/6000
RS/6000
RS/6000 SP
Scalable POWERparallel Systems

The following terms are trademarks of other companies:

Windows is a trademark of Microsoft Corporation.

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

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

C-bus is a trademark of Corollary, Inc.

Preface

This document provides a technical presentation support of Parallel System Support Programs (PSSP) Version 2, including mid-size foil pictures and related notes and text. This technical presentation includes the following topics:

- PSSP V2 overview
- PSSP V2 installation
- PSSP V2 partitioning
- HACWS
- Kerberos

This document is intended for specialists involved in PSSP V2 technical presentations to system programmers and administrators in charge of RS/6000 SP centers.

How This Document Is Organized

The document is organized as follows:

- Part 1, “PSSP V2 Overview”

  This chapter is an overview of PSSP Version 2. It includes a general presentation of new and improved functions.

- Part 2, “PSSP V2 Installation”

  This chapter describes the PSSP V2 installation process.

- Part 3, “PSSP V2 Partitioning”

  This chapter describes how to define and administer partitions on RS/6000 SP using the new functions included in PSSP Version 2.

- Part 4, “HACWS”

  This chapter is dedicated to the high-availability control workstation (HACWS) feature that allows RS/6000 SP administrators to avoid the control workstation being a single point of failure.

- Part 5, “Kerberos”

  This chapter presents Kerberos and its use to secure administration and monitoring commands that communicate between the control workstation and the RS/6000 SP nodes through the network.

Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this document.

- *IBM RISC System/6000 Scalable POWERparallel Systems: System Planning*, GC23-3902-01
- *IBM RISC System/6000 Scalable POWERparallel Systems: Administration Guide*, GC23-3897-01
- *IBM RISC System/6000 Scalable POWERparallel Systems: SP Installation Guide*, GC23-3898-01

- *IBM RISC System/6000 Scalable POWERparallel Systems: Diagnosis and Messages Guide*, GC23-3899-01
- *IBM RISC System/6000 Scalable POWERparallel Systems: Command and Technical Reference*, GC23-3900-01
- *NIM Reference Guide AIX V4.1*, SC23-2627-03

International Technical Support Organization Publications

A complete list of International Technical Support Organization publications, known as redbooks, with a brief description of each, may be found in:

> *International Technical Support Organization Bibliography of Redbooks,* GG24-3070.

To get a catalog of ITSO redbooks, VNET users may type:

```
TOOLS SENDTO WTSCPOK TOOLS REDBOOKS GET REDBOOKS CATALOG
```

A listing of all redbooks, sorted by category, may also be found on MKTTOOLS as ITSOCAT TXT. This package is updated monthly.

How to Order ITSO Redbooks

IBM employees in the USA may order ITSO books and CD-ROMs using PUBORDER. Customers in the USA may order by calling 1-800-879-2755 or by faxing 1-800-445-9269. Visa and MasterCard are accepted. Outside the USA, customers should contact their local IBM office. Guidance may be obtained by sending a PROFS note to BOOKSHOP at DKIBMVM1 or E-mail to bookshop@dk.ibm.com.

Customers may order hardcopy ITSO books individually or in customized sets, called GBOFs, which relate to specific functions of interest. IBM employees and customers may also order ITSO books in online format on CD-ROM collections, which contain redbooks on a variety of products.

ITSO Redbooks on the World Wide Web (WWW)

Internet users may find information about redbooks on the ITSO World Wide Web home page. To access the ITSO Web pages, point your Web browser to the following URL:

```
http://www.redbooks.ibm.com/redbooks
```

IBM employees may access LIST3820s of redbooks as well. Point your web browser to the IBM Redbooks home page:

```
http://w3.itsc.pok.ibm.com/redbooks/redbooks.html
```

Acknowledgments

This project was designed and managed by:

Endy Chiakpo	ITSO, Poughkeepsie Center
Michel Perraud	ITSO, Poughkeepsie Center

The authors of this document are:

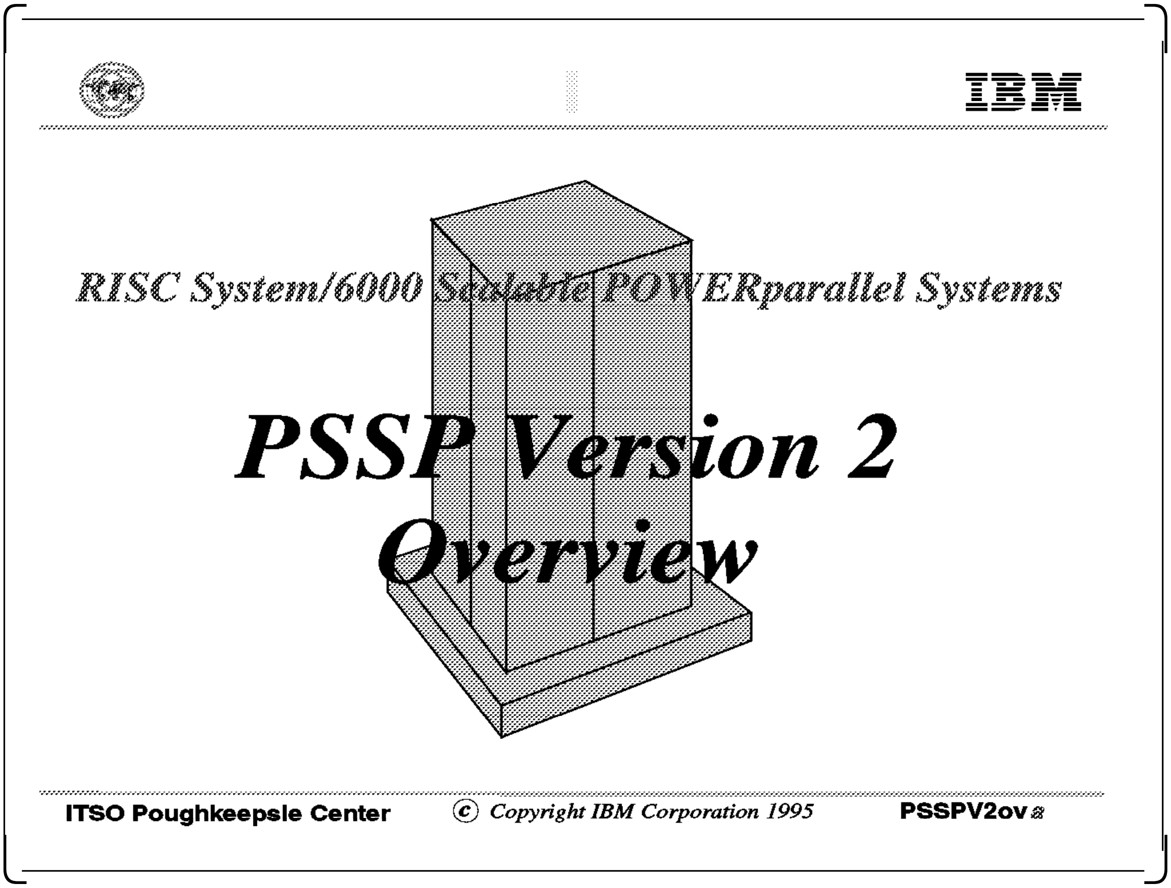
Giulia Caliari	IBM Italy
Henry Altaras	IBM Israel
Alexandre Blancke	IBM France
Mario Bono	IBM Australia
Franz Gerharter-Lueckl	IBM Austria

This publication is the result of a residency conducted at the International Technical Support Organization, Poughkeepsie Center.

Thanks to the following people for the invaluable advice and guidance provided in the production of this document:

Mary Ellen Coleman	IBM Poughkeepsie
Anna Fredricks	RISC System/6000 Division
Beth Lewis	IBM Poughkeepsie
Krystal Rothaupt	IBM Poughkeepsie
Bob Blackmore	IBM Poughkeepsie
Ken Briskey	IBM Poughkeepsie
Dave Folsom	IBM Poughkeepsie
James Gilman	IBM Poughkeepsie
Larry Parker	IBM Poughkeepsie
Skip Russel	IBM Poughkeepsie
John Simpson	IBM Poughkeepsie
Jim Still	IBM Poughkeepsie
Henry Zongaro	IBM Toronto Lab. Canada

Part 1. PSSP V2 Overview



RISC System/6000 Scalable POWERparallel Systems

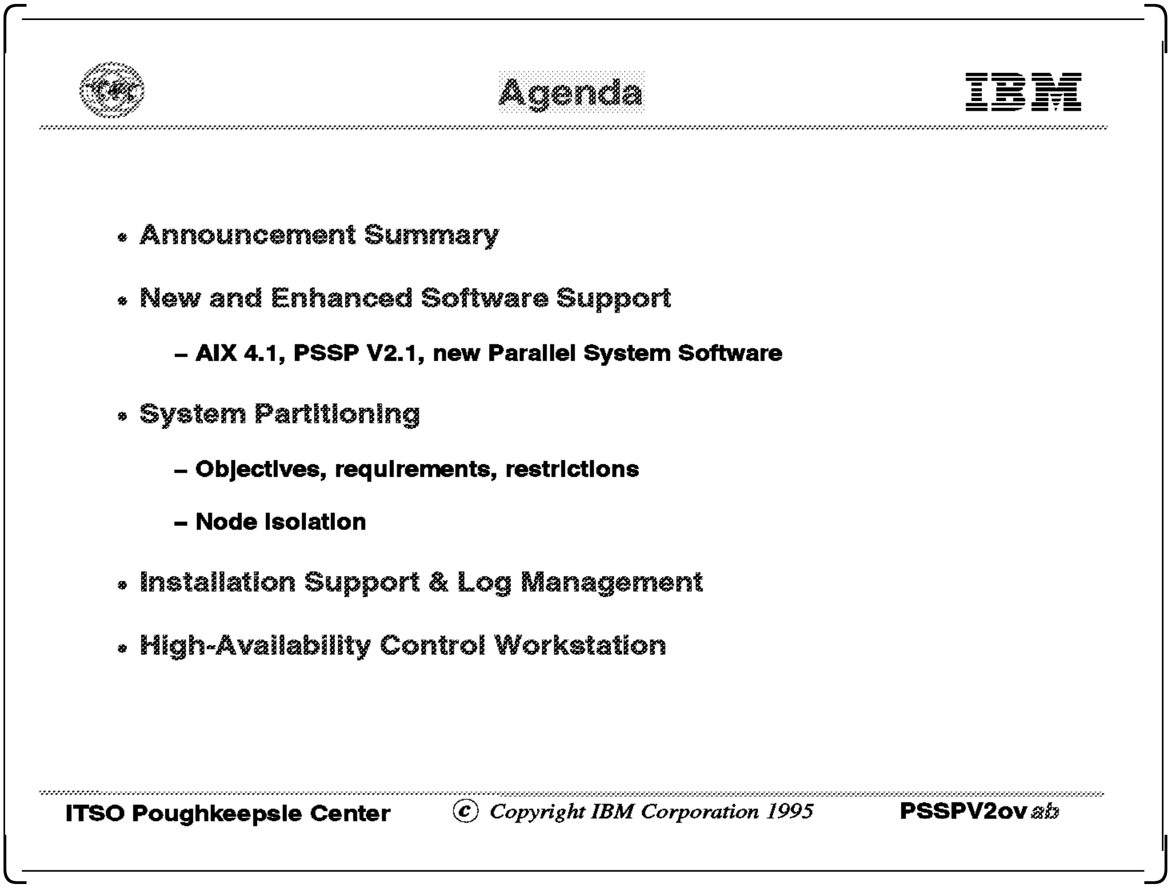
PSSP Version 2 Overview

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ov

This document is intended to provide specialists for AIX and RISC System/6000 Scalable POWERparallel (to be referred to in this book as SP) systems a transfer of knowledge and experience based on the recent June 1995 announcement by the POWERparallel Division of IBM Parallel System Support Programs Version 2. Other software products were also announced by IBM on June 19, 1995:

- IBM Parallel Environment Version 2, which includes the new message passing interface (MPI) library
- IBM PVMe Version 2, which provides new facilities, such as IP protocol support and parallel programs running on a mix of RS/6000 SP nodes and RS/6000 clustered workstations
- IBM Parallel Libraries (PESSL, POSL)
- IBM statement of direction to provide a High Performance Fortran (HPF) compiler. The HPF IBM compiler has been announced on December 5, 1995

Chapter 1. Topics Covered



Agenda

- Announcement Summary
- New and Enhanced Software Support
  - AIX 4.1, PSSP V2.1, new Parallel System Software
- System Partitioning
  - Objectives, requirements, restrictions
  - Node Isolation
- Installation Support & Log Management
- High-Availability Control Workstation

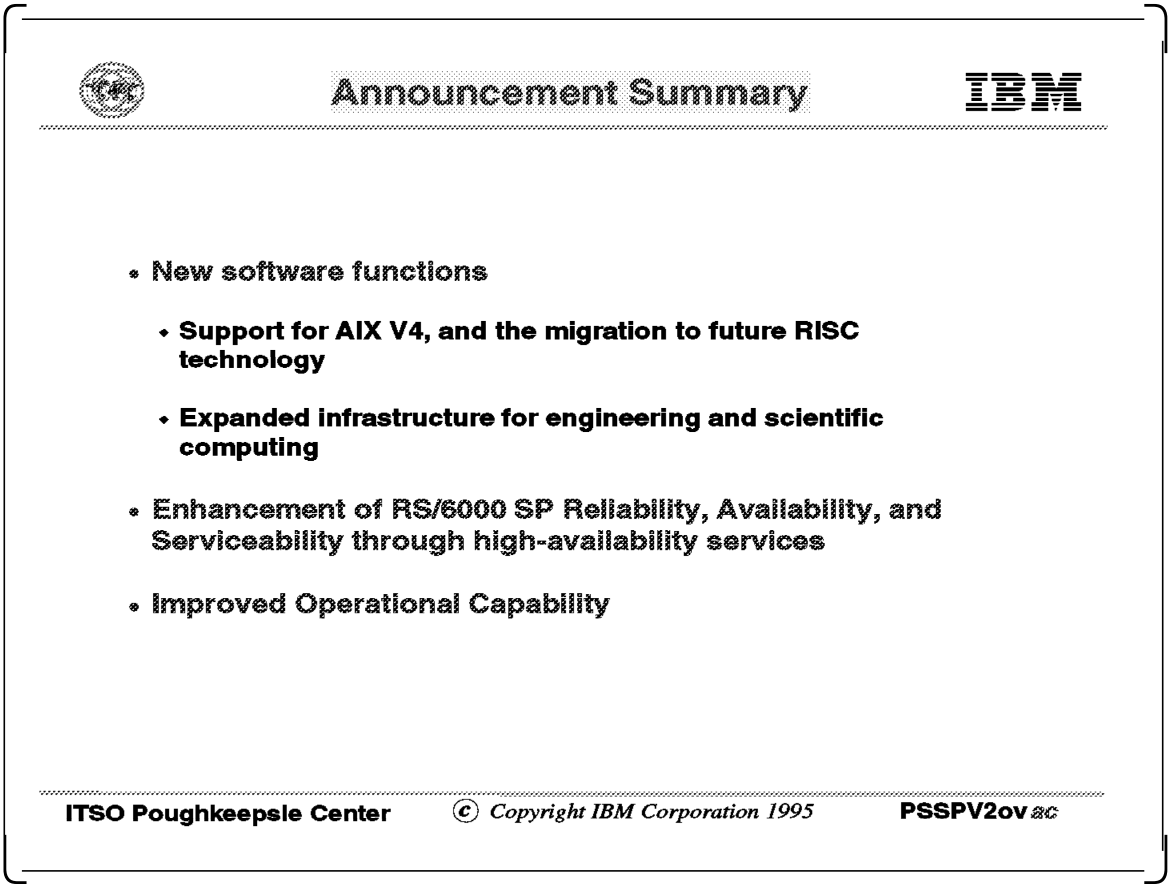
ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* PSSPV2ov*ab*

On June 19 1995, a significant release of the RS/6000 SP offering was announced. In addition to a continued close affinity with the RISC System/6000 product family, this release includes the following:

- Concurrency with AIX Version 4.1.
- New versions of Parallel System Support Programs and other software related to RS/6000 SP, such as IBM PE Version 2, IBM PVMe Version 2.
- System partitions for operations center flexibility and system migration.

  The design objectives, implementation requirements, and operational restrictions of system partitioning will be reviewed, along with the new node isolation feature, which allows non-disruptive node maintenance.
- Improved installation support is provided by the use of the AIX Version 4 Network Installation Management (NIM), and SP-specific log management is being integrated into the standard AIX Error Log.
- High availability services are being enhanced with the introduction of support for dual control workstations to back up the operations console.

1.1 Announcement Summary



Announcement Summary

IBM

- New software functions
  - Support for AIX V4, and the migration to future RISC technology
  - Expanded infrastructure for engineering and scientific computing
- Enhancement of RS/6000 SP Reliability, Availability, and Serviceability through high-availability services
- Improved Operational Capability

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ovac

The new Parallel System Support Programs (PSSP) V2.1 includes the following features:

1. AIX V4.1 support

   AIX V4.1.3 is being released on the SP in order to stay current with the product direction of the RISC System/6000 operating system. This will allow the SP system to exploit the specific benefits of AIX V4.1, and to also prepare for future hardware and software technology advances.

2. Enhancement to SP reliability, availability, and serviceability (RAS)

   - System partitioning support allowing several system partitions running AIX 3.2.5 or AIX 4.1.3 to run as logical systems in a single SP system. This provides the ability to test new software, and isolate workloads to dedicated resources.
   - Optional High-Availability Control Workstation (HACWS) feature for customers requiring high availability.
   - Node isolation, providing a mechanism to isolate nodes from the switch fabric for repair, replacement, or reboot without affecting end users, applications, or switch traffic on the remainder of the SP system or system partition.
   - New installation support that exploits the AIX 4.1 Network Installation Management (NIM) capability while maintaining current interfaces to improve administrative ease of use.

- Focus on standards for consistency in messages and error logging.
- Increased maximum ticket life (to 30 days) for Kerberos, and enhancements to address any host by any of its host names.

3. Performance improvements in IBM Virtual Shared Disk

   With standard VSD, database applications could only stripe data across multiple disks associated with one node. With Hashed Shared Disk (HSD), the capability to stripe across nodes has been added. This eliminates the manual effort required to spread large tables across multiple nodes and enhances disk utilization (and therefore performance) by allowing more simultaneous disk access. When data is written to HSD, it is put on disks in stripes of 4098 bytes. When HSD has written one stripe on one disk, it moves on to the next disk for the next stripe and repeats the process until all disks have a stripe on them, then it starts over again at the first stripe. When the database application wants to read n stripes from a Hashed Shared Disk, HSD can actually be accessing multiple stripes in parallel since they reside on different disks. HSD can therefore return more data to the application faster than serial reads.

4. Extended Support for Engineering and Scientific Computing

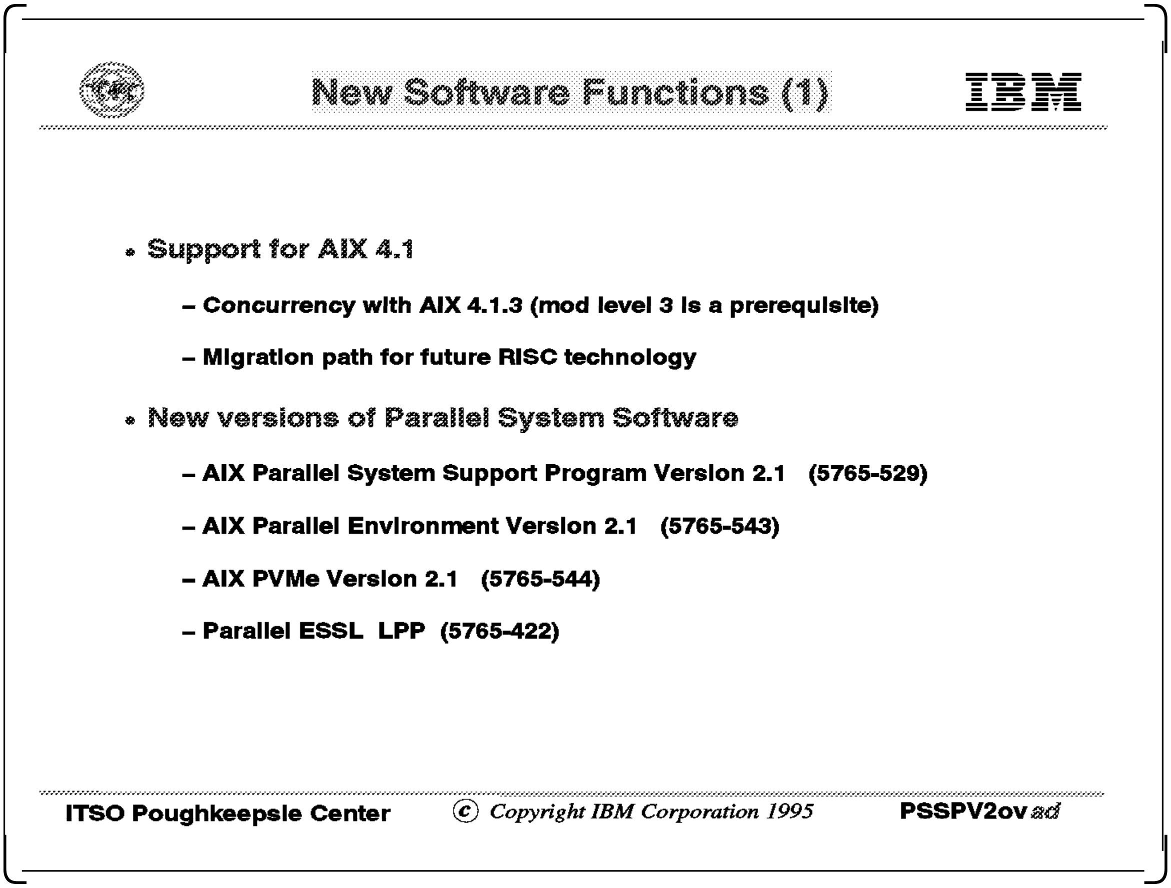
   The following software products have been upgraded with new functions to extend support for Engineering and Scientific Computing:

   - IBM Parallel Environment for AIX Version 2.1 provides support for parallel applications on AIX V4.1.3. It includes a full implementation of the Message Passing Interface (MPI) Standard, along with new parallel utilities and usability and performance improvements.

   - IBM Parallel ESSL for AIX V4 improves the performance of engineering and scientific applications on the RISC System/6000 Scalable POWERparallel (SP) systems.

   - IBM PVMe for AIX Version 2 provides user′s application source and object compatibility with the widely used Parallel Virtual Machine (PVM) Version 3.3 available from Oak Ridge National Laboratory. IBM PVMe for AIX supports parallel execution of applications on AIX Version 4.1.3.

1.2 New Software Functions



New Software Functions (1)

- **Support for AIX 4.1**
  - **Concurrency with AIX 4.1.3 (mod level 3 is a prerequisite)**
  - **Migration path for future RISC technology**
- **New versions of Parallel System Software**
  - **AIX Parallel System Support Program Version 2.1 (5765-529)**
  - **AIX Parallel Environment Version 2.1 (5765-543)**
  - **AIX PVMe Version 2.1 (5765-544)**
  - **Parallel ESSL LPP (5765-422)**

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* **PSSPV2ov***ad*

As already stated, the new version of Parallel System Support Programs Version 2.1 supports the latest level of AIX Version 4.1.3. This allows the RISC System/6000 Scalable POWERparallel (SP) system to incorporate the latest operating system technology, and because the same level of the operating system is available across the RISC family, features supported on RISC System/6000 workstations and servers are moveable to the SP system. Support for AIX 4.1 on SP systems is significant for the following reasons:

- Future symmetric multiprocessor enablement
- Common desktop environment (CDE) for clients
- File system enhancements, especially support for file systems greater than 2GB, and the Parallel I/O File System product
- Enhanced systems management (security, installation, print management, backup/restore, performance)
- As a base for new technology support (for example, system partitioning, High-Availability Control Work Station, new Serial Storage Architecture (SSD) technology and disk subsystems)

Further, new versions of the Parallel System Software were also announced; the following software products have been upgraded with new functions:

1. IBM Parallel Environment for AIX Version 2.1 provides support for parallel applications on AIX V4.1.3. It also:

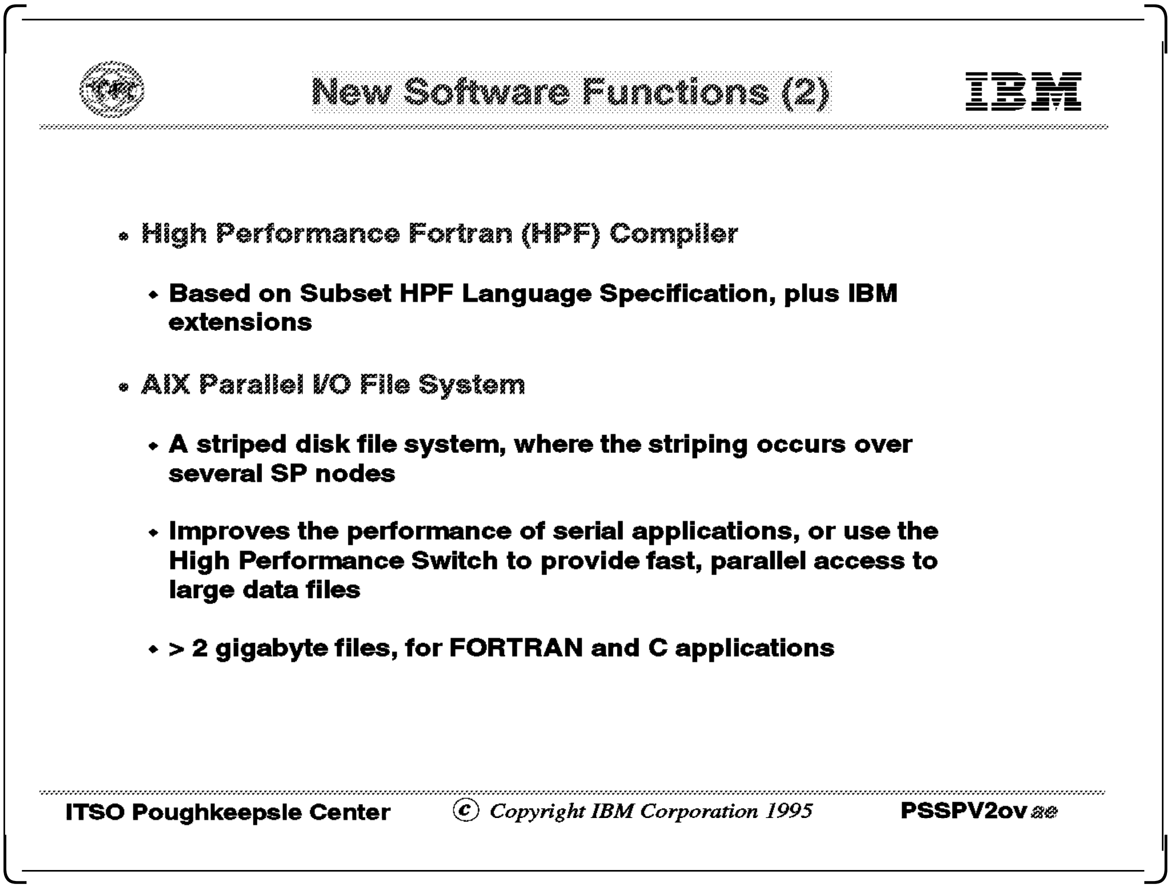
- Supports a full implementation of Message Passing Interface (MPI) Standard
- Includes new parallel utilities (copy, gather, and scatter)
- Usability and performance improvements to Visualization Tool and debuggers

2. IBM Parallel ESSL for AIX V4 improves the performance of engineering and scientific applications on the RISC System/6000 Scalable POWERparallel (SP) systems. Additional benefits of Parallel ESSL include the following:

   - Provides a parallel library tuned for performance on the SP with the High Performance Switch Adapter-2
   - Includes the ESSL/6000 product as part of Parallel ESSL
   - Allows licensing on a subset of your RS/6000 SP system
   - Supports the SPMD programming model under the IBM Parallel Environment (PE)
   - Includes the Basic Linear Algebra Communications Subprograms (BLACS), which use the MPI standard for communication between nodes
   - Fully compatible with de facto standard subroutine packages, for example, ScaLAPACK and PBLAS
   - Callable from FORTRAN, C, and C++

3. IBM PVMe for AIX Version 2 provides user′s application source and object compatibility with the widely used Parallel Virtual Machine (PVM) Version 3.3 available from Oak Ridge National Laboratory. IBM PVMe for AIX supports parallel execution of applications on AIX Version 4.1.3.

   Using IBM PVMe for AIX, you can migrate applications that contain PVM constructs from any of the many parallel computing systems supported by PVM Version 3.3 to RISC System/6000 Scalable POWERparallel Systems computers. This migration requires no changes to the PVM constructs; applications must simply be re-linked with the IBM PVMe for AIX libraries instead of the public domain PVM library.



New Software Functions (2)

- High Performance Fortran (HPF) Compiler
  - Based on Subset HPF Language Specification, plus IBM extensions
- AIX Parallel I/O File System
  - A striped disk file system, where the striping occurs over several SP nodes
  - Improves the performance of serial applications, or use the High Performance Switch to provide fast, parallel access to large data files
  - > 2 gigabyte files, for FORTRAN and C applications

ITSO Poughkeepsie Center   © *Copyright IBM Corporation 1995*   PSSPV2ov ae

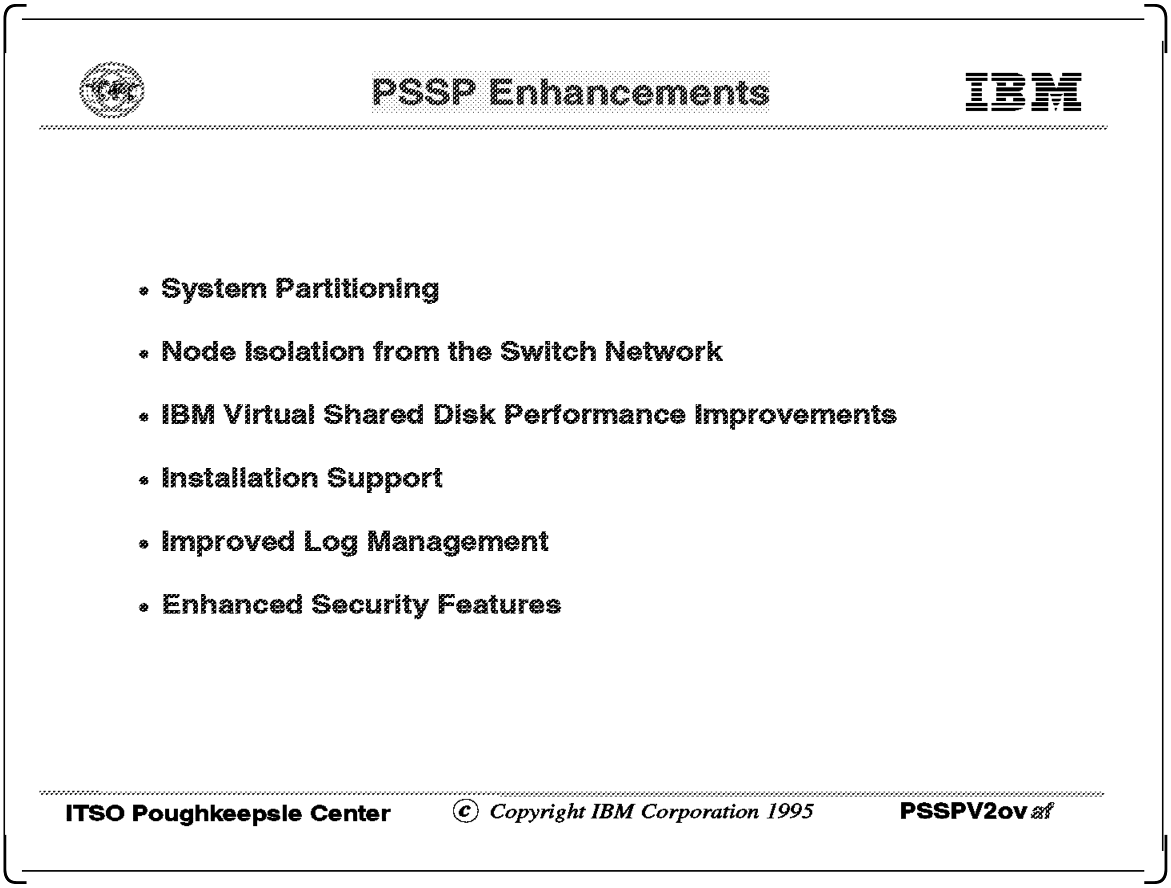
- On December 5, 1995, IBM also announced a High Performance Fortran (HPF) compiler that is be based on subset HPF, as defined by the "High Performance Fortran Language Specification, Version 1.1. Rice University, 1994." In addition to complying with these specifications, the IBM HPF compiler provides the following extensions:
  - PURE procedures
  - FORALL constructs and statements
  - Storage and sequence association, including the SEQUENCE directive (but not supporting mapping of sequenced variables)
  - HPF_LOCAL and HPF_SERIAL extrinsic kinds (on subroutines and functions only)
  - Selected features of both the HPF_LOCAL_LIBRARY and HPF_LIBRARY modules
  - Substantial parallel Fortran 90 support

  IBM′s HPF supports analysis and parallel execution of FORTRAN 77 DO loops as well as Fortran 90 array language.

- Also announced was the first release of the AIX Parallel I/O File System (PIOFS), which was designed for RISC System/6000 Scalable POWERparallel Systems and is capable of scaling in file input/output (I/O) performance, just as the RISC System/6000 SP scales in computing performance. Some of the features of PIOFS include:
  - Provides additional functions to allow larger than 2-gigabyte files using 64-bit file offsets
  - Provides capacity and performance from multiple file servers all accessible to the client through a high-performance network

- Eliminates bottlenecks by allowing tasks to read and write to separate portions of a file simultaneously
- Provides file checkpointing
- Provides file striping across multiple server nodes
- Provides data striping across multiple physical disks within storage nodes
- Coexists with other file servers and file systems
- Provides a FORTRAN and C end user interface for parallel I/O
- Allows the user to organize multiple, different logical views of the data

1.3 PSSP Enhancements



PSSP Enhancements

- System Partitioning
- Node Isolation from the Switch Network
- IBM Virtual Shared Disk Performance Improvements
- Installation Support
- Improved Log Management
- Enhanced Security Features

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ov af

This new version of PSSP offers a solid foundation on which to execute production applications on the IBM RISC System/6000 Scalable POWERparallel (SP) system.

As part of the POWERparallel development laboratory strategy, AIX 4.1.3 is being released on the SP in order to stay current with the product direction of the RISC System/6000 operating system. This release of PSSP for AIX focuses on administrative ease of use, high availability, and improved price/performance. The following are the enhancements:

- System Partitioning

  System partitioning enhances the overall availability of the SP system by providing the capability to divide the system into logical SP systems. System partitioning also provides multiple application environments in a single SP system. This allows exclusive use of a portion of the High Performance Switch for a single application environment. In addition, system partitioning provides the ability to operate some nodes at an AIX 3.2.5 production level, while migrating and testing applications and system software on AIX 4.1 nodes. Or, multiple AIX 4.1 system partitions may be defined, and may be used for applying service or new licensed programs to one, while running production workload on others. In all of these scenarios, the entire system is still monitored and controlled from a single point, the control workstation.

- Node Isolation from the Switch Network

  Nodes that are currently part of the switch network can be isolated from that network for repair, replacement, or reboot without affecting end users, applications, or switch traffic on the remainder of the SP system.

- Performance Improvements

  IBM Virtual Shared Disk performance is improved in this release. With standard VSD, database applications could only stripe data across multiple disks associated with one node. With Hashed Shared Disk (HSD), the capability to stripe across nodes has been added. This eliminates the manual effort required to spread large tables across multiple nodes and enhances disk utilization (and therefore performance) by allowing more simultaneous disk access. When data is written to HSD, it is put on disks in stripes of 4098 bytes. When HSD has written one stripe on one disk, it moves on to the next disk for the next stripe and repeats the process until all disks have a stripe on them, then it starts over again at the first stripe. When the database application wants to read n stripes from a Hashed Shared Disk, HSD can actually be accessing multiple stripes in parallel since they reside on different disks. HSD can therefore return more data to the application faster than serial reads.

- Installation Support

  Ease of administration is increased via the use of the AIX 4.1 Network Installation Management (NIM) function.

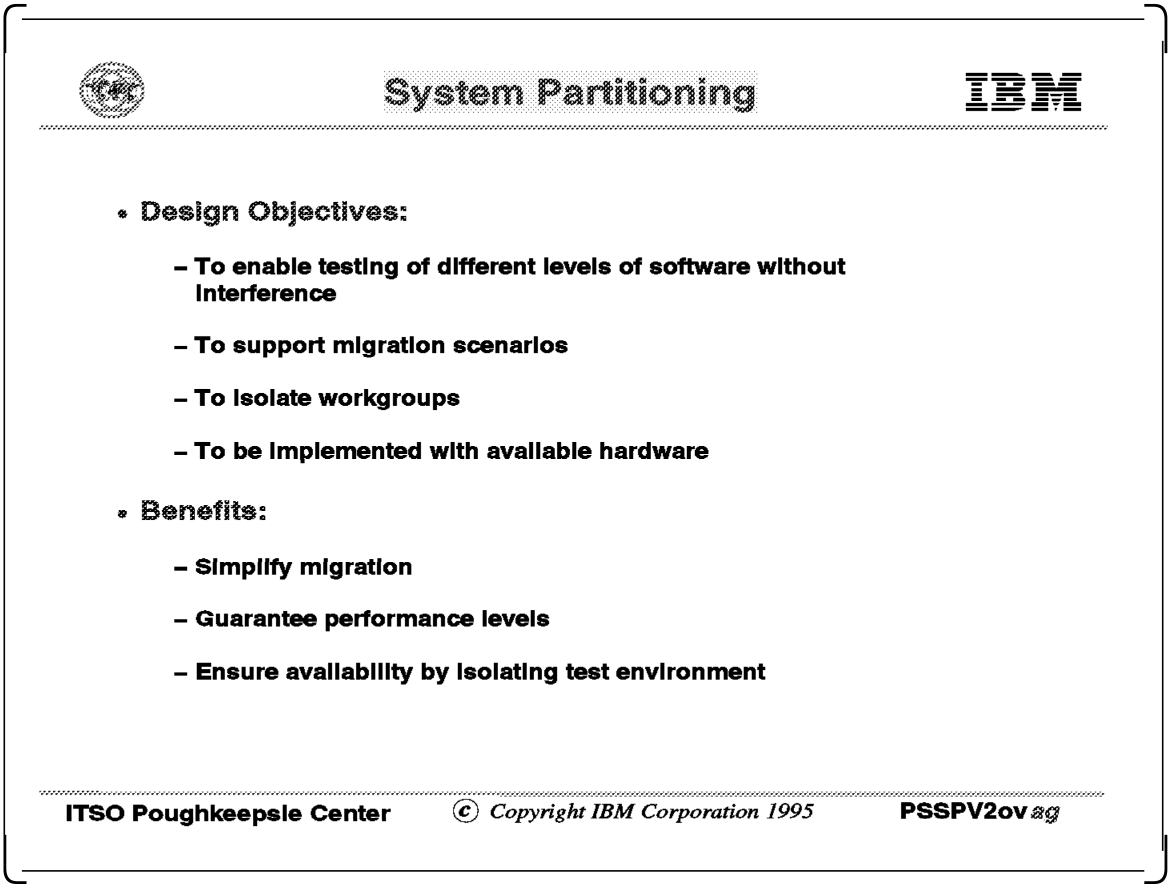
- Log Management

  Serviceability is significantly enhanced using the Log Management function. In multi-node configurations, the number of logs produced can be cumbersome. Log Management provides a single point of control to assist customers in configuring, archiving, and maintaining various system logs on individual nodes. This function also allows customers to specify additional logs that may be used by their applications.

- Enhanced Security Solution

  The Kerberos default ticket life has been increased from 21 hours to 30 days. The Kerberos support used by `sysctl` and remote commands is enhanced to allow addressing a host by any of its valid host names.

1.4 System Partitioning



System Partitioning

- Design Objectives:
  - To enable testing of different levels of software without interference
  - To support migration scenarios
  - To isolate workgroups
  - To be implemented with available hardware
- Benefits:
  - Simplify migration
  - Guarantee performance levels
  - Ensure availability by isolating test environment

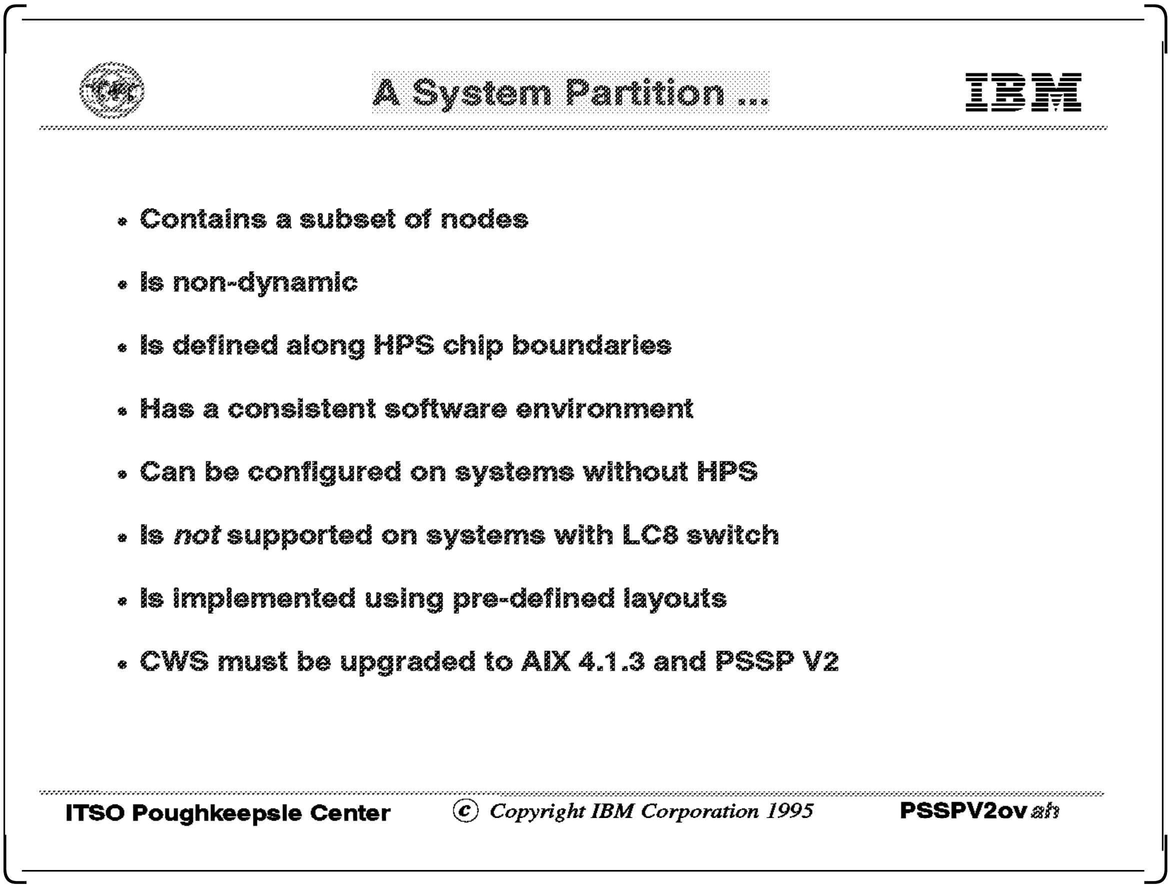
ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* PSSPV2ov*ag*

The design objectives of System Partitioning are to:

1. Provide the capability to test different levels of software in a non-interfering manner.
2. Support migration scenarios from previous PSSP levels of SP software to the current levels.
3. Make system partitions look like logical SP systems to software subsystems and users.
4. Not require any hardware changes.

The benefits that System Partitioning will provide include:

1. An improved ability to migrate to new software releases for system software, applications, database systems or tools.
2. Guaranteed service and performance levels by dedicating compute resources to specific SP customer clients.
3. Improved availability for production partitions by isolating them from test environments. For example, a new tape drive or RAID disk controller can be installed and checked out by using the test partition of nodes. Meanwhile, console operators and systems staff can manage both the production nodes groups and the test nodes groups as if they are a single system.



A System Partition ...

- Contains a subset of nodes
- Is non-dynamic
- Is defined along HPS chip boundaries
- Has a consistent software environment
- Can be configured on systems without HPS
- Is *not* supported on systems with LC8 switch
- Is implemented using pre-defined layouts
- CWS must be upgraded to AIX 4.1.3 and PSSP V2

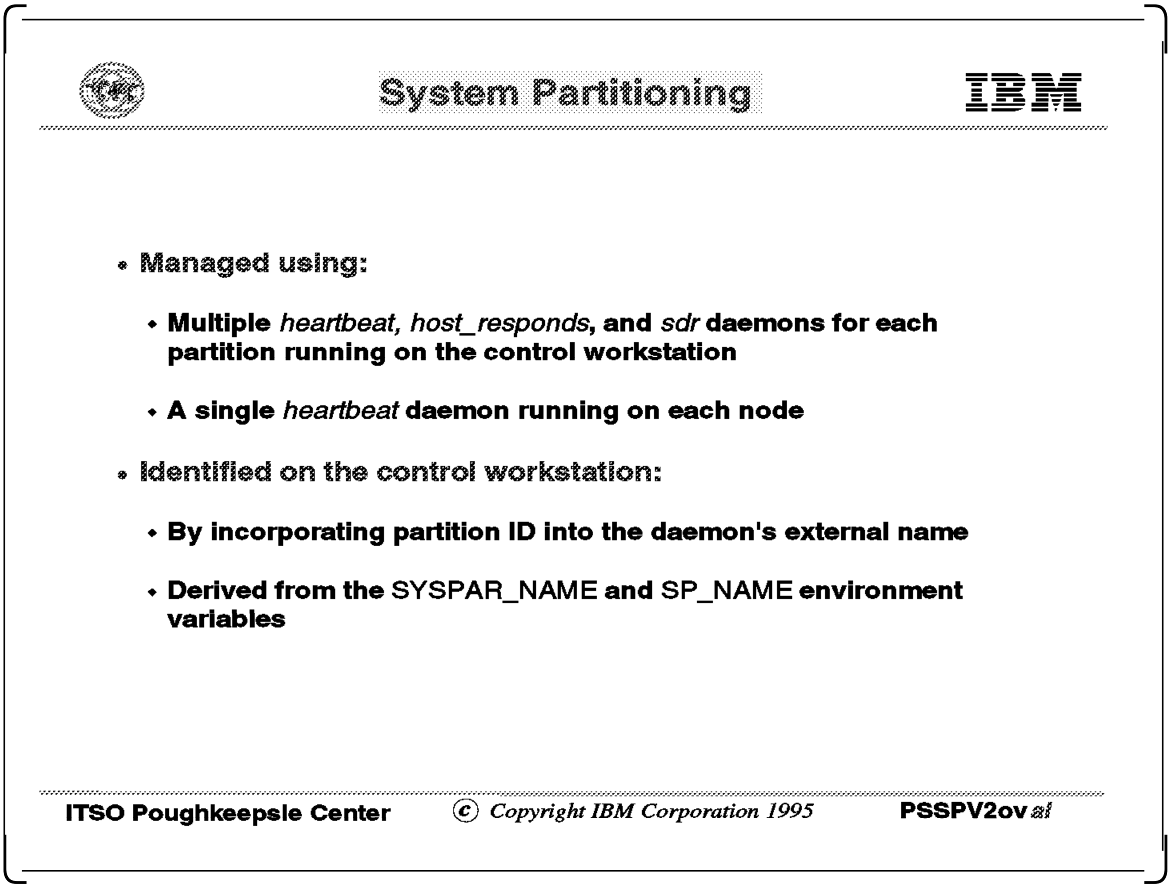
ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* PSSPV2ov*ah*

A system partition is a fairly static entity that contains a specified subset of nodes on a switch chip boundary and a consistent software environment further defined as follows:

- All nodes within a system partition are at the same release level of AIX.
- All nodes within a system partition are at the same release level of SP software.

In addition, partitions can only be defined along High Performance Switch chip boundaries. The smallest partition that can be defined comprises two slots. Each system partition has a logical switch and a primary node (used for switch initialization). The initialization and topology files are contained within a system partition. The layouts are predefined, and cannot be changed from the ones provided. Switch faults and message traffic are contained within a system partition.

System partitioning is *not* supported on systems that use the LC8 switch. The control workstation must be upgraded to AIX V4.1.3 and PSSP V2.1 to enable system partitioning.



System Partitioning

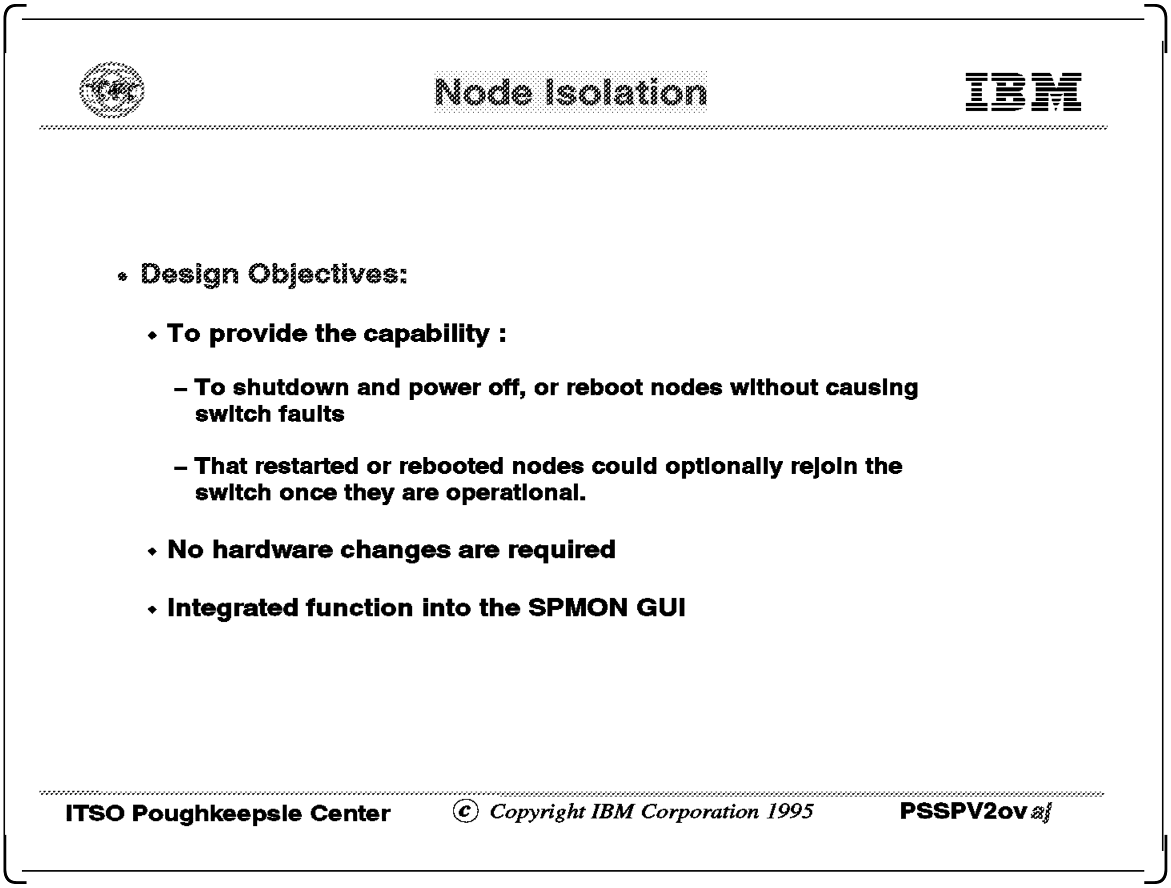
- **Managed using:**
  - **Multiple** ***heartbeat, host_responds*****, and** ***sdr*** **daemons for each partition running on the control workstation**
  - **A single** ***heartbeat*** **daemon running on each node**
- **Identified on the control workstation:**
  - **By incorporating partition ID into the daemon's external name**
  - **Derived from the** SYSPAR_NAME **and** SP_NAME **environment variables**

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* **PSSPV2ov** al

The management of system partitions on the control workstation is achieved through the use of multiple heartbeat, host_responds, and sdr daemons, one for each partition. On each node, only a single heartbeat daemon is configured to monitor the status of the node.

The administration of the partitions is achieved by incorporating the partition name into the daemons′ external name on the control workstation. Two environment variables (SP_NAME and SYSPAR_NAME), which are set on the control workstation, are used to discriminate between different partitions.

1.5 Node Isolation



Node Isolation

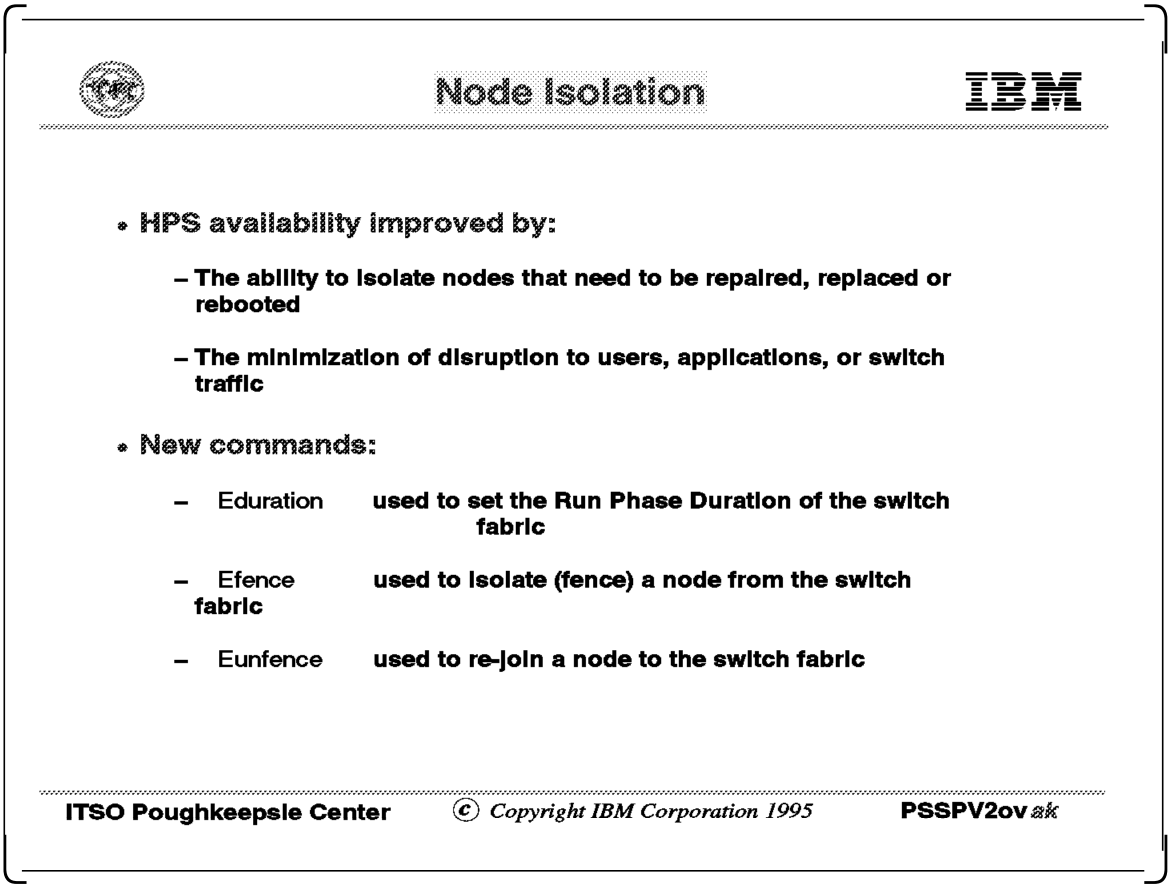
- Design Objectives:
  - To provide the capability :
    - To shutdown and power off, or reboot nodes without causing switch faults
    - That restarted or rebooted nodes could optionally rejoin the switch once they are operational.
  - No hardware changes are required
  - Integrated function into the SPMON GUI

ITSO Poughkeepsie Center ⓒ *Copyright IBM Corporation 1995* PSSPV2ov*af*

The following are the design objectives of node isolation:

- Provide the capability to shutdown, power off or reboot nodes without causing switch faults
- Allow nodes that are restarted to optionally rejoin the switch once they are operational

These functions were implemented without requiring any hardware changes, and they were integrated into the SPMON interface.



Node Isolation

- **HPS availability improved by:**
  - **The ability to isolate nodes that need to be repaired, replaced or rebooted**
  - **The minimization of disruption to users, applications, or switch traffic**
- **New commands:**
  - Eduration **used to set the Run Phase Duration of the switch fabric**
  - Efence **used to isolate (fence) a node from the switch fabric**
  - Eunfence **used to re-join a node to the switch fabric**

ITSO Poughkeepsie Center Ⓒ *Copyright IBM Corporation 1995* **PSSPV2ov**ak

Node isolation improves the availability of the switch during the time that nodes need to be shutdown or rebooted for any reason, including service. It provides the ability for nodes that are part of the switch network to be isolated from that network, preventing unnecessary faulting of the switch. When the node is operational again and the fault service daemon is running on that node, it can rejoin the switch fabric, assuming the switch has not been initialized since the node was isolated. The rejoining is done with a de-isolation request and can be configured to be automatic or manual.

The node isolation external interface is provided by the following three commands on the control workstation:

Command **Description**

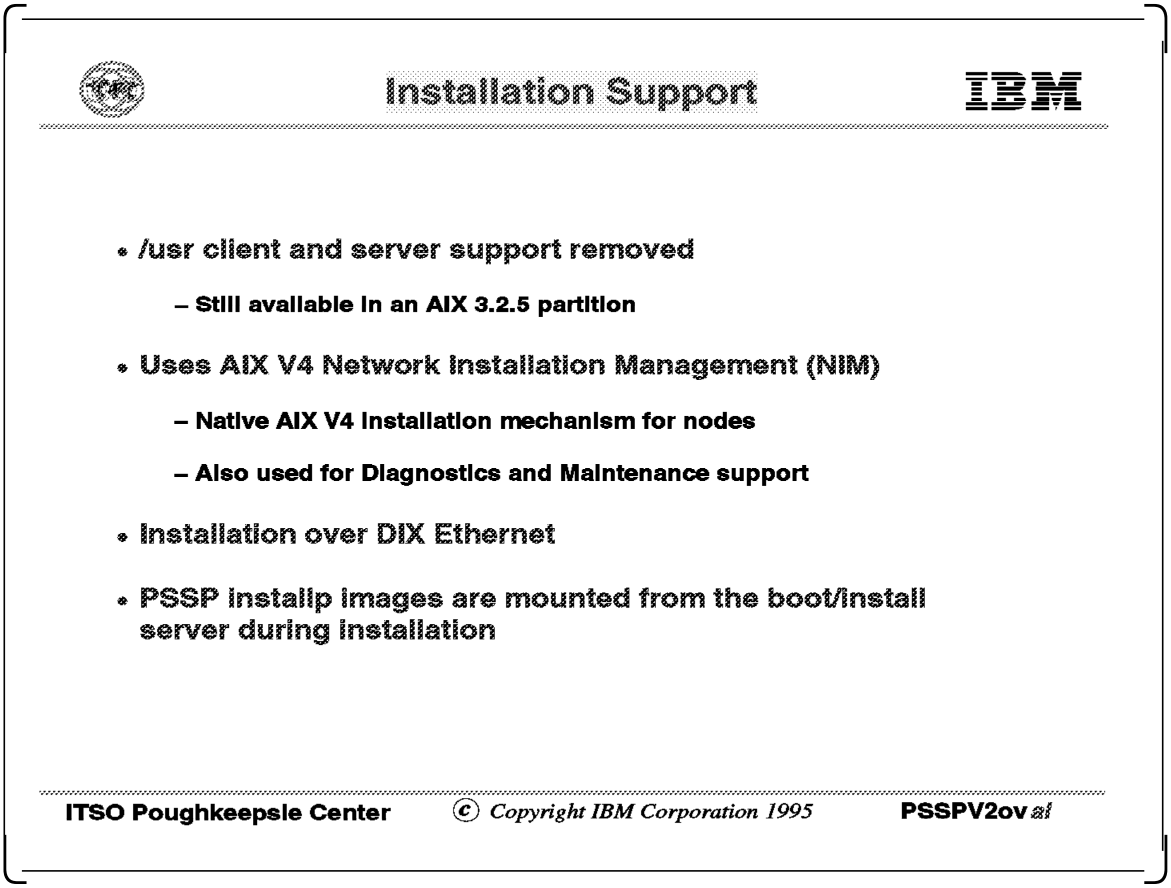
Eduration This is used to set the Run Phase Duration of the switch fabric. The duration of the run phase determines how quickly the system responds to node isolation requests. The default duration is two minutes.

Efence This is used to isolate (or fence) a node from the switch fabric.

Eunfence This is used to unfence a node and return it to the switch fabric.

Note: It is not possible to fence (or unfence) the node defined as the primary node.

1.6 Installation Support



Installation Support

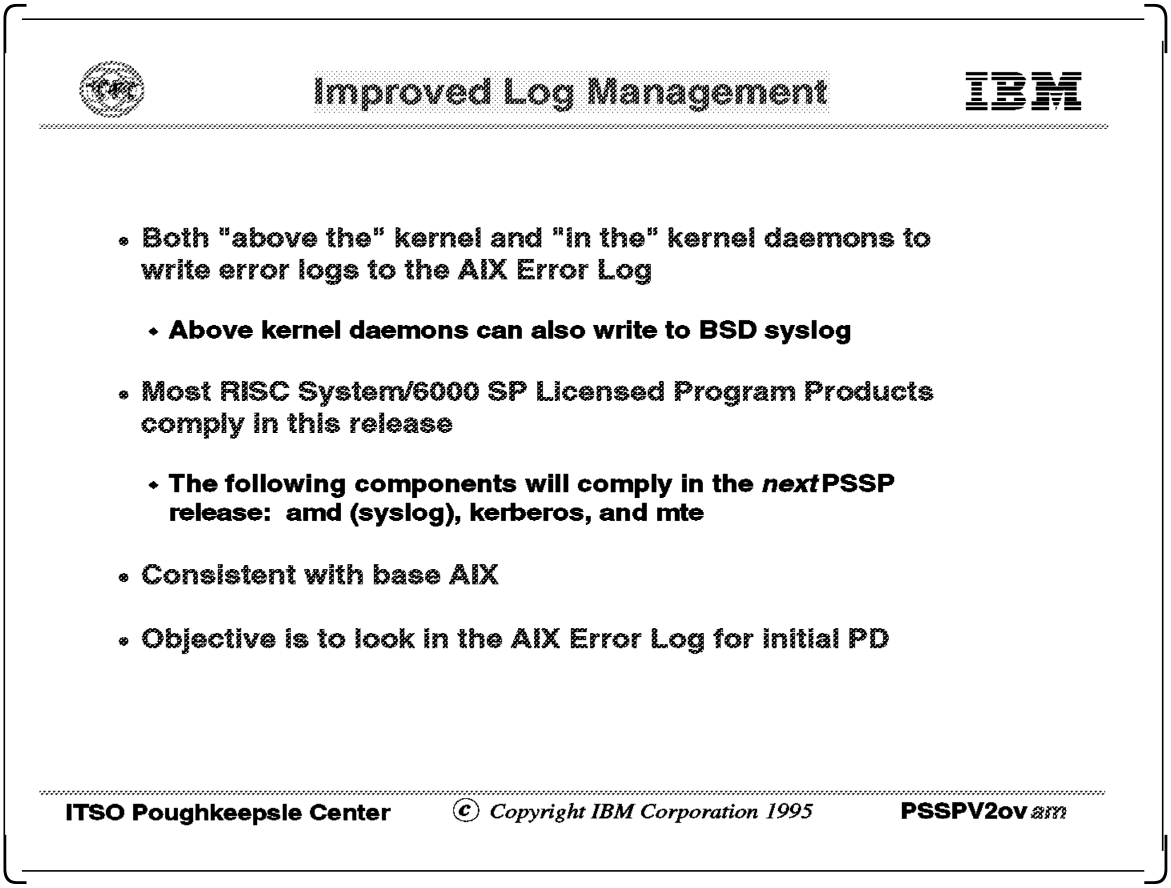
- /usr client and server support removed
  - Still available in an AIX 3.2.5 partition
- Uses AIX V4 Network Installation Management (NIM)
  - Native AIX V4 installation mechanism for nodes
  - Also used for Diagnostics and Maintenance support
- Installation over DIX Ethernet
- PSSP installp images are mounted from the boot/install server during installation

ITSO Poughkeepsie Center ⓒ *Copyright IBM Corporation 1995* PSSPV2ov*al*

There have been a number of significant changes in installation support for the Parallel System Support Programs (PSSP). These include:

- The removal of support for /usr client and /usr server, although it is still available in an AIX 3.2.5 partition.
- The use of AIX V4.1 Network Installation Management (NIM). This is the standard network installation tool for AIX 4.1. NIM is also used to provide diagnostics and maintenance support over the SP network.
- Support for installation over DIX Ethernet.
- Mounting the PSSP install images from the boot/install server during installation.
- Other changes include:
  - Changes to the customization scripts.
  - Customization no longer requires a reboot.
  - Nodes are no longer pre-installed for new systems.

1.7 Log Management



Improved Log Management

- Both "above the" kernel and "in the" kernel daemons to write error logs to the AIX Error Log
  - Above kernel daemons can also write to BSD syslog
- Most RISC System/6000 SP Licensed Program Products comply in this release
  - The following components will comply in the *next* PSSP release: amd (syslog), kerberos, and mte
- Consistent with base AIX
- Objective is to look in the AIX Error Log for initial PD

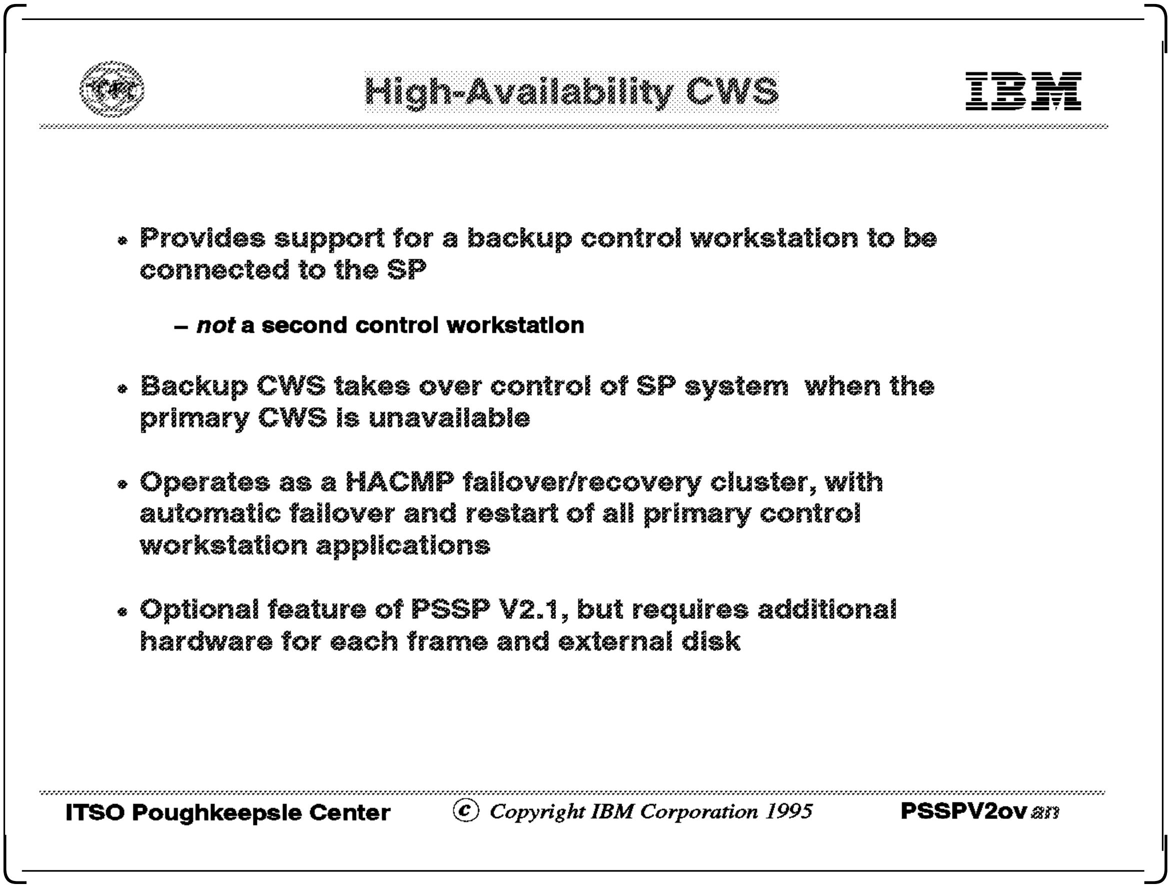
ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* PSSPV2ov*am*

The error logging policy that has been adopted by RS/6000 SP is that programs will write their error logs to the AIX Error Log. The objective is that all RS/6000 SP LPPs will comply by the *next* release of PSSP Version 2. In this release of PSSP V2, the only exceptions are public domain code (amd, kerberos, and mte).

The benefits gained by this new policy include:

- Easier management of error logs
- First step to First Failure Data Check (FFDC)
- Move from re-creation problem determination procedures to diagnostic-based procedures

1.8 High-Availability CWS



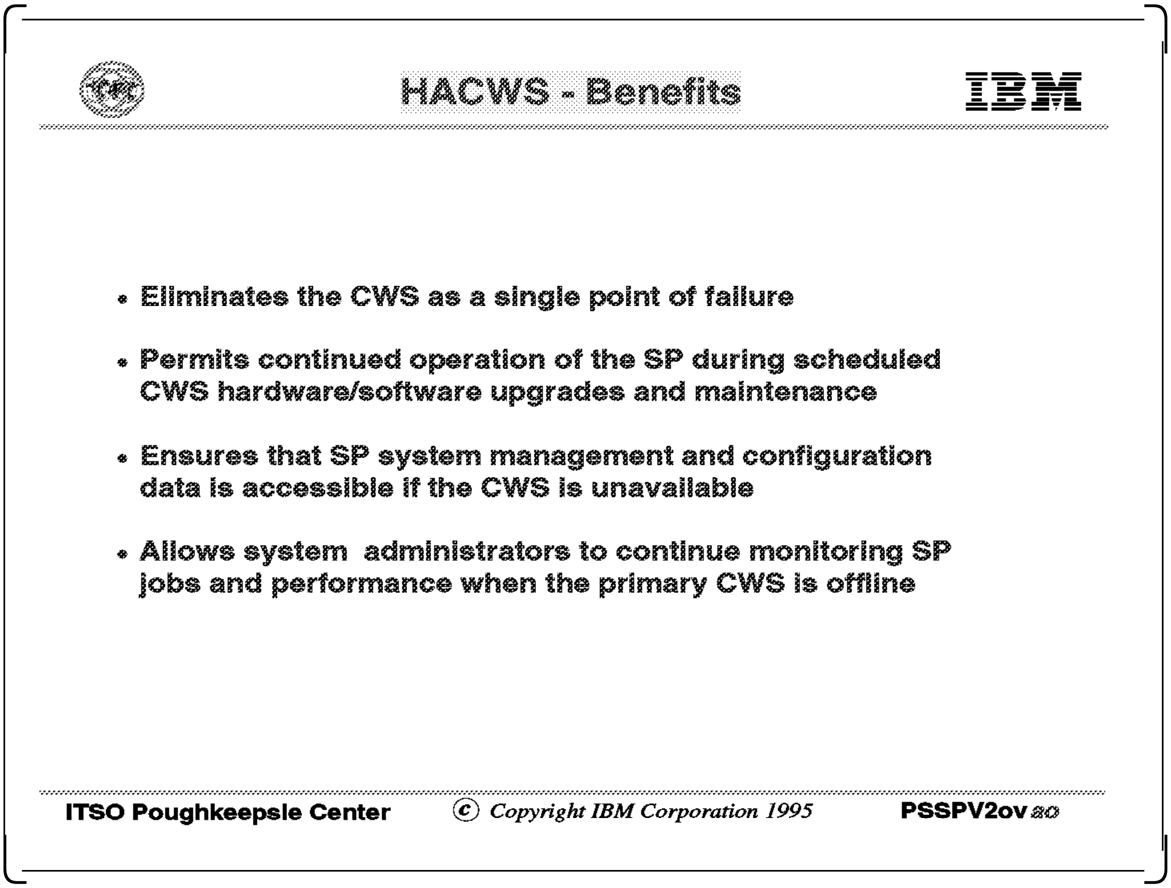
High-Availability CWS IBM

- **Provides support for a backup control workstation to be connected to the SP**
  - ***not*** **a second control workstation**
- **Backup CWS takes over control of SP system when the primary CWS is unavailable**
- **Operates as a HACMP failover/recovery cluster, with automatic failover and restart of all primary control workstation applications**
- **Optional feature of PSSP V2.1, but requires additional hardware for each frame and external disk**

ITSO Poughkeepsie Center ⓒ *Copyright IBM Corporation 1995* **PSSPV2ov***an*

The High Availability Control Workstation (HACWS) feature allows a backup control workstation to be connected to an SP machine. The backup control workstation takes over control of the SP system when the primary workstation is unavailable due to scheduled software upgrades or to unscheduled outages during normal operation. HACWS operates as part of an HACMP failover/recovery cluster, providing automated failover and restart of control workstation services. HACWS eliminates the control workstation as a single point of failure.

The HACWS feature is implemented with a dual RS-232 port on the SP frames. This is achieved by including a hot pluggable, dual-connector Y-cable as part of a new frame supervisor package. The hardware upgrade provides one connection each for the primary and backup control workstation. Only one of the tails will be active at any time.



HACWS - Benefits

- Eliminates the CWS as a single point of failure
- Permits continued operation of the SP during scheduled CWS hardware/software upgrades and maintenance
- Ensures that SP system management and configuration data is accessible if the CWS is unavailable
- Allows system administrators to continue monitoring SP jobs and performance when the primary CWS is offline

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* PSSPV2ov20

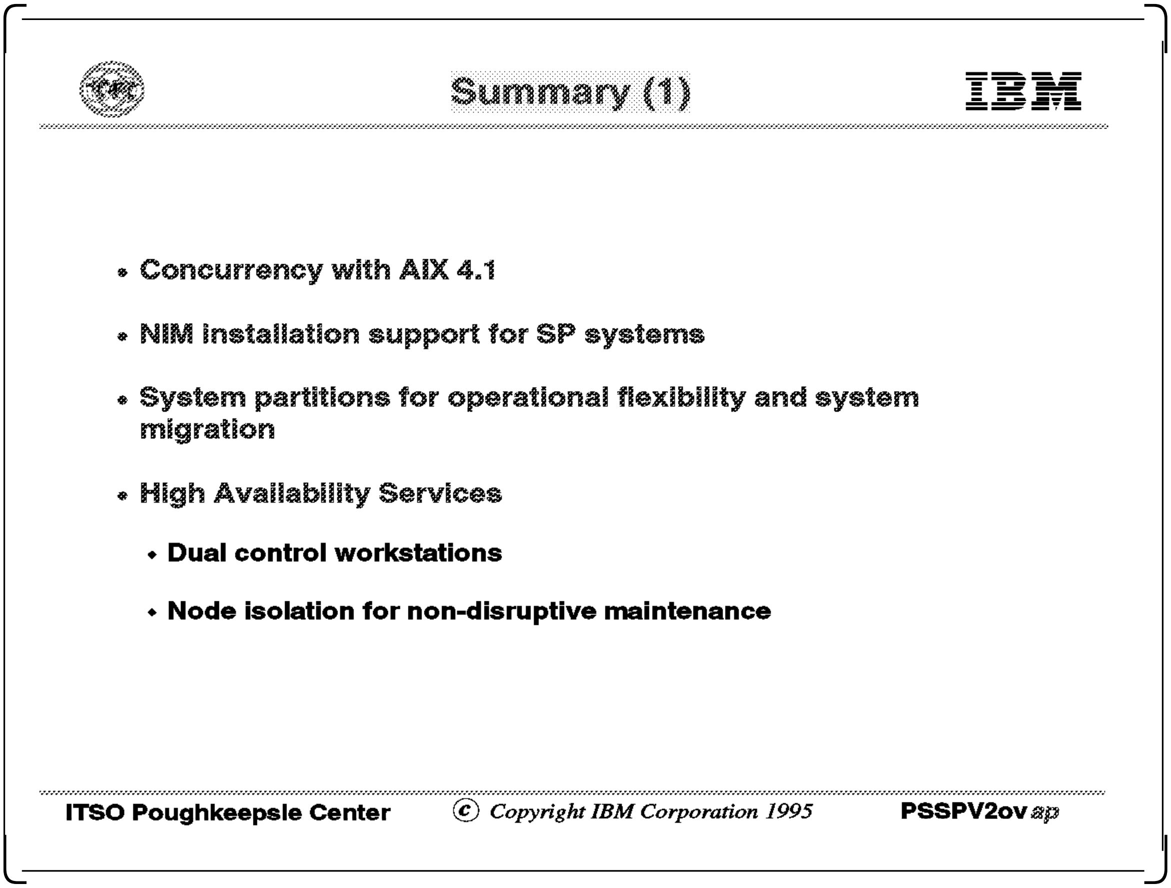
The following are the major benefits of implementing HACWS:

- Eliminates the control workstation as a single point of failure
- Allows customers to schedule hardware and software upgrades or maintenance for the primary control workstation
- Ensures SP system management and configuration data is accessible if the primary control workstation is down

However, there are also some restrictions.

- All nodes must be at AIX V4.1.3 level.
- Either the primary or backup control workstation provides all the function at one time — the load cannot be split across the two workstations.
- The backup control workstation cannot be used as the control workstation for another SP system.

1.9 Summary



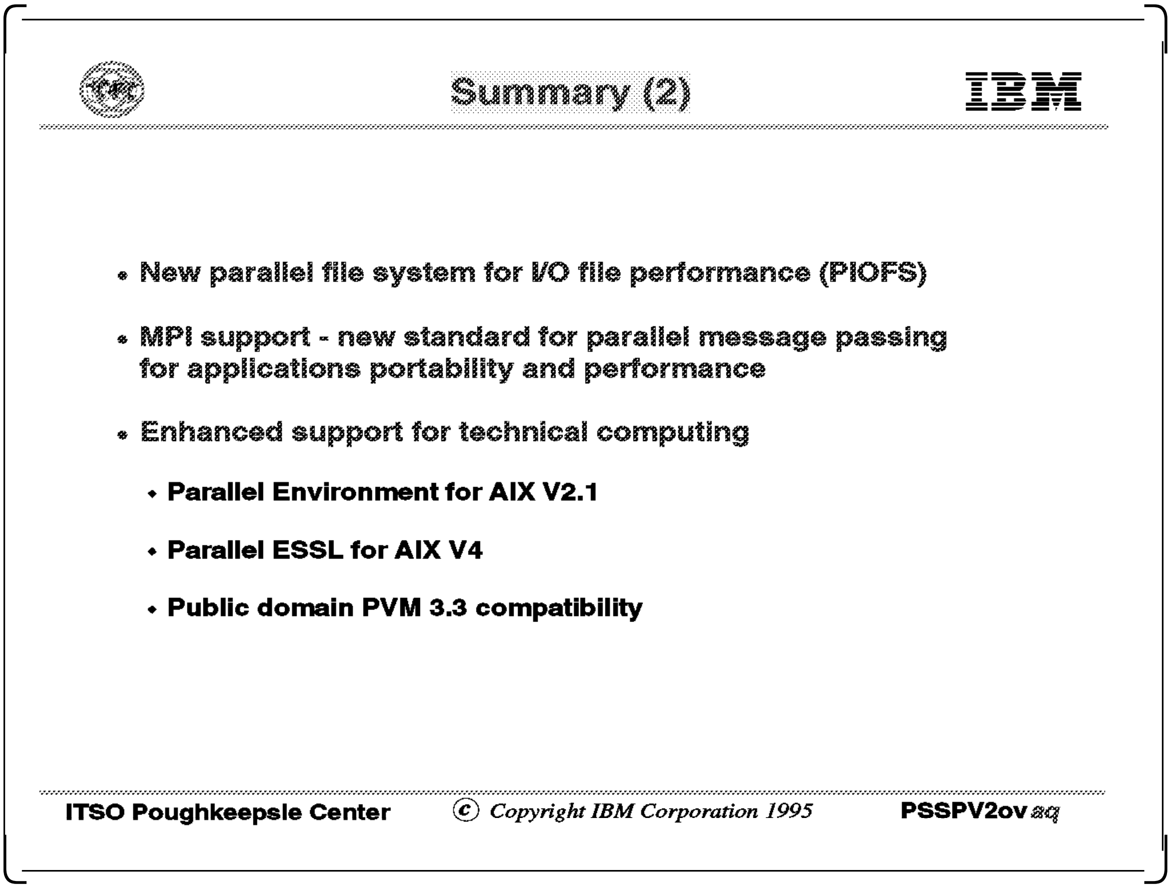
Summary (1)

- Concurrency with AIX 4.1
- NIM installation support for SP systems
- System partitions for operational flexibility and system migration
- High Availability Services
  - Dual control workstations
  - Node isolation for non-disruptive maintenance

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ovsp

To summarize, the key messages for this new level of software are as follows:

1. Keeping pace with current technology:
   - Concurrency with AIX 4.1
   - NIM installation support
2. Operational enhancements:
   - System partitioning
   - Node isolation
   - High-Availability control workstation



Summary (2)

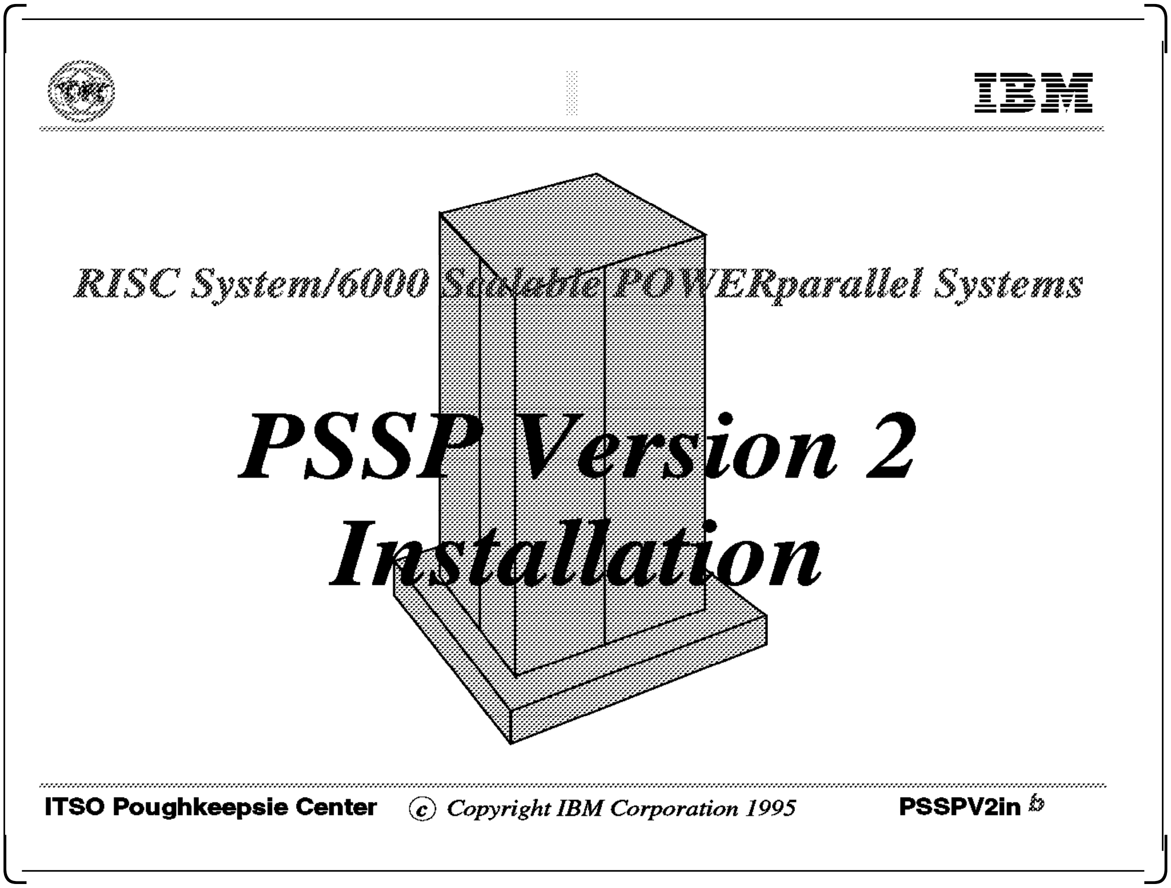
- New parallel file system for I/O file performance (PIOFS)
- MPI support - new standard for parallel message passing for applications portability and performance
- Enhanced support for technical computing
  - Parallel Environment for AIX V2.1
  - Parallel ESSL for AIX V4
  - Public domain PVM 3.3 compatibility

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* PSSPV2ov*aq*

3. Extending engineering and scientific computing capabilities:

- Parallel I/O File System
- Support for the new MPI standard
- New versions of:
  - IBM Parallel Environment for AIX
  - IBM Parallel ESSL for AIX
  - IBM PVMe for AIX (compatible with public domain PVM 3.3)

Part 2. PSSP V2 Installation



RISC System/6000 Scalable POWERparallel Systems

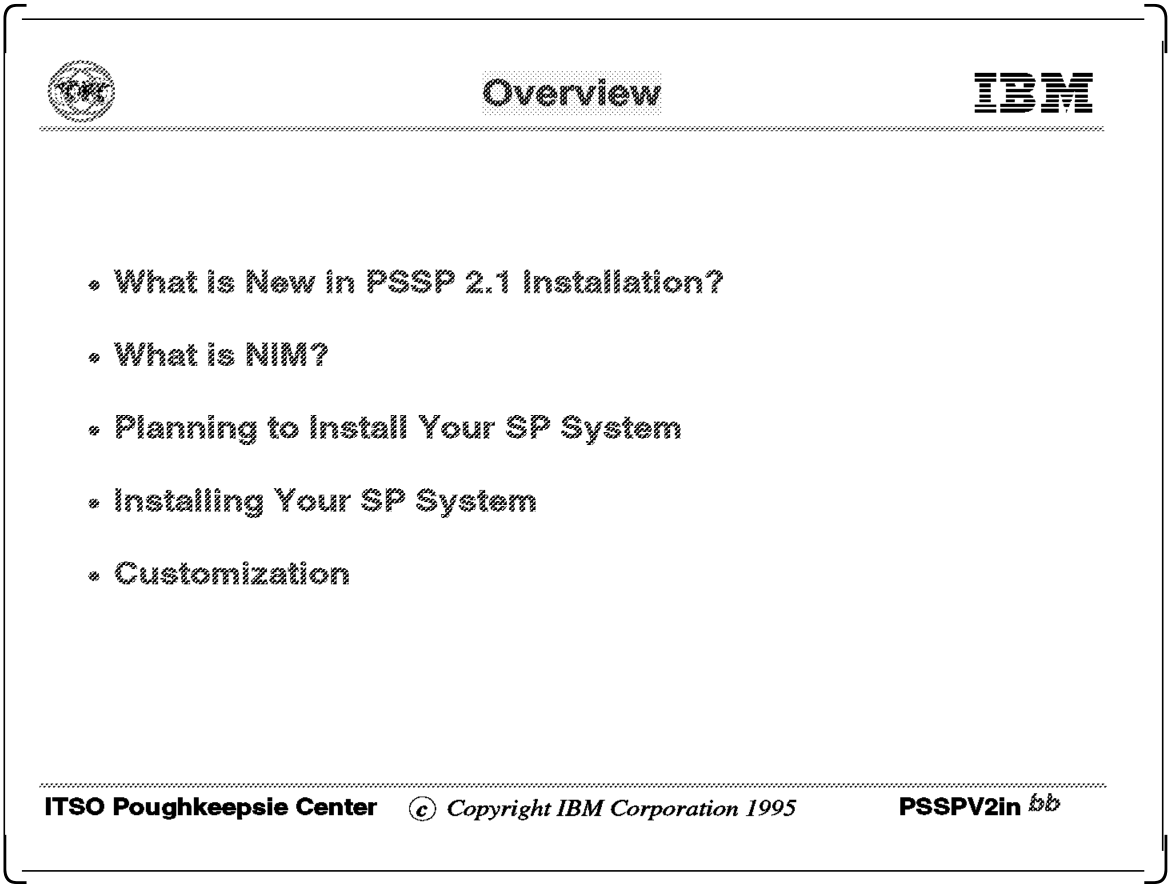
PSSP Version 2 Installation

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2in

In this chapter we present a standard installation of PSSP on an RS/6000 SP system. When you install the SP system you can partition your system or do a migration install. You will find specific information about partition management in Part 3, "PSSP V2 Partitioning" on page 75.

When installing PSSP Version 2 Release 1 on an existing RS/6000 SP running AIX 3.2.5 and PSSP Version 1 release 2, several scenarios are possible. They are well described in *SP Installation Guide*.

Chapter 2. Overview



Overview

IBM

- What is New in PSSP 2.1 Installation?
- What is NIM?
- Planning to Install Your SP System
- Installing Your SP System
- Customization

ITSO Poughkeepsie Center ⓒ Copyright IBM Corporation 1995 PSSPV2in bb

This section is an overview about PSSP Version 2 Release 1, and includes the following topics:

- Section 2.1, "What Is New in PSSP 2.1 Installation?" on page 26

  This section decribes the new features included in PSSP V2.

- Section 2.2, "What Is NIM?" on page 29

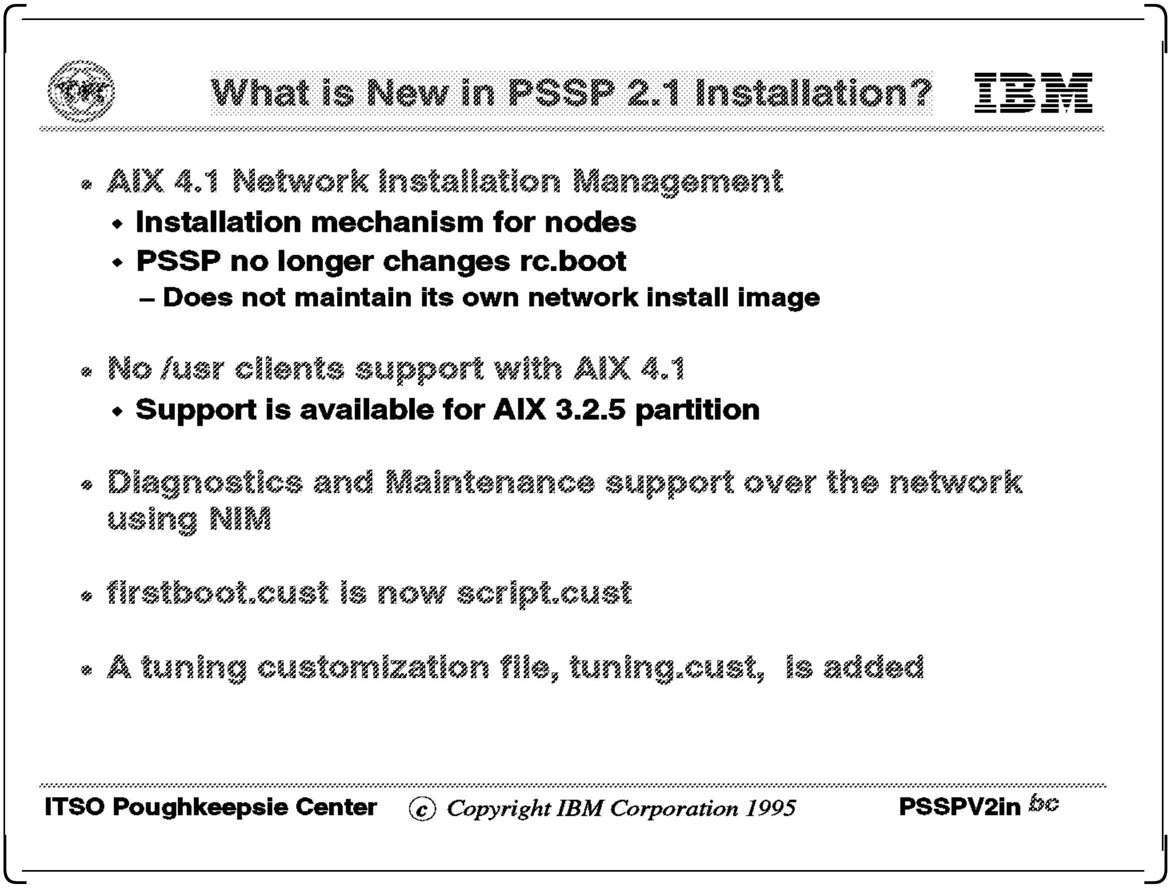
  This section provides some information about the way NIM is used to define and boot the RS/6000 SP nodes.

- Section 2.4, "Planning to Install your SP System" on page 33

  This section is devoted to the RS/6000 SP installation planning.

- Section Chapter 3, "Installing and Configuring the SP System" on page 35 This section describes the PSSP V2 installation process.
- Section Chapter 4, "Configure the CDE Desktop" on page 73 This section gives you some information about the PSSP V2 customization.

2.1 What Is New in PSSP 2.1 Installation?



What is New in PSSP 2.1 Installation?

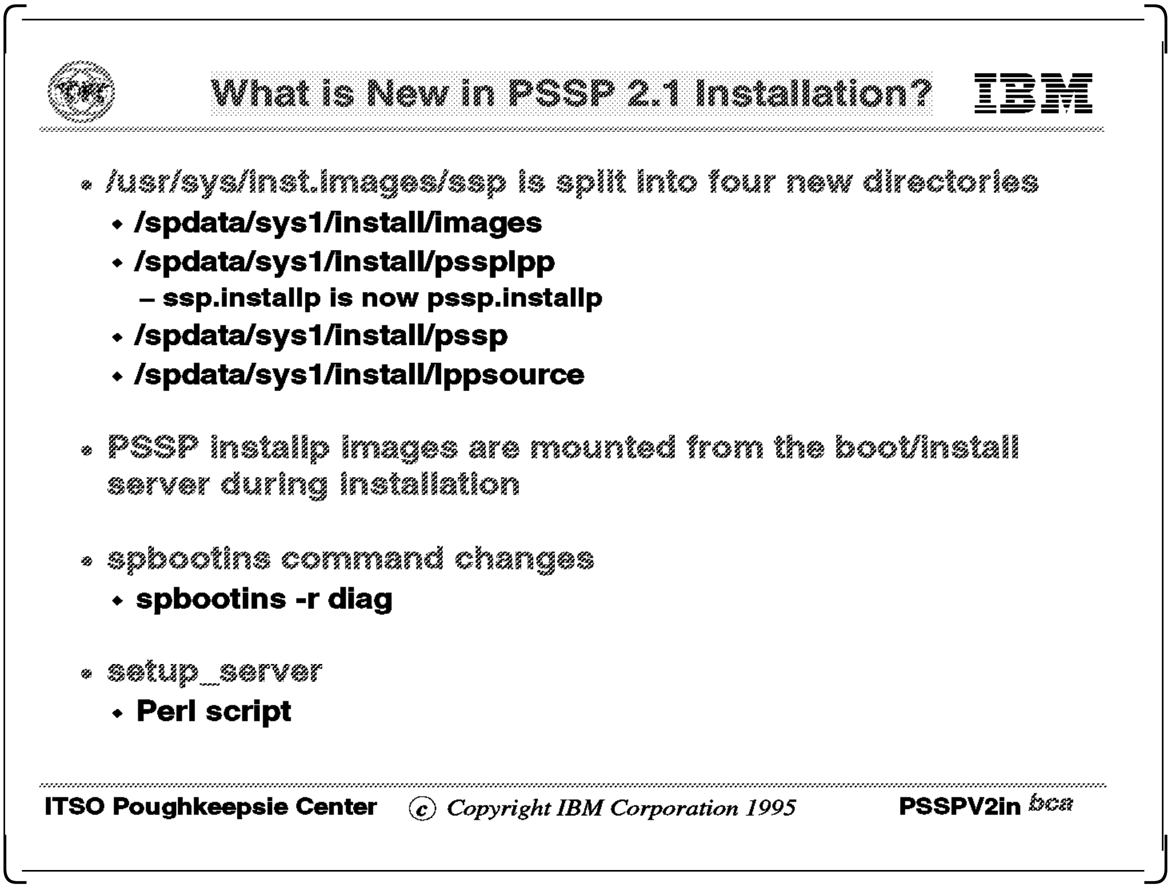
- AIX 4.1 Network Installation Management
  - Installation mechanism for nodes
  - PSSP no longer changes rc.boot
    - Does not maintain its own network install image
- No /usr clients support with AIX 4.1
  - Support is available for AIX 3.2.5 partition
- Diagnostics and Maintenance support over the network using NIM
- firstboot.cust is now script.cust
- A tuning customization file, tuning.cust, is added

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2in bc

PSSP Version 2 Release 1 includes new developments that provide more flexibilite and more robudtness to the RS/6000 SP management and operations:

- PSSP Version 2 is based on AIX 4.1.3, which provides a new mechanism for booting remote hosts through the network. This new facility is now used by PSSP for booting the RS/6000 SP nodes. A command, *setup_server*, gets the RS/6000 SP configuration from the SDR and puts this information into the NIM database in order to prepare the remote booting of nodes.
- To customize the nodes, the RS/6000 SP administrator will customize two scripts:
  - *script.cust*
  - *tuning.cust*

  These scripts will be executed on nodes during the network boot process and allow the administrator to complete the node customization and to execute the `no -o` commands to customize TCP/IP on the nodes.
- The /usr client support is maintained for AIX 3.2.5 partitions.



What is New in PSSP 2.1 Installation?

- /usr/sys/inst.images/ssp is split into four new directories
  - /spdata/sys1/install/images
  - /spdata/sys1/install/pssplpp
    - ssp.installp is now pssp.installp
  - /spdata/sys1/install/pssp
  - /spdata/sys1/install/lppsource
- PSSP installp images are mounted from the boot/install server during installation
- spbootins command changes
  - spbootins -r diag
- setup_server
  - Perl script

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2in bca

In PSSP V2, the former directories and subdirectories have been restructured. Now, any PSSP information needed to install and maintain the system on RS/6000 SP nodes are gathered in the same /spdata directory. So, the former /usr/sys/inst.images/ssp directory is splitted in:

/spdata/sys1/install/images
The AIX system images to be network booted on nodes are stored on this directory. For instance, when you install spimg, the default minimum AIX image is stored in this directory. If you want to create your own system images using mksysb (do not forget that you cannot do it on the control workstation after Kerberos is initialized), you will copy these images in the images directory.

/spdata/sys1/install/pssplpp
In this directory, you copy the PSSP V2 and related software installp file sets. You must rename the ssp.2.1.0.0 or current level as *ssp.installp*: the PSSP installation procedures will look for this file in place of the PSSP V2 installp file set as it is delivered.

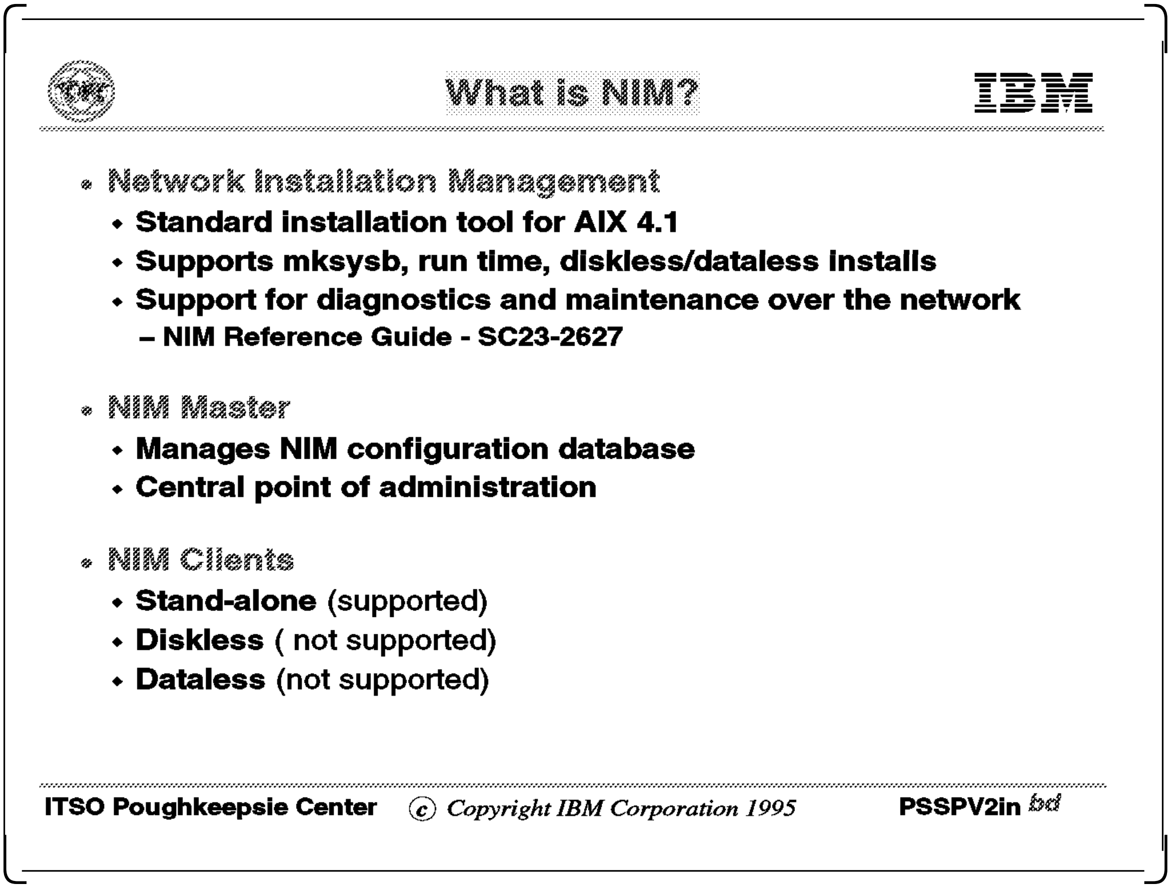
/spdata/sys1/install/pssp
This directory includes the NIM configuration data files.

/spdata/sys1/install/lppsource
In this directory, you will put the software installp file sets of required AIX 4.1.3 file sets.

The spbootins command includes a new value for the -r flag, *diag*. this parameter allows you to remotely run diagnostics on nodes. When a node is described this way, the next time it is booted, the diagnostics menu will be displayed on the tty window and the administrator will be able to run diagnostics on this node.

The setup_server command gets the node information from the SDR, puts them in the NIM database, and creates all files needed during the node network boot.

2.2 What Is NIM?



What is NIM?

- Network Installation Management
  - **Standard installation tool for AIX 4.1**
  - **Supports mksysb, run time, diskless/dataless installs**
  - **Support for diagnostics and maintenance over the network**
    - **NIM Reference Guide - SC23-2627**
- NIM Master
  - **Manages NIM configuration database**
  - **Central point of administration**
- NIM Clients
  - **Stand-alone** (supported)
  - **Diskless** (not supported)
  - **Dataless** (not supported)

ITSO Poughkeepsie Center ⓒ *Copyright IBM Corporation 1995* **PSSPV2in** *bd*

The *Network Install Management (NIM)* is a new standard installation tool in AIX 4.1. NIM provides the ability to install machines with software from a centrally managed repository in the network.

In the installation phase of SP nodes, the PSSP software uses the NIM process to restore an AIX system (*mksysb*) image to the hard disk of the node.

With NIM you have the possibility to diagnose and maintain the SP over the network. For more information on NIM please refer also to the *NIM Reference Guide*.

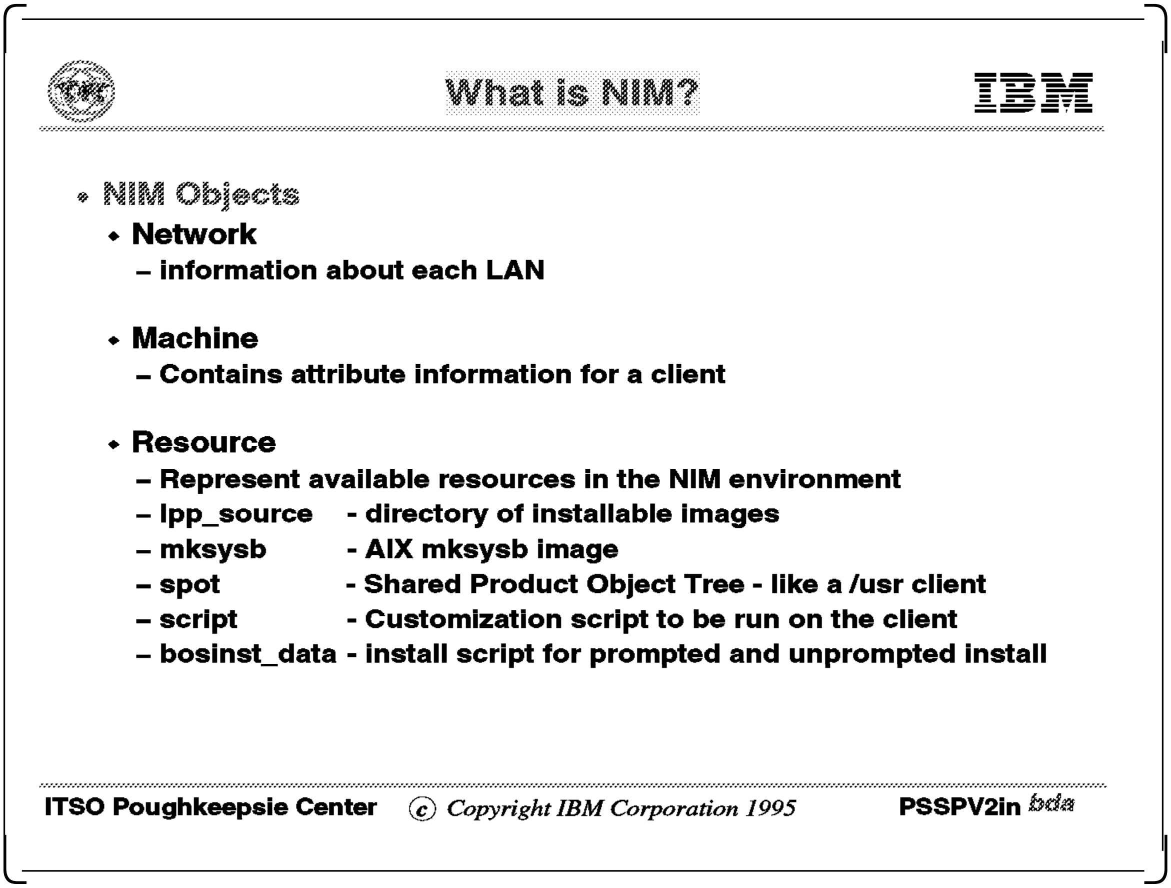
The *NIM Master* manages the NIM environment and may remotely execute commands on clients. A NIM Master is defined as an AIX 4.1 system with the NIM master file set (bos.sysmgt.nim.master) installed. The control workstation is the NIM master for a single frame system. What if you have more frames? In that case the configuration is different. Every first node in each frame is defined as a NIM Master for the rest of the nodes in that frame. And the control workstation is the NIM master for each first node in a frame. The NIM Master manages the NIM database and controls the overall environment. The NIM master has a *push* permission, therefore it can execute commands and initiate operations on other machines.

The NIM Clients are managed by a NIM master. NIM clients can request and initiate operations from the NIM master; this function is called *pull* permission. There are three different client configuration types:

- Stand_alone machines
- Diskless machines
- Dataless machines

PSSP Version 2 supports currently only stand_alone machines.

2.3 NIM Objects



What is NIM?

- NIM Objects
  - Network
    - information about each LAN
  - Machine
    - Contains attribute information for a client
  - Resource
    - Represent available resources in the NIM environment
    - lpp_source - directory of installable images
    - mksysb - AIX mksysb image
    - spot - Shared Product Object Tree - like a /usr client
    - script - Customization script to be run on the client
    - bosinst_data - install script for prompted and unprompted install

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* PSSPV2in bda

The NIM environment is defined by information that represents the physical topology of the network, including properties for each and every one of the NIM clients. This information is used for control and maintenance of the NIM environment. It is stored as *objects in the NIM database* on the NIM master.

The following are the three basic classes:

- Network
- Machine
- Resource

The *Network Class* has objects in the NIM database that represent information about each LAN that is part of the NIM environment. The information in a network object is the following:

- User defined name of the network
- Network types, like token-ring, Ethernet, or FDDI
- IP address
- Subnet mask
- Routing information

The *Machine Class* has objects in the NIM database about each client that participates in the NIM environment. The following information is stored in an object:

- Host name of a client network interface
- Network hardware address
- Network name (en0)
- Speed of the network interface (for token-ring only)

The *Resource Class* has objects in the NIM database that represent available resources in a NIM environment. These resources represent files, directories, and devices that have been made available by servers in the NIM environment. The following is a list of the most important resources:

Objects **Description**

lpp_source Represents the /spdata/sys1/install/lppsource directory containing optional software packages. The `installp` command is used in that directory to install AIX LPP images.

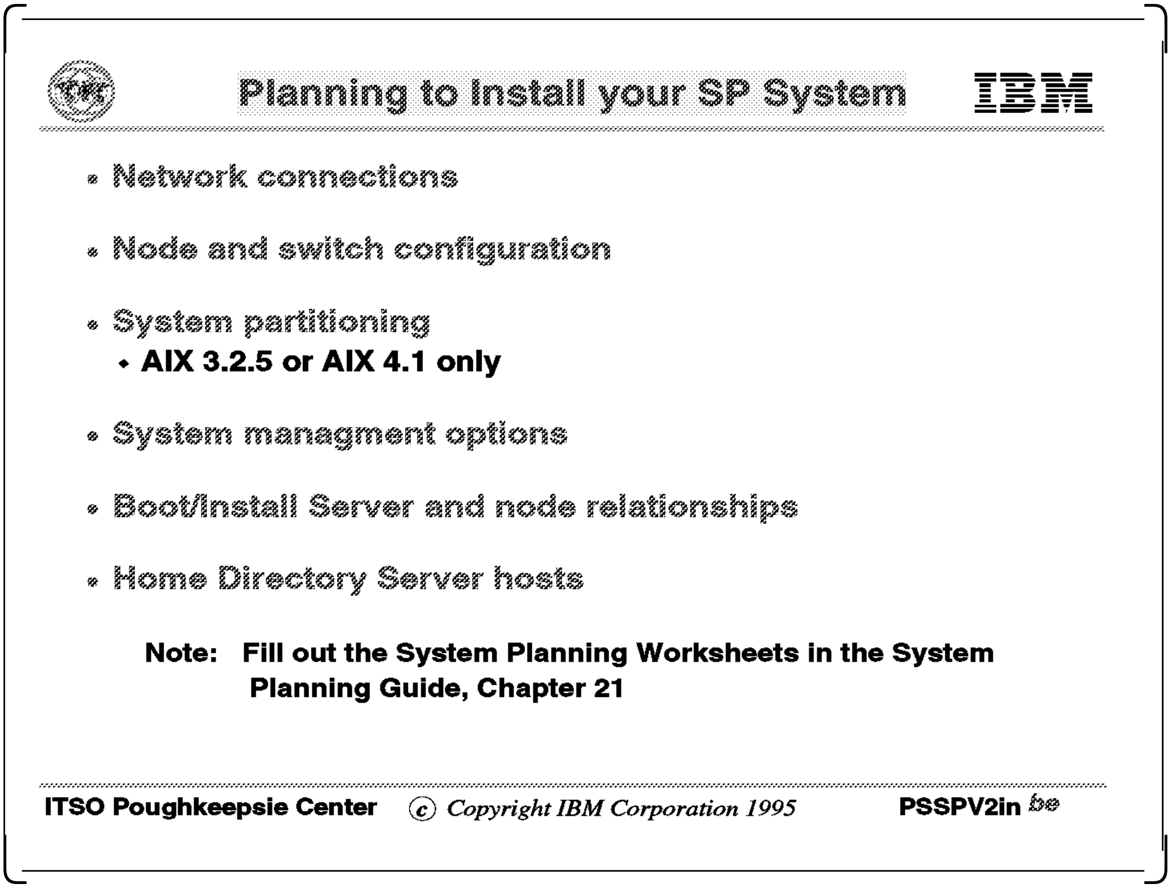
mksysb Represents a BOS image file created from a running machine. The mksysb object is used to restore sytem backups that were created with the `mksysb` command.

spot Represents a Shared Product Object Tree (SPOT), which is a directory containing the equivalent to a /usr file system.

script Represents PSSP (or user) defined shell scripts that can be used to perform additional configurations on a client.

bosinst_data The bosinst_data install script is used to set options for the BOS install program. Thereby it provides a way for an unattended installation process.

2.4 Planning to Install your SP System



Planning to Install your SP System

- Network connections
- Node and switch configuration
- System partitioning
  - AIX 3.2.5 or AIX 4.1 only
- System managment options
- Boot/Install Server and node relationships
- Home Directory Server hosts

Note: Fill out the System Planning Worksheets in the System Planning Guide, Chapter 21

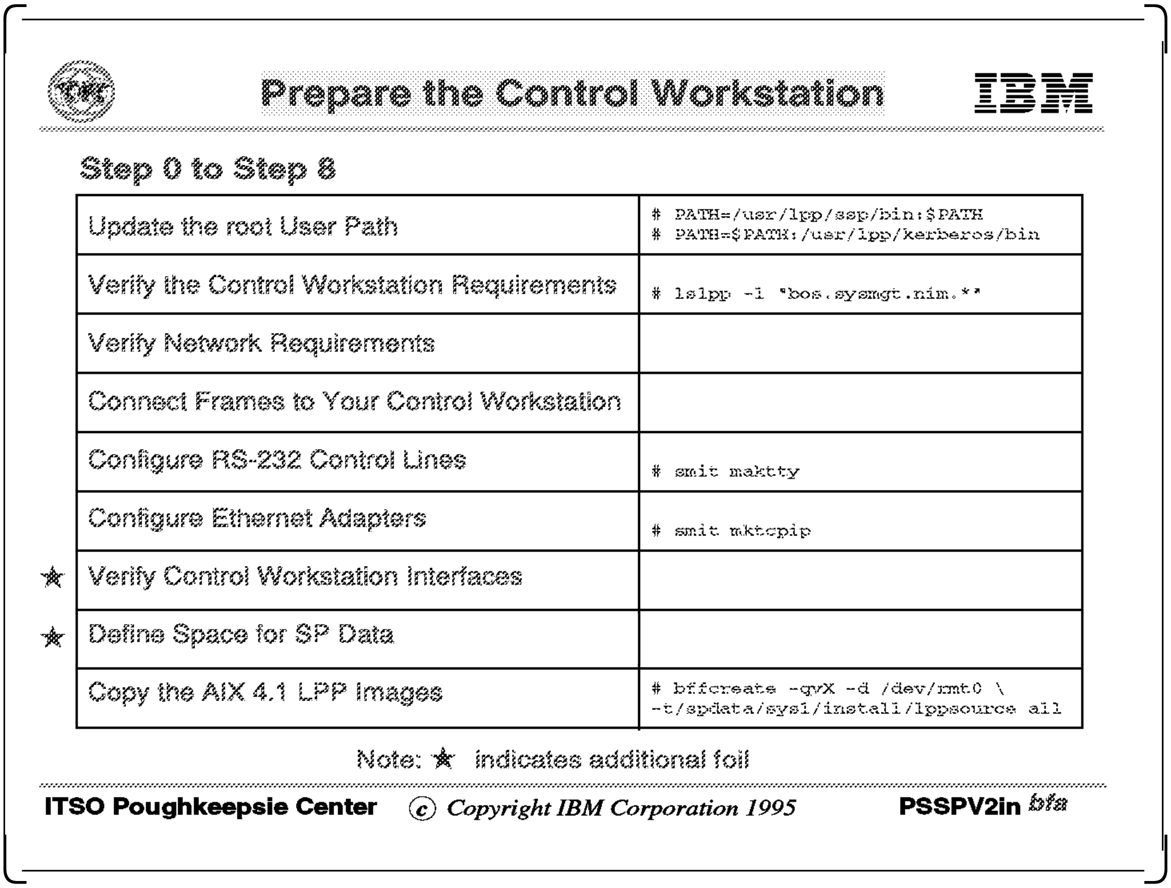
ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* PSSPV2in be

SP System Planning provides information about the preparation of PSSP installation. It includes the description of tasks that must be executed before installing PSSP V2 on the control workstation. The network configuration, together with the node and with configurations, is one of the most important tasks to be executed. You will find in *system Planning* the worksheets that you fill out to prepare the PSSP V2 installation.

Chapter 3. Installing and Configuring the SP System

The installation and configuration part uses the same step numbering as *SP Installation Guide*.

3.1 Prepare the Control Workstation



Prepare the Control Workstation

Step 0 to Step 8

	Update the root User Path	# PATH=/usr/lpp/ssp/bin:$PATH # PATH=$PATH:/usr/lpp/kerberos/bin
	Verify the Control Workstation Requirements	# lslpp -l "bos.sysmgt.nim.*"
	Verify Network Requirements	
	Connect Frames to Your Control Workstation	
	Configure RS-232 Control Lines	# smit maktty
	Configure Ethernet Adapters	# smit mktcpip
★	Verify Control Workstation Interfaces	
★	Define Space for SP Data	
	Copy the AIX 4.1 LPP Images	# bffcreate -qvX -d /dev/rmt0 \\ -t/spdata/sys1/install/lppsource all

Note: ★ indicates additional foil

ITSO Poughkeepsie Center ⓒ Copyright IBM Corporation 1995 PSSPV2in bfa

Here are the important steps to prepare a control workstation.

For the first step you have to update the root `.profile` and add the path for the PSSP software. Here is an example for a root .profile:

```
PATH=/usr/lpp/ssp/bin:/usr/lpp/ssp/kerberos/bin
PATH=$PATH:/usr/bin:/etc:/usr/sbin:/usr/ucb
PATH=$PATH:/usr/dt/bin:/usr/lpp/X11/bin:/sbin:
PATH=$PATH:$HOME/bin:/sbin:/bin
MANPATH=/usr/lpp/ssp/man:/u/loadl/man:/usr/man
ENV=/.kshrc
export PATH MANPATH ENV

if [ ! ″$DT″ ]
then
        if [ -s ″$MAIL″ ]
        then echo ″$MAILMSG″
        fi
fi
```

Since most customers will use the Common Desktop Environment (CDE) you should also update the root′s .dtprofile. The sample profile above conforms to CDE, and therefore you have to change only the last line in the dtprofile for the root user. Here you see the last five lines of /dtprofile:

```
#
#  if $HOME/.profile (.login) has been edited as described above, uncomment
#  the following line.

DTSOURCEPROFILE=true
```

For the second step we check if the correct level of AIX is installed and check if we have enough disk space. You should have about 4GB of free disk space. To check the correct level of AIX we use the command *lslpp*:

```
# lslpp -l bos.rte
  Fileset                 Level  State
  ----------------------------------------------
Path: /usr/lib/objrepos
  bos.rte               4.1.3.0  COMMITTED
Path: /etc/objrepos
  bos.rte               4.1.3.0  COMMITTED

# lslpp -l ″bos.sysmgt.nim.*″
Path: /usr/lib/objrepos
  bos.sysmgt.nim.client   4.1.3.0  COMMITTED
  bos.sysmgt.nim.master   4.1.3.0  COMMITTED
  bos.sysmgt.nim.spot     4.1.3.0  COMMITTED
Path: /etc/objrepos
  bos.sysmgt.nim.client   4.1.3.0  COMMITTED

# lslpp -l ″bos.net.*″
  Fileset                 Level  State
  ----------------------------------------------
Path: /usr/lib/objrepos
  bos.net.nfs.client      4.1.3.0  COMMITTED
  bos.net.nfs.server      4.1.0.0  COMMITTED
  bos.net.tcp.client      4.1.3.0  COMMITTED
  bos.net.tcp.server      4.1.3.0  COMMITTED
  bos.net.tcp.smit        4.1.3.0  COMMITTED
Path: /etc/objrepos
  bos.net.nfs.client      4.1.3.0  COMMITTED
  bos.net.tcp.client      4.1.3.0  COMMITTED
  bos.net.tcp.server      4.1.3.0  COMMITTED
```

The first command checks the version of the Base Operating System Runtime, which is 4.1.3. The second command checks if all needed components of the Network Install Manager (NIM) are installed. The NIM components that are needed are the following:

- NIM Client Tools
- NIM Master Tools
- NIM Shared Product Object Tree (SPOT)

The third command checks if the minimum for networking is installed. The minimum that you need on the control work station (CWS) is the following:

- Network File System (NFS) Client
- Network File System (NFS) Server
- TCP/IP Client Support
- TCP/IP Server
- TCP/IP SMIT Support

The third step is to connect all RS-232 and all Ethernet cables from the RS/6000 SP system frames to the control workstation.

The next step is to configure the serial lines. You can use smit maktty, or use the mkdev command. Here we have an example for mkdev:

```
# mkdev -c tty -t tty -s rs232 -p sa0 -w s1 -a speed=19200
tty0 Available
```

To configure the Ethernet adapter, you can use smit mktcpip or use the mktcpip command. For the example, we use the mktcpip command:

```
# /usr/sbin/mktcpip -h spcw0 -a 129.33.34.15 \
  -m 255.255.255.0 -i en0 -g 129.33.34.2 -t bnc
en0
spcw0
inet0 changed
en0 changed
inet0 changed
```

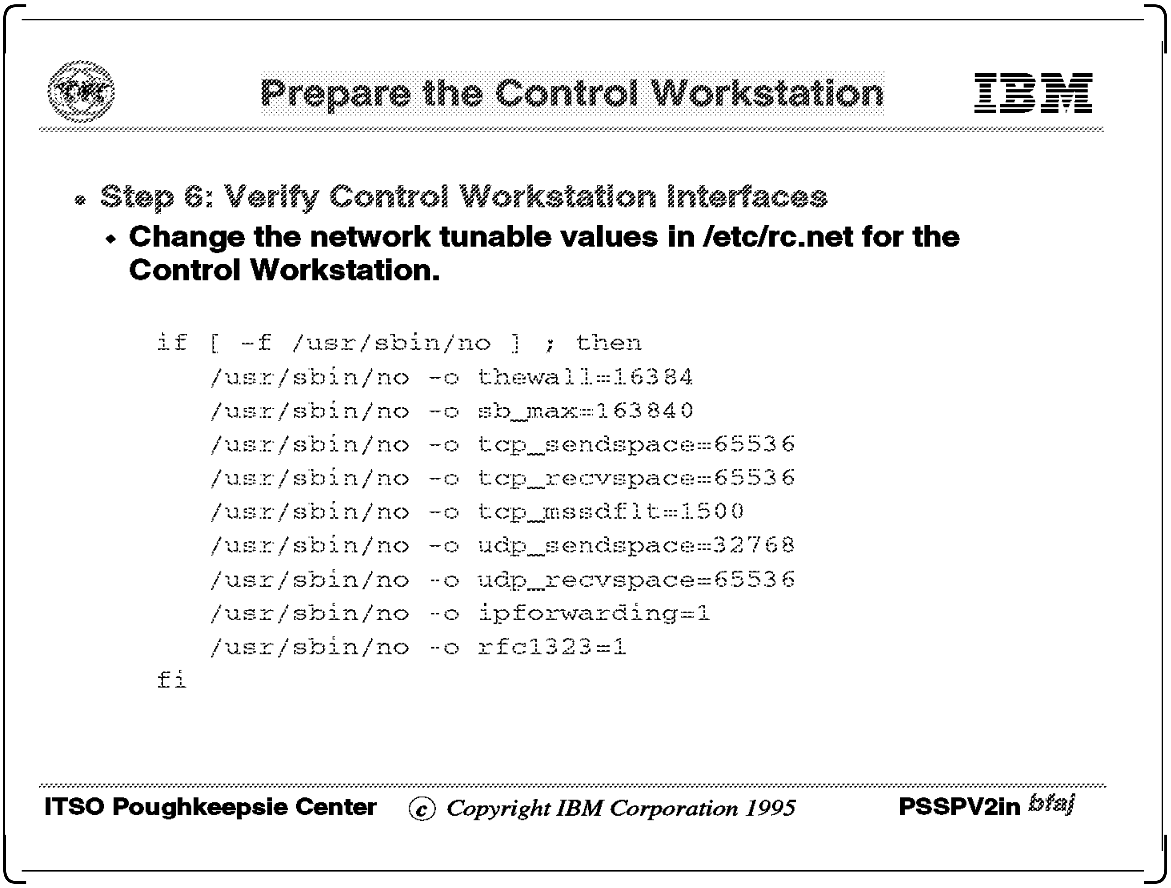
Step 6 will be discussed in section 3.1.1, “Verify Control Workstation Interfaces” on page 38.

For step 7, we have three foils that explain how to create a separate volume group for SP Data.

The last step in preparing the control workstation is to copy all AIX file sets to the /spdata/sys1/install/lppsource directory. Insert your AIX tape into the tape drive or your AIX CD ROM into your CD ROM player. You can use smit bffcreate or use the bffcreate command to copy the AIX LPP images to the hard disk. Depending on your AIX software, this step will take a couple of hours. The example shows how to copy all AIX LPPs from the tape drive rmt0 to the lppsource directory:

```
# cd /spdata/sys1/install/lppsource
# bffcreate -qvX -t. -d/dev/rmt0 all
# cd ..
# chmod -R a+r lppsource
```

3.1.1 Verify Control Workstation Interfaces



Prepare the Control Workstation

- **Step 6: Verify Control Workstation Interfaces**
  - **Change the network tunable values in /etc/rc.net for the Control Workstation.**

```
if [ -f /usr/sbin/no ] ; then
    /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=1500
    /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
fi
```

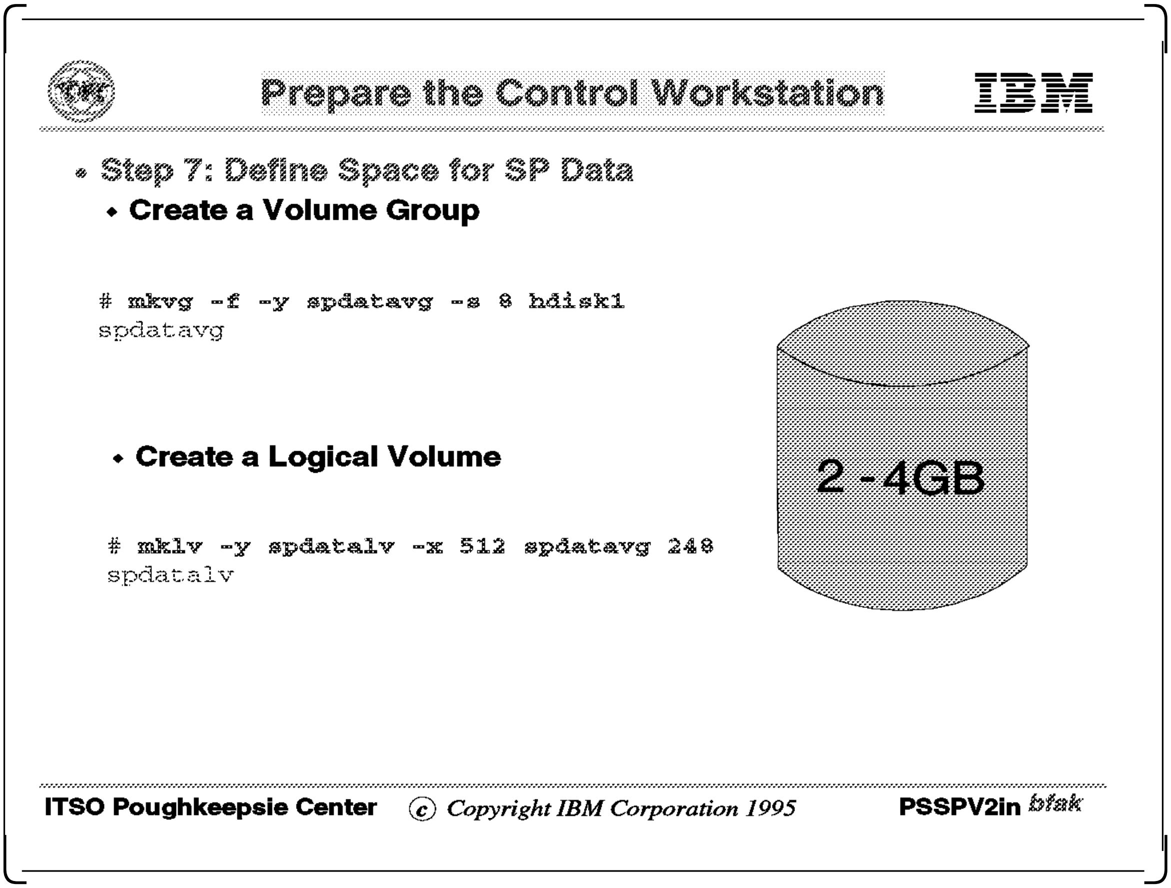
ITSO Poughkeepsie Center *© Copyright IBM Corporation 1995* **PSSPV2in** *bfaj*

Step 6 verifies the control workstation interfaces. Since we have just defined the Ethernet adapter, we will use ping to verify if the network is responding. For example:

```
# ping -c1 sp2cw0
PING spcw0.itsc.pok.ibm.com (129.33.34.15): 56 data bytes
64 bytes from 129.33.34.15: icmp_seq=0 ttl=255 time=0 ms

--- spcw0.itsc.pok.ibm.com ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0/0/0 ms
```

On the foil you see the changes of the network tunables at the bottom of the file /etc/rc.net. These values are recommended in the *SP Installation Guide*. The changes in /etc/rc.net are effective after the next system reboot.

3.1.2 Define Space for SP Data (Volume Group)



Prepare the Control Workstation

- Step 7: Define Space for SP Data
  - Create a Volume Group

```
# mkvg -f -y spdatavg -s 8 hdisk1
spdatavg
```

  - Create a Logical Volume

```
# mklv -y spdatalv -x 512 spdatavg 248
spdatalv
```

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2in bfak

Step 7 prepares a separate hard disk for the SP data. A separate volume group is recommended to keep your SP data from being dependent on your root volume group.

First we will create a volume group. You can use smit mkvg or use the mkvg command. In the example you see that we use a physical partition size of 8MB. The 8MB physical partition size is needed for hard disks that are bigger than 4GB.

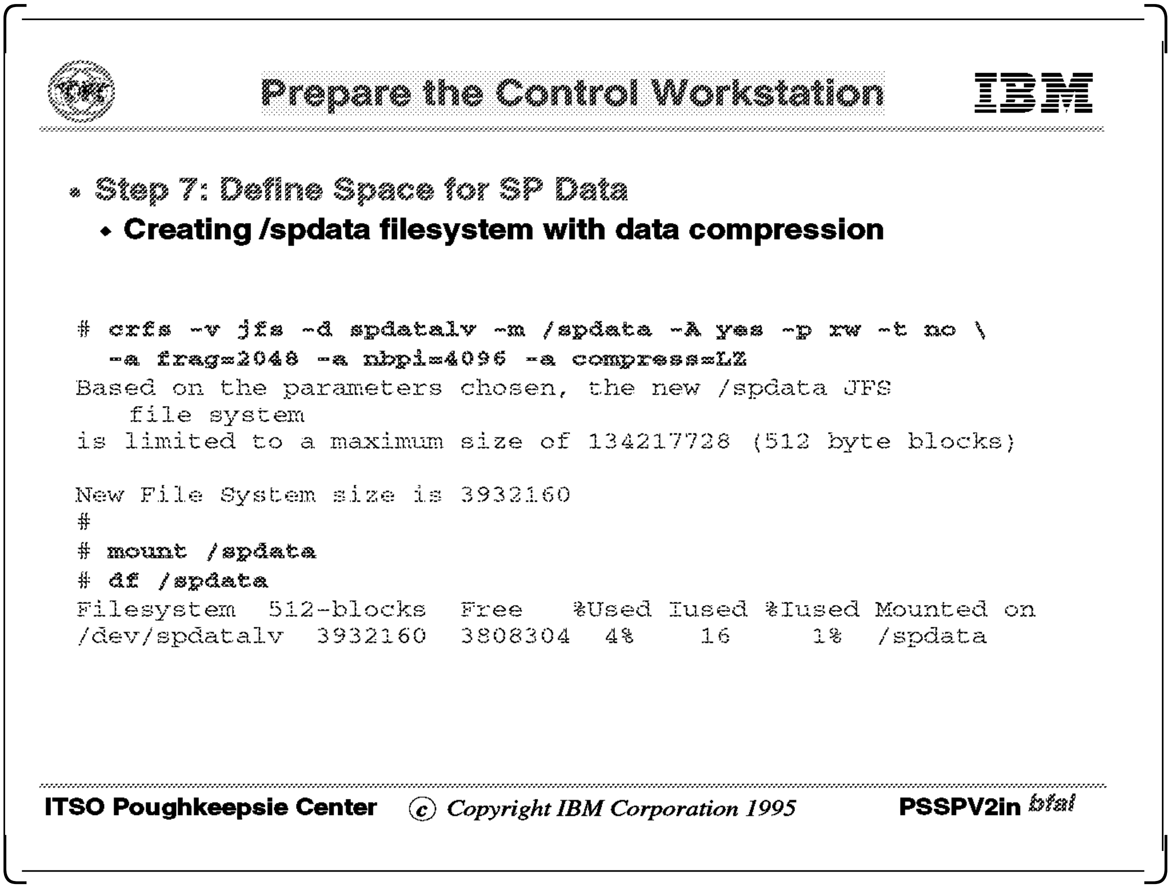
```
# mkvg -f -y spdatavg -s 8 hdisk1
spdatavg
```

Second we will create a logical volume. You can use smit mklv or use the mklv command. In the example we create a logical volume with the size of 248 physical partitions. On this 4GB hard drive, 248 physical partitions with 8MB each will add up to 1984MB of space for this logical volume. We left some physical partitions free for the journaled log file system.

```
# mklv -y spdatalv -x 512 spdatavg 248
spdatalv
```

The minimum disk space requirement for the lppsource directory is 450MB. It includes the disk space needed to store the minimal list of AIX 4.1.3 file sets (see *SP installation Guide* for more details).

3.1.3 Define Space for SP Data (File System)



Prepare the Control Workstation

- Step 7: Define Space for SP Data
  - **Creating /spdata filesystem with data compression**

```
# crfs -v jfs -d spdatalv -m /spdata -A yes -p rw -t no \
  -a frag=2048 -a nbpi=4096 -a compress=LZ
Based on the parameters chosen, the new /spdata JFS
   file system
is limited to a maximum size of 134217728 (512 byte blocks)

New File System size is 3932160
#
# mount /spdata
# df /spdata
Filesystem  512-blocks  Free    %Used Iused %Iused Mounted on
/dev/spdatalv  3932160  3808304  4%     16     1%  /spdata
```

ITSO Poughkeepsie Center   *Ⓒ Copyright IBM Corporation 1995*   PSSPV2in *bfal*

Third we create the /spdata file system with data compression. You might ask why use data compression?

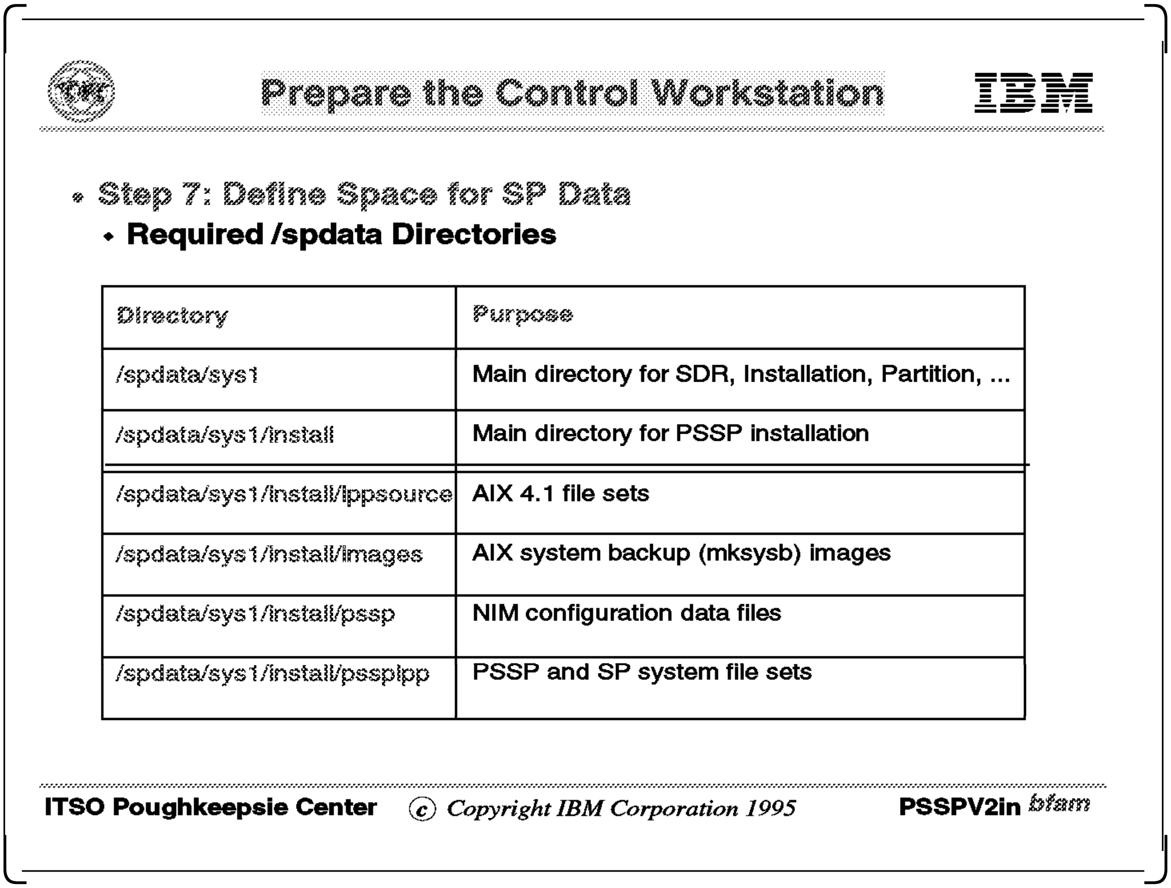
- It is free with AIX 4.1.
- You have over 2000 files in /spdata, most of which are ASCII files.

To create a file system, you can use smit crjfslv or the crfs command:

```
# crfs -v jfs -d spdatalv -m /spdata -A yes -p rw -t no \
  -a frag=2048 -a nbpi=4096 -a compress=LZ
Based on the parameters chosen, the new /spdata JFS file system
is limited to a maximum size of 134217728 (512 byte blocks)

New File System size is 3932160

# mount /spdata
# df /spdata
Filesystem    512-blocks      Free %Used    Iused %Iused Mounted on
/dev/spdatalv    3932160   3808304    4%       16     1% /spdata
```

With the mount command, we will mount the created file system, and we will verify the size of the file system with the df command.

3.1.4 Define Space for SP Data (directories)



Prepare the Control Workstation

- Step 7: Define Space for SP Data
  - Required /spdata Directories

Directory	Purpose
/spdata/sys1	Main directory for SDR, Installation, Partition, ...
/spdata/sys1/install	Main directory for PSSP installation
/spdata/sys1/install/lppsource	AIX 4.1 file sets
/spdata/sys1/install/images	AIX system backup (mksysb) images
/spdata/sys1/install/pssp	NIM configuration data files
/spdata/sys1/install/pssplpp	PSSP and SP system file sets

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* PSSPV2in *bfam*

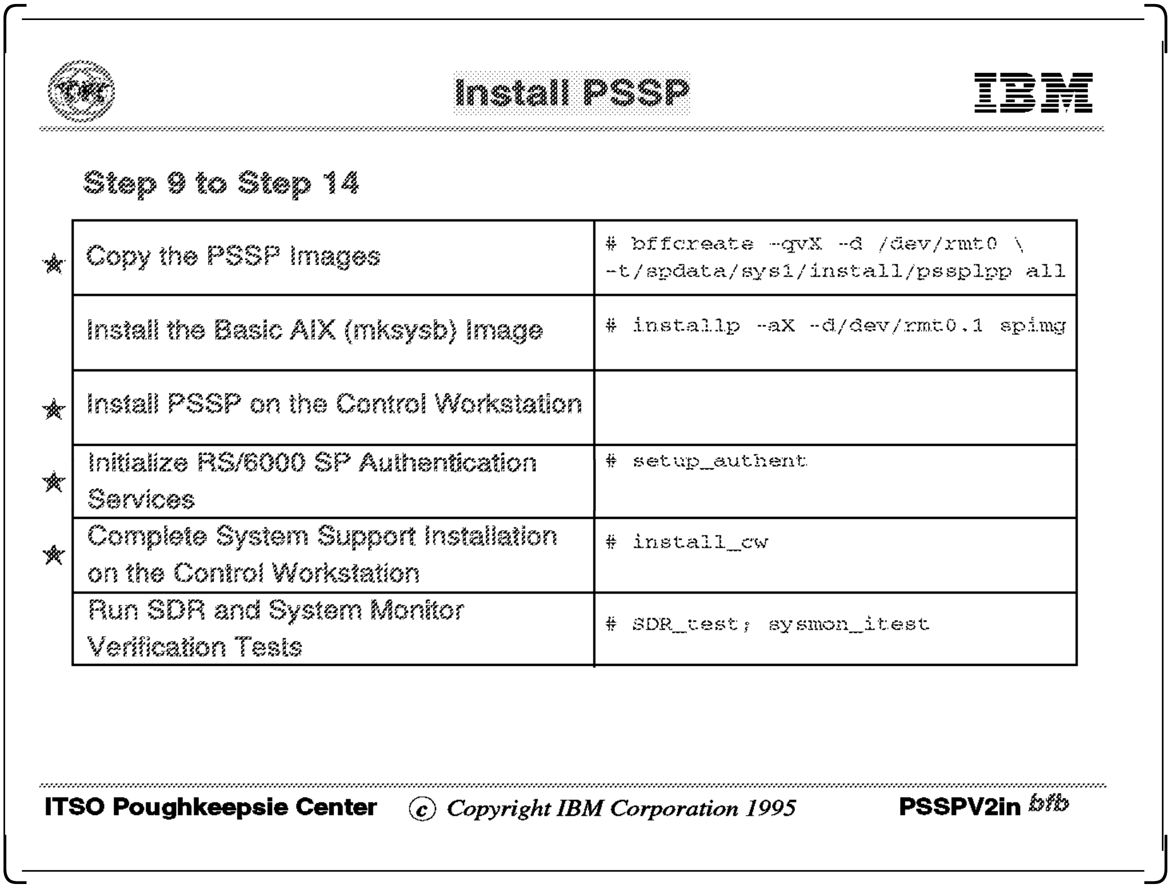
After you have mounted the new spdata file system, you have to create four different subdirectories for the installation. These directories are:

Directories	Description
/spdata/sys1	Main directory for SDR, SP monitor, log management, partition layout files, and installation
/spdata/sys1/install	Main directory for PSSP installation
/spdata/sys1/install/lppsource	Location of required AIX 4.1 file sets
/spdata/sys1/install/images	Location of AIX system backup (mksysb) images
/spdata/sys1/install/pssp	Location of NIM configuration data files
/spdata/sys1/install/pssplpp	Location of all PSSP and SP system file sets

To create these directories, you can use the following mkdir commands:

```
# mkdir -p /spdata/sys1/install/lppsource
# mkdir /spdata/sys1/install/images
# mkdir /spdata/sys1/install/pssp
# mkdir /spdata/sys1/install/pssplpp
# chmod -R a+rx /spdata
# chmod -R g+s /spdata
```

3.2 Install PSSP



Install PSSP

Step 9 to Step 14

★	Copy the PSSP Images	# bffcreate -qvX -d /dev/rmt0 \ -t/spdata/sys1/install/pssplpp all
	Install the Basic AIX (mksysb) Image	# installp -aX -d/dev/rmt0.1 spimg
★	Install PSSP on the Control Workstation	
★	Initialize RS/6000 SP Authentication Services	# setup_authent
★	Complete System Support Installation on the Control Workstation	# install_cw
	Run SDR and System Monitor Verification Tests	# SDR_test; sysmon_itest

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* **PSSPV2in** *bfb*

Step 9 will be discussed in 3.2.1, “Copy the PSSP Images” on page 43.

For step 10, we will install a basic AIX system image in the directory /spdata/sys1/install/images. The spimg installp image is provided with the SP system on a separate tape. You can also install your own AIX system backup (mksysb) image in the /spdata/sys1/install/images directory.

For the initial installation of your SP system we recommend using the spimg installp image. Follow the steps described below and you will save 150MB in /usr/lpp/spimg:

```
# ln -s /spdata/sys1/install/images /usr/lpp/spimg
# installp -aX -d/dev/rmt0.1 spimg
```

If you do not want to use the installp command directly, you can use the smit install_latest command.

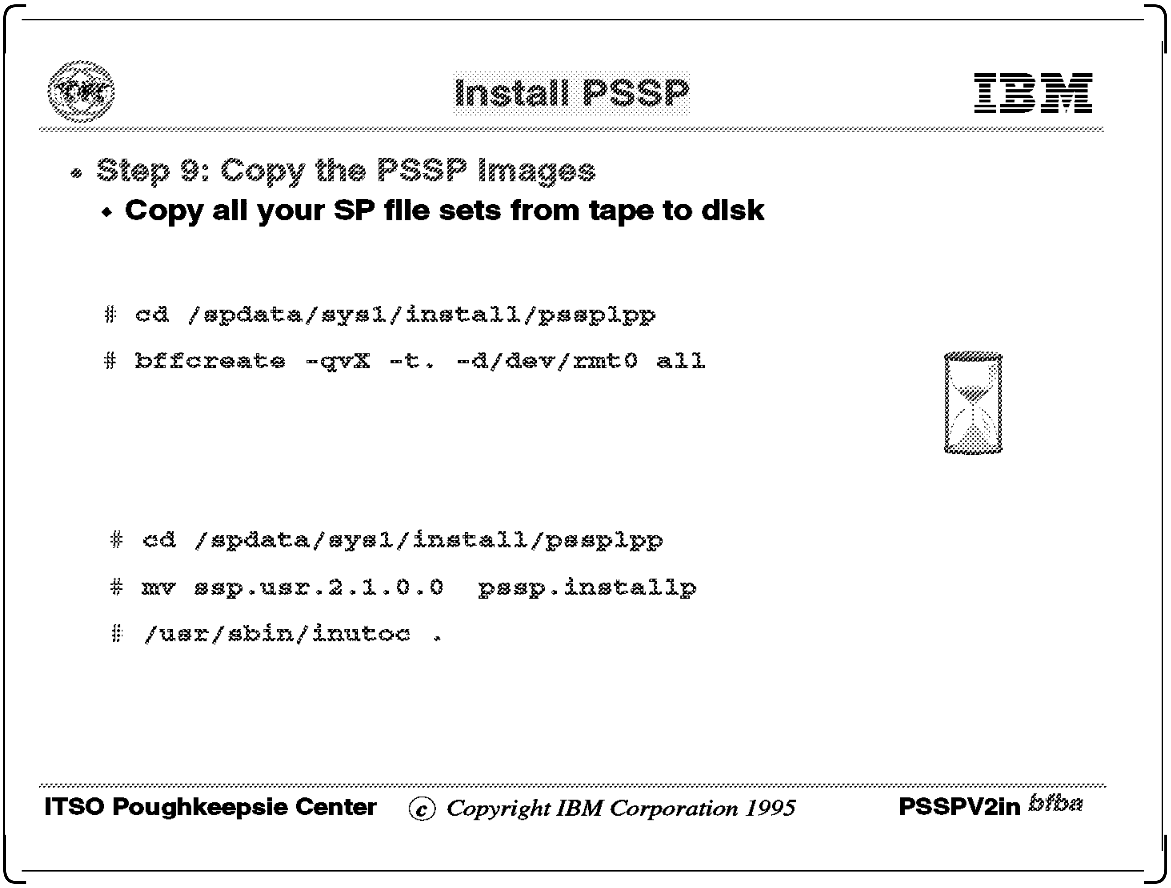
We have separate foils for the steps 11 to 13.

For step 14 we ran SDR_test and spmon_itest. The SDR_test verifies if the installation of System Data Repository (SDR) has completed successfully. The spmon_itest shell script is an installation verification program (IVP). If you want to know how this program works, then execute it in the following way:

```
ksh -x /usr/lpp/ssp/bin/spmon_itest
```

Ensure that the permission for /etc/rc.net is rwxr-xr--.

3.2.1 Copy the PSSP Images



Install PSSP

IBM

- Step 9: Copy the PSSP Images
  - **Copy all your SP file sets from tape to disk**

```
# cd /spdata/sys1/install/pssplpp
# bffcreate -qvX -t. -d/dev/rmt0 all
```

```
# cd /spdata/sys1/install/pssplpp
# mv ssp.usr.2.1.0.0  pssp.installp
# /usr/sbin/inutoc .
```

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* PSSPV2in *bfba*

For step 9, we will copy all PSSP file sets that you have to the /spdata/sys1/install/pssplpp directory. Insert your PSSP LPP tape into the tape drive. You can use smit bffcreate or use the bffcreate command to copy the PSSP LPP images to the hard disk:

- The pssp install image, which contains the PSSP options
- The ssp.csd install image (IBM Virtual Shared Disk)
- The ssp.mte install image (magnetic tape extension)

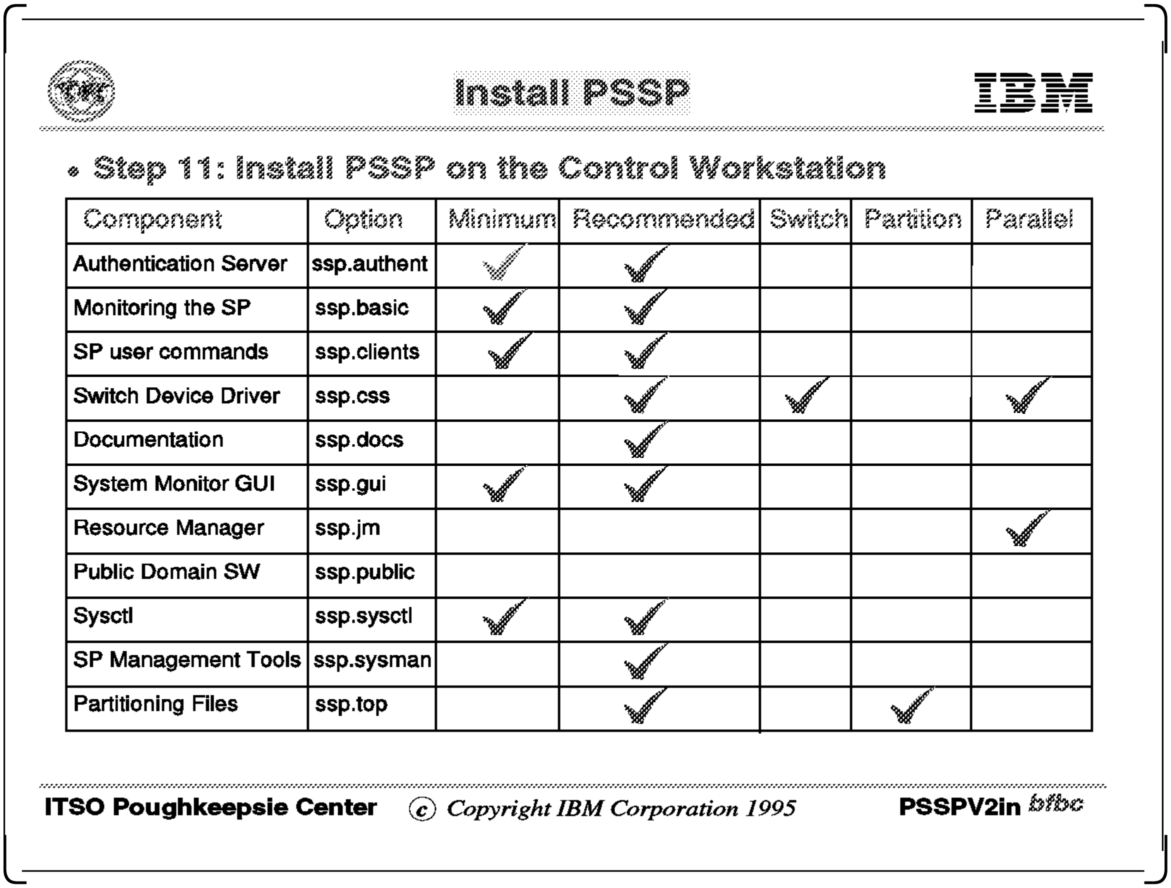
You will always need the *pssp* install image, which is named ssp.2.1.0.0. You will need the IBM Virtual Shared Disk when you use a parallel version of Oracle. The magnetic tape extension is needed only for the 3480 and 3490 tape systems.

The following example shows how to copy all PSSP LPPs from the tape drive rmt0 to the pssplpp directory:

```
# cd /spdata/sys1/install/pssplpp
# bffcreate -qvX -t. -d/dev/rmt0 all
# mv ssp.usr.2.1.0.0  pssp.installp
# /usr/sbin/inutoc .
# chmod -R a+r ..
```

When the transfer of the file sets is finished, you have to rename the ssp.usr.2.1.0.0 package to pssp.installp. This name change is very important otherwise the installation script would fail. After the name change, you have to update the table of contents file (.toc) with the inutoc command.

3.2.2 Install PSSP on the Control Workstation



Install PSSP

- Step 11: Install PSSP on the Control Workstation

Component	Option	Minimum	Recommended	Switch	Partition	Parallel
Authentication Server	ssp.authent	✓	✓			
Monitoring the SP	ssp.basic	✓	✓			
SP user commands	ssp.clients	✓	✓			
Switch Device Driver	ssp.css		✓	✓		✓
Documentation	ssp.docs		✓			
System Monitor GUI	ssp.gui	✓	✓			
Resource Manager	ssp.jm					✓
Public Domain SW	ssp.public					
Sysctl	ssp.sysctl	✓	✓			
SP Management Tools	ssp.sysman		✓			
Partitioning Files	ssp.top		✓		✓	

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* PSSPV2in *bfbc*

Here you see all components of the PSSP install image, which contains the following options:

ssp.authent Authentication server for SP authentication

ssp.basic Code for installing and monitoring the SP system, including:

- SP System Monitor
- System Data Repository (SDR)
- SMIT panels
- Installation and configuration commands

ssp.clients All user authentication commands, SP monitor command line interfaces, and logging daemon.

ssp.css Device drivers and High Performance Switch support

ssp.docs Man pages and online information

ssp.gui SP system monitor graphical user interface

ssp.jm Resource manager for parallel application scheduling

ssp.public Public domain source code for Amd, PERL, SUP, NTP, Tcl, TclX, TK-X11, and Expect

ssp.sysctl The Sysctl component

ssp.sysman SP system management tools including:

- User management support, such as the BSD automount (Amd)
- Print Support
- File Collections (SUP)
- Login Control

- Accounting Support
- Network Time Protocol (NTP)

ssp.top The system partitioning configuration directory and files

For a minimal installation you need the following components:

- Monitoring the SP (**ssp.basic**)
- SP user commands (**ssp.clients**)
- System Monitor GUI (**ssp.gui**)
- Sysctl (**ssp.sysctl**)

If you do not want to use AFS or your own Kerberos Version 4 as an authentication server, you will also need:

- Authentication Server (**ssp.authent**)

If you have a High Performance Switch, then you have to install the Switch Device Driver (**ssp.css**).

If you plan to partition your SP system, then you need the predefined partition files. Those files are part of the **ssp.top** option.

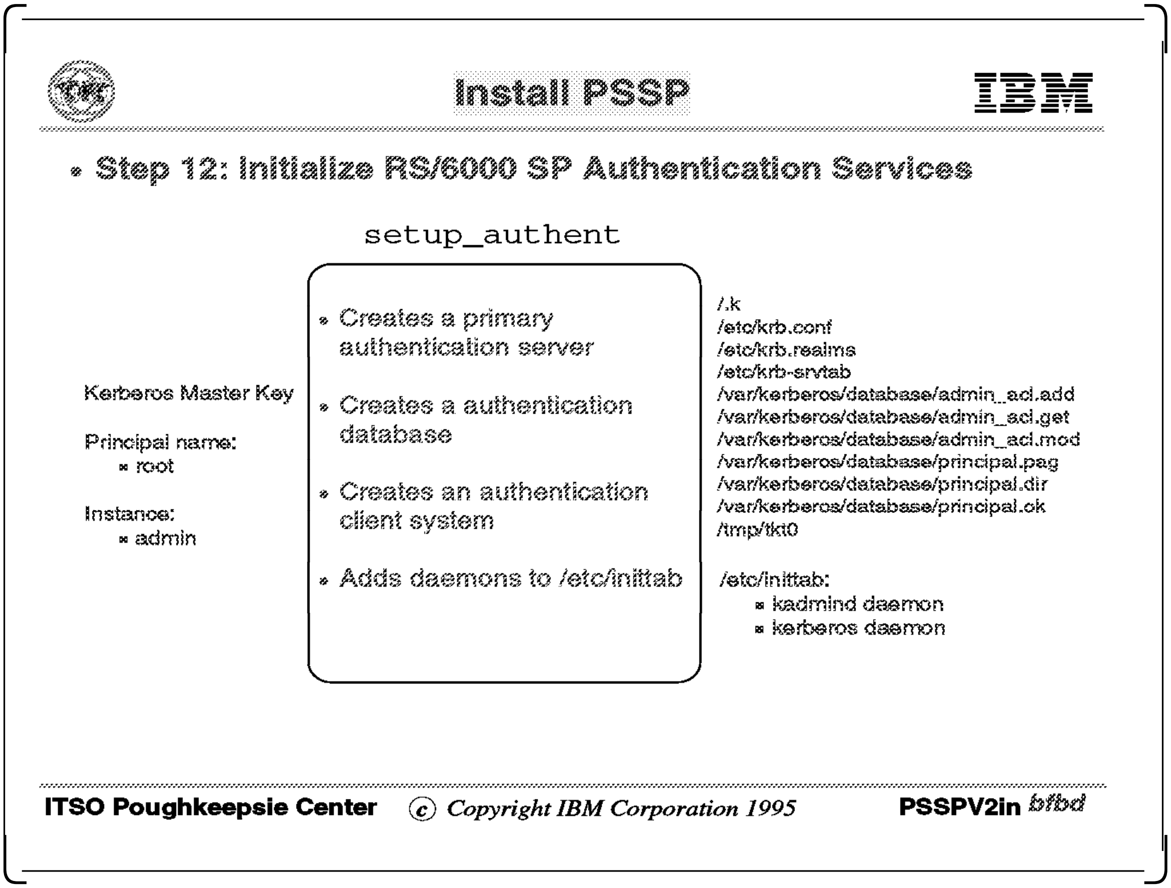
For parallel application scheduling, you need the Resource Manager (**ssp.jm**). To get the best performance for your parallel applications, a High Performance Switch is recommended.

Usually you will install all options of the PSSP install image on your control workstation. You can leave out the public domain source code. You can also skip the installation of the resource manager when there is no plan to run parallel applications.

The command for the recommended full installation will look like the following:

```
# cd /spdata/sys1/install/pssplpp
# installp -agX -d pssp.installp all
```

3.2.3 Initialize RS/6000 SP Authentication Services



Install PSSP

- Step 12: Initialize RS/6000 SP Authentication Services

setup_authent

Kerberos Master Key

Principal name:
- root

Instance:
- admin

- Creates a primary authentication server
- Creates a authentication database
- Creates an authentication client system
- Adds daemons to /etc/inittab

/.k
/etc/krb.conf
/etc/krb.realms
/etc/krb-srvtab
/var/kerberos/database/admin_acl.add
/var/kerberos/database/admin_acl.get
/var/kerberos/database/admin_acl.mod
/var/kerberos/database/principal.pag
/var/kerberos/database/principal.dir
/var/kerberos/database/principal.ok
/tmp/tkt0

/etc/inittab:
- kadmind daemon
- kerberos daemon

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2in *bfbd*

For this step you have to decide which authentication server you want to use. The following servers are possible:

- RS/6000 SP Authentication Server (ssp.authent)
- AFS
- MIT Kerberos Version 4

For this presentation we will only talk about the SP Authentication Server.

To initialize the SP authentication, you use the script **/usr/lpp/ssp/bin/setup_authent**. In the following example, we will show you the interaction with the setup_authent script:

```
# setup_authent
 ******************************************************
                  Creating the Kerberos Database
   ....
   ....
   .... see the kdb_init and kstash man pages.
 *******************************************************
You will be prompted for the database Master Password.
It is important that you NOT FORGET this password.

Enter Kerberos master key: KerberosMasterPasswd
Enter Kerberos master key: KerberosMasterPasswd


 ******************************************************
   Defining an Administrative Principal to Kerberos
   ....
   ....
   For more information see the kdb_edit man page.
 ******************************************************
Opening database...
Previous or default values are in [brackets] ,
enter return to leave the same, or new value.

Principal name: root
Instance: admin

<Not found>, Create [y] ? y

Principal: root, Instance: admin, kdc_key_ver: 1
New Password: RootAdminPasswd
Verifying, please re-enter
New Password: RootAdminPasswd

Principal′s new key version = 1 <Enter>
Expiration date (enter yyyy-mm-dd) [ 1999-12-31 ] ? <Enter>
Max ticket lifetime [ 255 ] ? <Enter>
Attributes [ 0 ] ? <Enter>
Edit O.K.
Principal name: <Enter>


 ******************************************************
           Logging into Kerberos as an admin user
   ....
   ....
       hardmon - for the System Monitor facilities
       rcmd    - for sysctl and Kerberos-authenticated rsh and rcp
   For more information, see the kinit man page.
 ******************************************************
Kerberos Initialization for ″root.admin″
Password: RootAdminPasswd

# klist
Ticket file:    /tmp/tkt0
Principal:      root.admin@ITSC.POK.IBM.COM

  Issued           Expires          Principal
Jul 14 08:57:42  Aug 13 08:57:42  krbtgt.ITSC.POK.IBM.COM@ITSC.POK.IBM.COM
```

How much is your system different after you have run setup_authent? First of all setup_authent creates a lot of files:

- /.k
- /etc/krb-srvtab
- /etc/krb.conf
- /etc/krb.realms
- /var/kerberos/database/admin_acl.add
- /var/kerberos/database/admin_acl.get
- /var/kerberos/database/admin_acl.mod

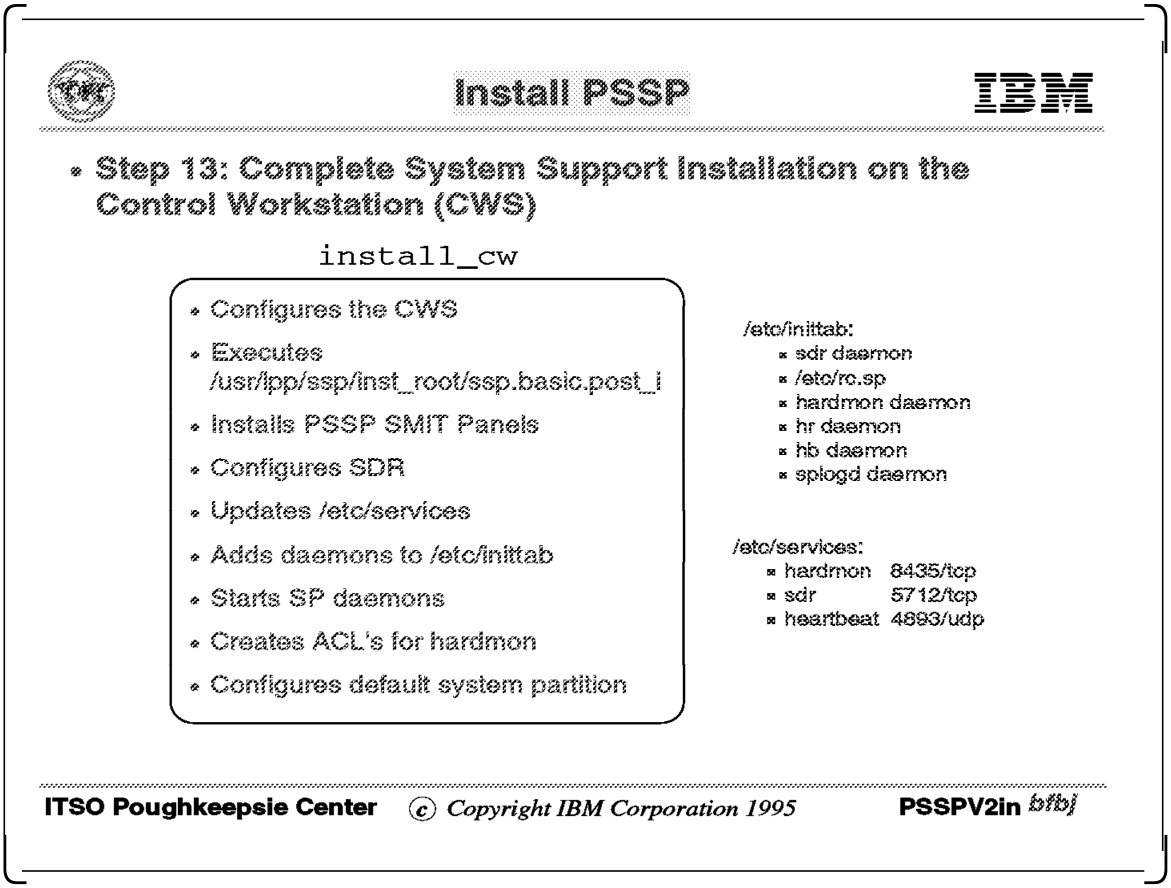
- /var/kerberos/database/principal.dir
- /var/kerberos/database/principal.ok
- /var/kerberos/database/principal.pag
- /tmp/tkt*

Second setup_authent adds following two daemons to **/etc/inittab**:

kadmind daemon
The kadmind daemon is the authentication database server for the password changing and administration tools. It uses the master key for authorization.

kerberos daemon
The kerberos daemon provides the authentication service and the ticket granting service to client programs that want to obtain tickets for authenticated services.

3.2.4 Complete System Support Installation on the Control Workstation



Install PSSP

- Step 13: Complete System Support Installation on the Control Workstation (CWS)

install_cw

- Configures the CWS
- Executes /usr/lpp/ssp/inst_root/ssp.basic.post_i
- Installs PSSP SMIT Panels
- Configures SDR
- Updates /etc/services
- Adds daemons to /etc/inittab
- Starts SP daemons
- Creates ACL's for hardmon
- Configures default system partition

/etc/inittab:
- sdr daemon
- /etc/rc.sp
- hardmon daemon
- hr daemon
- hb daemon
- splogd daemon

/etc/services:

hardmon	8435/tcp
sdr	5712/tcp
heartbeat	4893/udp

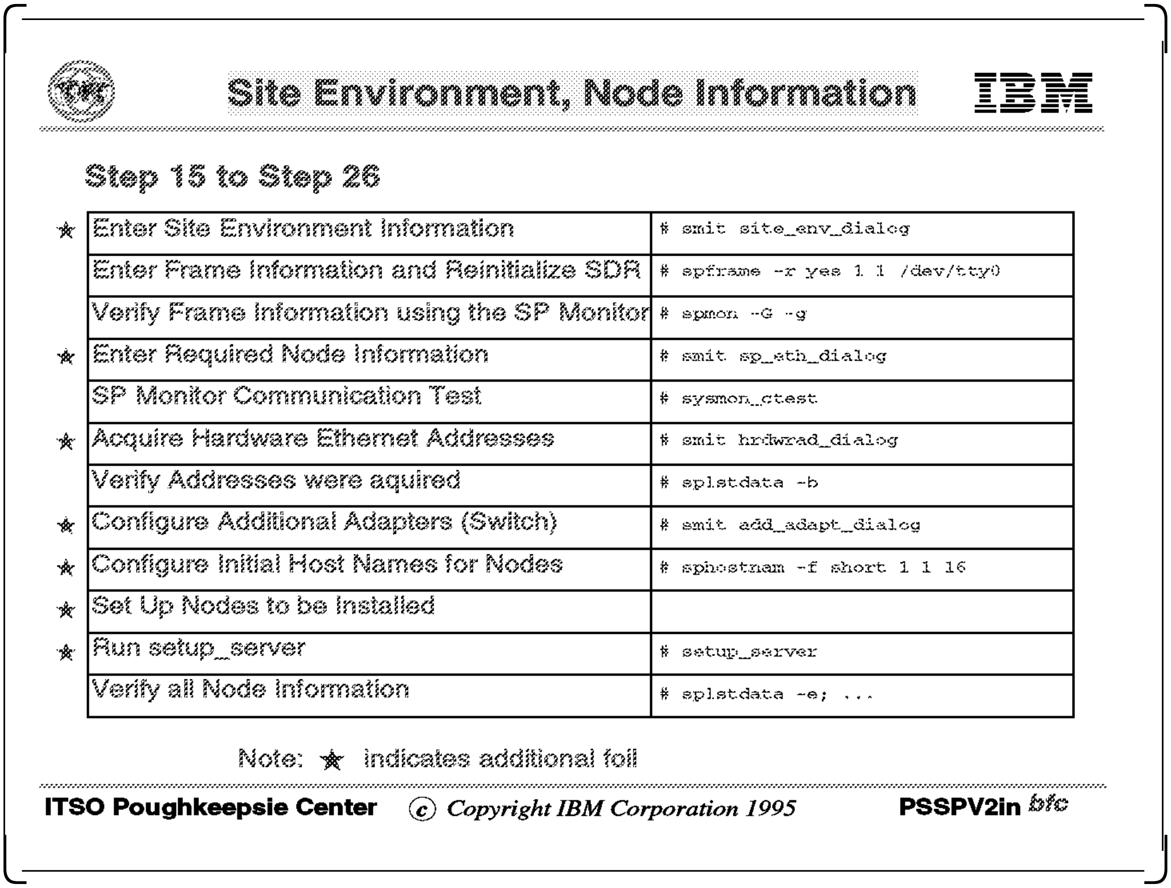
ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* PSSPV2in *bfbj*

The **install_cw** command completes the installation of PSSP (Parallel System Support Programs) on the control workstation (CWS). The install_cw command does the following:

- Configures the control workstation as **node number 0** and adds this information to the **CuAt** ODM database.
- Executes the shell script **/usr/lpp/ssp/inst_root/ssp.basic.post_i**. This shell script does the following:
  - Activates the portmap daemon in /etc/rc.tcpip.
  - Adds the **/etc/rc.sp** startup script to /etc/inittab.
  - Installs the error message templates for the **errpt** command.
  - Creates the **log directories** in /var/adm/SPlogs.
  - Adds the SMIT Panels for the Parallel System Support Programs.
  - Adds port numbers to **/etc/services** for hardmon, sdr, and heartbeat.
  - Creates the hardmon access control list (ACL) file.
  - Adds the following daemons to /etc/inittab:
    - **sdr daemon**
    - **hardmon daemon**
    - **host response (hrd) daemon**
    - **heart beat (hbd) daemon**

  - Starts the sdrd, hbd, hrd, and hardmon daemon.
  - Calls the perl script **/usr/lpp/ssp/install/bin/SDR_init**, which creates the SDR and sets up a default system partition.
- Adds the **splogd daemon** to /etc/inittab and starts the splogd daemon.
- Sets the **authentication server attribute** in the SDR to reflect the kind of authentication server environment.

3.3 Site Environment, Node Information



Site Environment, Node Information

Step 15 to Step 26

★	Enter Site Environment Information	# smit site_env_dialog
	Enter Frame Information and Reinitialize SDR	# spframe -r yes 1 1 /dev/tty0
	Verify Frame Information using the SP Monitor	# spmon -G -g
★	Enter Required Node Information	# smit sp_eth_dialog
	SP Monitor Communication Test	# sysmon_ctest
★	Acquire Hardware Ethernet Addresses	# smit hrdwrad_dialog
	Verify Addresses were aquired	# splstdata -b
★	Configure Additional Adapters (Switch)	# smit add_adapt_dialog
★	Configure Initial Host Names for Nodes	# sphostnam -f short 1 1 16
★	Set Up Nodes to be Installed	
★	Run setup_server	# setup_server
	Verify all Node Information	# splstdata -e; ...

Note: ★ indicates additional foil

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2in bfc

You can skip step 15 if you do not want to change any defaults for the site environment. We will talk about the different ways to set up the site environment in section 3.3.1, “Enter Site Environment Information” on page 54.

For step 16, you create frame objects in the SDR for each frame in your SP system. The SDR will also be reinitialized, resulting in the creation of node objects for each frame. You can use the **smit frame_dialog** panel or use the **spframe** command. Here is an example of one frame. The serial line from the frame is attached to the tty port /dev/tty0:

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

Step 17 verifies the frame information with the **spmon** command.

```
 # spmon -G -g
   →All Node Summary Display
      →3DigitDisplay
```

Step 18 will be discussed in section 3.3.2, “Enter Required Node Information” on page 56.

Step 19 checks for correct installation of the SP System Monitor. You can run these tests with **smit smonc_verify** or use the **spmon_ctest** command. For debugging, you can use this command in the following way:

```
 # ksh -x /usr/lpp/ssp/bin/sysmon_ctest
```

Step 20 will be discussed in section 3.3.3, “Acquire Hardware Ethernet Addresses” on page 58.

Step 21 verifies if hardware Ethernet addresses were correctly acquired. You can use **smit list_node_bootins** or the **splstdata** command to display the SDR boot install data:

```
 # splstdata -b
```

Step 22 is an optional step. You use this step if you have a switch or other network adapters in your nodes. We will discuss the details in section 3.3.4, “Configure Additional Adapters for Nodes” on page 59.

Step 23 is also an optional step, and we will discuss this step in section 3.3.5, “Configure Initial Host Names for Nodes” on page 60.

We will talk about step 24 and step 25 in section 3.3.6, “Run setup_server on the Control Workstation” on page 61.

The last step on this foil verifies all node information. To check the information, we use the **splstdata** command.

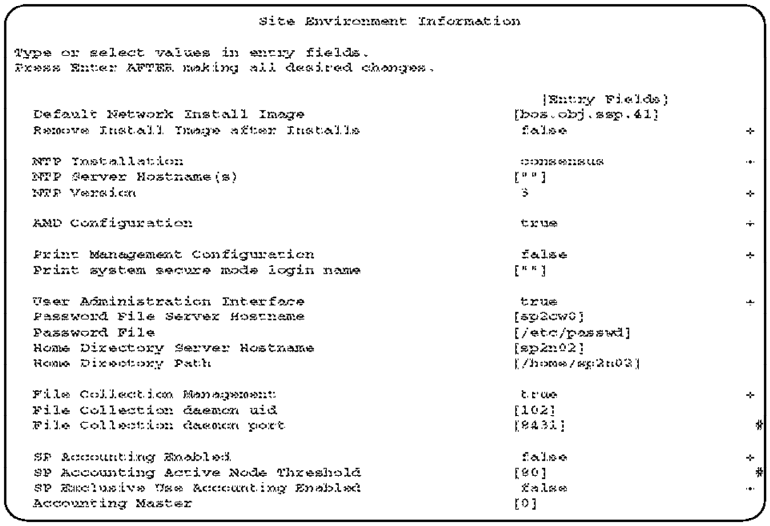
Table 1. Splstdata Options and the Corresponding SMIT Fastpath Panels

splstdata options	**SMIT fastpath**	**Description**
splstdata -e	smit list_sp	Displays the SDR site environment data
splstdata -f	smit list_frame	Displays the SDR frame data
splstdata -n -G	smit list_node_config	Displays the SDR node data
splstdata -a -G	smit list_lan	Displays the SDR adapter data
splstdata -b -G	smit list_node_bootins	Displays the SDR boot/install data

3.3.1 Enter Site Environment Information

Site Environment, Node Information

- **Step 15: Enter Site Environment Information**



```
                        Site Environment Information

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

                                                      [Entry Fields]
  Default Network Install Image                      [bos.obj.ssp.41]
  Remove Install Image after Installs                 false            +

  NTP Installation                                    consensus        +
  NTP Server Hostname(s)                             [""]
  NTP Version                                         3                +

  AMD Configuration                                   true             +

  Print Management Configuration                      false            +
  Print system secure mode login name                [""]

  User Administration Interface                       true             +
  Password File Server Hostname                      [sp2cw0]
  Password File                                      [/etc/passwd]
  Home Directory Server Hostname                     [sp2n02]
  Home Directory Path                                [/home/sp2n02]

  File Collection Management                          true             +
  File Collection daemon uid                         [102]
  File Collection daemon port                        [8431]             #

  SP Accounting Enabled                               false            +
  SP Accounting Active Node Threshold                [80]               #
  SP Exclusive Use Accounting Enabled                 false            +
  Accounting Master                                  [0]
```

ITSO Poughkeepsie Center *ⓒ Copyright IBM Corporation 1995* **PSSPV2in** *bfca*

In this foil you see the smit panel for the **site environment data**. The smit fastpath for this panel is: **smit site_env_dialog**. Before you see this screen, the shell script **/usr/lpp/ssp/bin/discover** is executed to fill in the default values on the smit panel.

The site environment data has the following entries:

Network Install Image
The default network install image is bos.obj.ssp.41 and it is located in the /spdata/sys1/install/lppsource directory.

NTP
The default values are consensus and version 3. This default causes the control workstation and file servers to attempt to generate a consensus time based on their own date settings.

Auto Mount Daemon (Amd)
The default is to use the auto mount daemon to mount the user′s home directory when the user logs into the system.

Print Management
Specify true only when you want to disable remote printing and more security for print jobs.

User Administration Interface
Specify true if you want to have the **9076 SP Users** smit menu added to your smit interface.

Password File Server
This is usually the control workstation.

Home Directory Server
You should never specify the control workstation as your home directory server. The control workstation is busy to keep all daemons running and is usually not a fast machine. Either you specify a node with a lot of disk space as a file server, or you have no home directory server and every user has the home directory on his own workstation. When you specify a node, you can also use the switch for auto mounting. This improves greatly the NFS performance. When you distribute the home directories to a lot of workstations, those workstations do not need Amd. You only have to export **/home/your_workstation_hostname** on every workstation.

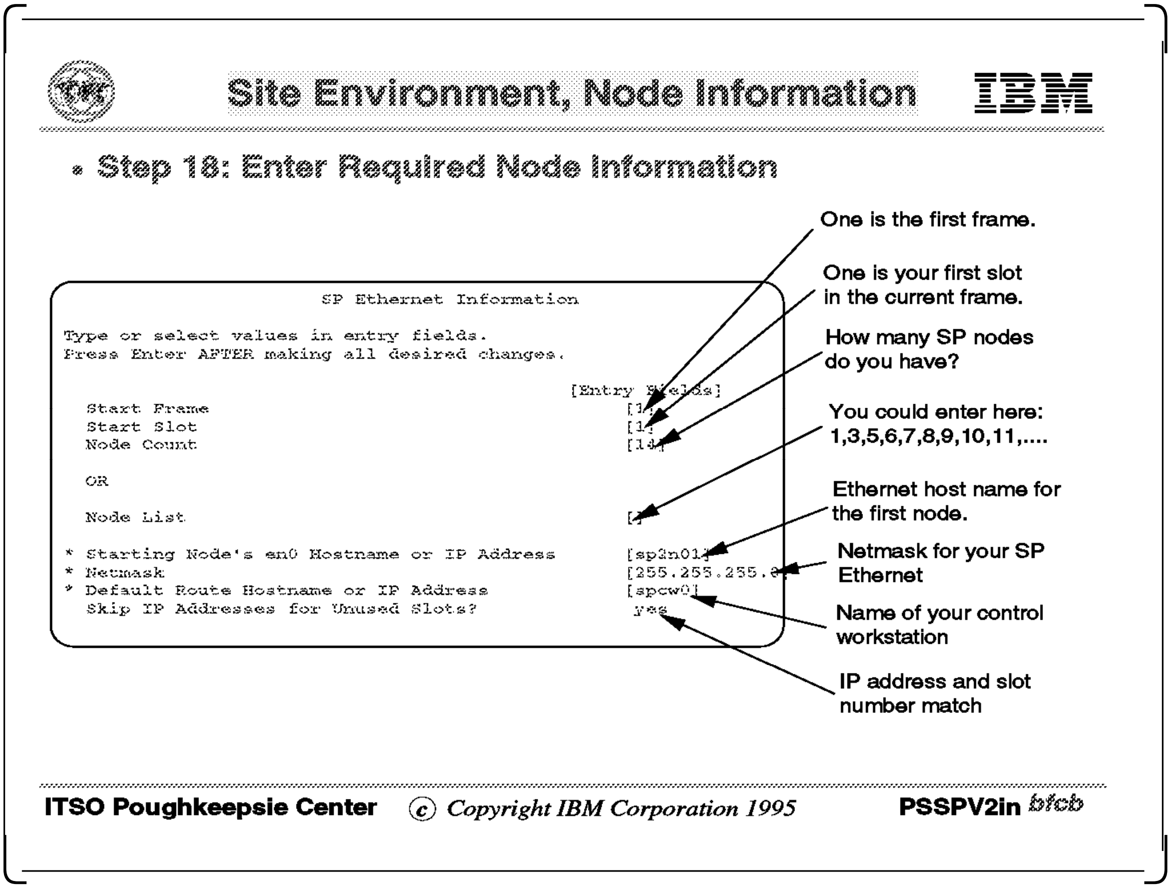
File Collections
When you use Amd you have to set file collection management to true. This ensures that the following files will be requested by every node:

- /etc/passwd
- /etc/group
- /etc/security/group
- /etc/security/passwd
- /etc/amd/amd-maps/amd.*****

The nodes request these files ten minutes after a full hour.

SP Accounting
The default is to disable the SP specific accounting

3.3.2 Enter Required Node Information



Site Environment, Node Information

- **Step 18: Enter Required Node Information**

```
                        SP Ethernet Information

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

                                                         [Entry Fields]
  Start Frame                                                [1]
  Start Slot                                                 [1]
  Node Count                                                 [14]

  OR

  Node List                                                  []

* Starting Node's en0 Hostname or IP Address                 [sp2n01]
* Netmask                                                    [255.255.255.0
* Default Route Hostname or IP Address                       [spcw0]
  Skip IP Addresses for Unused Slots?                         yes
```

One is the first frame.

One is your first slot in the current frame.

How many SP nodes do you have?

You could enter here: 1,3,5,6,7,8,9,10,11,....

Ethernet host name for the first node.

Netmask for your SP Ethernet

Name of your control workstation

IP address and slot number match

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2in bfcb

For this step you need your node configuration worksheet. The worksheet is in the *SP System Planning Guide*. Be sure that the information on the worksheet is up to date. This step adds Ethernet IP information to the node objects in the SDR.

For example, you have 14 nodes in your system. You could fill in the first three fields in the following way:

Frame Enter 1 for your first frame, 2 for your second frame and for the first setup enter 1.

Slot The slot number is relative to the frame number and for the first setup you should always enter 1.

Count You enter the number of nodes. In this example we use a node count of 14, but you should use as many nodes as you have.

Node List The node list panel is mostly used to query information about particular nodes. It is impractical to use it in this step when you have more than ten nodes.

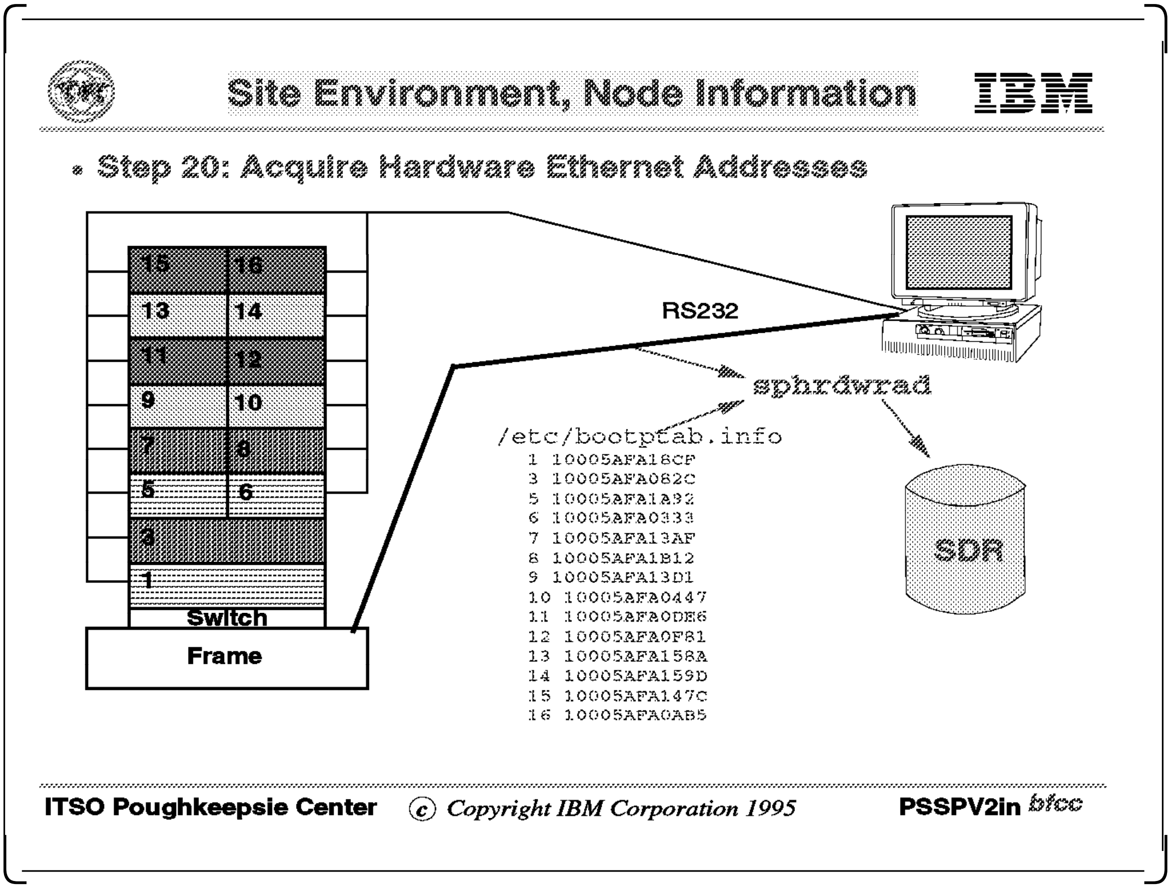
Starting Node In this field you enter the host name of your first node, like sp2n01. This name has to be known by the nameserver or has to be defined in /etc/hosts. If you have not defined any name, then you have to use the IP address.

Netmask This is your default netmask for you IP net.

Default Route Here you enter your default gateway computer, which most of the time is the control workstation.

Skip IP Addresses This field is used to indicate whether or not IP addresses should be skipped when the process encounters an unused slot. You can only have unused slots when your frame has wide nodes. We recommend using *yes* so that the IP addresses correspond to the slots in the frame.

3.3.3 Acquire Hardware Ethernet Addresses



Site Environment, Node Information

- **Step 20: Acquire Hardware Ethernet Addresses**

```
/etc/bootptab.info
  1 10005AFA18CF
  3 10005AFA082C
  5 10005AFA1A92
  6 10005AFA0333
  7 10005AFA13AF
  8 10005AFA1B12
  9 10005AFA13D1
  10 10005AFA0447
  11 10005AFA0DE6
  12 10005AFA0F81
  13 10005AFA158A
  14 10005AFA159D
  15 10005AFA147C
  16 10005AFA0AB5
```

ITSO Poughkeepsie Center ⓒ *Copyright IBM Corporation 1995* PSSPV2in *bfcc*

This step gets the hardware Ethernet addresses for the en0 adapters for your nodes. This information will be stored in the node object in the SDR and will be put on request in the **/etc/bootptab** file. There are different ways to tell the control workstation which hardware addresses you have on the nodes:

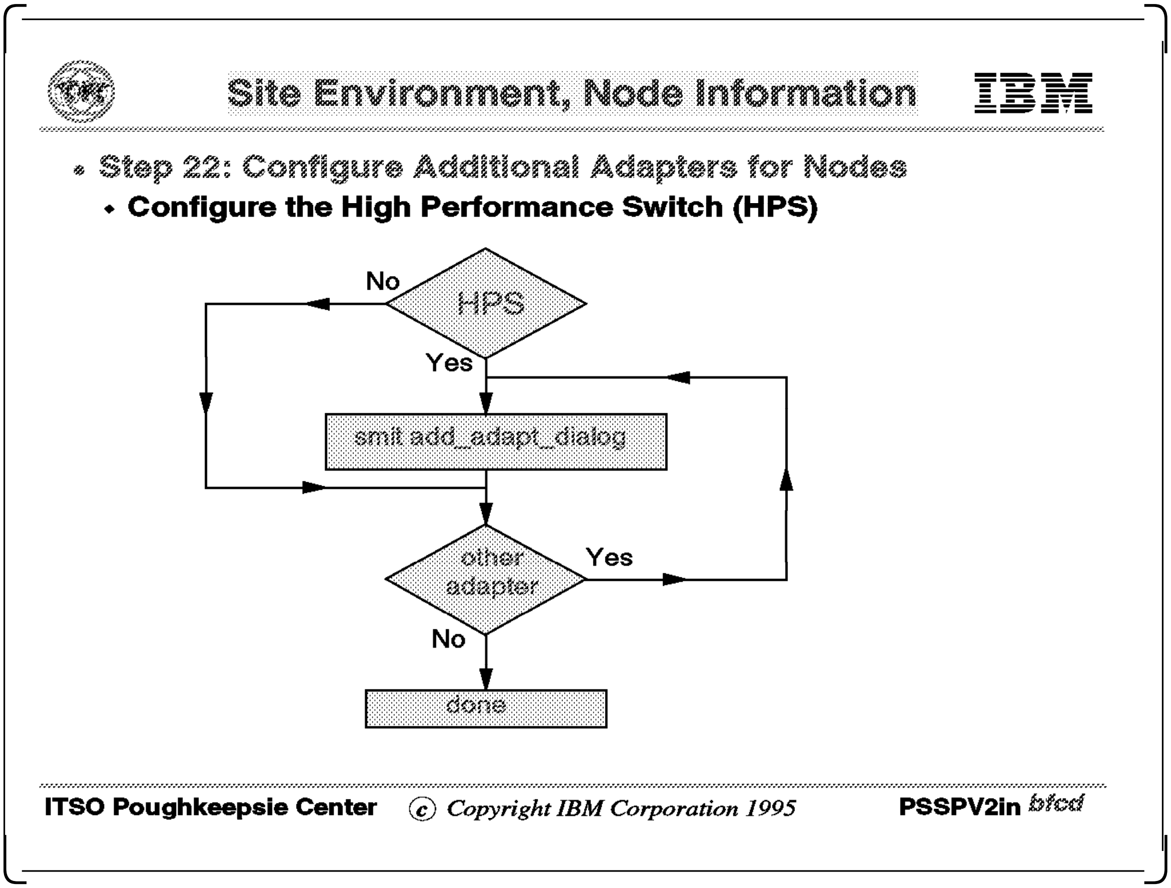
1. The **sphrdwrad** command acquires the information from every node.
2. The sphrdwrad command reads the hardware addresses from the **/etc/bootptab.info** file. This file exists only when you entered the hardware addresses manually with your favorite editor.

To fill the node object in the SDR, you can use **smit hrdwrad_dialog** or use the sphrdwrad command directly to get all hardware Ethernet addresses:

```
# sphrdwrad 1 1 rest
```

The sphrdwrad command first checks the /etc/bootptab.info file. If this file does not exist or the file has only entries for some nodes, the sphrdwrad command will shut down and start the missing nodes to acquire the hardware Ethernet address. The shutdown and restart of a node take some time, and therefore you speed up the process when the information is in the /etc/bootptab.info file.

3.3.4 Configure Additional Adapters for Nodes



Site Environment, Node Information

- Step 22: Configure Additional Adapters for Nodes
  - **Configure the High Performance Switch (HPS)**

ITSO Poughkeepsie Center *© Copyright IBM Corporation 1995* **PSSPV2in** *bfcd*

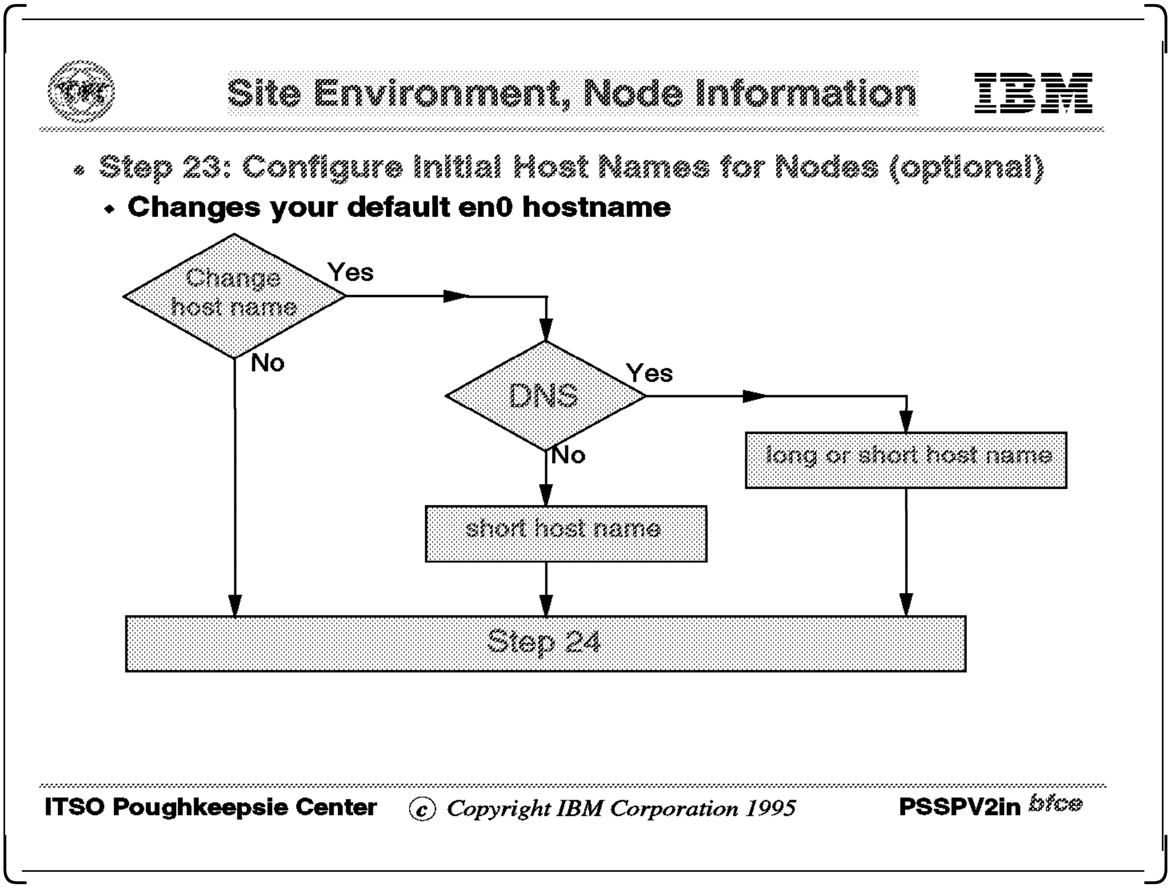
You perform this optional step when you have a switch or additional adapters in your node. To perform this step, you use **spadaptrs** or you use **smit add_adapt_dialog**. The preferred way of entering the information is to use smit.

We said already that skipping IP addresses is useful when you have wide nodes because the IP addresses correspond to the slots in the frame. Therefore we recommend changing the default values for the switch interface to:

- Set the skip IP Addresses (-s) flag to **yes**.
- Set the Enable ARP (-a) flag to **yes**.
- Set the Use Switch Node Numbers (-n) flag to **no**.

You specify any other adapter that you have in your node in the same way as the switch adapter. But before you go from one adapter to the next, you have to exit the smit add_adapt_dialog panel, otherwise the script **/usr/lpp/ssp/bin/discover** will not be executed, and you might not be able to specify any other adapter.

3.3.5 Configure Initial Host Names for Nodes



Site Environment, Node Information

- Step 23: Configure Initial Host Names for Nodes (optional)
  - **Changes your default en0 hostname**

Change host name

Yes

No

DNS

Yes

No

long or short host name

short host name

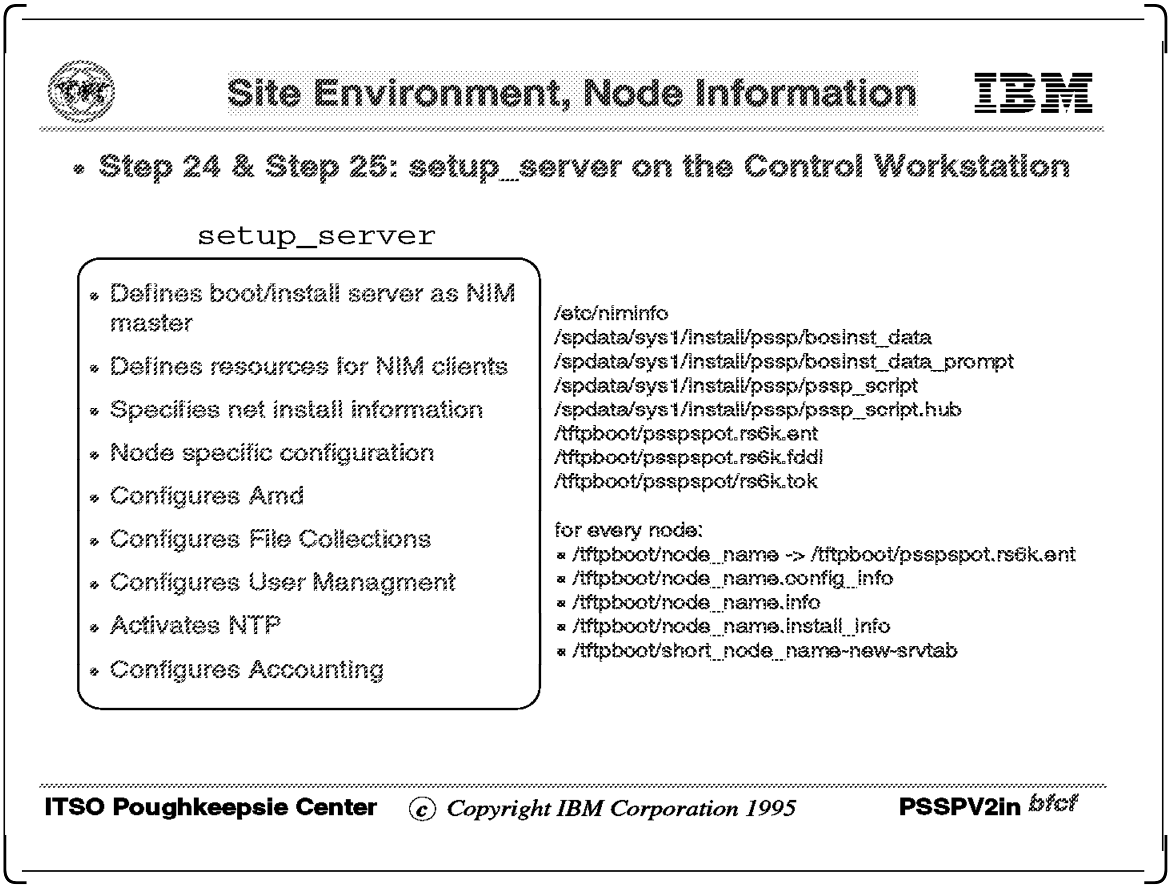
Step 24

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* **PSSPV2in** *bfce*

The first question is, do you want to change the host name? If you want the default host name on the nodes to match the **en0 adapter** name, then you can skip this step.

The second question is, do you want long or short host names? A long host name is the short host name plus the domain name. If you have no **Domain Name Server (DNS)** and only the **/etc/hosts** file, then use short host names. If you have a name server, then you can choose either the long or the short host name. When you have a mixed environment of workstations, where you can only select eight characters for the host name (like HP-workstations), you will mostly prefer to have short names on all machines.

3.3.6 Run setup_server on the Control Workstation



Site Environment, Node Information

- Step 24 & Step 25: setup_server on the Control Workstation

setup_server

- Defines boot/install server as NIM master
- Defines resources for NIM clients
- Specifies net install information
- Node specific configuration
- Configures Amd
- Configures File Collections
- Configures User Managment
- Activates NTP
- Configures Accounting

/etc/niminfo
/spdata/sys1/install/pssp/bosinst_data
/spdata/sys1/install/pssp/bosinst_data_prompt
/spdata/sys1/install/pssp/pssp_script
/spdata/sys1/install/pssp/pssp_script.hub
/tftpboot/psspspot.rs6k.ent
/tftpboot/psspspot.rs6k.fddi
/tftpboot/psspspot/rs6k.tok

for every node:
- /tftpboot/node_name -> /tftpboot/psspspot.rs6k.ent
- /tftpboot/node_name.config_info
- /tftpboot/node_name.info
- /tftpboot/node_name.install_info
- /tftpboot/short_node_name-new-srvtab

ITSO Poughkeepsie Center ⓒ *Copyright IBM Corporation 1995* PSSPV2in *bfcf*

We combined step 24 and step 25 because it is better to run the **setup_server** command before setting up nodes to be installed. The first time the setup_server runs, it takes approximately 20 to 30 minutes to configure the control workstation as a NIM Master and to configure the nodes as NIM Clients. After that we set up the nodes to be installed and run setup_server again.

The setup_server command was changed from a shell script into a perl script. This improves very much the scalability of this script, and the perl script has now more than 3000 lines of code.

The first time you run setup_server it will do the following on each boot/install server (usually the control workstation):

- Defines the boot/install servers as NIM master.
- Defines the other nodes as NIM clients.
- Allocates the NIM resources necessary for each NIM client.
- Creates the **/tftpboot/node_hostname.install_info** file.
- Creates the **/tftpboot/node_hostname.config_info** file.
- Creates the **/tftpboot/short_hostname-new-srvtab** file. This file is the authentication server key file.

- The following options will be configured when you have selected them in your site environment:
  - Auto mount daemon (Amd)
  - File Collection with the Software Update Protocol (SUP) daemon
  - User Management
  - Network Time Protocol (NTP)
  - Print Service
  - Accounting

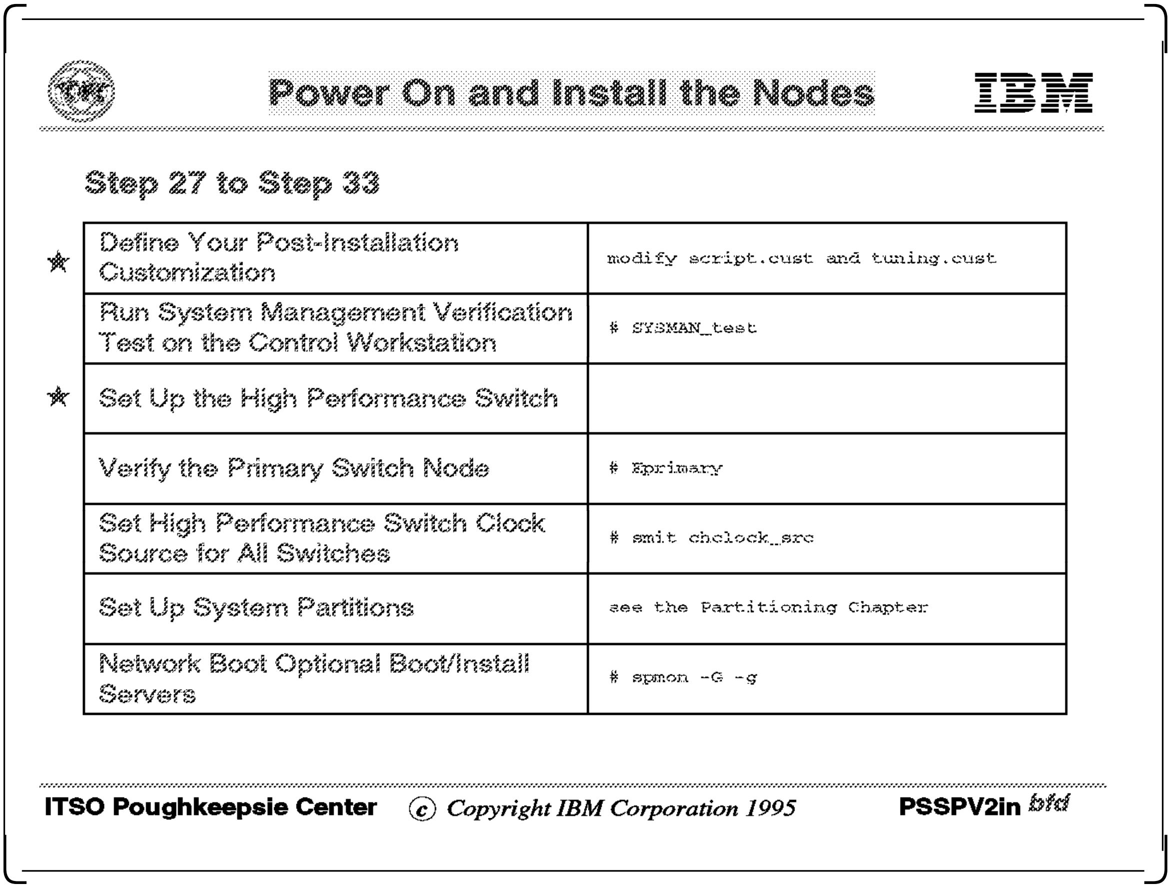
  These options will be available the next time you reboot your control workstation.

After **setup_server** has executed without any errors, you define which nodes should be installed or which nodes should boot from their disks. At the end of your configuration, you run setup_server again. Here we have a sample output:

```
# setup_server
setup_server command results from sp2cw0

setup_server: Starting setup_server

setup_server: Running services_config script to configure SSP services.
              This may take a few minutes...

setup_server: Getting Node object information from the SDR

setup_server: Creating Node arrays for processing

setup_server: Getting SP Object information from the SDR

setup_server: Performing Control Workstation setup

setup_server: Checking kerberos setup

setup_server: Checking to see if this system is an install server

setup_server: Checking to see if bos.sysmgt.nim.master is installed
setup_server: bos.sysmgt.nim.master is installed.

setup_server: Checking to see if bos.sysmgt.nim.spot is installed
setup_server: bos.sysmgt.nim.spot is installed.

setup_server: NIM master is configured

setup_server: Checking the NIM master resources -
              lpp_source, spot, mksysb, bosinst.data, and script
setup_server: spot exists
setup_server: bosinst_data noprompt resource exists
setup_server: bosinst_data prompt resource exists
setup_server: script resource exists


setup_server: Checking NIM client - sp2n01
setup_server: Checking NIM client allocations - sp2n01
setup_server: Creating the /tftpboot/sp2n01.itsc.pok.ibm.com.install_info file
setup_server: Creating the /tftpboot/sp2n01.itsc.pok.ibm.com.config_info file
setup_server: Creating/Verifying /tftpboot/sp2n01-new-srvtab

setup_server: Checking NIM client - sp2n03
```

```
setup_server: Checking NIM client allocations - sp2n03
setup_server: Creating the /tftpboot/sp2n03.itsc.pok.ibm.com.install_info file
setup_server: Creating the /tftpboot/sp2n03.itsc.pok.ibm.com.config_info file
setup_server: Creating/Verifying /tftpboot/sp2n03-new-srvtab

setup_server: Checking NIM client - sp2n05
setup_server: Checking NIM client allocations - sp2n05
setup_server: Creating the /tftpboot/sp2n05.itsc.pok.ibm.com.install_info file
setup_server: Creating the /tftpboot/sp2n05.itsc.pok.ibm.com.config_info file
setup_server: Creating/Verifying /tftpboot/sp2n05-new-srvtab

setup_server: Checking NIM client - sp2n06
setup_server: Checking NIM client allocations - sp2n06
setup_server: Creating the /tftpboot/sp2n06.itsc.pok.ibm.com.install_info file
setup_server: Creating the /tftpboot/sp2n06.itsc.pok.ibm.com.config_info file
setup_server: Creating/Verifying /tftpboot/sp2n06-new-srvtab

.
.
.
.

setup_server: Checking NIM client - sp2n16
setup_server: Checking NIM client allocations - sp2n16
setup_server: Allocating resources for client sp2n16
setup_server: Creating the /tftpboot/sp2n16.itsc.pok.ibm.com.install_info file
setup_server: Creating the /tftpboot/sp2n16.itsc.pok.ibm.com.config_info file
setup_server: Creating/Verifying /tftpboot/sp2n16-new-srvtab
```

3.4 Power On and Install the Nodes



Power On and Install the Nodes

Step 27 to Step 33

★	Define Your Post-Installation Customization	modify script.cust and tuning.cust
	Run System Management Verification Test on the Control Workstation	# SYSMAN_test
★	Set Up the High Performance Switch	
	Verify the Primary Switch Node	# Eprimary
	Set High Performance Switch Clock Source for All Switches	# smit chclock_src
	Set Up System Partitions	see the Partitioning Chapter
	Network Boot Optional Boot/Install Servers	# spmon -G -g

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2in bfd

Step 27 will be discussed in section 3.4.2, “Define Your Post-Installation Customization” on page 68.

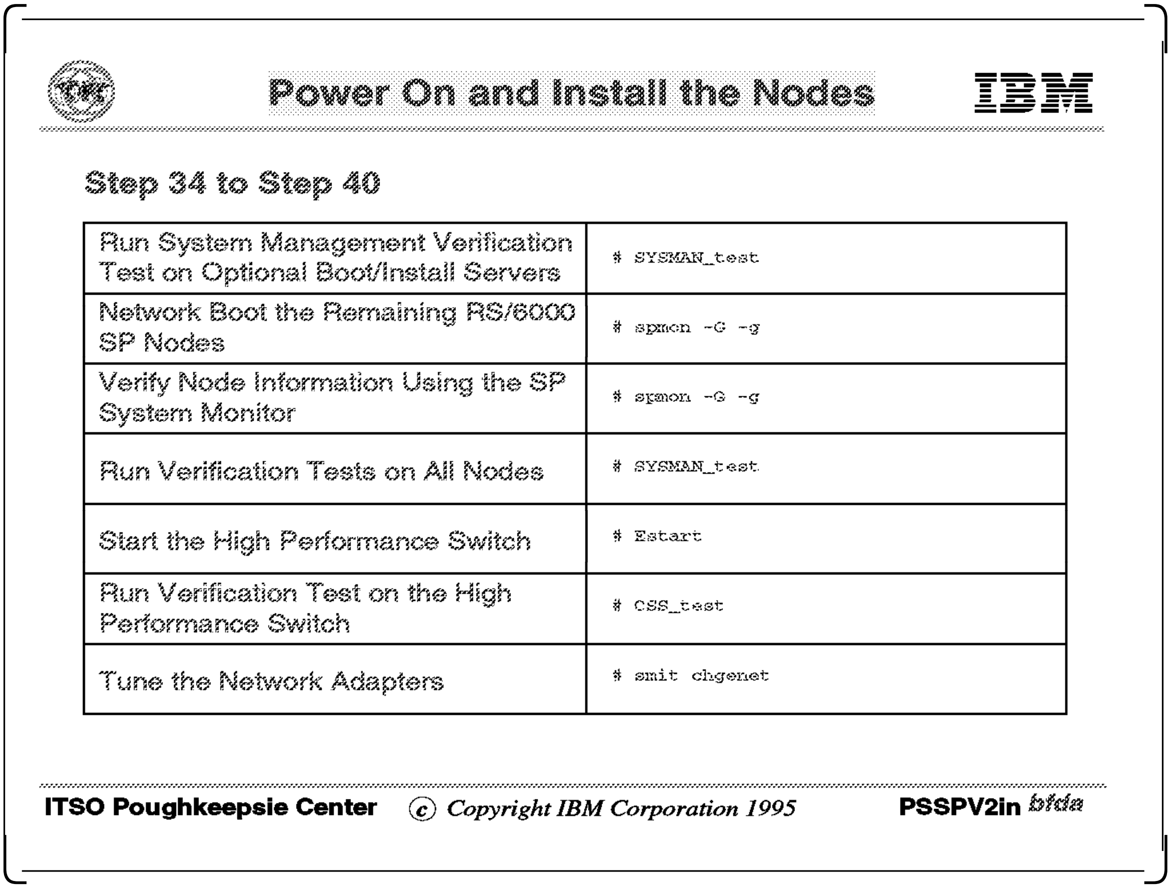
Step 28 runs the system management verification test on the control workstation. You can use the **SYSMAN_test** command or **smit sman_verify** to check if your systems management software is set up correctly. In this example we use the ksh shell to run the SYSMAN_test command:

```
# ksh -x /usr/lpp/ssp/bin/SYSMAN_test
# pg /var/adm/SPlogs/SYSMAN_test.log
```

Verify at all times the SYSMAN log. If you see an error like The amd daemon is not running, but the Amd option was configured, reboot your control workstation.

Step 29 to step 33 will be discussed in section 3.4.3, “Set Up the High Performance Switch” on page 70.

3.4.1 Step 34 to Step 40



Power On and Install the Nodes

IBM

Step 34 to Step 40

Run System Management Verification Test on Optional Boot/Install Servers	# SYSMAN_test
Network Boot the Remaining RS/6000 SP Nodes	# spmon -G -g
Verify Node Information Using the SP System Monitor	# spmon -G -g
Run Verification Tests on All Nodes	# SYSMAN_test
Start the High Performance Switch	# Estart
Run Verification Test on the High Performance Switch	# CSS_test
Tune the Network Adapters	# smit chgenet

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2in bfda

Step 34 uses the same procedures as step 28. You use either the **SYSMAN_test** command or use **smit sman_verify** to check if your boot/install servers are powered up and that the PSSP software is installed correctly. Here we show you the SYSMAN_test command again:

```
# SYSMAN_test
# pg /var/adm/SPlogs/SYSMAN_test.log
```

In step 35 you repeat the procedures used in step 33, but this time you net boot the remaining nodes in your SP system. Here are the procedures for network boot:

```
# spmon -G -g
   →SP →Global Controls
      →Select all the remaining nodes for the network boot
         →Net Boot
            →Do Command
```

In step 36 you verify the node information with the SP System Monitor. Here are the steps to check the hostResponds and powerLED indicators:

```
# spmon -G -g
  →SP →All Node Summary Display
     →hostResponds
        →Display all-node summary
     →powerLED
        →Display all-node summary
```

Check if all indicators in the hostResponds window and all indicators in the powerLED window are green.

Step 37 is the same as step 34. Follow the procedures in step 34 and verify if all your nodes are working.

In step 38 you start the High Performance Switch with **smit start_switch** or with the **Estart** command. Here we have an example:

```
# Estart
```

If you have partitions, then you should use the Estart command in the following way to start the switch on all nodes:

```
# export SP_NAME=hostname_partition_1
# Estart
# export SP_NAME=hostname_partition_2
# Estart
# export SP_NAME=hostname_partition_3
# Estart
```

Step 39 verifies the High Performance Switch with the **CSS_test** command. If you prefer to use smit, you can use **smit css_verify**. Here is an example for CSS_test:

```
# CSS_test
```

If you have partitions, then you would use the command in the following way:

```
# export SP_NAME=hostname_partition_1
# CSS_test
# export SP_NAME=hostname_partition_2
# CSS_test
# export SP_NAME=hostname_partition_3
# CSS_test
```

The last step is tuning the network adapters. First, we increase the transmit queue size of the network adapter to the maximum, and next we increase the receive queue size (does not apply for built in Ethernet adapters) to the maximum. We have also to change the MBUF AIX system parameter so that enough real memory is available for the network. Here is our example to tune the Ethernet adapter ent0 on the control workstation and on all nodes (values for the build-in Ethernet adapter):

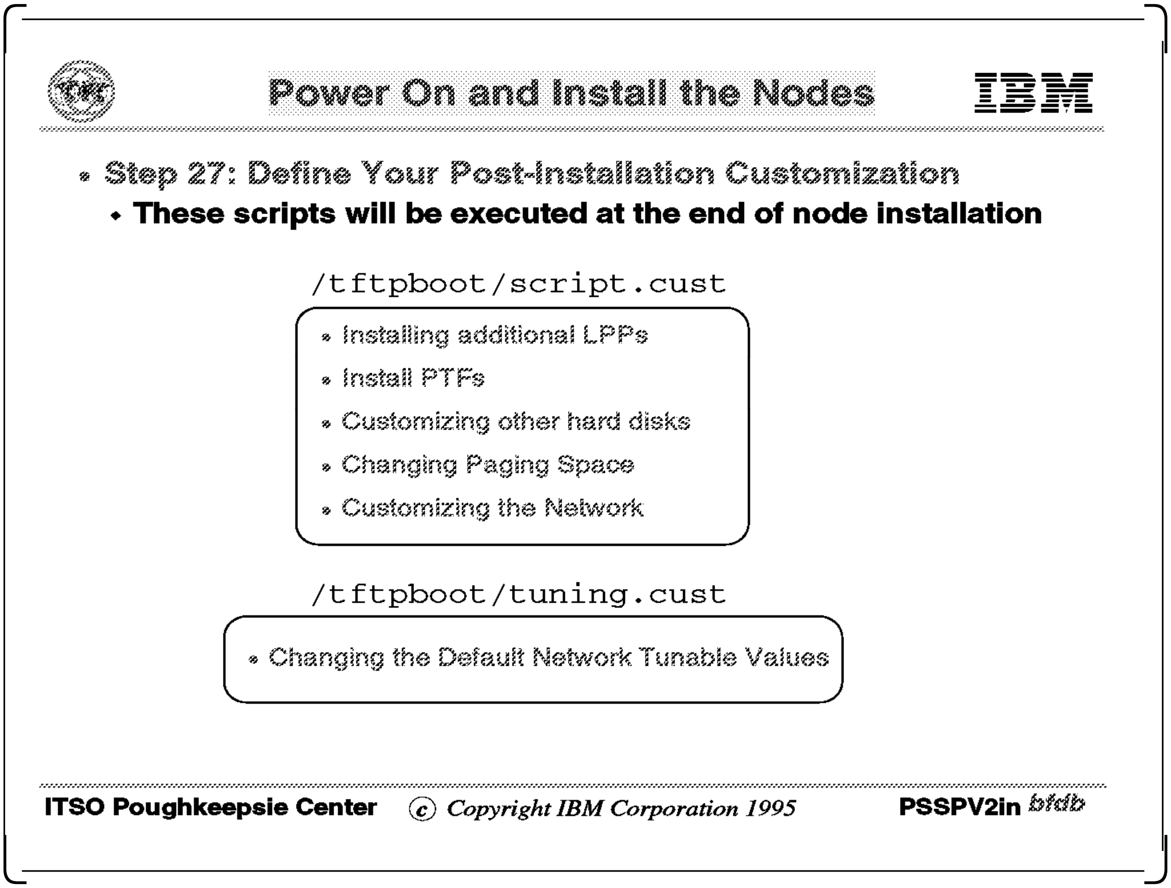
```
# chdev -l ent0 -a xmt_que_size=150 -P
ent0 changed
# dsh -G -a /usr/sbin/chdev -l ent0 -a xmt_que_size=150 -P
sp2en01: ent0 changed
sp2en02: ent0 changed
sp2en05: ent0 changed
sp2en06: ent0 changed
sp2en09: ent0 changed
sp2en11: ent0 changed
sp2en03: ent0 changed
sp2en04: ent0 changed
```

```
sp2en07: ent0 changed
sp2en08: ent0 changed
sp2en10: ent0 changed
sp2en12: ent0 changed
# chdev -l sys0 -a maxmbuf=16384
sys0 changed
# dsh -G -a /usr/sbin/chdev -l sys0 -a maxmbuf=16384
sp2en01: sys0 changed
sp2en02: sys0 changed
sp2en05: sys0 changed
sp2en06: sys0 changed
sp2en09: sys0 changed
sp2en11: sys0 changed
sp2en03: sys0 changed
sp2en04: sys0 changed
sp2en07: sys0 changed
sp2en08: sys0 changed
sp2en10: sys0 changed
sp2en12: sys0 changed
```

If you have wide nodes or an Ethernet adapter card on the control workstation, then you can change also the receiving queue size on the adapter:

```
# chdev -l ent0 -a xmt_que_size=150 -a rec_que_size=150 -P
ent0 changed
```

3.4.2 Define Your Post-Installation Customization



Power On and Install the Nodes

- Step 27: Define Your Post-Installation Customization
  - **These scripts will be executed at the end of node installation**

/tftpboot/script.cust

- Installing additional LPPs
- Install PTFs
- Customizing other hard disks
- Changing Paging Space
- Customizing the Network

/tftpboot/tuning.cust

- Changing the Default Network Tunable Values

ITSO Poughkeepsie Center Ⓒ *Copyright IBM Corporation 1995* **PSSPV2in** *bfdb*

With this step you want to make additional customization on your nodes. There are two scripts that are used for this customization:

- **script.cust**
- **tuning.cust**

Copy the file **/usr/lpp/ssp/samples/script.cust** to **/tftpboot** and modify this script with your favorite editor. Uncomment and change the lines in the script to perform the following customization steps on the nodes:

- Install additional LPPs, such as a Fortran compiler.
- Install your latest PTFs.
- Import other hard disks with data.
- Increase the default paging space.
- Increase the maximum number of users that can be concurrently logged on.
- If you use a name server, copy the /etc/resolv.conf file to every node.
- If you use Network Information Service (NIS), configure NIS.

You should copy the second file **/usr/lpp/ssp/samples/tuning.cust** to **/tftpboot** and modify the values. The **/etc/rc.sp** startup script will execute the

/tftpboot/tuning.cust at every system boot at that node. Here we have network option values for four configurations:

1. Example for an SP system with best switch performance:

   ```
   if [ -f /usr/sbin/no ] ; then
           /usr/sbin/no -o thewall=16384
           /usr/sbin/no -o sb_max=655360
           /usr/sbin/no -o tcp_sendspace=262144
           /usr/sbin/no -o tcp_recvspace=262144
           /usr/sbin/no -o tcp_mssdflt=4096
           /usr/sbin/no -o udp_sendspace=65536
           /usr/sbin/no -o udp_recvspace=655360
           /usr/sbin/no -o ipforwarding=1
           /usr/sbin/no -o rfc1323=1
   fi
   ```

2. Example for an SP system with no switch and best Ethernet performance:

   ```
   if [ -f /usr/sbin/no ] ; then
           /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=1500
           /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
   fi
   ```

3. Example for an SP system with Ethernet and thin nodes:

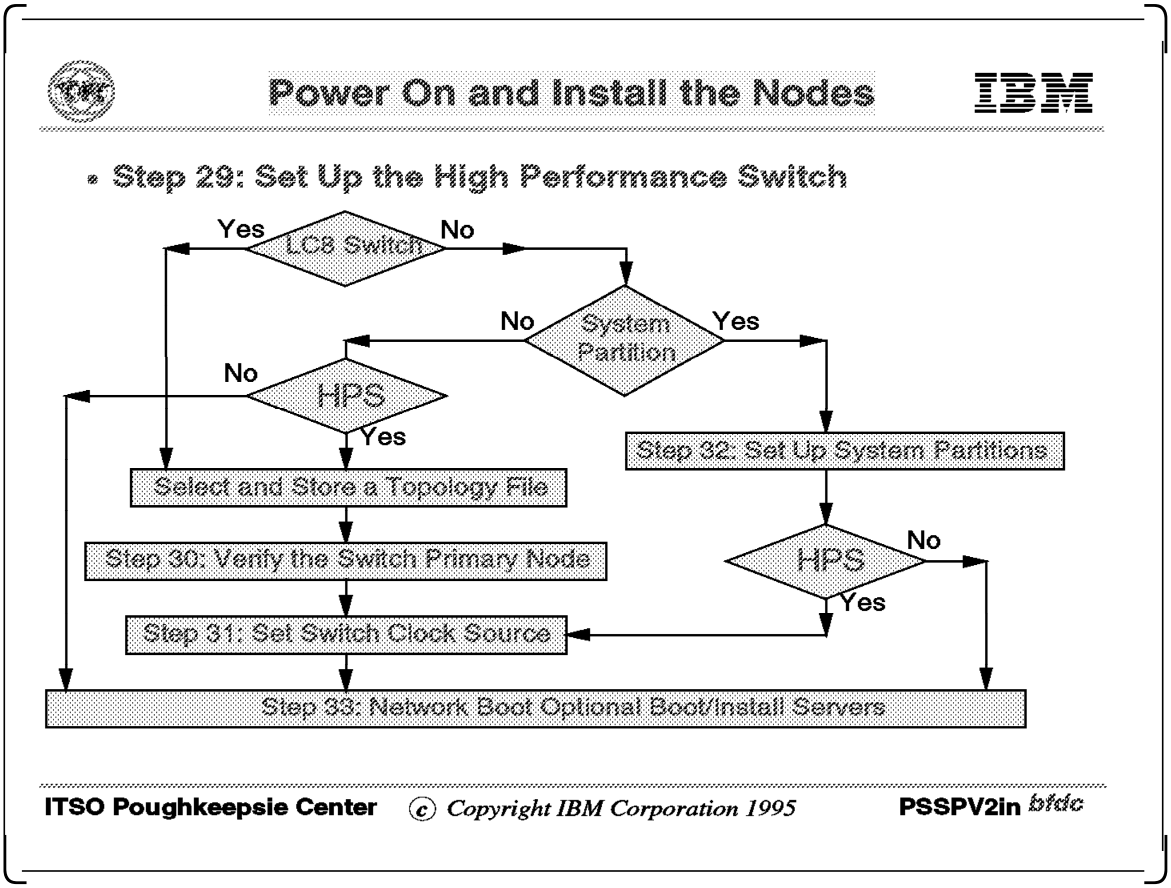
   ```
   if [ -f /usr/sbin/no ] ; then
           /usr/sbin/no -o thewall=16384
           /usr/sbin/no -o sb_max=442368
           /usr/sbin/no -o tcp_sendspace=221184
           /usr/sbin/no -o tcp_recvspace=221184
           /usr/sbin/no -o tcp_mssdflt=1500
           /usr/sbin/no -o udp_sendspace=59392
           /usr/sbin/no -o udp_recvspace=221184
           /usr/sbin/no -o ipforwarding=1
           /usr/sbin/no -o rfc1323=1
   fi
   ```

4. Example for an SP system with Ethernet and wide nodes:

   ```
   if [ -f /usr/sbin/no ] ; then
           /usr/sbin/no -o thewall=16384
           /usr/sbin/no -o sb_max=475236
           /usr/sbin/no -o tcp_sendspace=237568
           /usr/sbin/no -o tcp_recvspace=237568
           /usr/sbin/no -o tcp_mssdflt=1500
           /usr/sbin/no -o udp_sendspace=59392
           /usr/sbin/no -o udp_recvspace=237568
           /usr/sbin/no -o ipforwarding=1
           /usr/sbin/no -o rfc1323=1
   fi
   ```

The meaning of the options are explained in the /tftpboot/tuning.script file. These values should improve your performance, but you can try some other values that are better suited for your environment.

3.4.3 Set Up the High Performance Switch



Power On and Install the Nodes

- Step 29: Set Up the High Performance Switch

LC8 Switch
Yes
No
System Partition
No
Yes
HPS
No
Yes
Select and Store a Topology File
Step 30: Verify the Switch Primary Node
Step 31: Set Switch Clock Source
Step 32: Set Up System Partitions
HPS
No
Yes
Step 33: Network Boot Optional Boot/Install Servers

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2in bfdc

For this step you have to make some decisions. The first question is, do you have a LC8 (low cost) switch? If the answer is yes, then you cannot partition that switch and would select and store the LC8 topology file. If the answer is no, than you can think about system partitioning.

The second question is, do you want partitions other than your default partition in your system? If you say yes, then you would go directly to step 32: Set Up System Partitions. If you say no, then the next question would be, do you have a High Performance Switch? If you do not have the High Performance Switch, go to step 33: Network Boot Optional Boot/Install Servers. If you have a switch, then you have to select and store a topology file.

There are two ways that you can store a topology file: you can use the **Eannotator** command or use **smit annotator**. Here we have the smit menu for a one frame system:

```
                         Topology File Annotator

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

                                                   [Entry Fields]
 * Input Topology File Name          [/etc/SP/expected.top.1nsb.0isb.0] +/
 * Retrieve Input File from SDR      [no]                               +
   Output Topology File Name         [/etc/SP/expected.top.annotated]   +/
 * Save Output File to SDR           [yes]                              +
```

The switch topology file should be annotated before storing it in the SDR. The next part of the High Performance Switch set up is to store the switch topology file in the SDR. You can use **smit etopology_store** or use the **Etopology** command. Here we have an example for a one frame system:

```
# /usr/lpp/ssp/bin/Etopology /etc/SP/expected.top.1nsb.0isb.0
```

Step 30: When you are finished with the switch setup, verify the switch primary node. To verify the primary node, use the following command:

```
# Eprimary
1
```

This command returned 1 as the primary node. You should not change the switch primary node unless the current primary node is unavailable, powered off, or being serviced.

Step 31: With this step you initialize the clock source for all switches. When you have a HPS switch or an LC8 switch, you have to perform this step. You can use **smit chclock_src** or use the **Eclock** command. Here we have an example for a 16 node switch:

```
# /usr/lpp/ssp/bin/Eclock -f /etc/SP/Eclock.top.1nsb.0isb.0
```

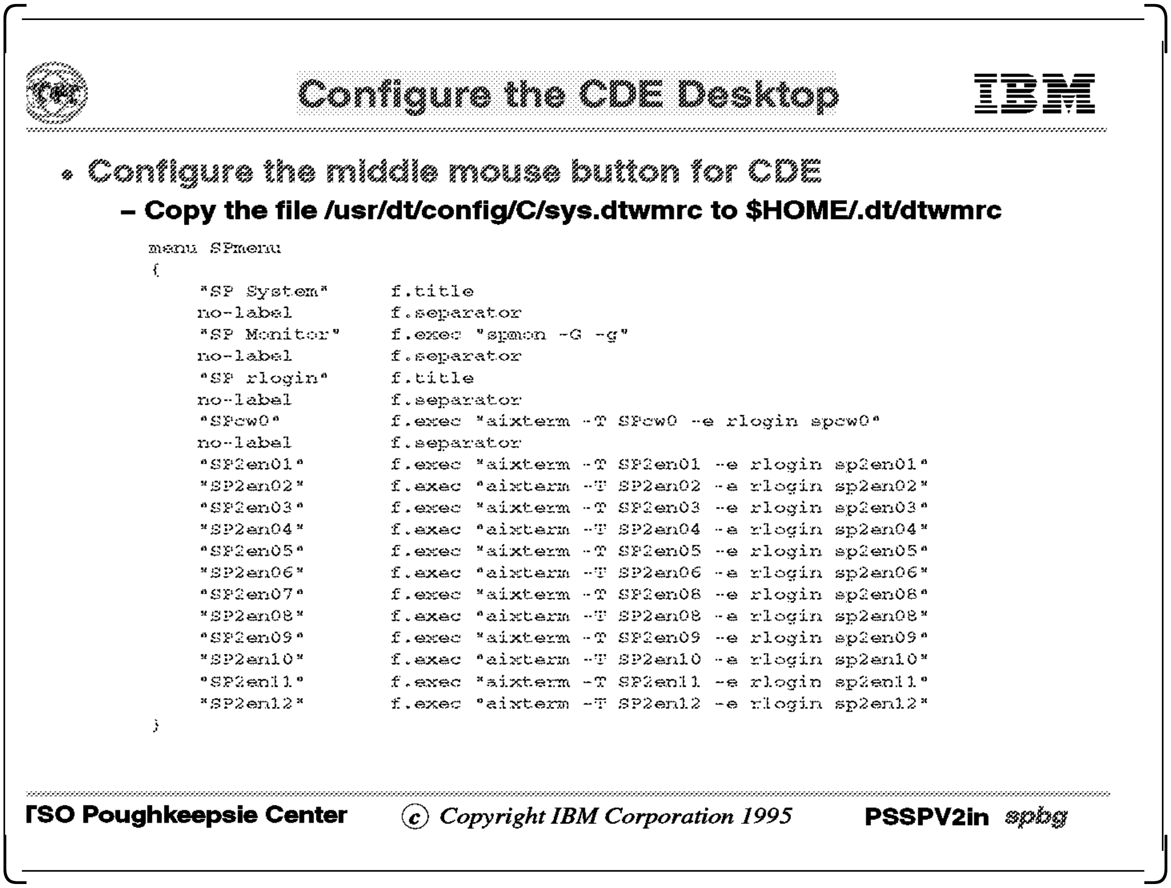
Step 32: We will talk about System Partitioning in chapter Part 3, “PSSP V2 Partitioning” on page 75. You should also remember that you can partition your system at a later time; you do not have to do it as part of this installation.

Step 33: In this step we use the same procedures as in step 35. The difference is that we will network boot the boot/install servers on the nodes, and in step 35 we will network boot all other nodes. Here are the procedures for network boot:

```
# spmon -G -g
  →SP →All Node Summary Display
     →3DigitDisplay
  →SP →Global Controls
     →Select the nodes for the network boot
        →Net Boot
           →Do Command
```

Now the installation process will start and take about 30 to 40 minutes. There is a very nice time table for the network installation progress in the *SP Installation Guide*.

Chapter 4. Configure the CDE Desktop



Configure the CDE Desktop

- Configure the middle mouse button for CDE
  - Copy the file /usr/dt/config/C/sys.dtwmrc to $HOME/.dt/dtwmrc

```
menu SPmenu
{
    "SP System"     f.title
    no-label        f.separator
    "SP Monitor"    f.exec "spmon -G -g"
    no-label        f.separator
    "SP rlogin"     f.title
    no-label        f.separator
    "SPcw0"         f.exec "aixterm -T SPcw0 -e rlogin spcw0"
    no-label        f.separator
    "SP2en01"       f.exec "aixterm -T SP2en01 -e rlogin sp2en01"
    "SP2en02"       f.exec "aixterm -T SP2en02 -e rlogin sp2en02"
    "SP2en03"       f.exec "aixterm -T SP2en03 -e rlogin sp2en03"
    "SP2en04"       f.exec "aixterm -T SP2en04 -e rlogin sp2en04"
    "SP2en05"       f.exec "aixterm -T SP2en05 -e rlogin sp2en05"
    "SP2en06"       f.exec "aixterm -T SP2en06 -e rlogin sp2en06"
    "SP2en07"       f.exec "aixterm -T SP2en08 -e rlogin sp2en08"
    "SP2en08"       f.exec "aixterm -T SP2en08 -e rlogin sp2en08"
    "SP2en09"       f.exec "aixterm -T SP2en09 -e rlogin sp2en09"
    "SP2en10"       f.exec "aixterm -T SP2en10 -e rlogin sp2en10"
    "SP2en11"       f.exec "aixterm -T SP2en11 -e rlogin sp2en11"
    "SP2en12"       f.exec "aixterm -T SP2en12 -e rlogin sp2en12"
}
```

TSO Poughkeepsie Center   Ⓒ *Copyright IBM Corporation 1995*   **PSSPV2in** spbg

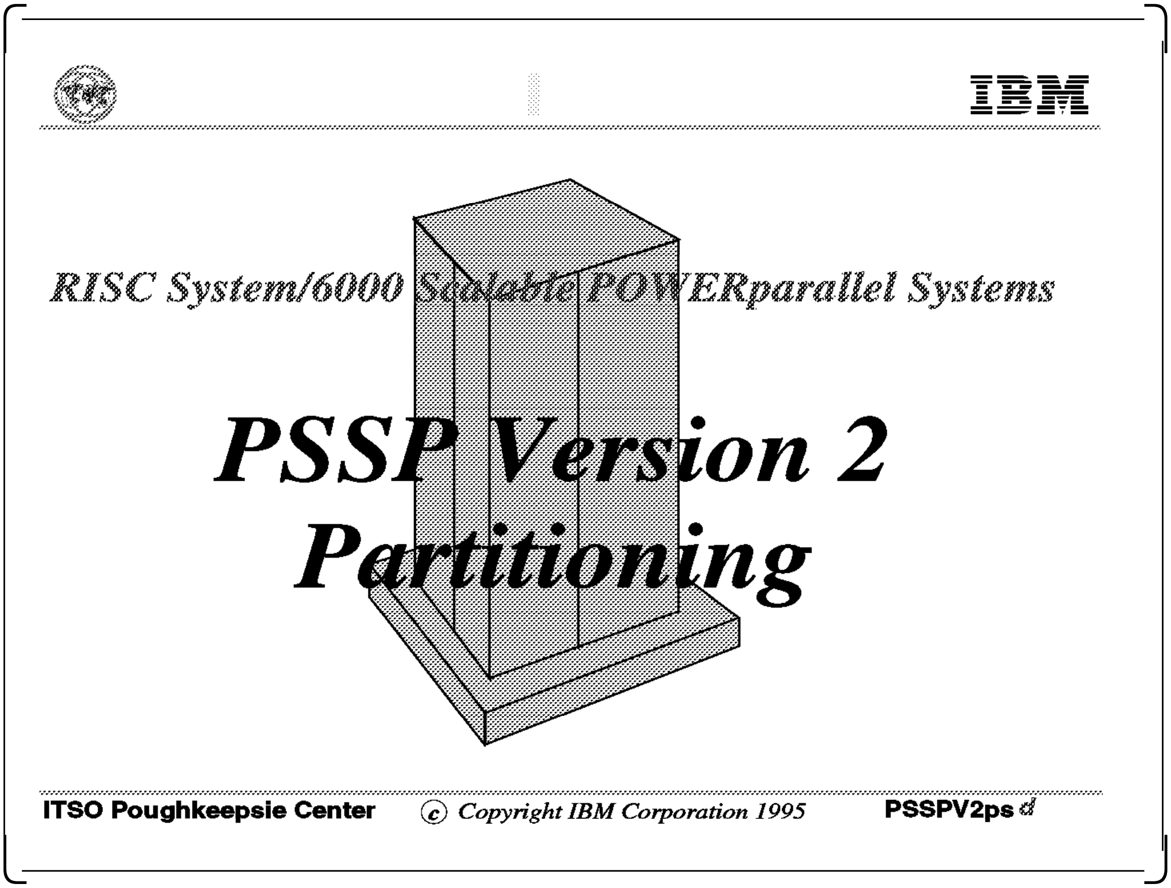
Most AIX 4.1 customers use the **Common Desktop Environment (CDE)** on the control workstation. In this chapter we show you how to configure the middle mouse button. First of all you have to copy the file **/usr/dt/config/C/sys.dtwmrc** to **$HOME/.dt/dtwmrc**. Then add the following bold line to activate the middle mouse button:

```
###
#   Mouse Button Bindings Description
###
Buttons DtButtonBindings
{
    <Btn1Down>        root                  f.marquee_selection
    <Btn2Down>        root                  f.menu  SPMenu
    <Btn3Down>        root                  f.menu  DtRootMenu
    <Btn1Down>        frame|icon            f.raise
    <Btn3Down>        frame|icon            f.post_wmenu
    Alt<Btn1Down>     icon|window           f.move
    Alt<Btn3Down>     window                f.minimize
}
```

Finally add the following lines to your $HOME/.dt/dtwmrc file and restart the CDE Workspace Manager.

```
#============================================================================
# SPMenu
#============================================================================
menu SPmenu
{
    "SP System"     f.title
    no-label        f.separator
    "SP Monitor"    f.exec "spmon -G -g"
    no-label        f.separator
    "SP rlogin"     f.title
    no-label        f.separator
    "SPcw0"         f.exec "aixterm -T SPcw0 -e rlogin spcw0"
    no-label        f.separator
    "SP2en01"       f.exec "aixterm -T SP2en01 -e rlogin sp2en01"
    "SP2en02"       f.exec "aixterm -T SP2en02 -e rlogin sp2en02"
    "SP2en03"       f.exec "aixterm -T SP2en03 -e rlogin sp2en03"
    "SP2en04"       f.exec "aixterm -T SP2en04 -e rlogin sp2en04"
    "SP2en05"       f.exec "aixterm -T SP2en05 -e rlogin sp2en05"
    "SP2en06"       f.exec "aixterm -T SP2en06 -e rlogin sp2en06"
    "SP2en07"       f.exec "aixterm -T SP2en08 -e rlogin sp2en08"
    "SP2en08"       f.exec "aixterm -T SP2en08 -e rlogin sp2en08"
    "SP2en09"       f.exec "aixterm -T SP2en09 -e rlogin sp2en09"
    "SP2en10"       f.exec "aixterm -T SP2en10 -e rlogin sp2en10"
    "SP2en11"       f.exec "aixterm -T SP2en11 -e rlogin sp2en11"
    "SP2en12"       f.exec "aixterm -T SP2en12 -e rlogin sp2en12"
}
```

Part 3. PSSP V2 Partitioning



RISC System/6000 Scalable POWERparallel Systems

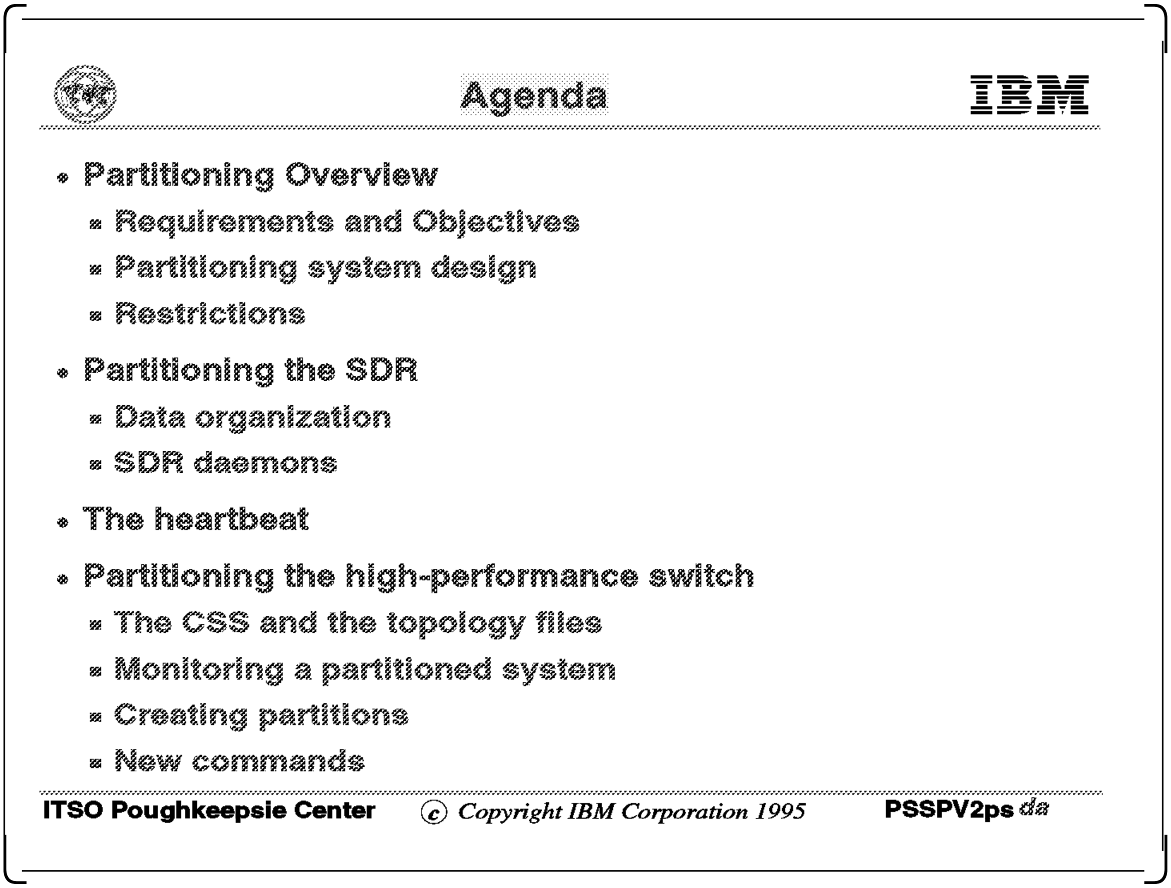
PSSP Version 2 Partitioning

ITSO Poughkeepsie Center   (c) Copyright IBM Corporation 1995   PSSPV2ps

Part 3, "PSSP V2 Partitioning" describes the new system partitioning feature available in PSSP V2, which allows the RS/6000 SP adminsitrator to define and manage logical partitions.

Following chapters describe the design, the components and the definition of partitions.

Chapter 5. Covered Topics



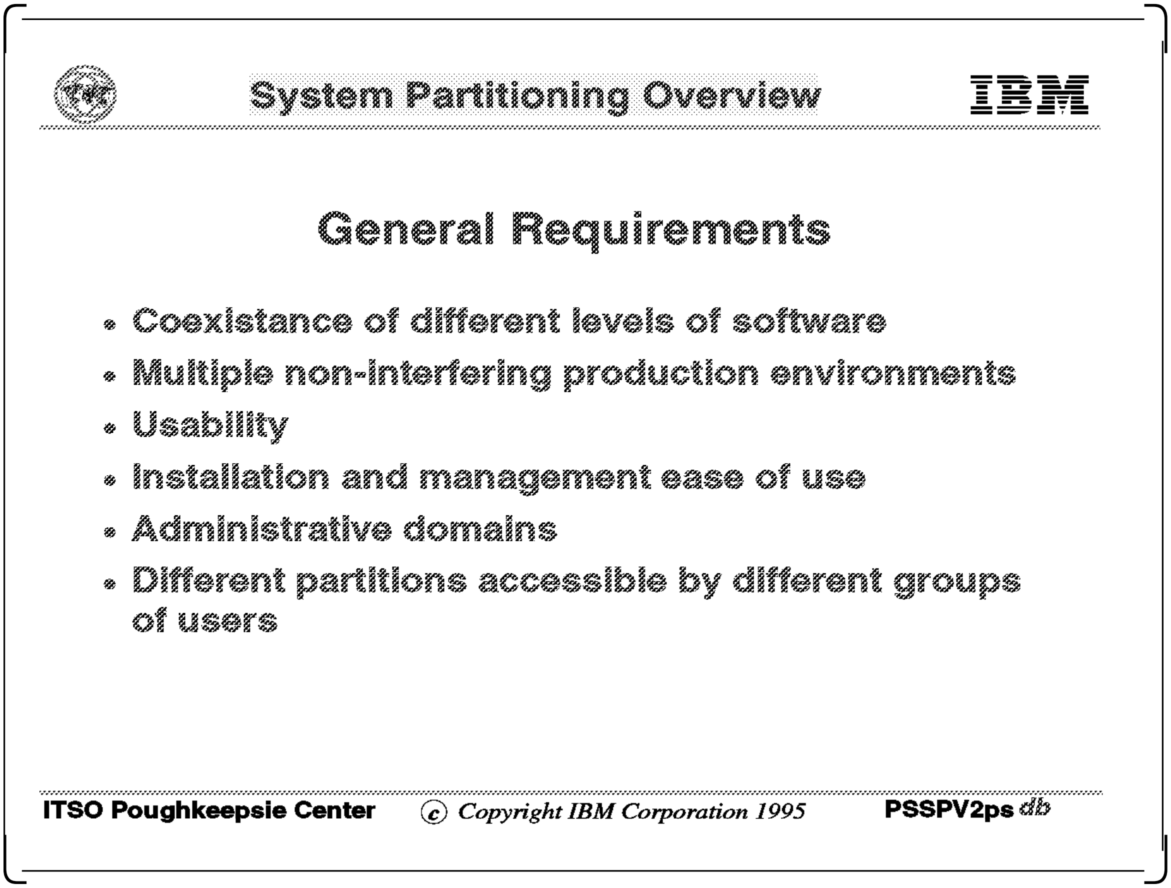
Agenda

- Partitioning Overview
  - Requirements and Objectives
  - Partitioning system design
  - Restrictions
- Partitioning the SDR
  - Data organization
  - SDR daemons
- The heartbeat
- Partitioning the high-performance switch
  - The CSS and the topology files
  - Monitoring a partitioned system
  - Creating partitions
  - New commands

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ps da

With the term *partitioning*, we mean the capability of splitting the resources of a computing system for various purposes, in particular to facilitate management functions. The structure of the RS/6000 SP system provides the opportunity to partition the system into groups of nodes. In this chapter we discuss the PSSP implementation of *system partitioning*.

Chapter 6. General Requirements



System Partitioning Overview

General Requirements

- Coexistance of different levels of software
- Multiple non-interfering production environments
- Usability
- Installation and management ease of use
- Administrative domains
- Different partitions accessible by different groups of users

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ps *db*

The following different requirements lead to the idea of *system partition*:

- The possibility of running different levels of software on logically distinct parts of the system.

  Different levels of software means a different level of the operating system, of the SP software, of the LPPs or any other applications. The following two advantages came from this approach:

  - New levels of software can be tested on a specific partition without affecting the production workload running on the other partitions.
  - A migration path is provided to gradually upgrade software.

- The possibility of setting up multiple production or development environments, without them interfering with one another.

  The workload running on a partition should not affect the workload running on the others. Also, failure conditions or maintenance operations within a partition should not affect the other partitions.

- The system administrator must be able to create, delete or change partitions.

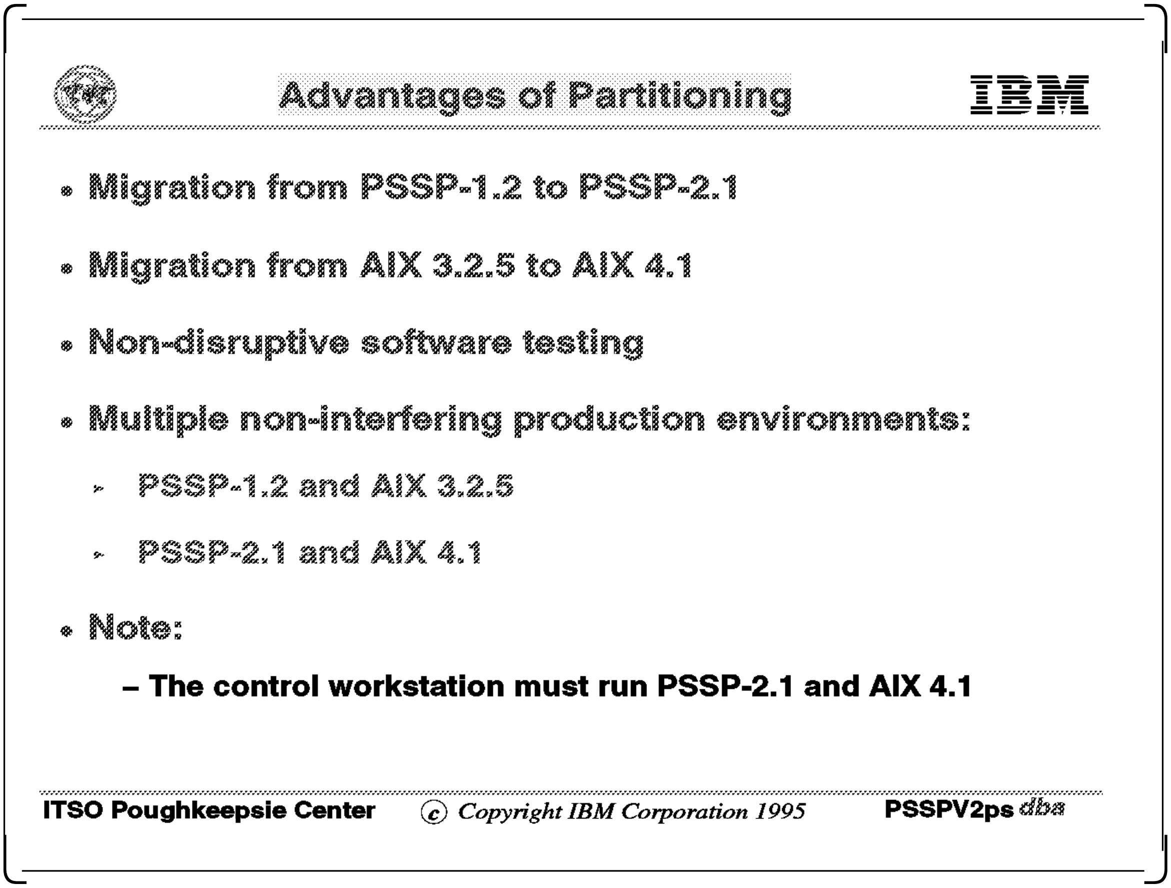
Each partition looks like a single SP system, and the management and configuration tasks can be performed either on the whole system or on a specific partition.

- The possibilty of installing, managing and customizing different partitions as if they were distinct SP systems.

  This implies, for example, customizing a set of nodes with a specific combination of software, or monitoring a single partition separately from the other parts of the system.

- The possibilty of setting up multiple administrative domains, and assigning the authority over each domain to different system administrators, who typically belong to different departments.
- The possibility of defining users, groups and account workspaces specific to each partition. This not only means that some users are not allowed to access some nodes, but that some users are only defined in a given partition, as well as their workspace.

6.1 System Partitioning Objectives



Advantages of Partitioning

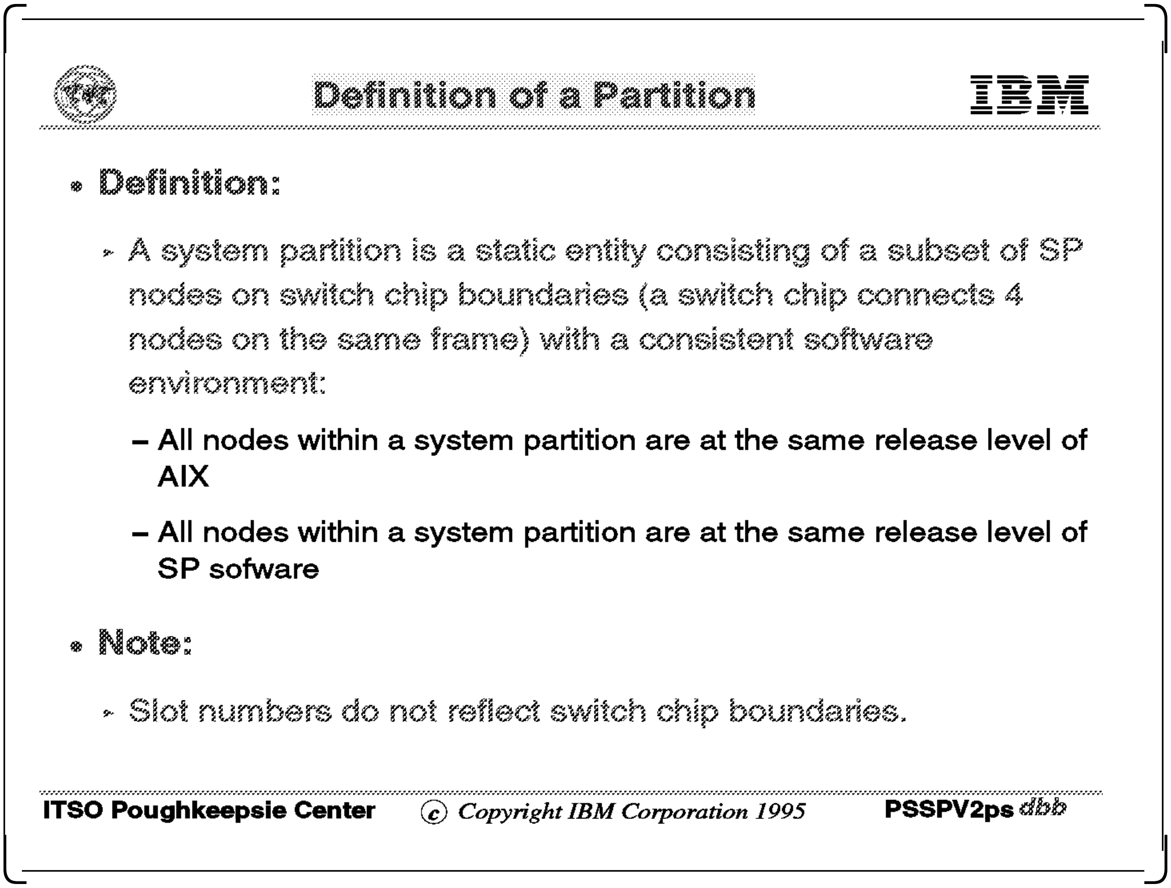
- Migration from PSSP-1.2 to PSSP-2.1
- Migration from AIX 3.2.5 to AIX 4.1
- Non-disruptive software testing
- Multiple non-interfering production environments:
  - PSSP-1.2 and AIX 3.2.5
  - PSSP-2.1 and AIX 4.1
- Note:
  - **The control workstation must run PSSP-2.1 and AIX 4.1**

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* PSSPV2ps dba

The following are the issues addressed by the PSSP implementation of the *system partition*:

- Provide the possibility:
  - To run different levels of software in different environments
  - To set up different production environments
  - To monitor separately different parts of the system
  - To isolate the switch traffic in different parts of the system
- Provide a migration path. It is basically intended for systems running AIX 3.2.5 and PSSP 1.2 that must be upgraded to AIX 4.1.3 and PSSP 2. Many customers are expected to require such a feature, since many licensed products are not yet available on AIX 4.1.3, as well as many customer applications still must be ported to the new AIX.
- No hardware modifications should be done to the nodes running the old software. PTFs must be applied to the SP nodes if they will continue to run AIX 3.2.5. Also, no further customization should be required.

6.2 Definitions



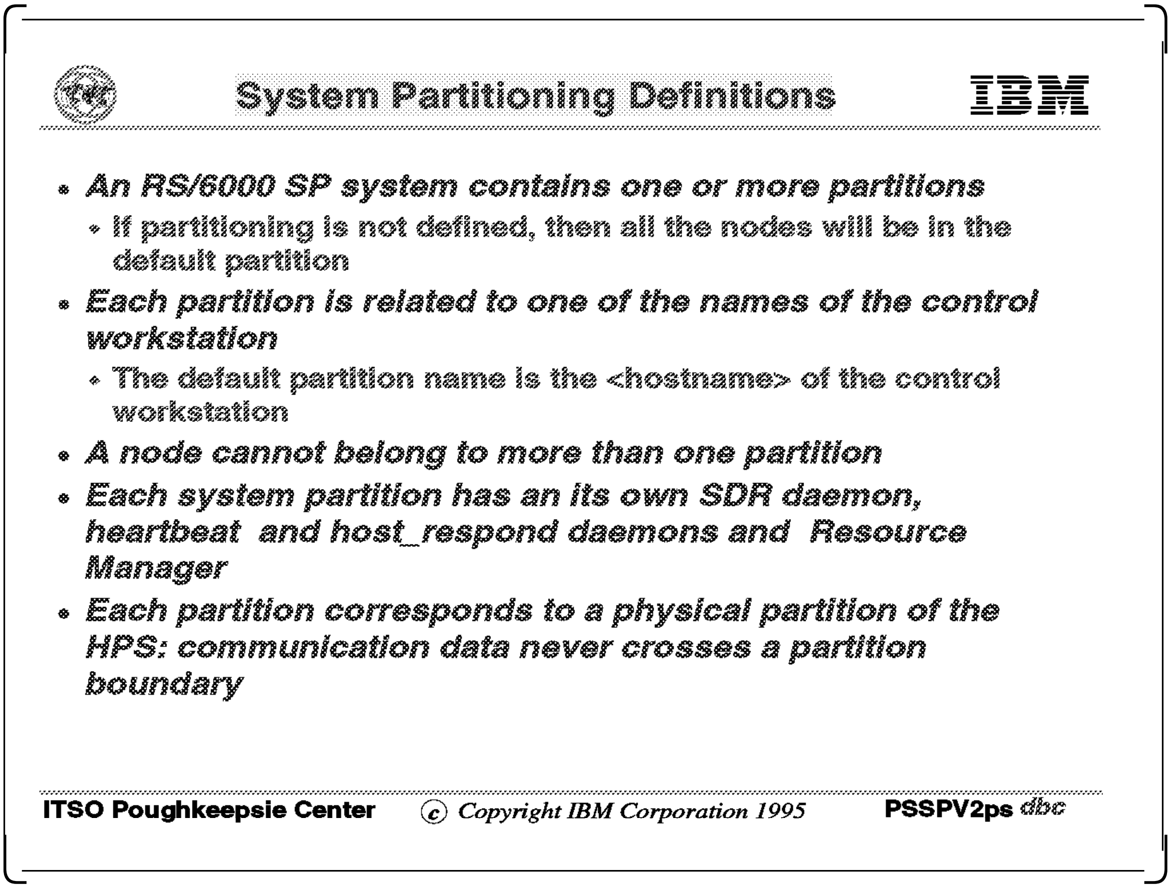
Definition of a Partition

- **Definition:**
  - A system partition is a static entity consisting of a subset of SP nodes on switch chip boundaries (a switch chip connects 4 nodes on the same frame) with a consistent software environment:
    - All nodes within a system partition are at the same release level of AIX
    - All nodes within a system partition are at the same release level of SP sofware
- **Note:**
  - Slot numbers do not reflect switch chip boundaries.

ITSO Poughkeepsie Center ⓒ *Copyright IBM Corporation 1995* PSSPV2ps *dbb*

- The level of AIX and PSSP must be consistent within a partition.
- Partitioning the system means also partitioning the High Performance Switch, so that all data exchanges over the HPS network will be actually contained within partitions. This prevents nodes connected to the same switch chip to belong to different partitions. The number of possible configurations is limited, and configuration files for all the allowed configurations are provided (this will be explained in detail in 8.4, “The CSS and the Topologies” on page 114). These files reflect the hardware connections as they have been setup at the hardware installation time by the CEs.

  Note: The clock source is still unique for a whole HPS system.
- Partitioning is intended to be almost static: it′s not feasible to change the partitions′ set up depending on the workload progress on the machine or on a time basis. The system administrator should monitor the system occupancy by the users in advance and then plan carefully for partitioning.



System Partitioning Definitions

- *An RS/6000 SP system contains one or more partitions*
  - If partitioning is not defined, then all the nodes will be in the default partition
- *Each partition is related to one of the names of the control workstation*
  - The default partition name is the <hostname> of the control workstation
- *A node cannot belong to more than one partition*
- *Each system partition has an its own SDR daemon, heartbeat and host_respond daemons and Resource Manager*
- *Each partition corresponds to a physical partition of the HPS: communication data never crosses a partition boundary*

ITSO Poughkeepsie Center ⓒ *Copyright IBM Corporation 1995* **PSSPV2ps** *dbc*

- Each partition in the system has a specific name, which corresponds to one of the names/addresses of the control workstation over the primary network adapter. If you do not set up any partitions, all the nodes will be in the default partition, whose name is the *hostname* of the control workstation.

- To make each partition look like a single SP system, some functions must be replicated on the control workstation. There are as many sdrd, hbd, hrd daemons running on the control workstation as there are partitions you set up. Each daemon monitors and serves a single partition, and any requests coming from a node are sent to the daemons responsible for the partition the node belongs to. The same happens for the Resource Manager daemons.

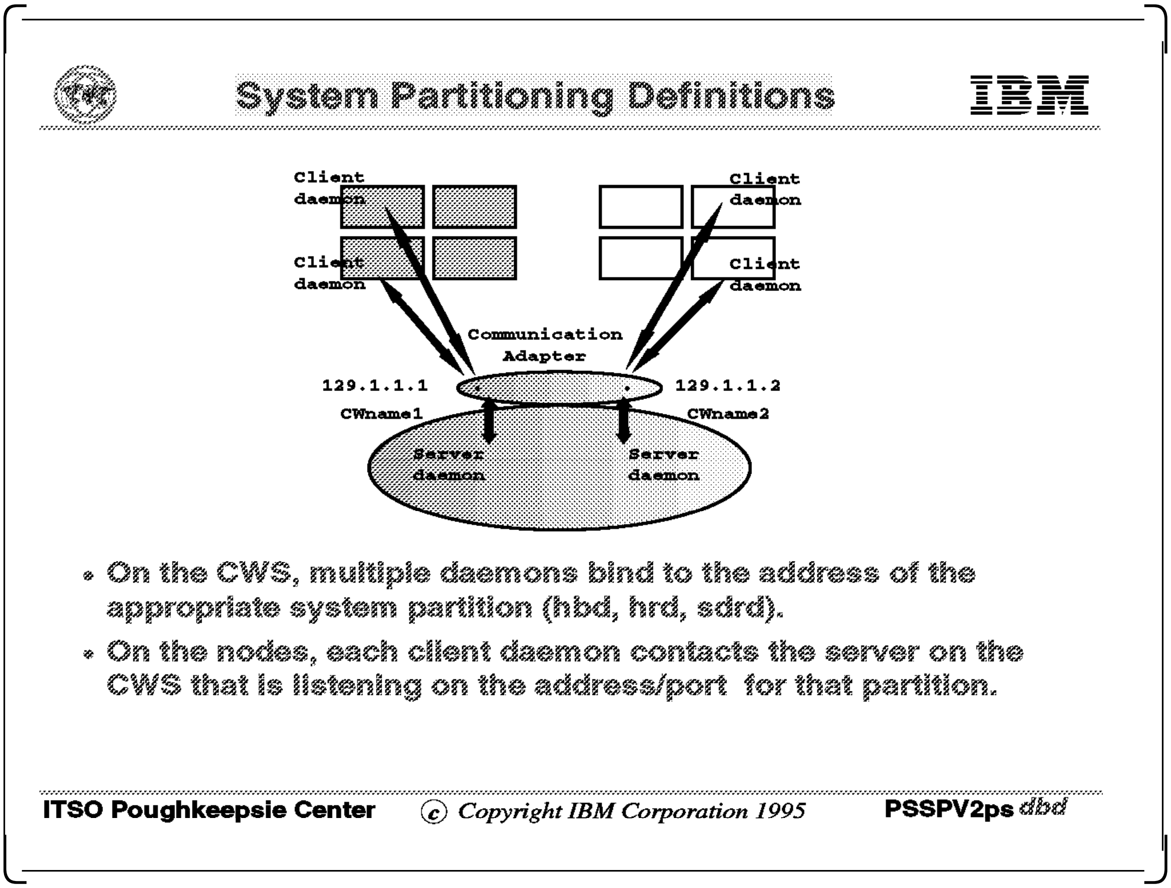
  Substantial modifications have been required both to the SDR data organization and to the SDR, hbd, hdr, jmd daemons. We are going to describe the new implementation of each component in the following sections.

Notes:

1. The sdr daemon is responsible for managing the SP database, the SDR, which contains all the configuration information of the system. Requests to update the SDR (coming from the system administrator, when he invokes an SP command, or from other daemons) are addressed to the sdr daemon. It also replies to information requests coming from other daemons, commands or applications.

2. The hbd daemon monitors the status of nodes and of the CWS via the SP Ethernet. It provides this information to the **sdrd**, to the host_respond daemon, and to the daemons serving the VSD.

3. The host_respond daemon (**hrd**) directly updates the host_responds class in the SDR.

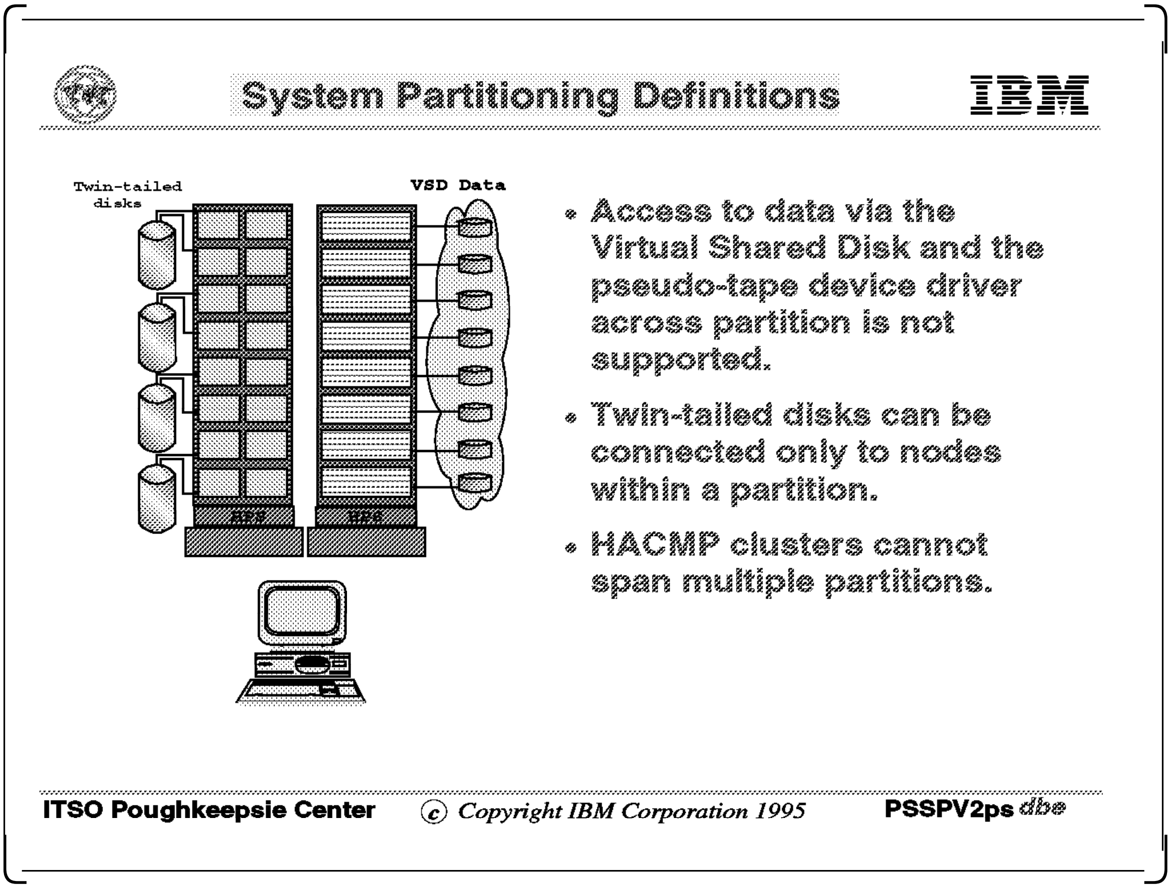


System Partitioning Definitions

- On the CWS, multiple daemons bind to the address of the appropriate system partition (hbd, hrd, sdrd).
- On the nodes, each client daemon contacts the server on the CWS that is listening on the address/port for that partition.

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ps dbd

- On the CWS, the sdr, the heartbeat and the host_responds systems provide the capability for multiple daemons (one for each system partition) to coexist on the control workstation.
- On the nodes, the hbd daemon as well as client applications are able to connect to the daemons responsible for the partiton the node belongs to.



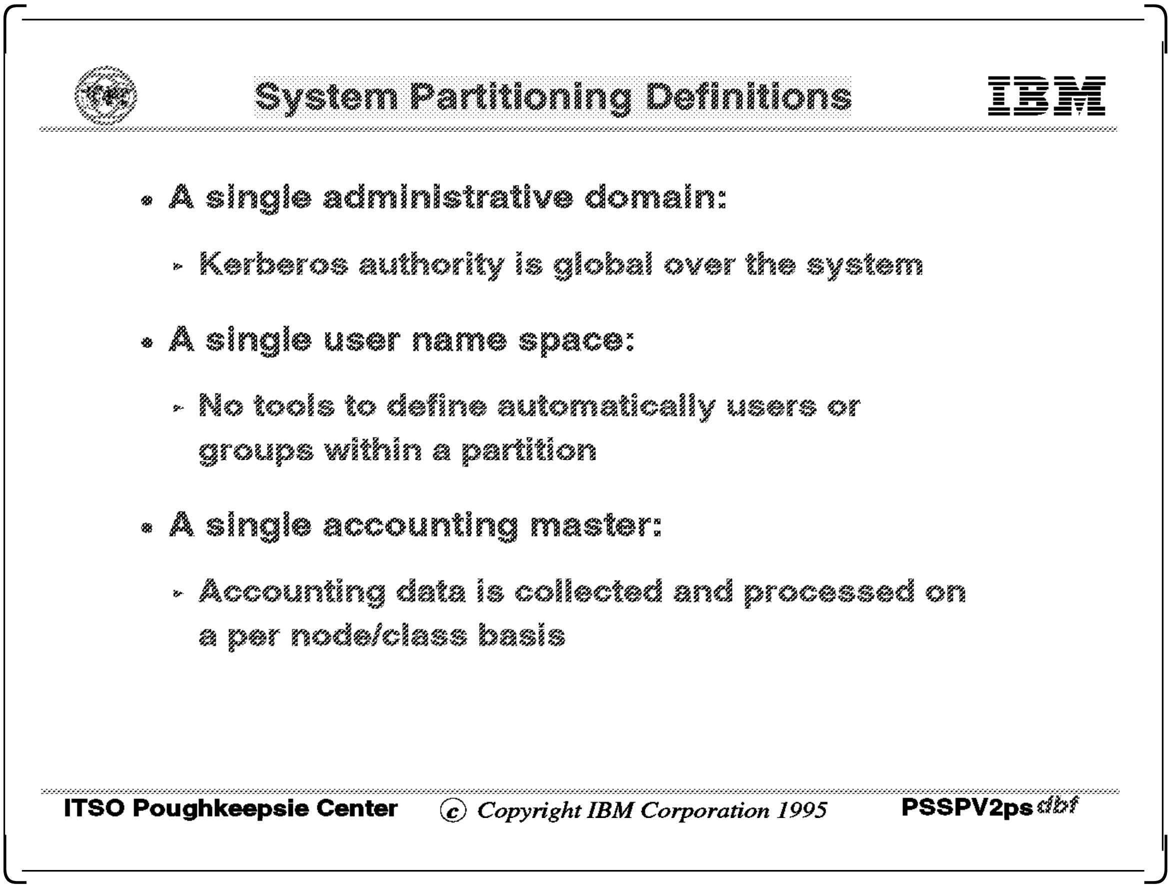
IBM
Twin-tailed disks
VSD Data
HPS
HPS

System Partitioning Definitions

- Access to data via the Virtual Shared Disk and the pseudo-tape device driver across partition is not supported.
- Twin-tailed disks can be connected only to nodes within a partition.
- HACMP clusters cannot span multiple partitions.

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ps dbe

The items listed above could be considered either *definitions* or *restrictions*. Actually, they seem to be consistent with the System Partitioning design. If you split the system into almost independent subsystems, to run different applications environments, you probably do not require features such as VSD data sharing or logical volumes definitions across partitions.



System Partitioning Definitions

- A single administrative domain:
  - Kerberos authority is global over the system
- A single user name space:
  - No tools to define automatically users or groups within a partition
- A single accounting master:
  - Accounting data is collected and processed on a per node/class basis

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ps dbf

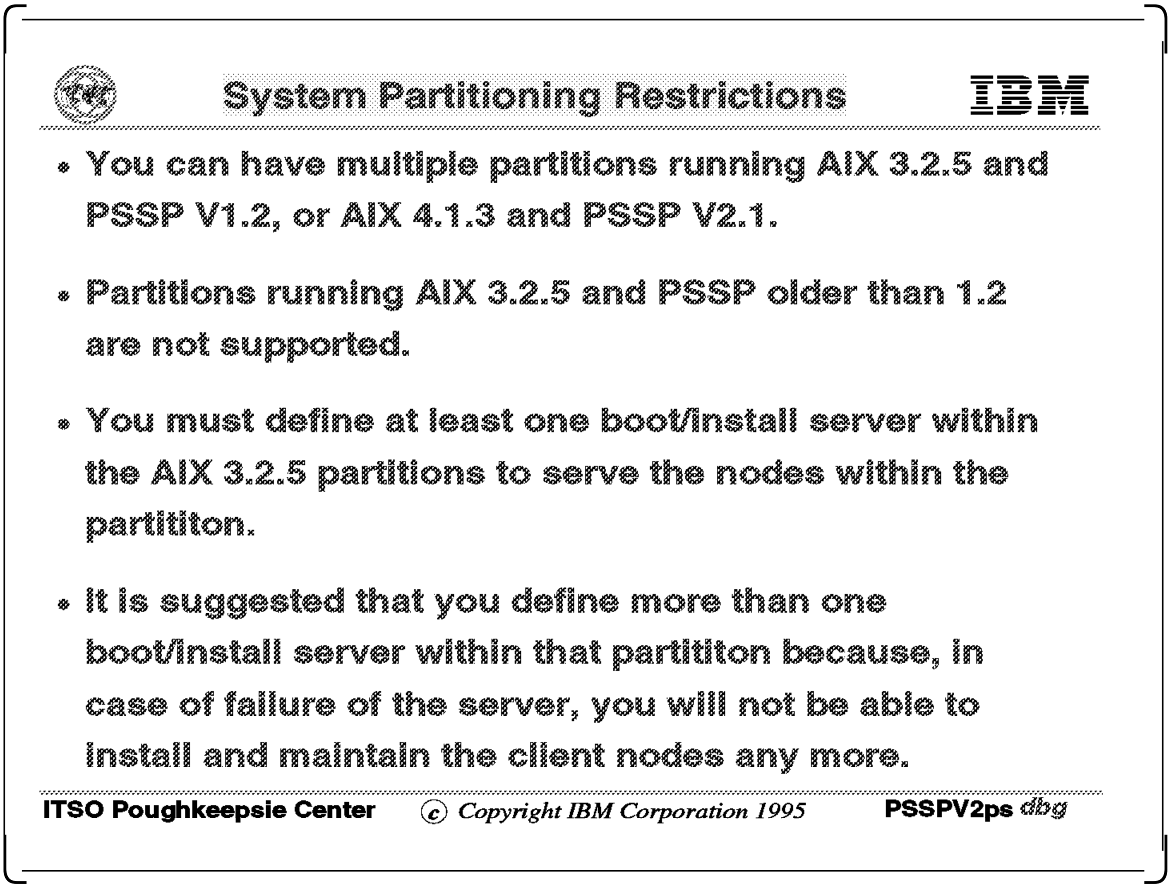
- Kerberos authority is global to the system. This means that you don′t have the capability to set up multiple independent administrative domains, and assign domains to different system administrators.

  You can (as with the previous version of PSSP) add administrative users to the Kerberos database and allow them to access the *hardmon* and *rcmd* services with same permissions as root, but then all of them will be able to access the services on a system wide basis.

- One user name space in the system means that partitioning does not affect the groups/users management.

  You can configure either your NIS environment or the File Collections, or you can use the *login control* tool to restrict users′ access to subsets of nodes, but this kind of setup is strictly related to the *nodes* rather than to a *partition*. For example, if you decide to change the machine partitioning, you also have to modify the user management configuration to reflect the new situation

- Again, no tools are provided to collect accounting data on a partition basis. You can still assign nodes belonging to the same partition to the same accounting class in order to distinguish resource consumption on different sets of nodes, but you are responsible for doing that (it is not automatic when you create a partition). You are also responsible for changing the accounting setup, if necessary, when you change the system partitioning configuration.



System Partitioning Restrictions

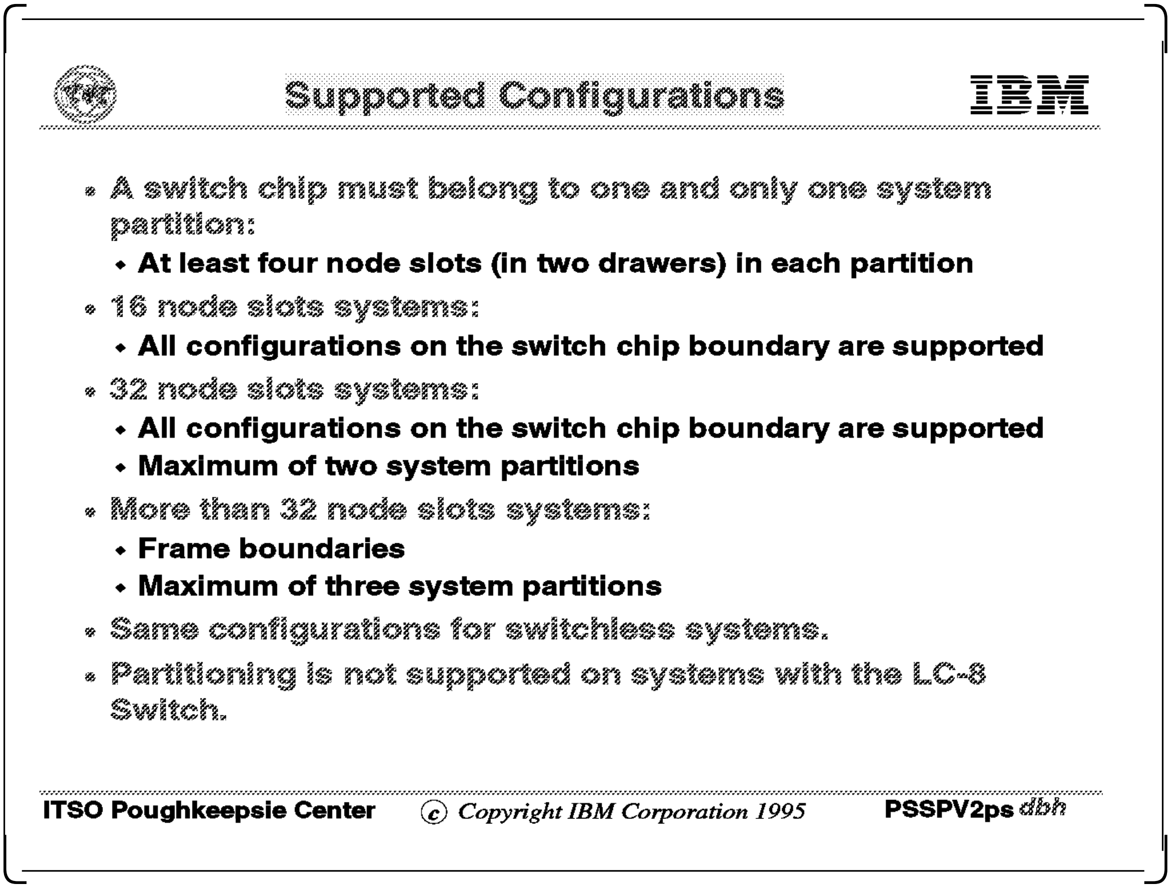
- You can have multiple partitions running AIX 3.2.5 and PSSP V1.2, or AIX 4.1.3 and PSSP V2.1.
- Partitions running AIX 3.2.5 and PSSP older than 1.2 are not supported.
- You must define at least one boot/install server within the AIX 3.2.5 partitions to serve the nodes within the partititon.
- It is suggested that you define more than one boot/install server within that partititon because, in case of failure of the server, you will not be able to install and maintain the client nodes any more.

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* PSSPV2ps *dbg*

- On the CWS only one heartbeat daemon can run in compatibility mode and talk to PSSP 1.2 daemons.
- Since the CWS is running AIX 4.1.3, you cannot perform network installations from the CWS to the AIX 3.2.5 nodes; you need an installation server in the AIX 3.2.5 partitions.

  If you have more than one installation server within the partition you are guaranteed in case of failure of one of them, and you still have the possibility to reinstall any of them whenever you need to do that.

6.3 Supported Configurations



Supported Configurations

- A switch chip must belong to one and only one system partition:
  - **At least four node slots (in two drawers) in each partition**
- 16 node slots systems:
  - **All configurations on the switch chip boundary are supported**
- 32 node slots systems:
  - **All configurations on the switch chip boundary are supported**
  - **Maximum of two system partitions**
- More than 32 node slots systems:
  - **Frame boundaries**
  - **Maximum of three system partitions**
- Same configurations for switchless systems.
- Partitioning is not supported on systems with the LC-8 Switch.

ITSO Poughkeepsie Center ⓒ *Copyright IBM Corporation 1995* **PSSPV2ps** *dbh*

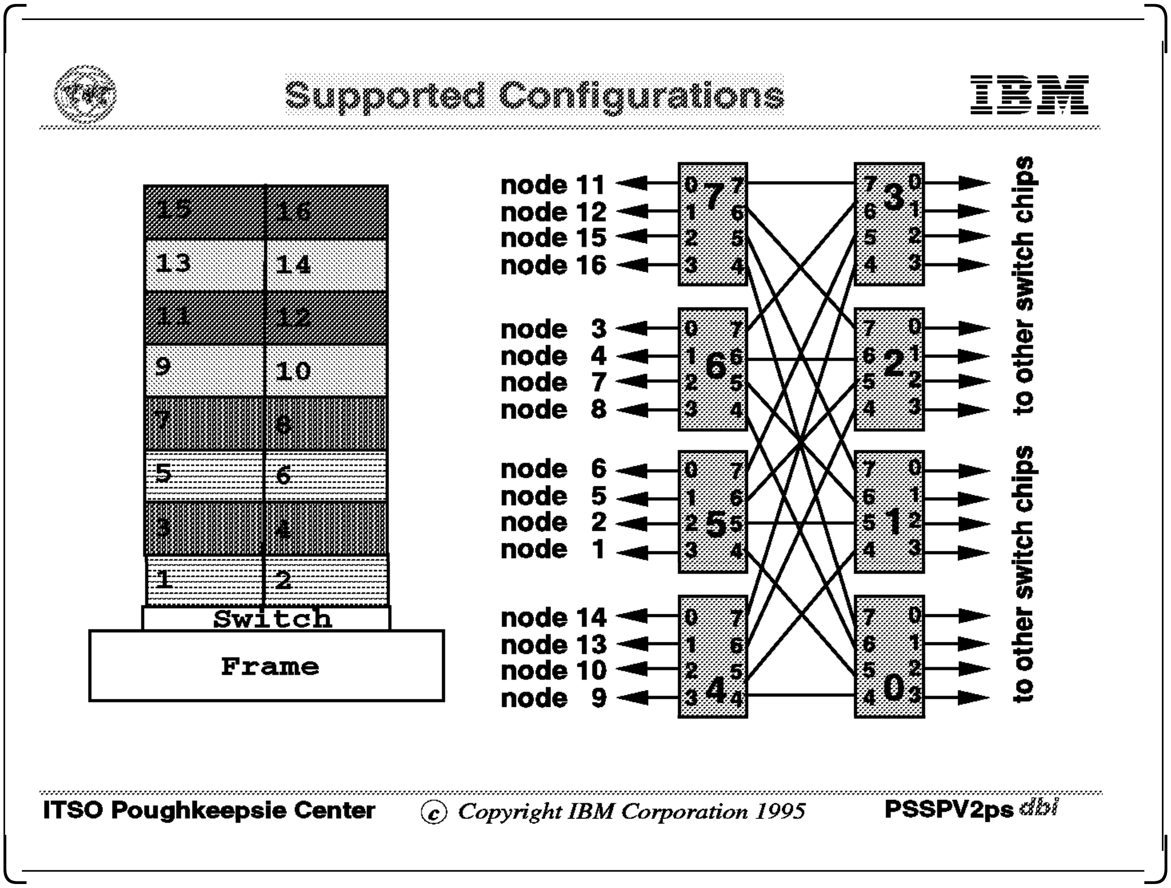
Partitioning the HPS is a critical issue. Not all possible configurations are supported; both performance and reliability factors have been taken into account when the predefined configurations have been prepared.

- The switch chip boundary requirement guarantees that network traffic over the switch in one partition does not interfere with traffic in another partition.
- On a 16 slot system, all configurations on a switch chip boundary are supported. Depending on the configuration, you could experience slightly worse performance over the HPS. One example is the 8-node partition in a 4_4_8 configuration. This happens because the internal links switch redundancy in that partition is only half the redundancy you have on a non-partitioned system (usually you have at least four paths between any pair of nodes, now it is only two). However the performance decrease is expected to be quite small (10-15 %) on average.
- On larger systems, only two (32-way system) or three (more that 32-way system) system partitions are supported so far (the amount of possible configurations and layouts would be unmanageable for those system).

  System partitioning configurations, including more partitions than those supported, can be available as an RPQ, and in that case performance and reliability implications may be discussed directly with the developers.
- The same predefined configurations apply to the switchless systems. Adding a switch in the future does not imply rebuilding the system partitioning.

- Partitioning is not supported over the light switch because it consists of a single chip.

6.4 Supported Configurations



Supported Configurations

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ps dbi

This picture shows how nodes are connected to the HPS in a frame. Ports on the right side are used to connect the switch board to other switch boards in multi-frame systems (they are unused in a single frame system).

The arrangement of cables is the reason why nodes plugged into non-contiguous drawers are connected to the same switch. Cables coming from odd drawers reach the bottom of the frame on the left, while cables coming from even drawers go along the right side.

We include here the annotated topology file for a 16-way system, describing how the node switch adapters are connected to the switch ports, how the switch chips are connected to each other, and which ports are unused.

```
# Node to Switch Connections

  Switch board
  |Switch chip
  || Chip port
  || |   Switch node_number           node_number
  || |      |                              |
  || |      |           N. of connector the cable is plugged into
  || |      |                   |          |
s 15 3  tb0 0 0     L01-S00-BH-J18 to L01-N1
s 15 2  tb0 1 0     L01-S00-BH-J16 to L01-N2
```

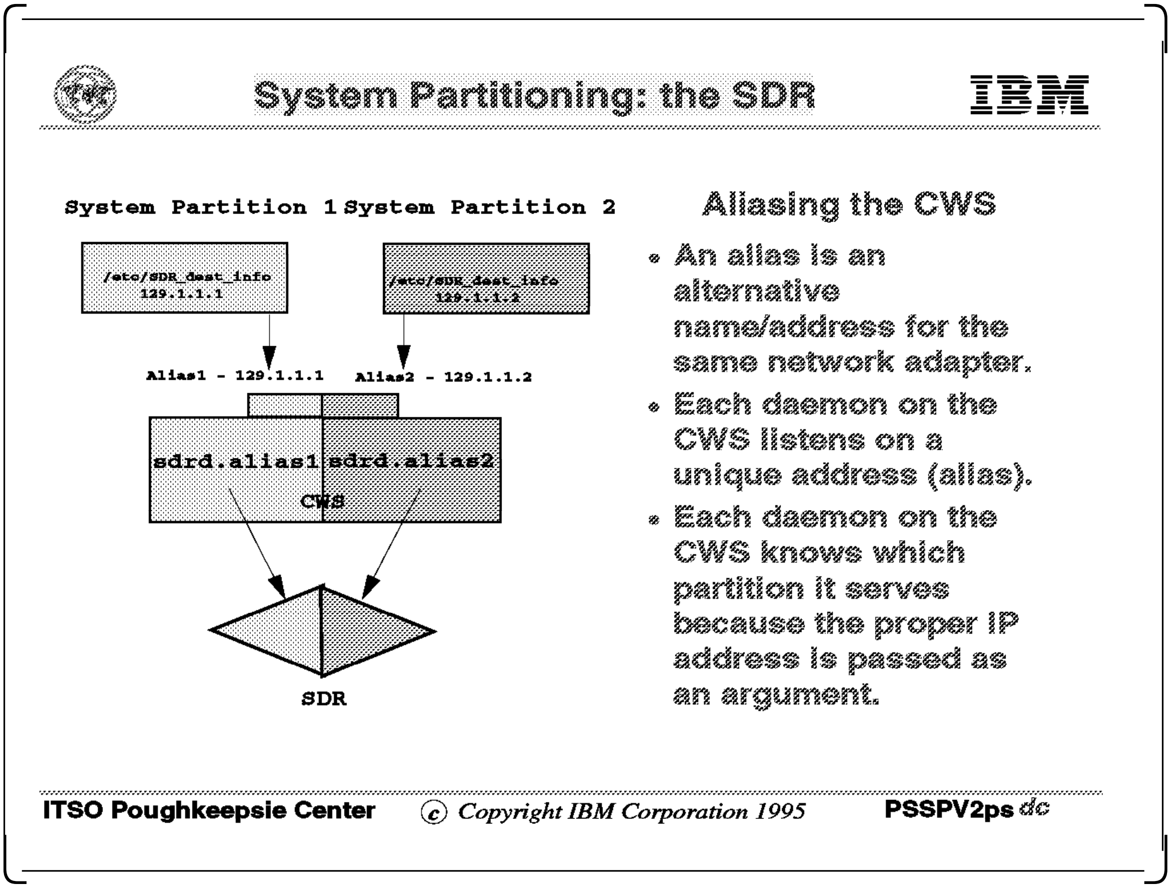
```
s 16 0  tb0 2 0     L01-S00-BH-J20 to L01-N3
s 16 1  tb0 3 0     L01-S00-BH-J22 to L01-N4
s 15 1  tb0 4 0     L01-S00-BH-J14 to L01-N5
s 15 0  tb0 5 0     L01-S00-BH-J12 to L01-N6
s 16 2  tb0 6 0     L01-S00-BH-J24 to L01-N7
s 16 3  tb0 7 0     L01-S00-BH-J26 to L01-N8
s 14 3  tb0 8 0     L01-S00-BH-J10 to L01-N9
s 14 2  tb0 9 0     L01-S00-BH-J8  to L01-N10
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 14 1  tb0 12 0    L01-S00-BH-J6  to L01-N13
s 14 0  tb0 13 0    L01-S00-BH-J4  to L01-N14
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

  Switch board
  |Switch chip
  || Chip port
  || |
  || |      Switch board
  || |      |Switch chip
  || |      || Chip port
  || |      || |
s 14 7    s 13 4    L01-S00-SC
s 14 6    s 12 4    L01-S00-SC
s 14 5    s 11 4    L01-S00-SC
s 14 4    s 10 4    L01-S00-SC
s 15 7    s 13 5    L01-S00-SC
s 15 6    s 12 5    L01-S00-SC
s 15 5    s 11 5    L01-S00-SC
s 15 4    s 10 5    L01-S00-SC
s 16 7    s 13 6    L01-S00-SC
s 16 6    s 12 6    L01-S00-SC
s 16 5    s 11 6    L01-S00-SC
s 16 4    s 10 6    L01-S00-SC
s 17 7    s 13 7    L01-S00-SC
s 17 6    s 12 7    L01-S00-SC
s 17 5    s 11 7    L01-S00-SC
s 17 4    s 10 7    L01-S00-SC

# L01 switch 1 wrapped ports

s 13 3 s 13 3     L01-S00-BH-J3  to L01-S00-BH-J3
s 13 2 s 13 2     L01-S00-BH-J5  to L01-S00-BH-J5
s 13 1 s 13 1     L01-S00-BH-J7  to L01-S00-BH-J7
s 13 0 s 13 0     L01-S00-BH-J9  to L01-S00-BH-J9
s 12 3 s 12 3     L01-S00-BH-J11 to L01-S00-BH-J11
s 12 2 s 12 2     L01-S00-BH-J13 to L01-S00-BH-J13
s 12 1 s 12 1     L01-S00-BH-J15 to L01-S00-BH-J15
s 12 0 s 12 0     L01-S00-BH-J17 to L01-S00-BH-J17
s 11 3 s 11 3     L01-S00-BH-J19 to L01-S00-BH-J19
s 11 2 s 11 2     L01-S00-BH-J21 to L01-S00-BH-J21
s 11 1 s 11 1     L01-S00-BH-J23 to L01-S00-BH-J23
s 11 0 s 11 0     L01-S00-BH-J25 to L01-S00-BH-J25
s 10 3 s 10 3     L01-S00-BH-J27 to L01-S00-BH-J27
s 10 2 s 10 2     L01-S00-BH-J29 to L01-S00-BH-J29
```

```
s 10 1 s 10 1     L01-S00-BH-J31 to L01-S00-BH-J31
s 10 0 s 10 0     L01-S00-BH-J33 to L01-S00-BH-J33
```

In case you need to change the connections to the HPS (for example, one switch port is out of service), you also have to change the topology file for that partition and the ODM attributes on the nodes. You can create a modified copy of the topology file and place it in the */etc/SP* directory on the primary node. In fact, if a */etc/SP/expected.top* file exists on the primary node, then it overrides the topology information that is stored into the SDR.

When changing configurations, you must make sure that you remove the */etc/SP/expected.top* files from the old primary nodes.

Chapter 7. Partitioning the SDR



System Partitioning: the SDR

Aliasing the CWS

- An alias is an alternative name/address for the same network adapter.
- Each daemon on the CWS listens on a unique address (alias).
- Each daemon on the CWS knows which partition it serves because the proper IP address is passed as an argument.

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ps dc

The term alias is the alternate name/address for an adapter. We will use the above foil to describe how this is used.

- On the nodes, the correct path for the requests to the SDR is contained in the */etc/SDR_dest_info* file. The file actually contains two entries, as follows:

  ```
  default: <IP address>
  primary: <IP address>
  ```

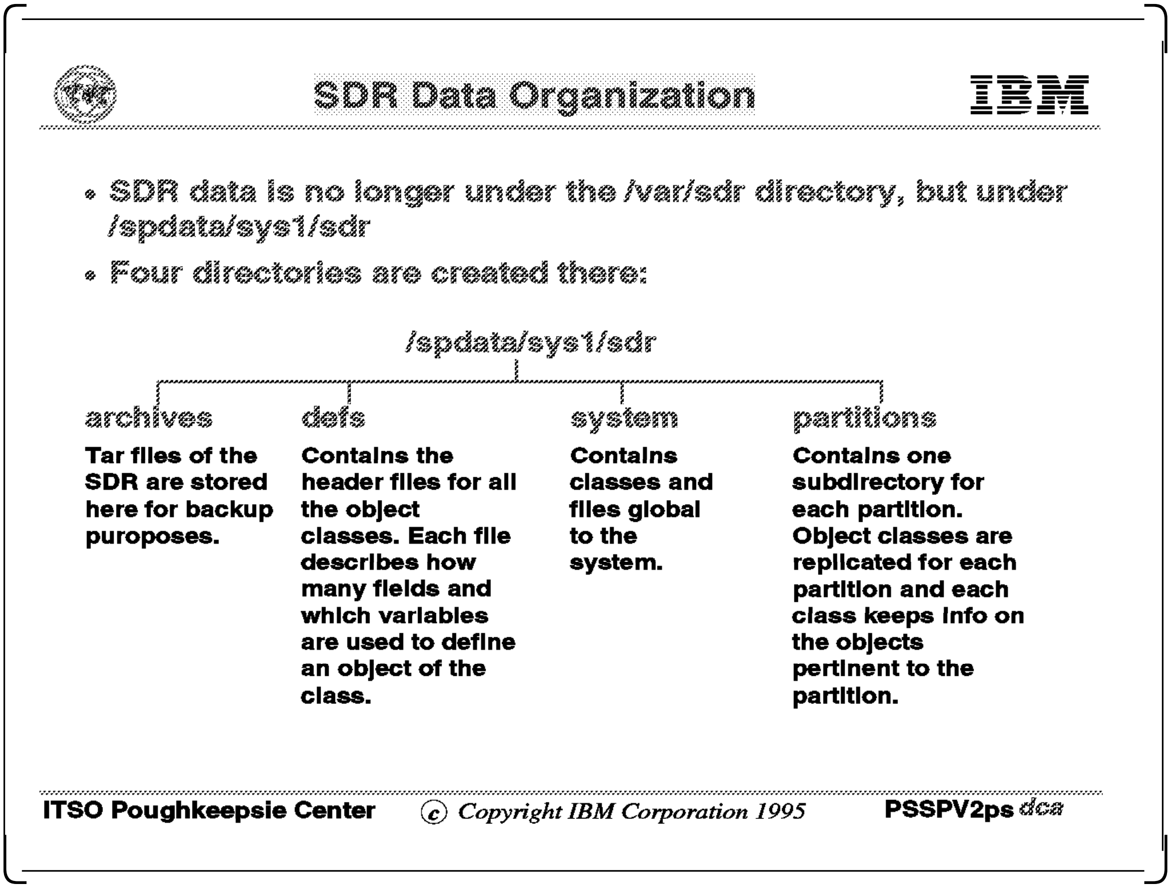
  where *primary* is the IP address (alias or real in the case of the default partition) of the specific alias for the partition, *default* is the real IP address of the control workstation and the address of the default partition. It is used in case the primary destination identifier is no longer valid (for example, the partition has been deleted) to find out the correct information.

  On the PSSP 1.2 nodes, with AIX 3.2.5 partitions, the */etc/SDR_dest_info* file contains the name and address of the CWS, that is, the name of the default partition, so that the path to SDR is guaranteed on those nodes.

  The SP_NAME environment variable is usually unset (you can set it to address a query or a command to a specific partition).

- On the CWS, each sdr daemon receives a parameter at startup time, that is the IP address (alias or real) or name of the CWS referring to a specific partition, so that it is able to determine which partition to operate on.

7.1 SDR Data Organization



SDR Data Organization

- SDR data is no longer under the /var/sdr directory, but under /spdata/sys1/sdr
- Four directories are created there:

/spdata/sys1/sdr

archives
Tar files of the SDR are stored here for backup puroposes.

defs
Contains the header files for all the object classes. Each file describes how many fields and which variables are used to define an object of the class.

system
Contains classes and files global to the system.

partitions
Contains one subdirectory for each partition. Object classes are replicated for each partition and each class keeps info on the objects pertinent to the partition.

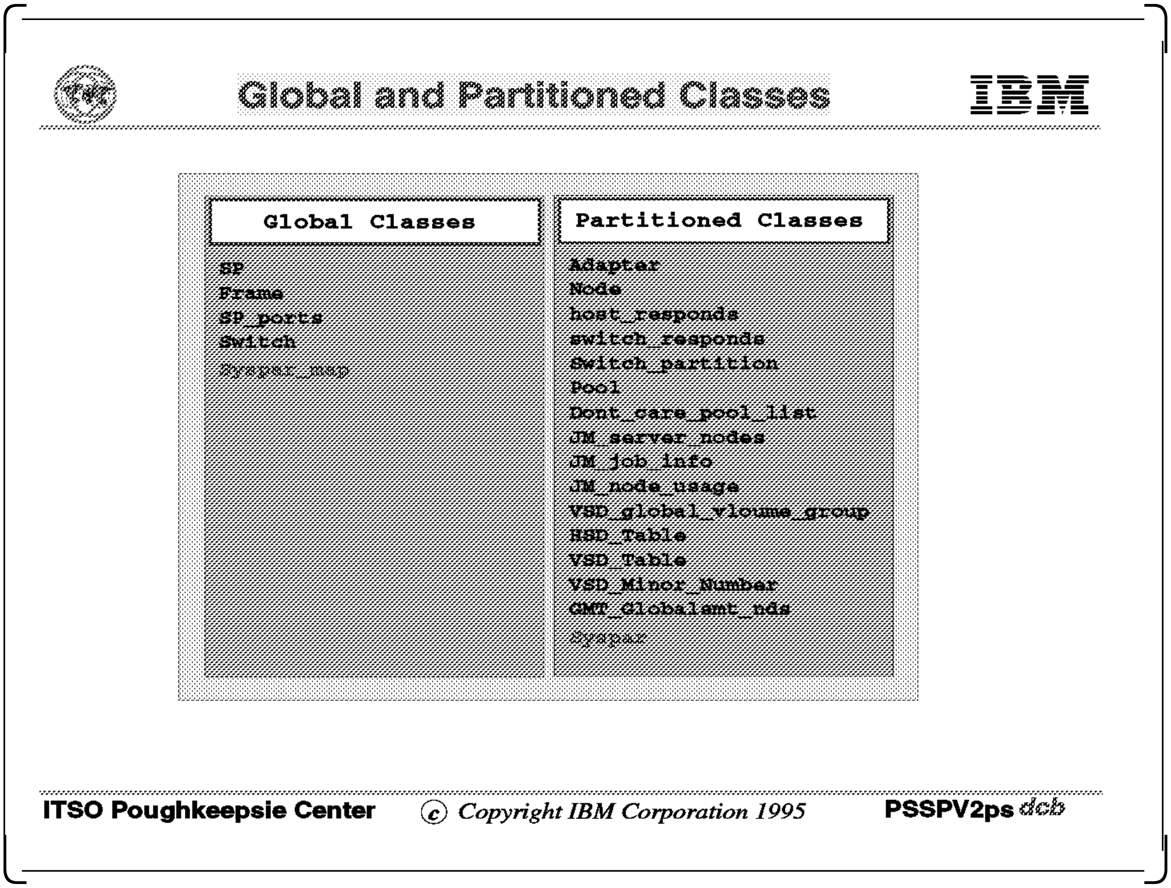
ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ps dca

The SDR contains data about the entire SP system. Most of the data is contained within a system partition, and some of the data (the data that is most pertinent to the physical configuration and to the hardware) is global data for the entire system.

The SDR is divided into object classes that are global and object classes that are partitioned, which means that for a given partitioned object class, some objects represent data associated with one system partition, and some objects represent data associated with another partition. For example, the *Node* class is divided into subsets, representing each system partition, that contain the node objects for the nodes in that particular system partition.

All global data is accessible from any system partition. Partitioned data is usually accessed within the current partition, but data queries involving other partitions are satisfied as well.

7.2 Global and Partitioned Classes



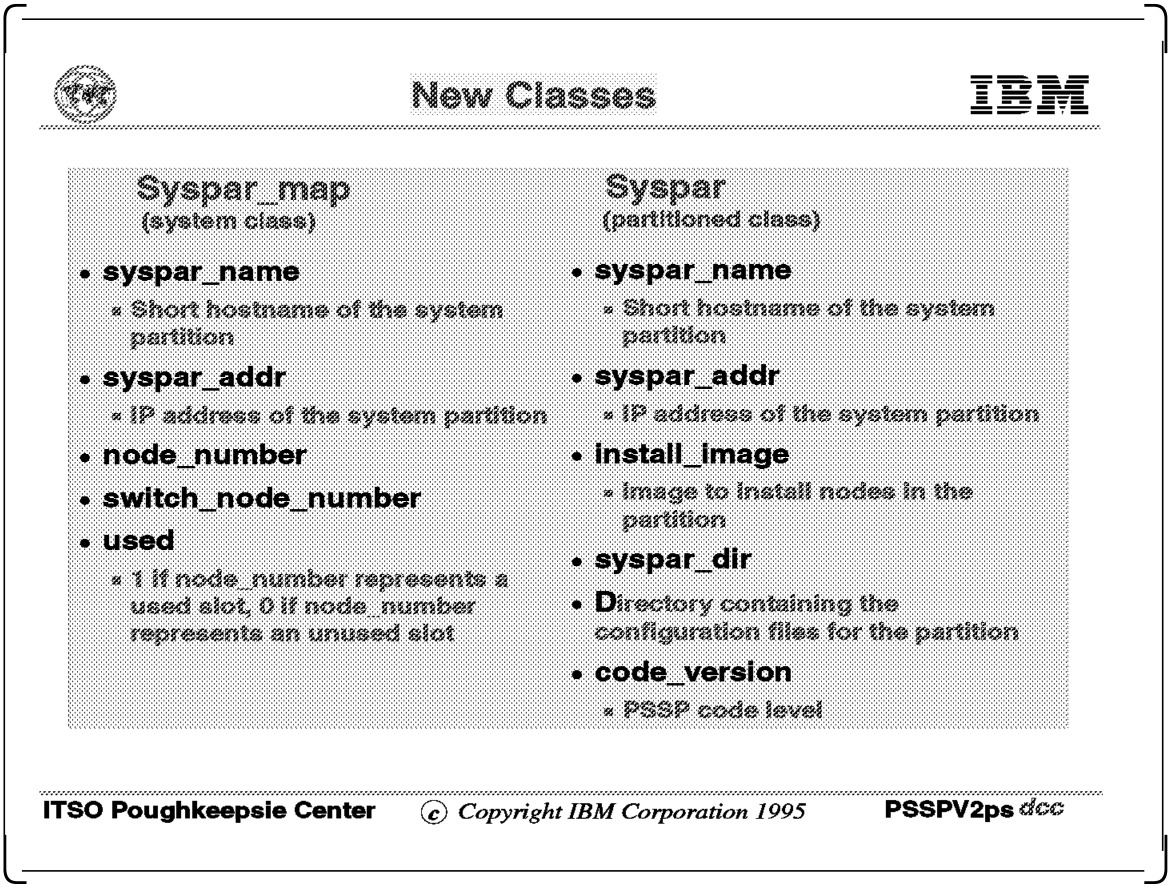
Global and Partitioned Classes

Global Classes	Partitioned Classes
SP	Adapter
Frame	Node
SP_ports	host_responds
Switch	switch_responds
Syspar_map	Switch_partition
	Pool
	Dont_care_pool_list
	JM_server_nodes
	JM_job_info
	JM_node_usage
	VSD_global_vloume_group
	HSD_Table
	VSD_Table
	VSD_Minor_Number
	GMT_Globalamt_nds
	Syspar

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* PSSPV2ps *dcb*

The table shows the global classes and the partitioned classes. There are two new classes (*Syspar_map* and *Syspar*), which are described in the next foil.

7.3 New Classes



New Classes

Syspar_map
(system class)

- **syspar_name**
  - Short hostname of the system partition
- **syspar_addr**
  - IP address of the system partition
- **node_number**
- **switch_node_number**
- **used**
  - 1 if node_number represents a used slot, 0 if node_number represents an unused slot

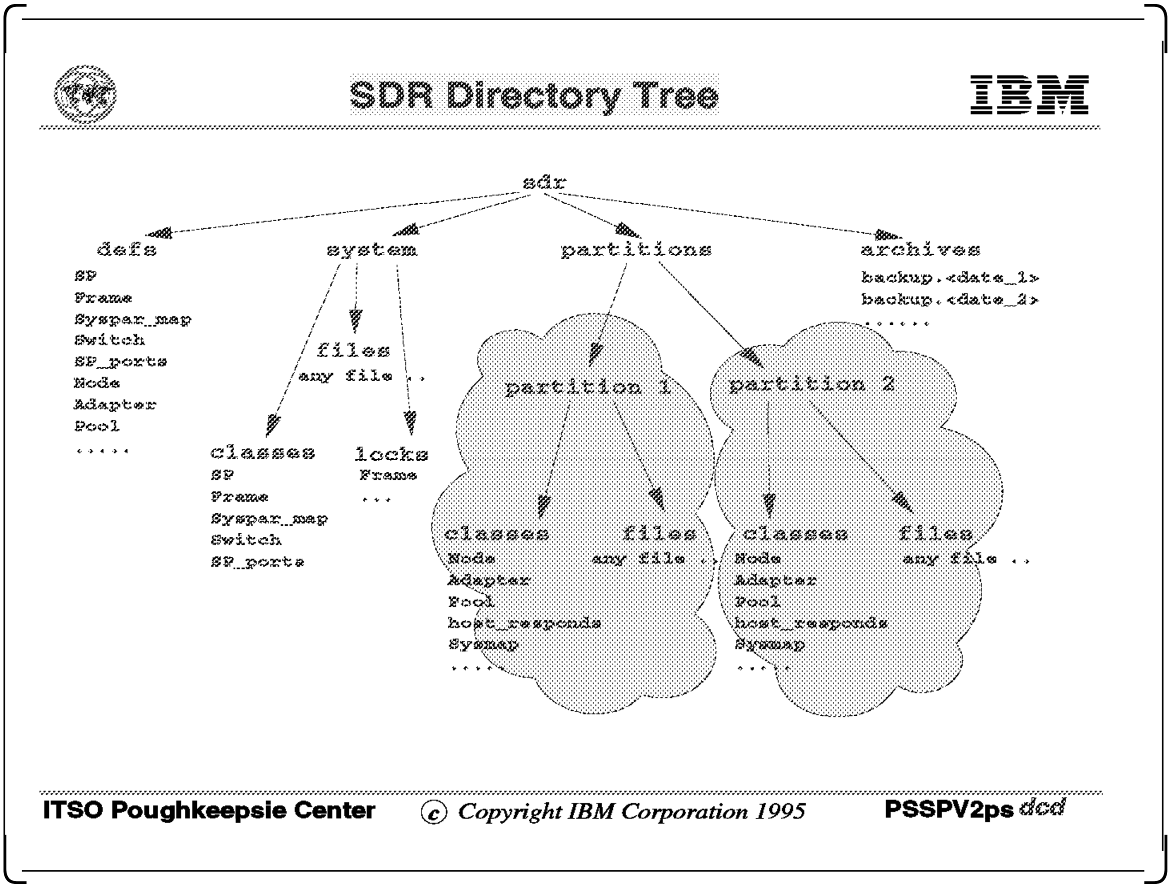
Syspar
(partitioned class)

- **syspar_name**
  - Short hostname of the system partition
- **syspar_addr**
  - IP address of the system partition
- **install_image**
  - Image to install nodes in the partition
- **syspar_dir**
- Directory containing the configuration files for the partition
- **code_version**
  - PSSP code level

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ps dcc

- The *Syspar_map* class is a global class that describes the mapping of nodes onto partitions. It contains one object for each node. For each node, the object contains the name and the IP address of the partition the node belongs to, and the node number and the switch node number and flag, indicating whether the slot is used by a real node or not (for example, even slots in drawers occupied by wide nodes are unused).
- The *Syspar* class is a partitioned class, containing a single object. Attributes of this object are the customization parameters of the partition (PSSP level, primary node, and so on).

7.4 SDR Directory Tree



SDR Directory Tree

ITSO Poughkeepsie Center ⓒ *Copyright IBM Corporation 1995* PSSPV2ps *dcd*

sdr/defs
Stores the class definition (the first line in the old sdr files in PSSP 1.2). It looks like:

```
I1=node_number I2=host_respond
```

A ″g″ prefixes the header for system classes, a ″p″ prefixes it for partition classes.

sdr/system/classes
Stores system classes. File name is same as class name.

sdr/system/files
Store system files.

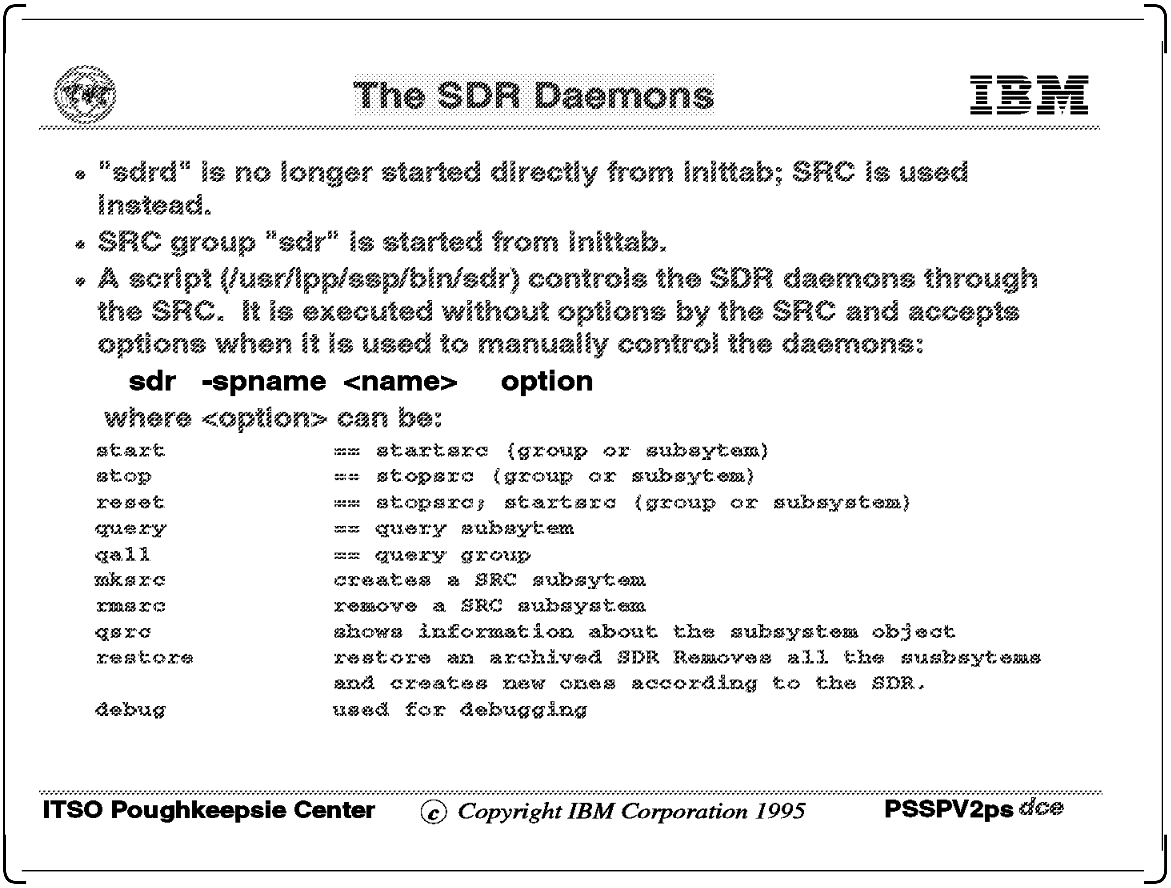
sdr/system/locks
Store persistent locks for system classes. See 7.7, “New SDR Commands” on page 102.

sdr/partitions/<IP_address>/classes
Store class data for partition whose IP address is *<IP_address>*.

sdr/partitions/<IP_address>/files
Store partition files for partition whose IP address is *<IP_address>*.

sdr/archives
Store archive files.

7.5 The SDR Daemons



The SDR Daemons

- "sdrd" is no longer started directly from inittab; SRC is used instead.
- SRC group "sdr" is started from inittab.
- A script (/usr/lpp/ssp/bin/sdr) controls the SDR daemons through the SRC. It is executed without options by the SRC and accepts options when it is used to manually control the daemons:

sdr -spname <name> option

where <option> can be:

```
start          == startsrc (group or subsytem)
stop           == stopsrc (group or subsytem)
reset          == stopsrc; startsrc (group or subsystem)
query          == query subsytem
qall           == query group
mksrc          creates a SRC subsytem
rmsrc          remove a SRC subsystem
qsrc           shows information about the subsystem object
restore        restore an archived SDR Removes all the susbsytems
               and creates new ones according to the SDR.
debug          used for debugging
```

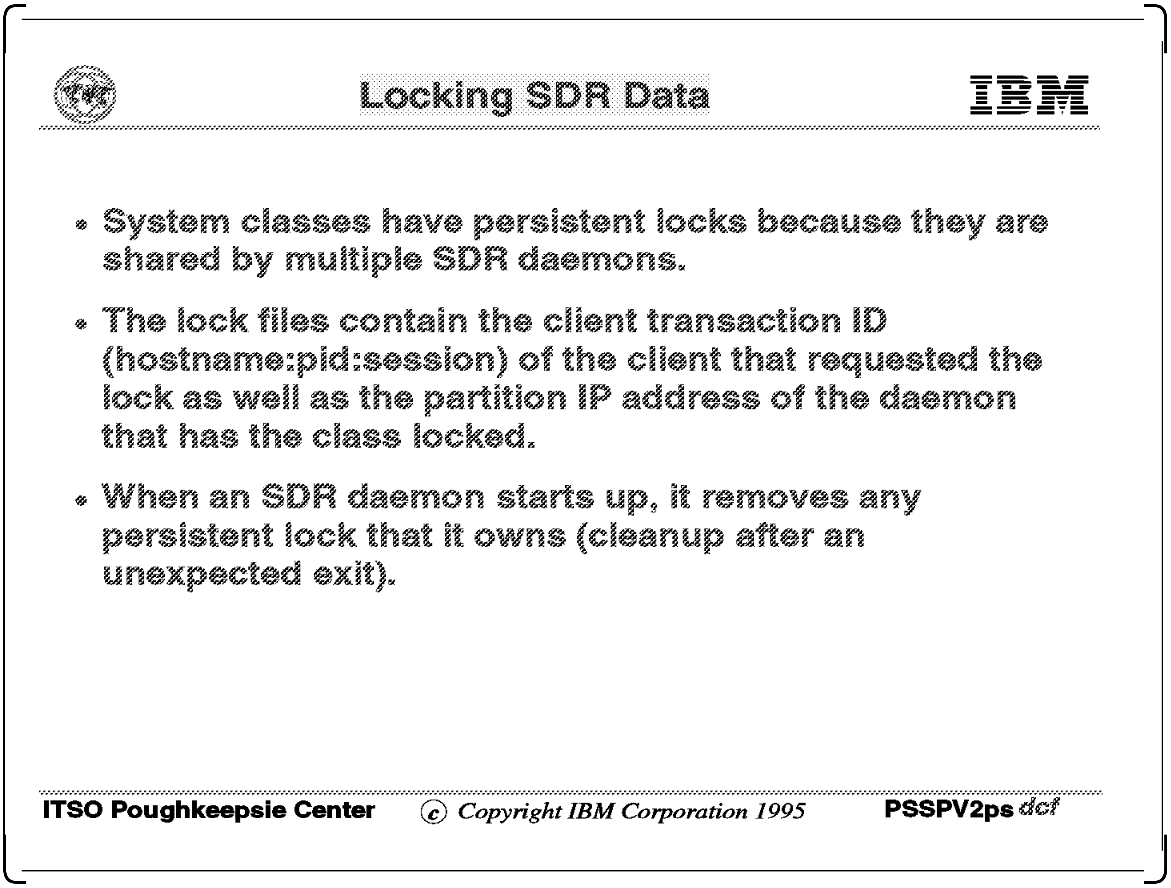
ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ps dce

One SDR server exists for each partition. Each server can modify objects in a system class or objects within its partition, while *query* requests can get information either about the local partition, or about other partitions, or about the whole system.

The SRC subsystem is used in place of inittab to start the daemons. The SRC system is much more robust in dealing with multiple daemons of the same type. An entry in inittab makes the SRC to start the SDR daemons.

The *sdr* script is used by the SRC to start the daemons, but it can also be used to manually control the daemons. This script accepts the *-spname* option, sets the SP_NAME variable and then starts the daemons. When used for manual control, it also takes options that cause specific actions to be taken.

7.6 SDR Locks



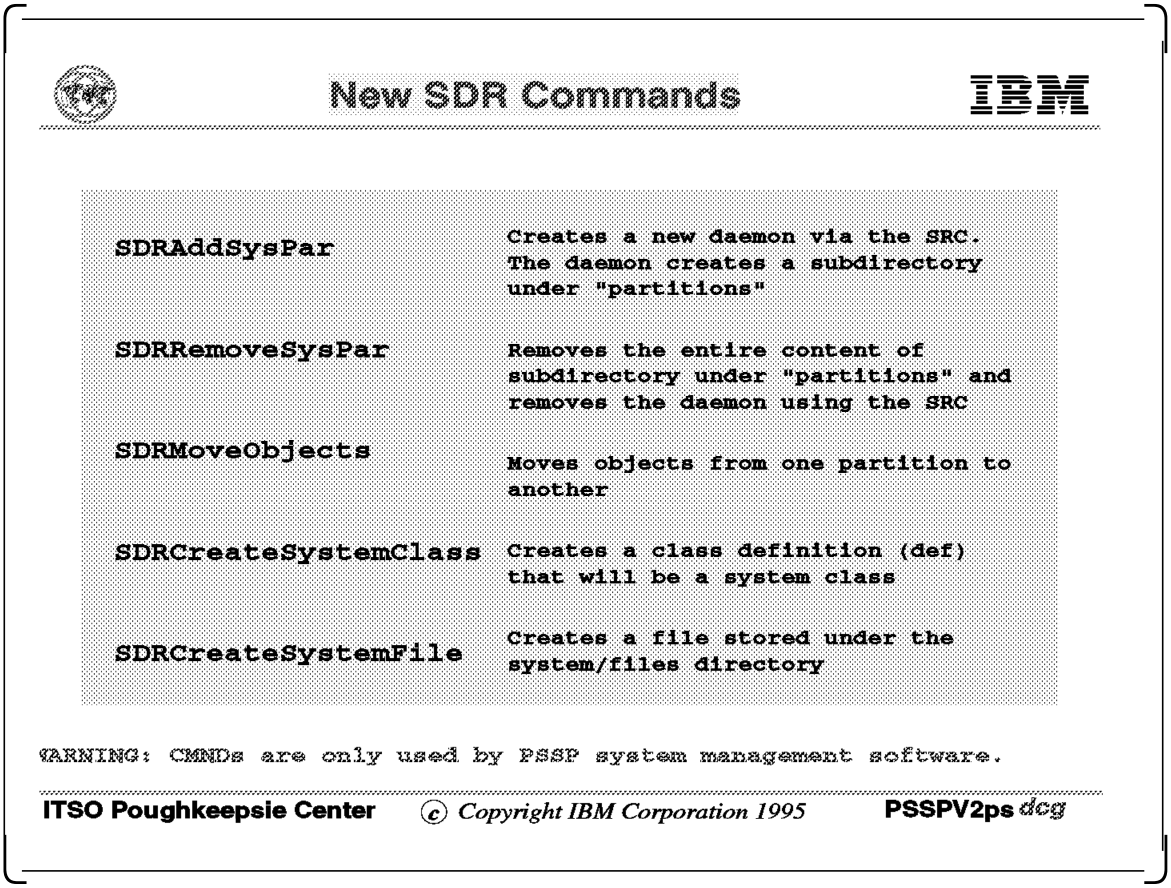
Locking SDR Data

- System classes have persistent locks because they are shared by multiple SDR daemons.
- The lock files contain the client transaction ID (hostname:pid:session) of the client that requested the lock as well as the partition IP address of the daemon that has the class locked.
- When an SDR daemon starts up, it removes any persistent lock that it owns (cleanup after an unexpected exit).

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* PSSPV2ps *dcf*

Having multiple SDR servers creates the need for persistent locks on system (global) data. This is implemented using lock files; there is one lock (potentially) for each system class. This is an advisory lock; it will go away if the SDR daemon dies or is restarted.

7.7 New SDR Commands



New SDR Commands

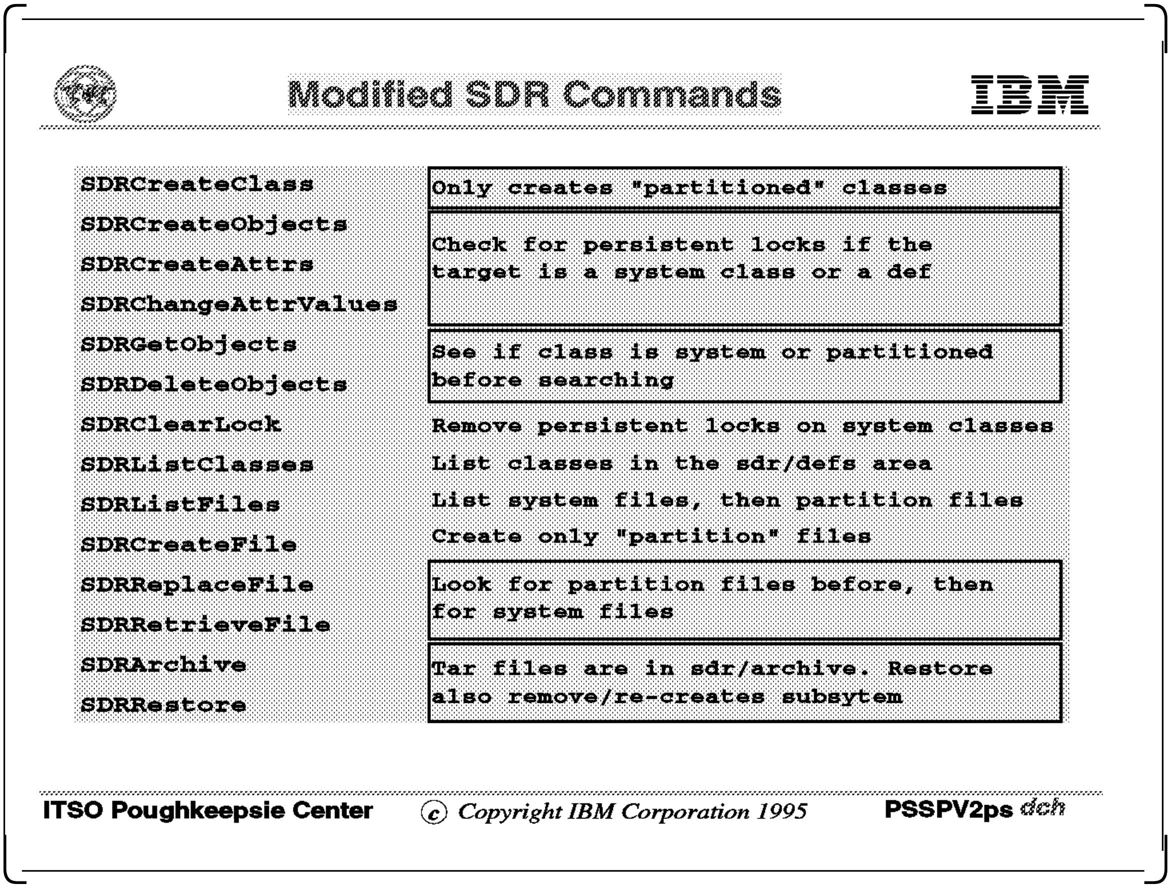
SDRAddSysPar	Creates a new daemon via the SRC. The daemon creates a subdirectory under "partitions"
SDRRemoveSysPar	Removes the entire content of subdirectory under "partitions" and removes the daemon using the SRC
SDRMoveObjects	Moves objects from one partition to another
SDRCreateSystemClass	Creates a class definition (def) that will be a system class
SDRCreateSystemFile	Creates a file stored under the system/files directory

WARNING: CMNDs are only used by PSSP system management software.

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ps dcg

Attention: These commands are not to be used by the administrators. They are only used by PSSP system management software.

7.8 Modified SDR Commands



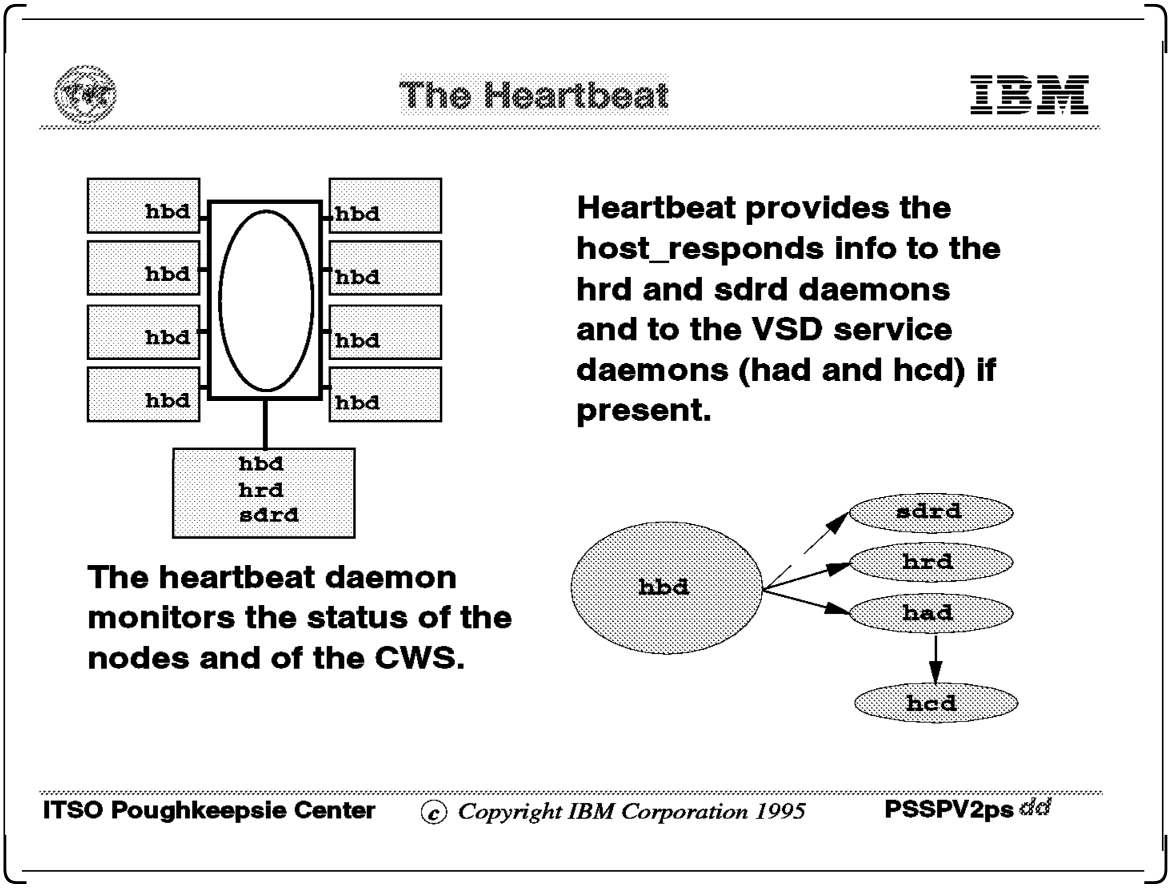
Modified SDR Commands

SDRCreateClass	Only creates "partitioned" classes
SDRCreateObjects	Check for persistent locks if the target is a system class or a def
SDRCreateAttrs	
SDRChangeAttrValues	
SDRGetObjects	See if class is system or partitioned before searching
SDRDeleteObjects	
SDRClearLock	Remove persistent locks on system classes
SDRListClasses	List classes in the sdr/defs area
SDRListFiles	List system files, then partition files
SDRCreateFile	Create only "partition" files
SDRReplaceFile	Look for partition files before, then for system files
SDRRetrieveFile	
SDRArchive	Tar files are in sdr/archive. Restore also remove/re-creates subsytem
SDRRestore	

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ps dch

There is no difference in the SDR commands used to retrieve data. Instead, when a new class has to be defined, two different functions will be used to create a system class or a partitioned class (so far, the SDR is the only application that create classes, but in principle, other applications could exploit the SDR database and interface too). Once the definition of a new class has been added (under the *defs* directory), that definition is available for all the partitions. So that a partitioned class, for example, needs only to be created once, although objects for that class could not even exist in a partition. When a partition is created, it automatically gets all the partitioned classes that have been created. The same applies to system files and partitioned files retrieve or creation tasks.

Chapter 8. The Heartbeat



The Heartbeat

Heartbeat provides the host_responds info to the hrd and sdrd daemons and to the VSD service daemons (had and hcd) if present.

The heartbeat daemon monitors the status of the nodes and of the CWS.

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ps dd

The heartbeat server monitors the health of the nodes and the control workstation. One instance of the *hbd* runs on each node and on the CWS; each node talks with the neighbor on the left and the neighbor on the right via a socket connection over the SP Ethernet, to detect possible failures either on the nodes or the network. If a node does not respond after a certain time and after a number of retries, it is excluded from the list of active nodes; when a node becomes active again (either rebooted or the network becomes active), it contacts the other nodes to be included again in the ring. One daemon plays a master role (the *group leader*), and a *quorum* is defined to prevent the ring from being divided into sub-rings.

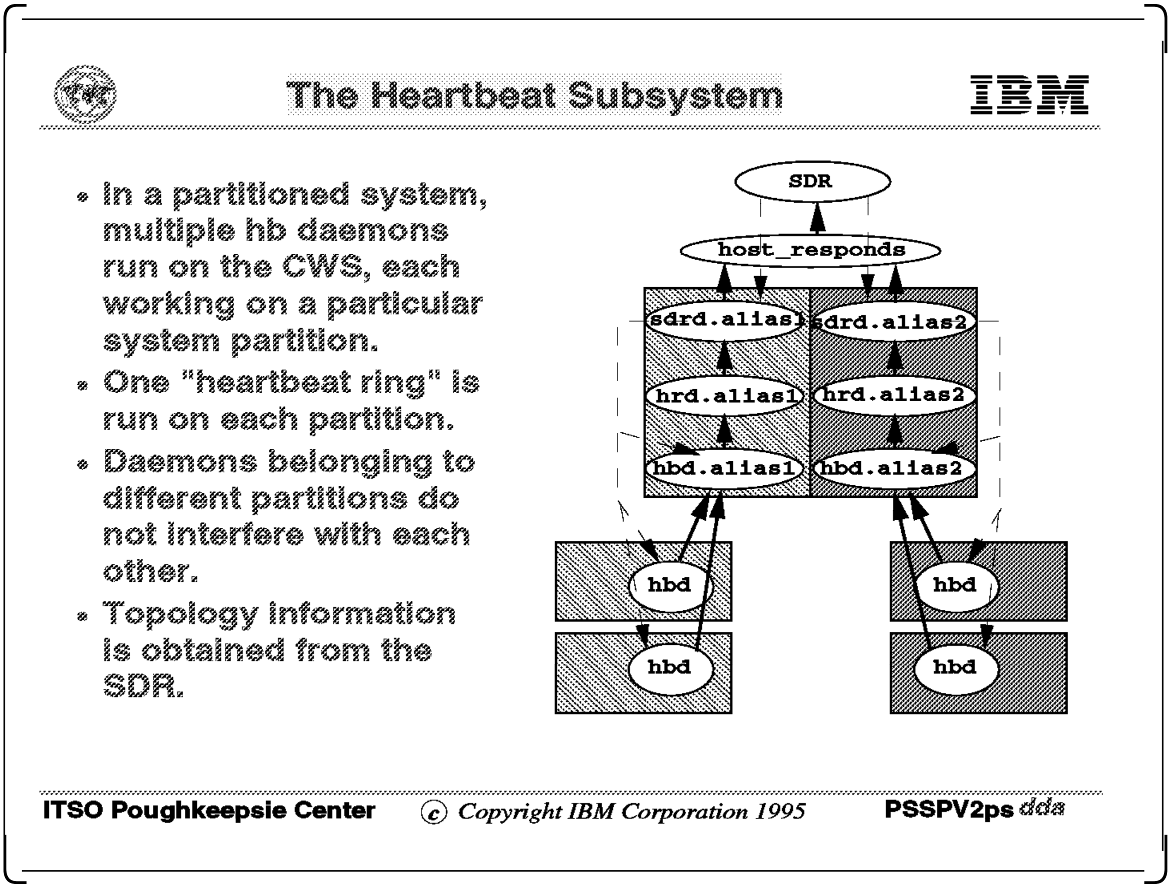
The hearbeat daemon provides the *host_responds* information to the host_responds daemon and to the SDR. There is only one instance of the **hrd** running on the CWS. The hr daemon can monitor the *host_responds* in two ways: the default method is to use the heartbeat system (gets the updates from the **hbd** running on the CWS).

The heartbeat daemon also provides input to the Recoverable Virtual Shared Disk service daemons (**had** and **hcd**) if the product is installed. When the **hb** script is executed, it triggers the **hbd** with a **-p0** or **-p1** flag depending on the existence of the */usr/lpp/csd/bin/ha_vsd* file. If the */usr/lpp/csd/bin/ha_vsd* exist, the **-p1** flag is used which indicates to the **hbd** to wait for the VSD service daemons. The VSD service daemons are namely;

- High availability daemon (had)
- Oracle connection manager (hcd)

When the **had** is triggered, it has the responsibility of initializing the **hcd**. When this sequence of operation is completed, the **hbd** resume it′s normal operation. Conversely, if the */usr/lpp/csd/bin/ha_vsd* file does not exist, the **-p0** flag is used and the **hbd** bypasses the wait or suspense state and resume it′s normal operation immediately.

8.1 The Heartbeat Subsystem



The Heartbeat Subsystem

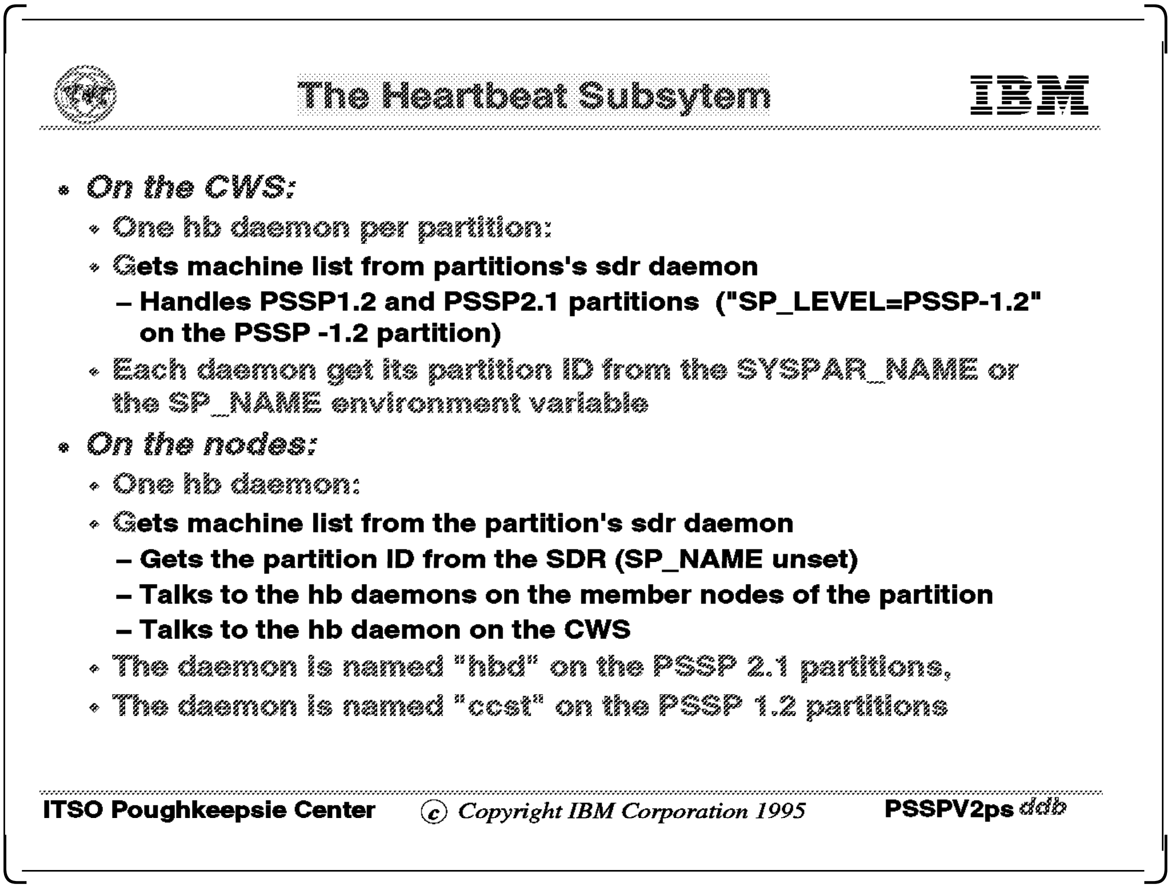
- In a partitioned system, multiple hb daemons run on the CWS, each working on a particular system partition.
- One "heartbeat ring" is run on each partition.
- Daemons belonging to different partitions do not interfere with each other.
- Topology information is obtained from the SDR.

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ps dda

In a partitioned system:

- Multiple heartbeat daemons exist on the CWS; the heartbeat daemon running on each node contacts the **hbd** for the partition it belongs to. Similarly, each **hrd** running on the CWS connects to the **hbd** and **sdrd** for the partition it belongs to.
- If the system is partitioned, one heartbeat ring is run on each partition.
- The configuration information (a list of all the nodes in the heartbeat ring) is provided by the SDR.

Note: All the daemons running on the CWS are uniquely identified because the partition name is appended to the process name.



The Heartbeat Subsytem

IBM

- *On the CWS:*
  - One hb daemon per partition:
  - **Gets machine list from partitions's sdr daemon**
    - **Handles PSSP1.2 and PSSP2.1 partitions ("SP_LEVEL=PSSP-1.2" on the PSSP -1.2 partition)**
  - Each daemon get its partition ID from the SYSPAR_NAME or the SP_NAME environment variable
- *On the nodes:*
  - One hb daemon:
  - **Gets machine list from the partition's sdr daemon**
    - **Gets the partition ID from the SDR (SP_NAME unset)**
    - **Talks to the hb daemons on the member nodes of the partition**
    - **Talks to the hb daemon on the CWS**
  - The daemon is named "hbd" on the PSSP 2.1 partitions,
  - The daemon is named "ccst" on the PSSP 1.2 partitions

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* **PSSPV2ps** *ddb*

The hb shell script is used to start the heartbeat and pass flags and parameters to it. It also produces the list of configured nodes by reading the SDR and passes it to the daemon.

- The SYSPAR_NAME variable holds the name of the system partition. The source of SYSPAR_NAME may be:

  1. The value of the *-spname* parameter passed to the hb script. In this case, SP_NAME is redefined by the script to be consistent with the -spname SYSPAR_NAME value.

  2. The SP_NAME environment variable, if exported and non-NULL. In this case, SYSPAR_NAME is set to be consistent with SP_NAME (to force SYSPAR_NAME to be NULL, you must use -spname '').

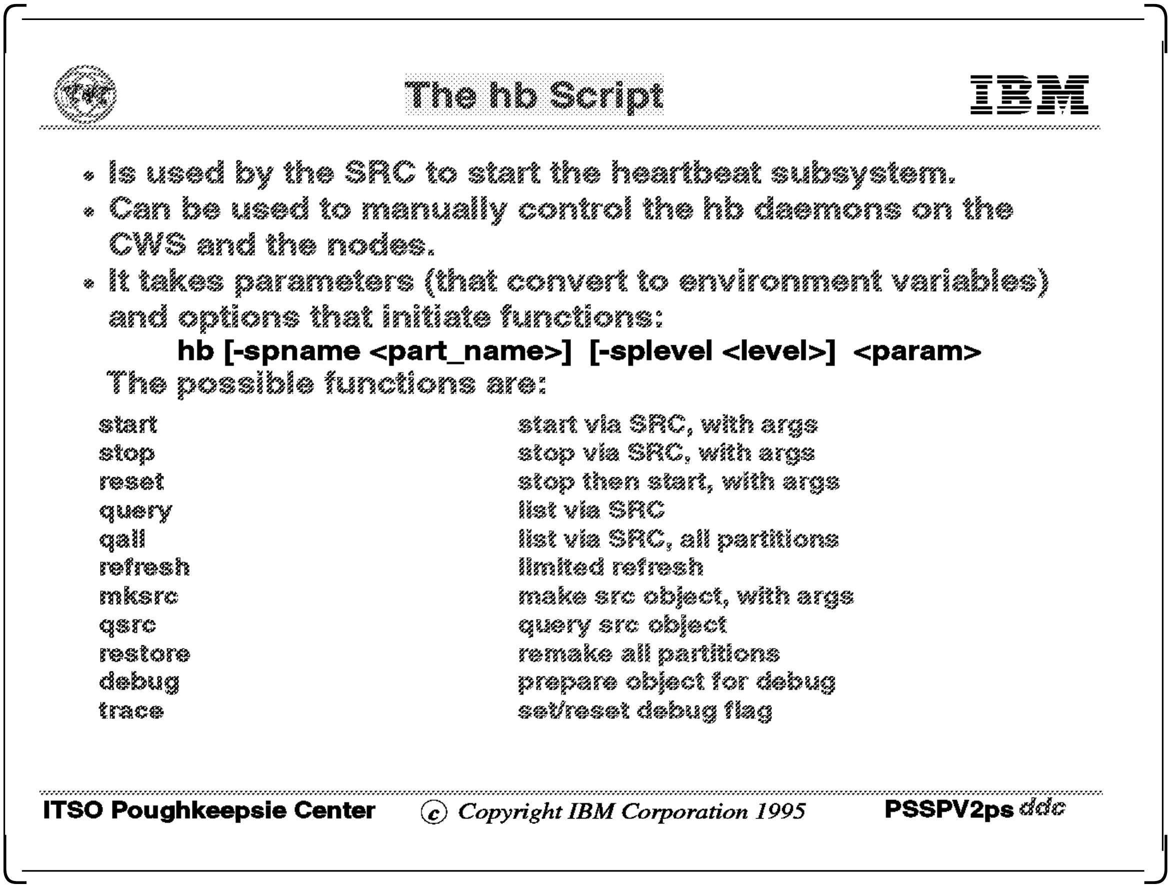
  3. The qualifier of the entry in the ODM for this SRC subsystem if there is exactly one such entry. On the rack nodes, it gives a NULL value for SYSPAR_NAME. On the CWS it will contain the name of the default partition. The value chosen for SYSPAR_NAME then determines SP_NAME.

  4. The name of the default system partition, when there are multiple SRC subsystem entries in the ODM. This is an expected case on the CWS, which may be hosting multiple partitions. SP_NAME is left unset in this case.

- A second parameter allows you to specify the PSSP level. In fact, for the coexistence of PSSP 2.1 and PSSP 1.2, the following mechanisms are adopted:

- The **hbd** daemon running on the AIX 3.2.5 nodes is a the PSSP 1.2 executable.
- The **hbd** daemon running on the CWS for the AIX 3.2.5 partition is the PSSP 2.1 executable, which runs in compatibility mode (triggered by SP_LEVEL=1.2).

8.2 The hb Script



The hb Script

- Is used by the SRC to start the heartbeat subsystem.
- Can be used to manually control the hb daemons on the CWS and the nodes.
- It takes parameters (that convert to environment variables) and options that initiate functions:

hb [-spname <part_name>] [-splevel <level>] <param>

The possible functions are:

start	start via SRC, with args
stop	stop via SRC, with args
reset	stop then start, with args
query	list via SRC
qall	list via SRC, all partitions
refresh	limited refresh
mksrc	make src object, with args
qsrc	query src object
restore	remake all partitions
debug	prepare object for debug
trace	set/reset debug flag

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* **PSSPV2ps** *ddc*

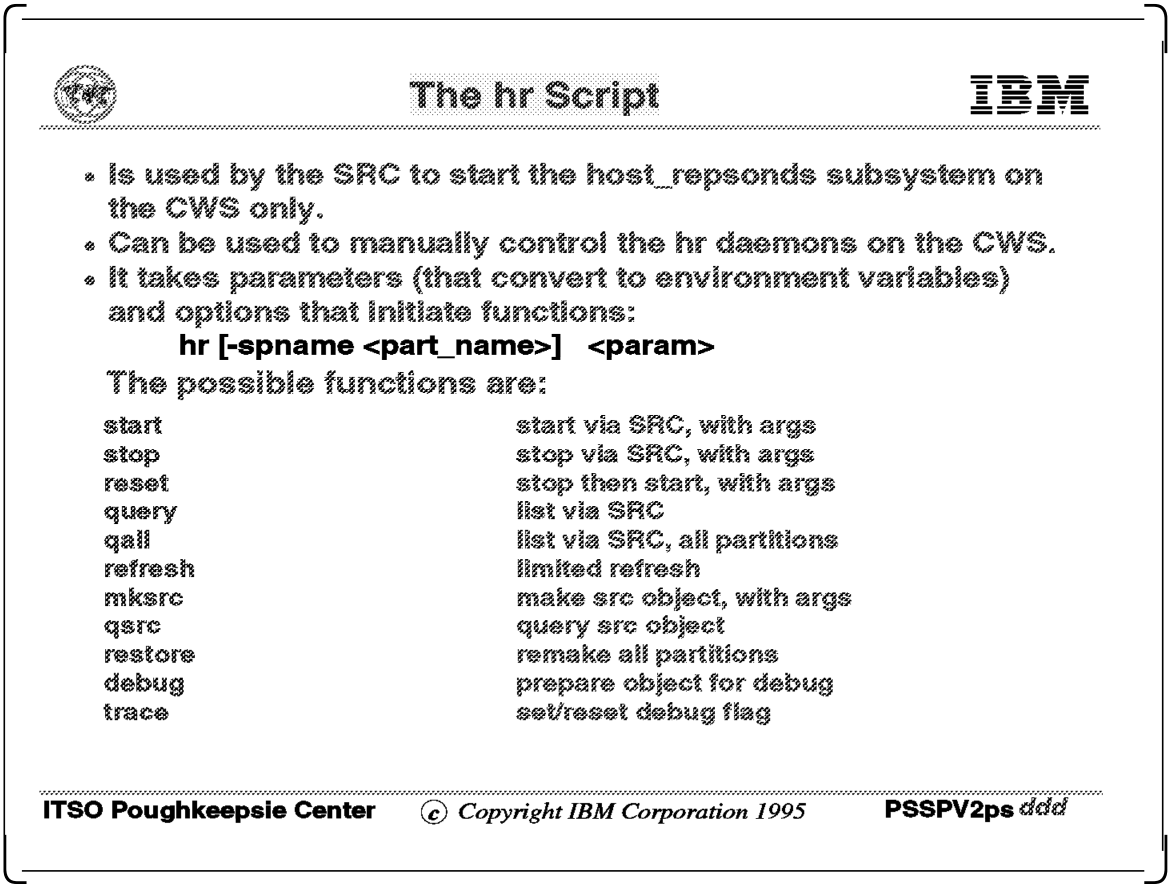
In PSSP 2.1, the **hb** and the **hr** daemons, as well as the **sdr** daemons are managed via the SRC. In PSSP 1.2, each daemon was started directly by the *init* process; one entry in **inittab** for each of the three daemons specified the name of the executable and the arguments and also the *respawn* attribute. In order to refresh a daemon, for example, you had to locate the process using the **ps** and the **grep** commands kill the process, and then a new daemon was started automatically, beacuse of the *respawn* attribute in *inittab*.

The new approach has many of the following advantages:

- The SRC still offers the *respawn* capability, should the daemon exit (this capability was already available with the old method).
- The SRC objects have both a specific name and a group name. For example, **hb** is the group name for all the heartbeat daemons, while **hb.sp2cw0** and **hb.sp2encw0** are the specific names of two subsystems, representing the heartbeat daemons running in two different partitions. The global name is used by *init* to start all the daemons in a group. The specific name can be used to monitor a single subsytem.
- The SRC offers convenient commands to monitor the subsystems: **llssrc**, **stopsrc**, **startsrc**, **refresh**.
- The SRC enforces that only one copy of a specific object is running at any time.

- Arguments and attributes for the subsystem are specified in the SRC object, and additional arguments can be appended when the object is started.
- The SRC can control an object with either signals or sockets. For the PSSP daemons, the socket protocol is used.

8.3 The hr Script



The hr Script

- Is used by the SRC to start the host_repsonds subsystem on the CWS only.
- Can be used to manually control the hr daemons on the CWS.
- It takes parameters (that convert to environment variables) and options that initiate functions:

  hr [-spname <part_name>] <param>

  The possible functions are:

start	start via SRC, with args
stop	stop via SRC, with args
reset	stop then start, with args
query	list via SRC
qall	list via SRC, all partitions
refresh	limited refresh
mksrc	make src object, with args
qsrc	query src object
restore	remake all partitions
debug	prepare object for debug
trace	set/reset debug flag

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* **PSSPV2ps** *ddd*

The commands description on this page also refers to the previous foil (the same functions are available). Moreover, the hr script uses the same mechanism to set the SYSPAR_NAME and SP_NAME variables that the hb script uses. The **hrd** daemon does not uses the *-sp_level* flag, and it only runs on the CWS.

The following control parameters are accepted:

start (or resume) Uses startsrc to start up the daemon.

stop (or quiesce) Uses stopsrc to stop the daemon.

reset Uses stopsrc, then startsrc to restart the daemon.

query Uses lssrc (and lssrc -l) to query the daemon.

qall Same as above, but for all partitions (this is better than lssrc -g, which does not take -l).

refresh Uses the refresh command to request a daemon refresh.

mksrc Uses mkssys to create an src subsystem object.

qsrc Uses odmget to show the object′s detailed definition.

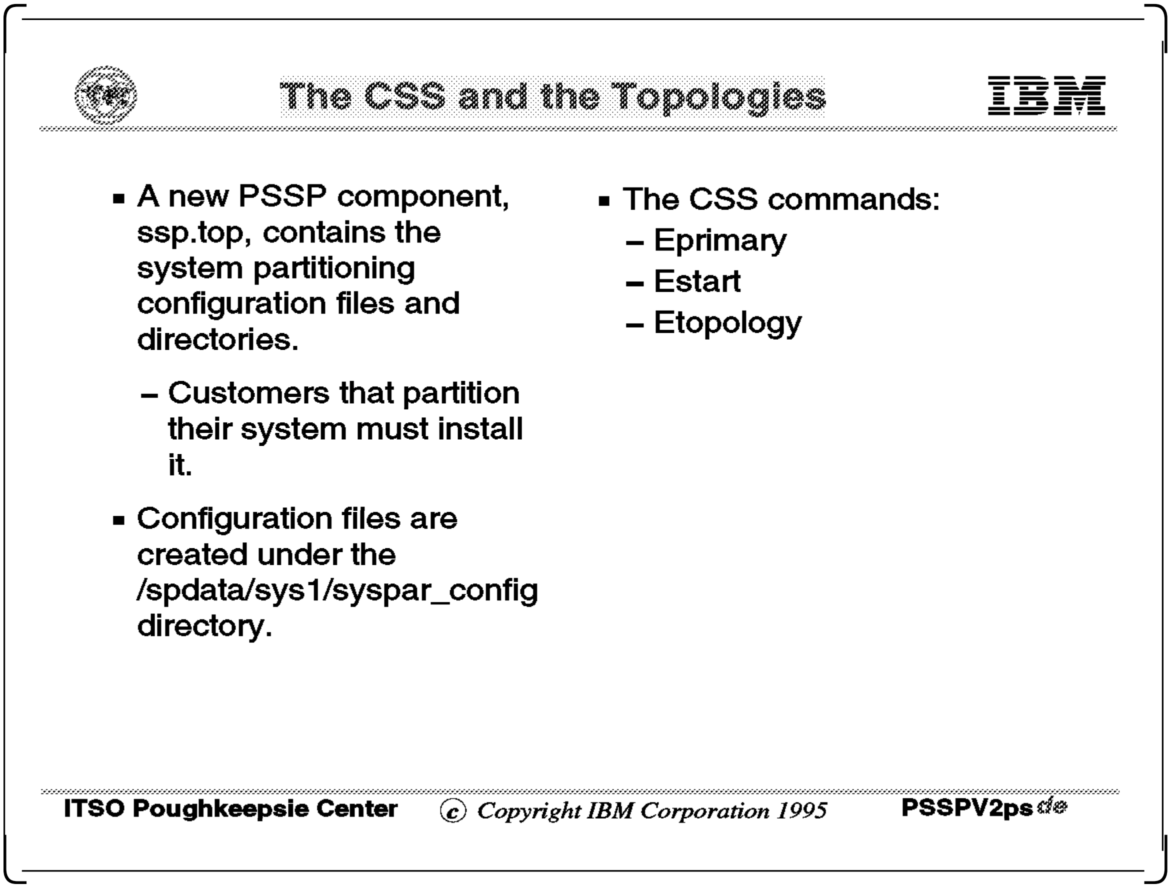
rmsrc Uses rmssys to remove an src subsystem object.

restore Removes all entries for the subsystem, then creates new ones based on values from the SDR, and starts them. The result is that the daemons agree with the SDR.

debug Uses chssys to modify the definition of the src subsystem object to make debugging easier (turns respawning off, sends stdout and stderr to /dev/console). Use *debug off* or *mksrc* to go back to normal.

trace Uses traceson/tracesoff to request daemon tracing. Specify *trace* or *trace x*, where x is anything but off to request tracing; *trace off* to stop it.

8.4 The CSS and the Topologies



The CSS and the Topologies

IBM

- A new PSSP component, ssp.top, contains the system partitioning configuration files and directories.
  - Customers that partition their system must install it.
- Configuration files are created under the /spdata/sys1/syspar_config directory.

- The CSS commands:
  - Eprimary
  - Estart
  - Etopology

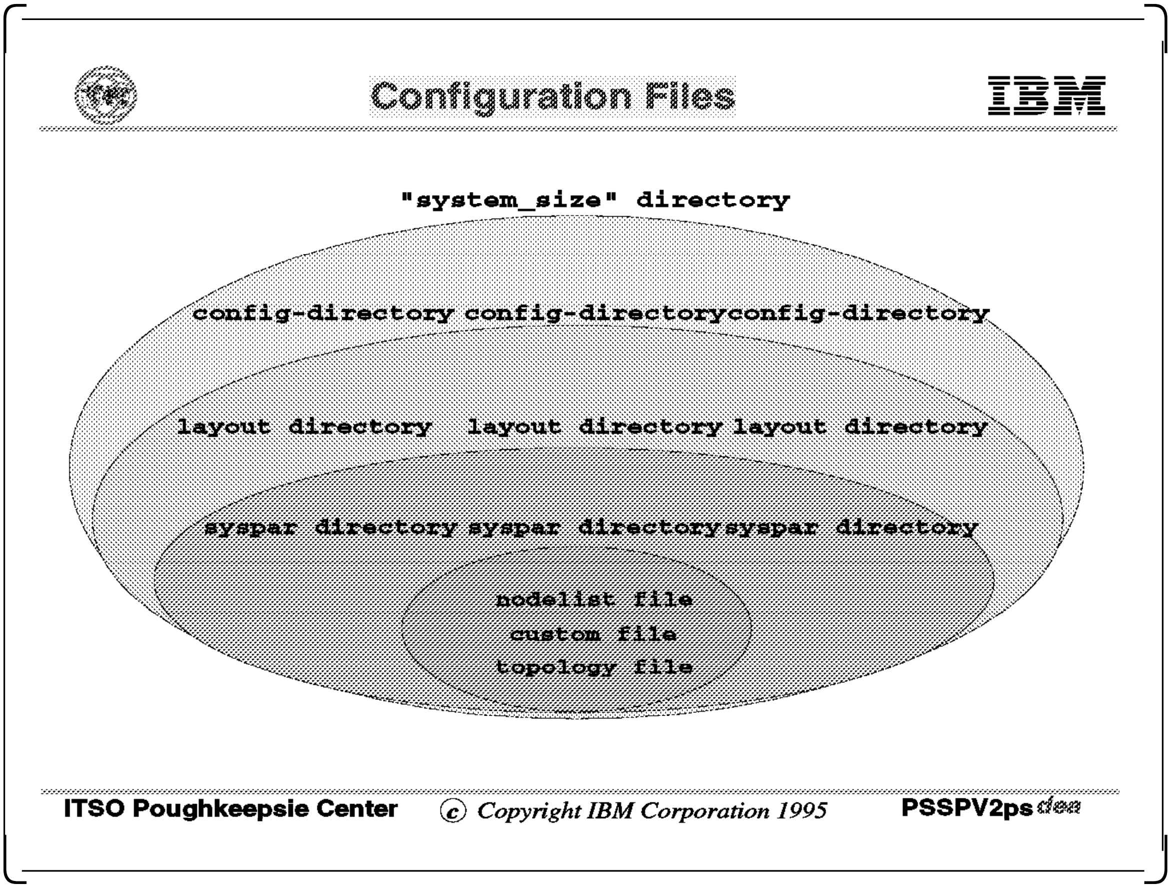
ITSO Poughkeepsie Center

PSSPV2ps de

Because a partitioned system subdivides the nodes and parts of the HPS switch network, these configuration directories define the different structures of each unique subdivision. Each switch partition requires a specific topology file that defines the subset of switch boards, chips and links that belong to the partition.

When a system configuration is applied to SDR, the topology files are copied to their corresponding partition′s in *sdr/partition/<IP address>/files* directory. Later, the primary node on each partition retrieves the correct topology file from the partition SDR, to initialize its portion of the switch.

8.5 Configuration Files



Configuration Files

"system_size" directory

config-directory config-directoryconfig-directory

layout directory layout directory layout directory

syspar directory syspar directorysyspar directory

nodelist file
custom file
topology file

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* PSSPV2ps *dea*

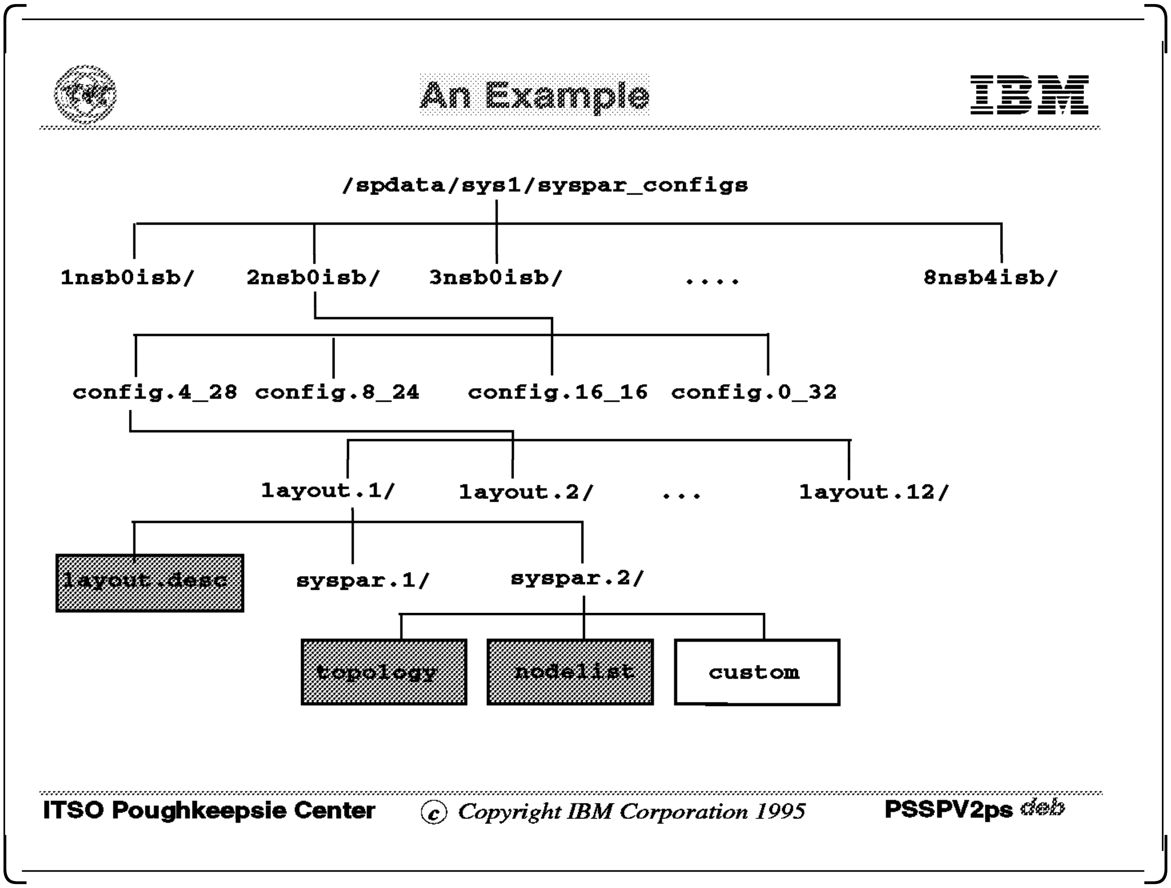
A physical RS/6000 SP system can be divided, given certain constraints, into a number of configurations.

- A directory exists for each of the nine system sizes:

16 slots	1nsb0isb
32 slots	2nsb0isb
...	...
128 slots	8nsb4isb

- Within each directory, there is a group of configuration directories representing a generic partitioning of the system (for example, for a 16 slot system, you have the following choices: 16, 4_12, 8_8, 4_4_8, 4_4_4_4).
- Within each configuration directory, there are specific node layouts for each generic partitioning. For the 4_12 configuration, for example, you have four possible layouts, depending on which set of four nodes connected to the same switch chip you put in the 4-nodes partition.
- Within each layout directory, there are the system partition directories. Each of these directories contains the files for a particular partition.

8.5.1 An Example



An Example

/spdata/sys1/syspar_configs

1nsb0isb/ 2nsb0isb/ 3nsb0isb/ 8nsb4isb/

config.4_28 config.8_24 config.16_16 config.0_32

layout.1/ layout.2/ ... layout.12/

layout.desc syspar.1/ syspar.2/

topology nodelist custom

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* PSSPV2ps deb

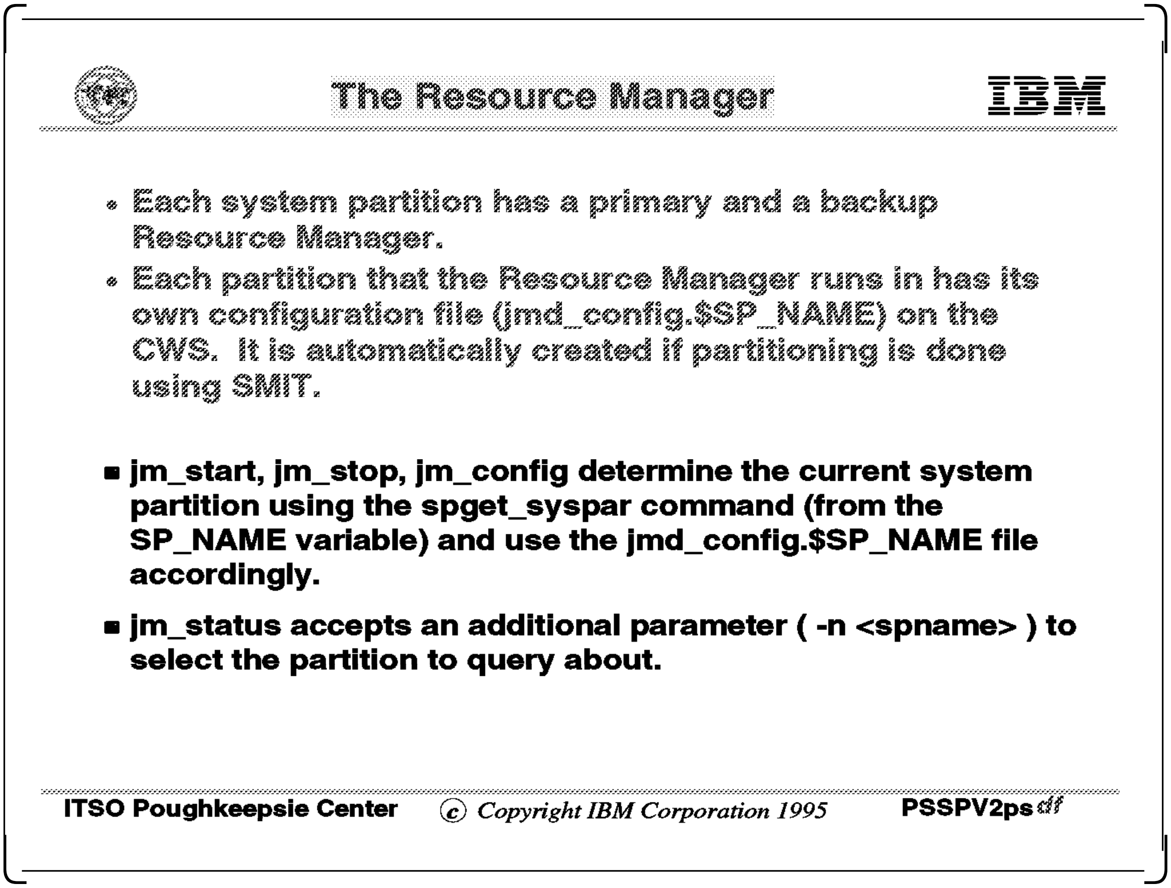
This foil presents the directory tree where you find the reference files for defining your partitions. You can retrieve the files that describe your configuration as follows:

/spdata/sys1/syspar_configs
This is the directory that include all possible partition configurations. Let′s suppoe you r RS/6000 SP is a 2-frame system, certainly without intermediate switch board. You will select your partition description in the directory `/2nsb0isb`.

/spdata/sys1/syspar_configs/2nsb0isb
This directory contains all the possible partitions you can define on your 32-node RS/6000 SP. Let′s suppose your choice is a 4-node partition plus a 28-node partition: the related layouts are in the config.4_28 directory.

/spdata/sys1/syspar_configs/2nsb.0isb/config.4_28
This directory contains the possible layouts for a `config.4_28` partition configuration. There are 12 possibilities, according to the nodes you want to dedicate to your partitions. Each layout directory includes a layout description and one directory per partition. In these directories you find the topology files and the nodelist file of your partition.

8.6 The Resource Manager



The Resource Manager

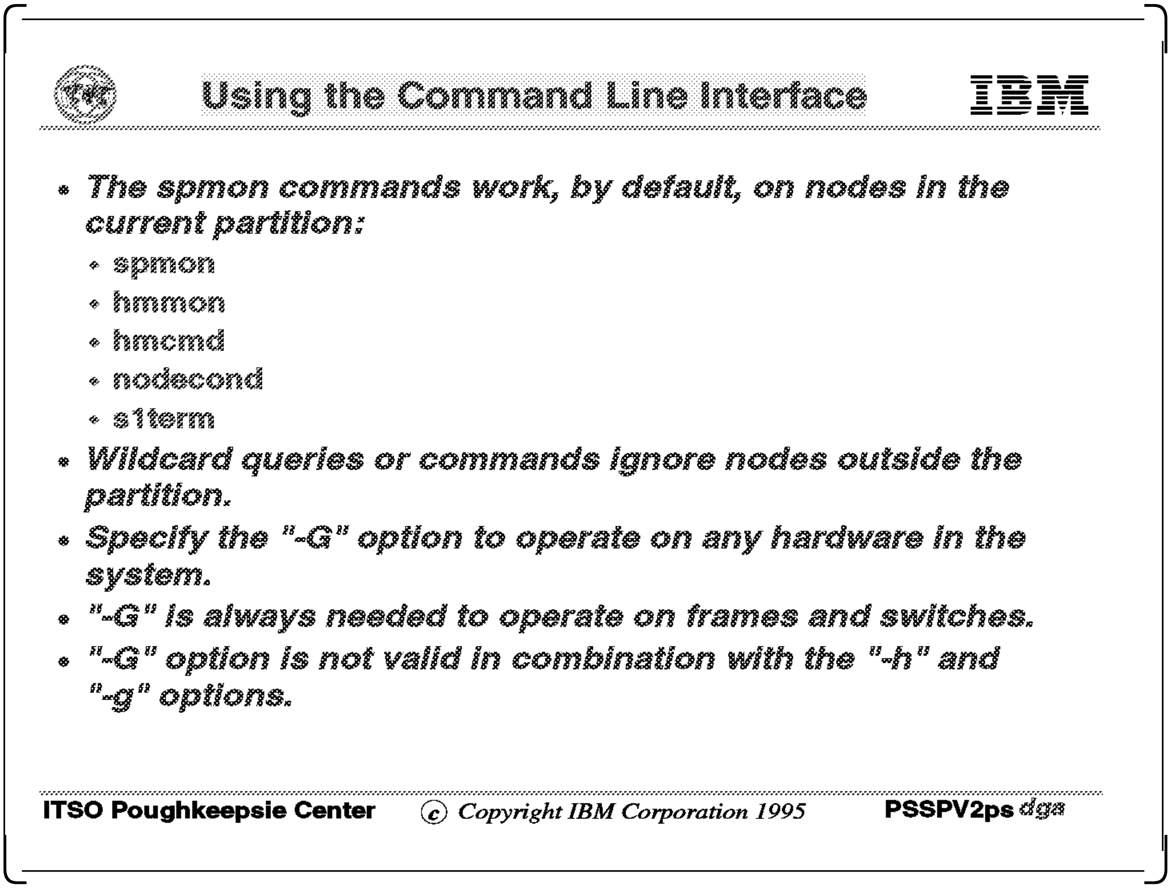
- Each system partition has a primary and a backup Resource Manager.
- Each partition that the Resource Manager runs in has its own configuration file (jmd_config.$SP_NAME) on the CWS. It is automatically created if partitioning is done using SMIT.

- jm_start, jm_stop, jm_config determine the current system partition using the spget_syspar command (from the SP_NAME variable) and use the jmd_config.$SP_NAME file accordingly.
- jm_status accepts an additional parameter (-n <spname>) to select the partition to query about.

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* PSSPV2ps*df*

The Resource Manager views each system partition as a logical SP. All previous functionalities are supported within an individual partition. Each system partition has its own configuration file.

8.6.1 Monitoring System Partitions



Using the Command Line Interface

- ***The spmon commands work, by default, on nodes in the current partition:***
  - spmon
  - hmmon
  - hmcmd
  - nodecond
  - s1term
- ***Wildcard queries or commands ignore nodes outside the partition.***
- ***Specify the "-G" option to operate on any hardware in the system.***
- ***"-G" is always needed to operate on frames and switches.***
- ***"-G" option is not valid in combination with the "-h" and "-g" options.***

ITSO Poughkeepsie Center ⓒ *Copyright IBM Corporation 1995* PSSPV2ps dga

The spmon command line interface consists of the following commands:

- spmon
- hmmon
- hmcmd
- nodecond
- s1term

Each of these works, by default, on nodes in the current partition. The current partition can be obtained by issuing the SDRGetPartition command.
Each of these commands has a **-G** option that removes the partition barriers and allows the command to operate on any hardware in the SP.
The **-G** option is always needed to monitor and control any frame or switch hardware. Frames and switches should be thought of as being outside any partitions.

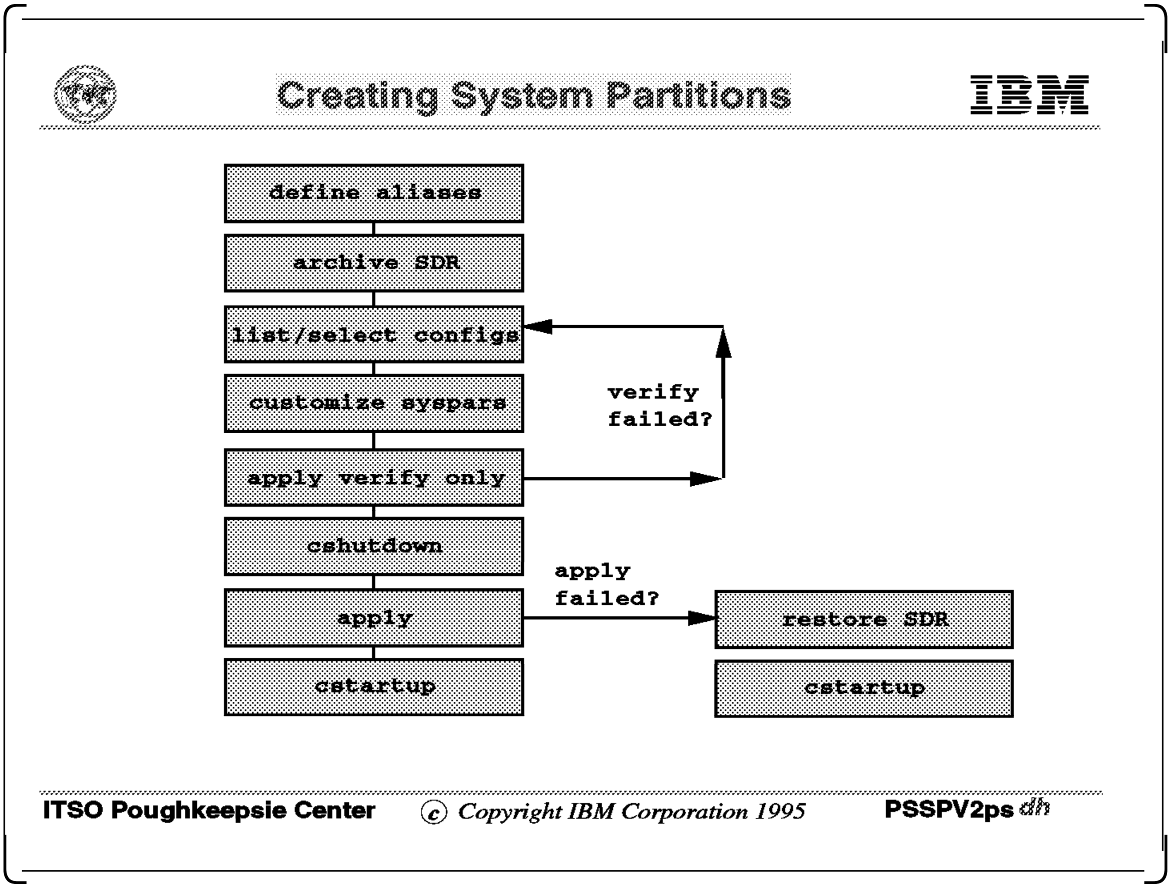
The System Monitor GUI and command line interface provide a view of the SP system as a set of system partitions, which can be monitored and controlled separately.

The Main Menu contains the additional View pulldown menu. When selected it lists all the defined partitions and the *Global* choice. You can select one partition, or *Global* to define the target of all the actions that you are going to perform. Also the Global Commands, Display Layout and All Node Summary menus have the View menu, to restrict the access/control to a single partition or to the whole system.

The View Menu is visible only if multiple partitions are defined.

The Frame Layout and Switch Layout are accessible only if you have chosen the *Global* view of the system.

Chapter 9. Creating System Partitions



Creating System Partitions

define aliases

archive SDR

list/select configs

customize syspars

apply verify only

verify failed?

cshutdown

apply

apply failed?

restore SDR

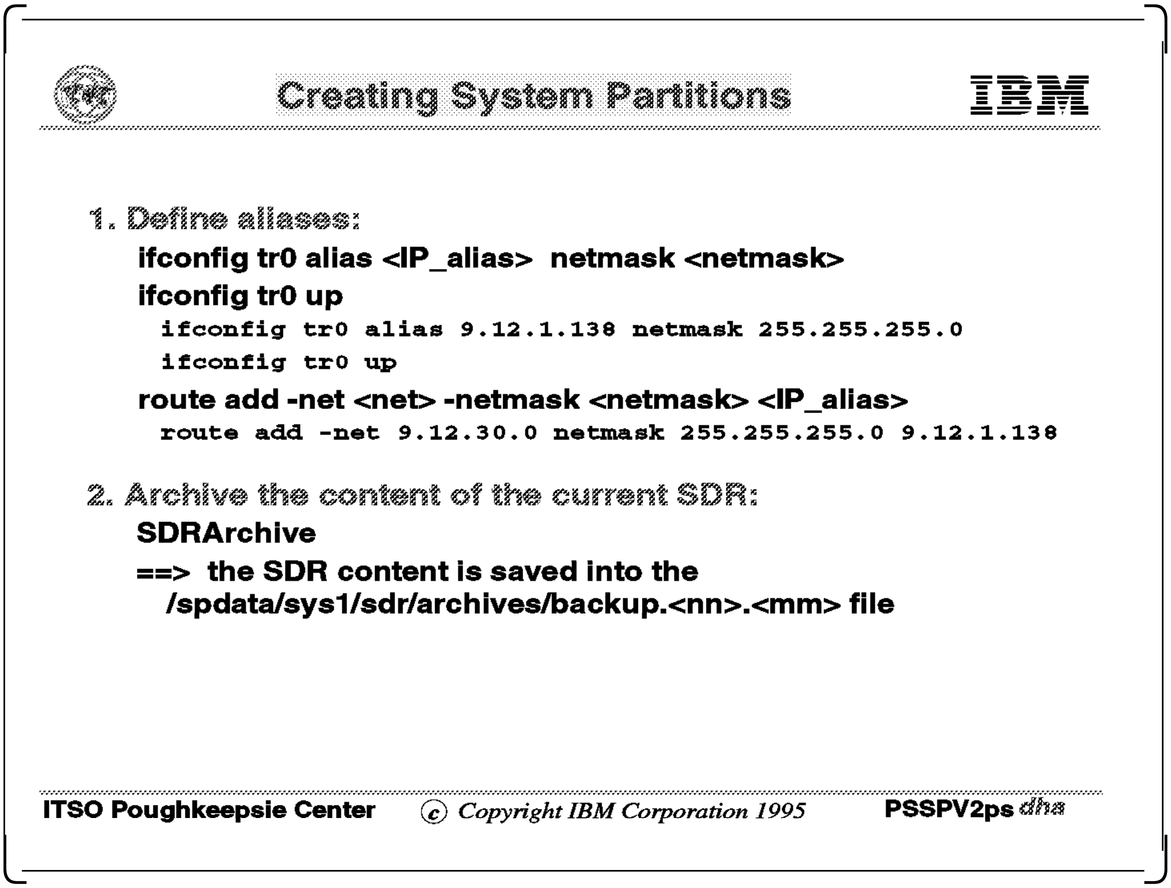
cstartup

cstartup

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ps dh

System partitioning configuration occurs after initial installation of PSSP 2.1 on the control workstation. However, you will follow the same procedure if you want to repartition the system (for example, you want multiple partitions to merge in one).

The picture shows the flow diagram for the process. Each step is described in detail in the next foils. All but the first step can be performed via the SMIT interface; we will show the SP command, however, to give a better understanding of what happens at each step.



Creating System Partitions

1. Define aliases:

   ifconfig tr0 alias <IP_alias> netmask <netmask>

   ifconfig tr0 up

   ```
   ifconfig tr0 alias 9.12.1.138 netmask 255.255.255.0
   ifconfig tr0 up
   ```

   route add -net <net> -netmask <netmask> <IP_alias>

   ```
   route add -net 9.12.30.0 netmask 255.255.255.0 9.12.1.138
   ```

2. Archive the content of the current SDR:

   SDRArchive

   ==> the SDR content is saved into the /spdata/sys1/sdr/archives/backup.<nn>.<mm> file

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* **PSSPV2ps** *dha*

- The system administrator sets up alias names and IP addresses representing each system partition (other than the default). This includes:

  1. Defining the alias to AIX using the **ifconfig** command with the **alias** parameter:

     ```
     ifonfig tr0 alias <IP alias> netmask <netmask>
     ifonfig tr0 up
     ```

  2. Adding the alias name to the name resolution mechanism (in fact the search for the aliases is made by parsing the output of the *netstat -nr* command in the syspar script files)

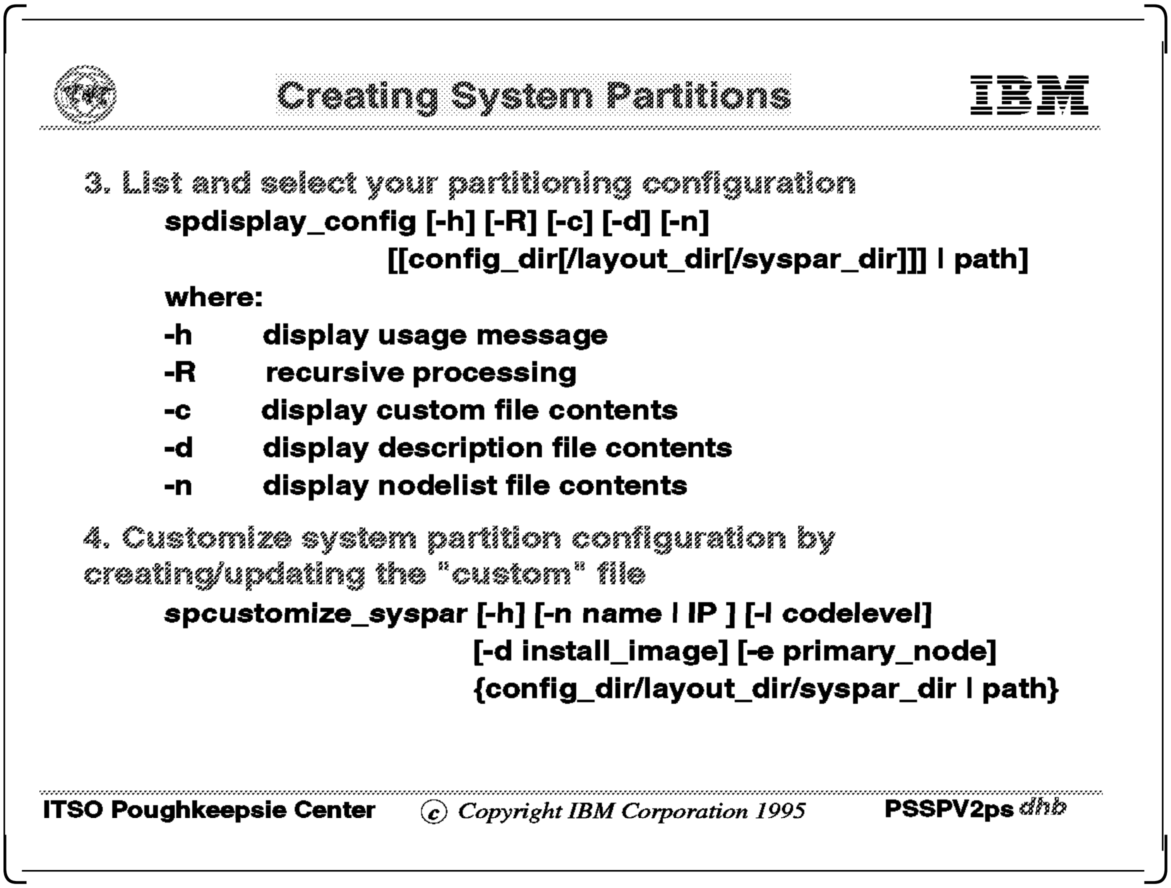
     ```
     route add -net <net> -netmask <netmask> <IP alias>
     ```

  Notes:

  1. You should add the commands above to your *rc.net* file, so that they are executed at each boot of the CWS.

  2. The design of the SP is such that it is only possible to define aliases for the network interface corresponding to the *hostname* of the CWS.

- The SDR must be archived so that if any errors occur during the following steps, the system administrator restores it and starts the configuration over.



Creating System Partitions

3. List and select your partitioning configuration

spdisplay_config [-h] [-R] [-c] [-d] [-n]
[[config_dir[/layout_dir[/syspar_dir]]] | path]

where:

-h display usage message
-R recursive processing
-c display custom file contents
-d display description file contents
-n display nodelist file contents

4. Customize system partition configuration by creating/updating the "custom" file

spcustomize_syspar [-h] [-n name | IP] [-l codelevel]
[-d install_image] [-e primary_node]
{config_dir/layout_dir/syspar_dir | path}

ITSO Poughkeepsie Center *© Copyright IBM Corporation 1995* PSSPV2ps *dhb*

- Use the **spdisplay_config** command to look at all the possible partitions and configurations for your system. Depending upon the options and operand specified, the information displayed is at the configuration, layout, partition or customization level. For example, if you don′t supply the last parameter, all the available configurations are shown, so that you obtain (in this case the *system_size* is 1nsb.0isb.0):

  ```
  config_16
  config_4_12
  config_4_4_4_4
  config_4_4_8
  config_8_8
  ```

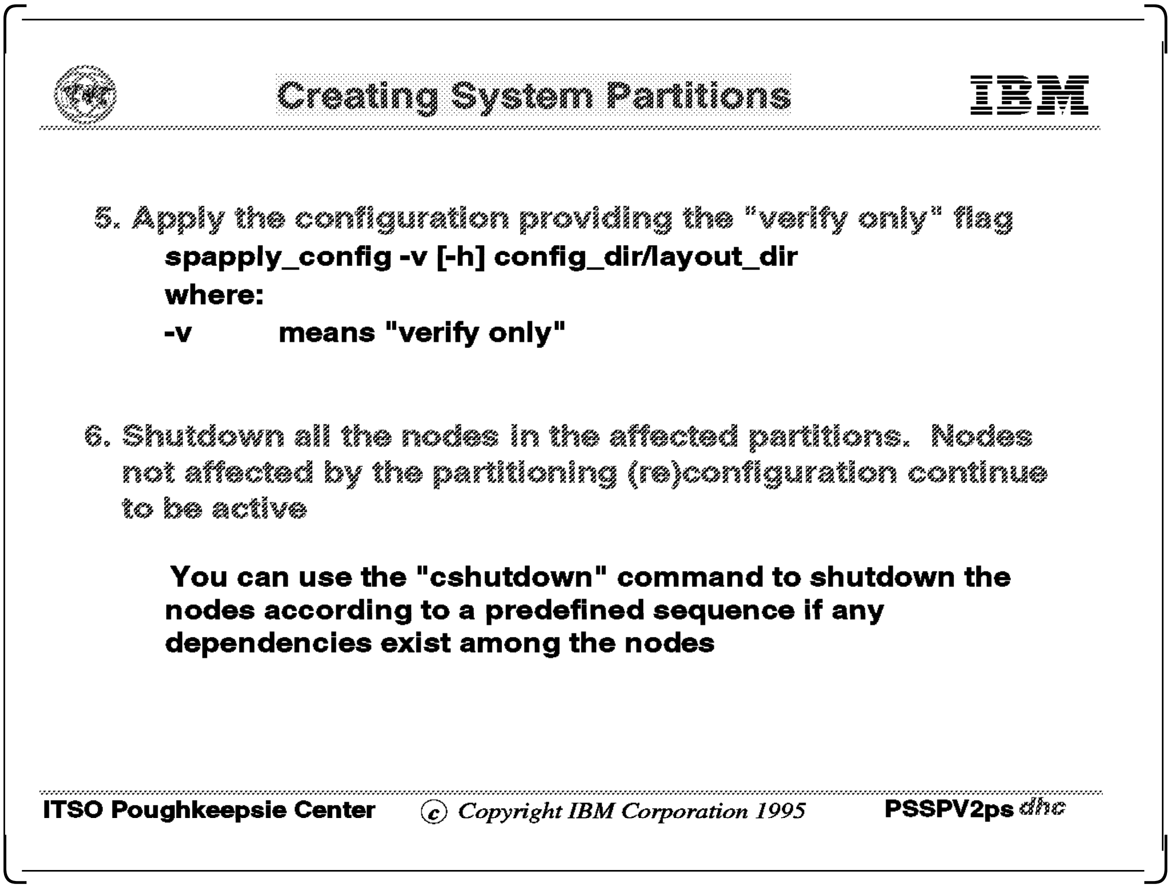
  If you specify *config.4_12* in the last parameter, you obtain the list of all the available layouts for that configuration:

  ```
  layout.1
  layout.2
  layout.3
  layout.4
  ```

  This command only displays data that is applicable to the system it is executed on (the system size is automatically selected).

  Note: This command does not show the current syspar info from the SDR; the **splstdata** command must be used for that purpose.

- This new command customizes a system partition configuration file. You are requested to enter the partition name/IP addrress, the PSSP code level, the partition default installation image (optional - defaults to the default SP install image) and the primary node for the partition (optional - defaults to the first node in the *nodelist*). This information is saved in the **custom** file, so data is persistent. Subsequent partitioning tasks that apply the same configuration to the system can bypass this step if customization data does not need to be changed. Customization is permitted for any partition in any valid configuration.



Creating System Partitions

5. Apply the configuration providing the "verify only" flag

   spapply_config -v [-h] config_dir/layout_dir

   where:

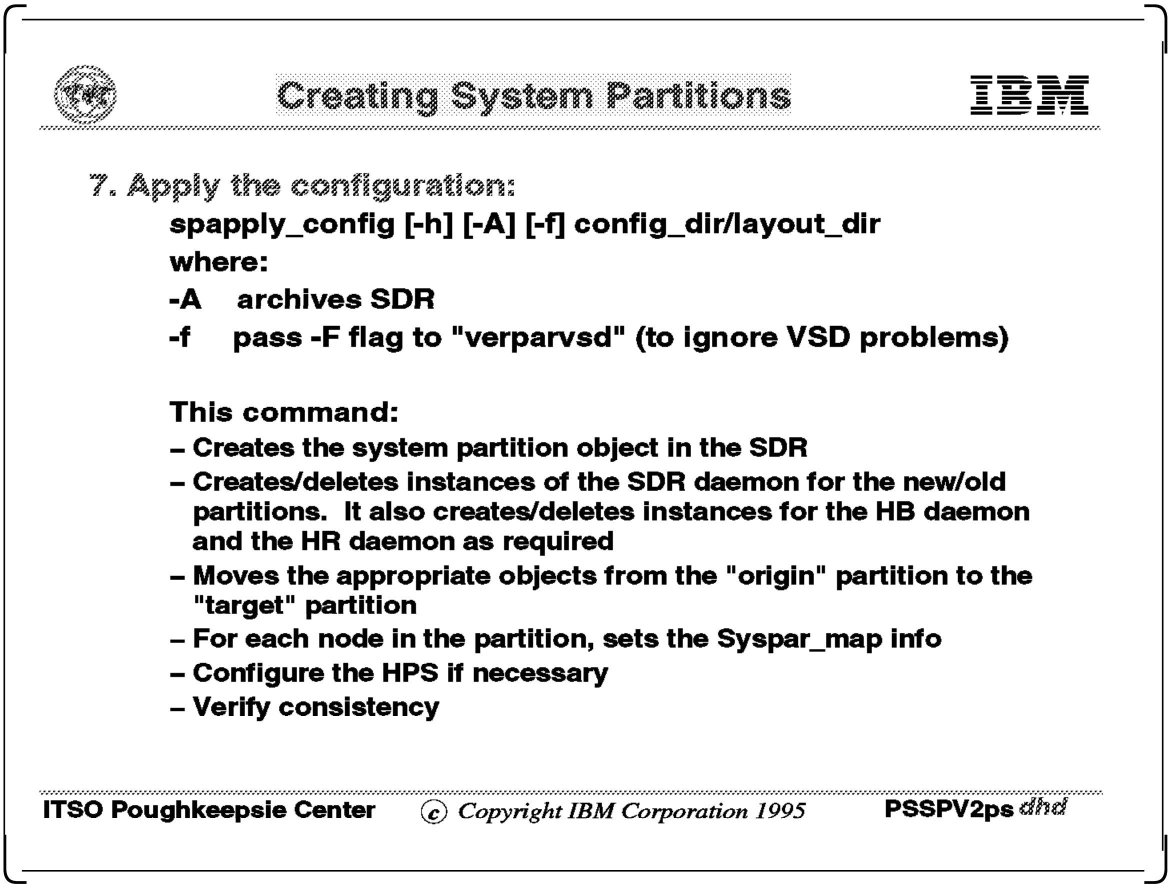
   -v means "verify only"

6. Shutdown all the nodes in the affected partitions. Nodes not affected by the partitioning (re)configuration continue to be active

   You can use the "cshutdown" command to shutdown the nodes according to a predefined sequence if any dependencies exist among the nodes

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* PSSPV2ps *dhc*

- The *spapply_config -v* command:
  - Verifies that the files *nodelist*, *topology* and *custom* exist for all the partitions
  - Verifies that the content of the *custom* files is consistent
  - Verifies that the SDR is reachable and that its content is consistent
- Shutdown the affected nodes in the proper sequence.



Creating System Partitions

7. Apply the configuration:

spapply_config [-h] [-A] [-f] config_dir/layout_dir

where:

-A archives SDR

-f pass -F flag to "verparvsd" (to ignore VSD problems)

This command:

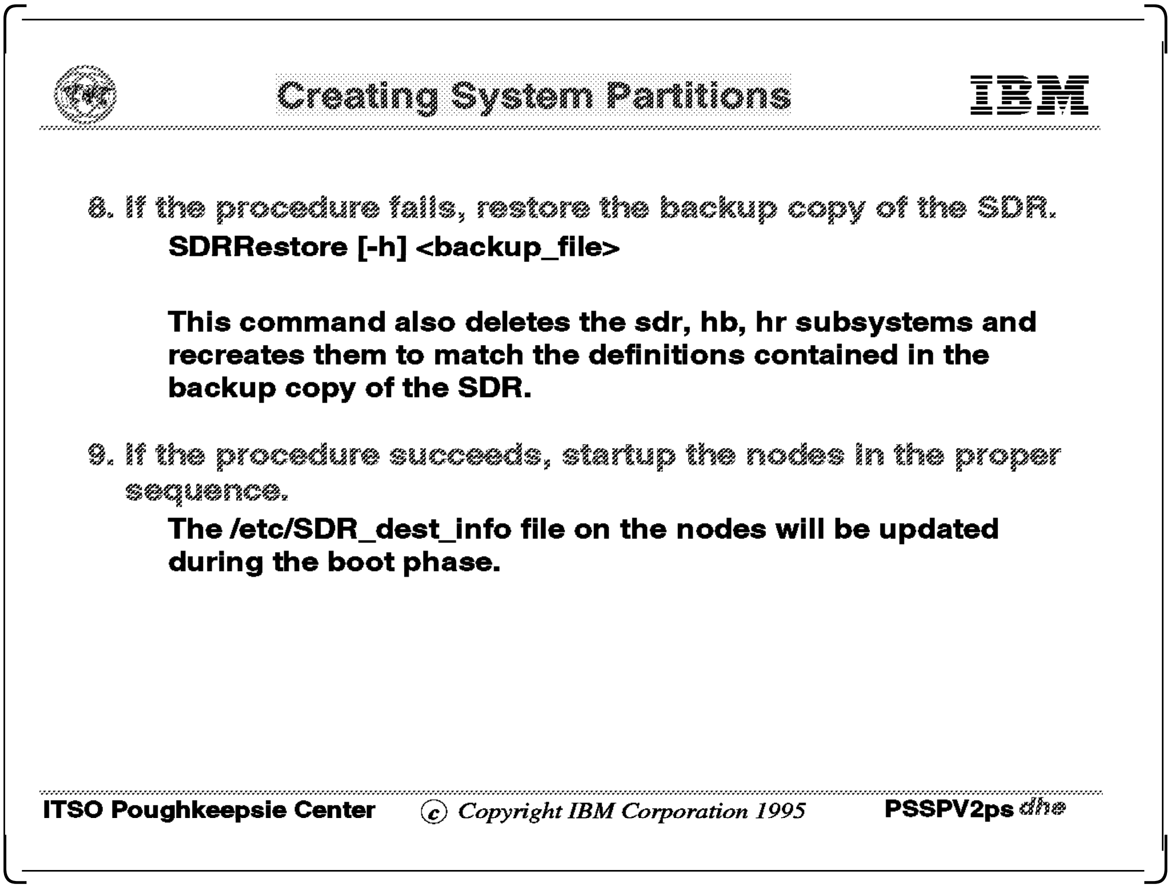
- **Creates the system partition object in the SDR**
- **Creates/deletes instances of the SDR daemon for the new/old partitions. It also creates/deletes instances for the HB daemon and the HR daemon as required**
- **Moves the appropriate objects from the "origin" partition to the "target" partition**
- **For each node in the partition, sets the Syspar_map info**
- **Configure the HPS if necessary**
- **Verify consistency**

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* **PSSPV2ps** *dhd*

The *spapply_config* command commits the configuration setup you have defined:

- For each new partition, creates the system_partition directory and files in the SDR.
- For each affected partition, moves the objects of the Node, Adapter, switch_responds, host_responds class, from their original partition to the new one (SDRMoveObjects).
- For all nodes in the new partition, sets the name in the Syspar_map object class for this node number to the name of the new partition.
- Deletes the **sdr** subsystems for the affected partitions, creates and starts the *sdr* daemons (**sdr -mksrc ..**, **sdr -start ..**) according to the configuration.
- Perform the same step above, for the **hb** and the **hr** subsystems.
- If the system has the switch, configures it (**Eprimary, Ennotator, Etopolgy**).
- Validates the VSD configuration and moves VSD objects to the new partition. The **checkVSD** routine is invoked to determine the impact upon the VSD subsytem of applying the specified configuration layout. If the -F flag is used, possible inconsistencies are recorded and corrected for the new partition layout.

If you are moving back to the default partition, redundant sdr, hb and hr subsystems are removed; node objects are moved back to the default partition, and redundant partitions are removed.



Creating System Partitions

8. If the procedure fails, restore the backup copy of the SDR.

   SDRRestore [-h] <backup_file>

   This command also deletes the sdr, hb, hr subsystems and recreates them to match the definitions contained in the backup copy of the SDR.

9. If the procedure succeeds, startup the nodes in the proper sequence.

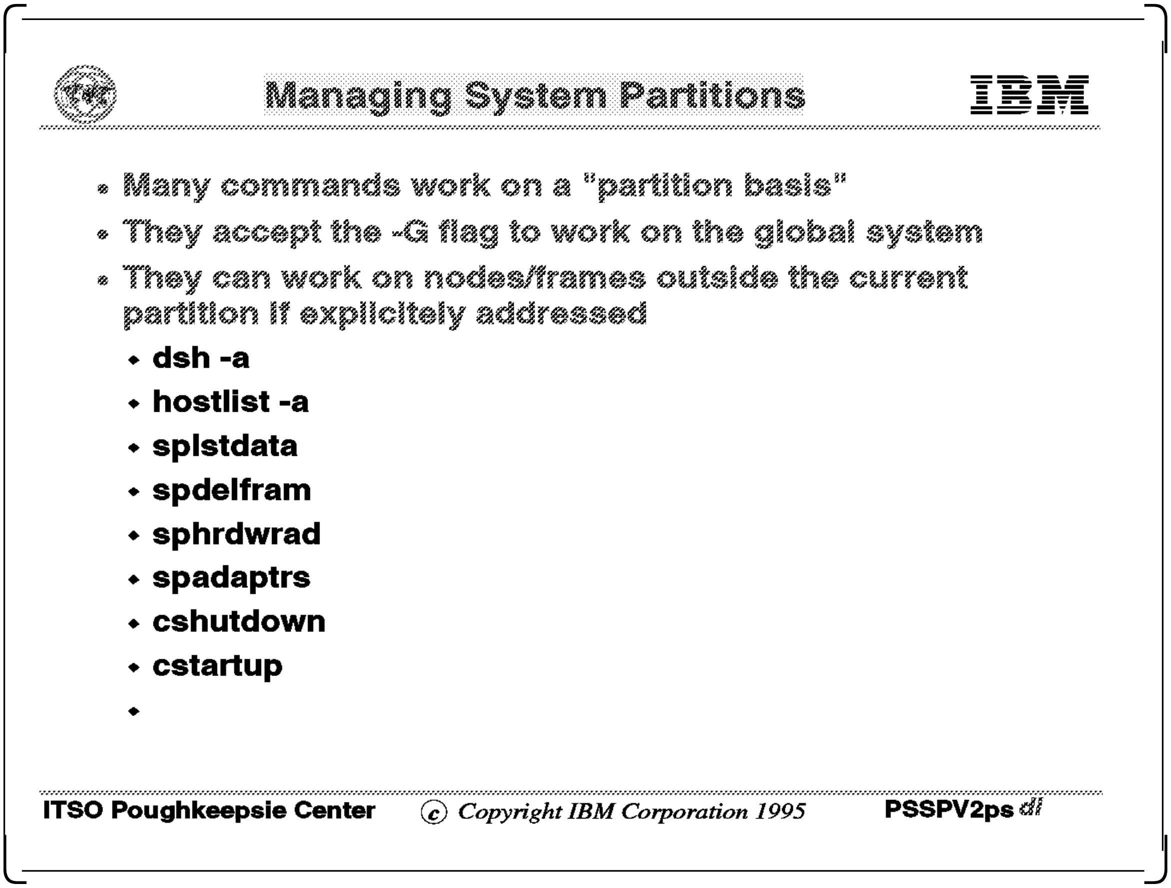
   The /etc/SDR_dest_info file on the nodes will be updated during the boot phase.

ITSO Poughkeepsie Center Ⓒ *Copyright IBM Corporation 1995* **PSSPV2ps** *dhe*

- If something goes wrong, even though step 5 was successful, you can restore the previous configuration, so that the system will be up and running (but with the old setup) while you figure out and fix the cause of failure.
- When nodes are rebooted, the *rc.sp* script does the following:
  - Gets the default partition name from */etc/SDR_dest_info* file.
  - Contacts the SDR to get the real primary name, that is, the partition name for this node.
  - Checks to see if the real name is the same as the primary in */etc/SDR_dest_info* file.
  - Updates */etc/SDR_dest_info* if necessary.

  Note: This check is done whenever the node is rebooted.

9.1 Managing System Partitions



Managing System Partitions

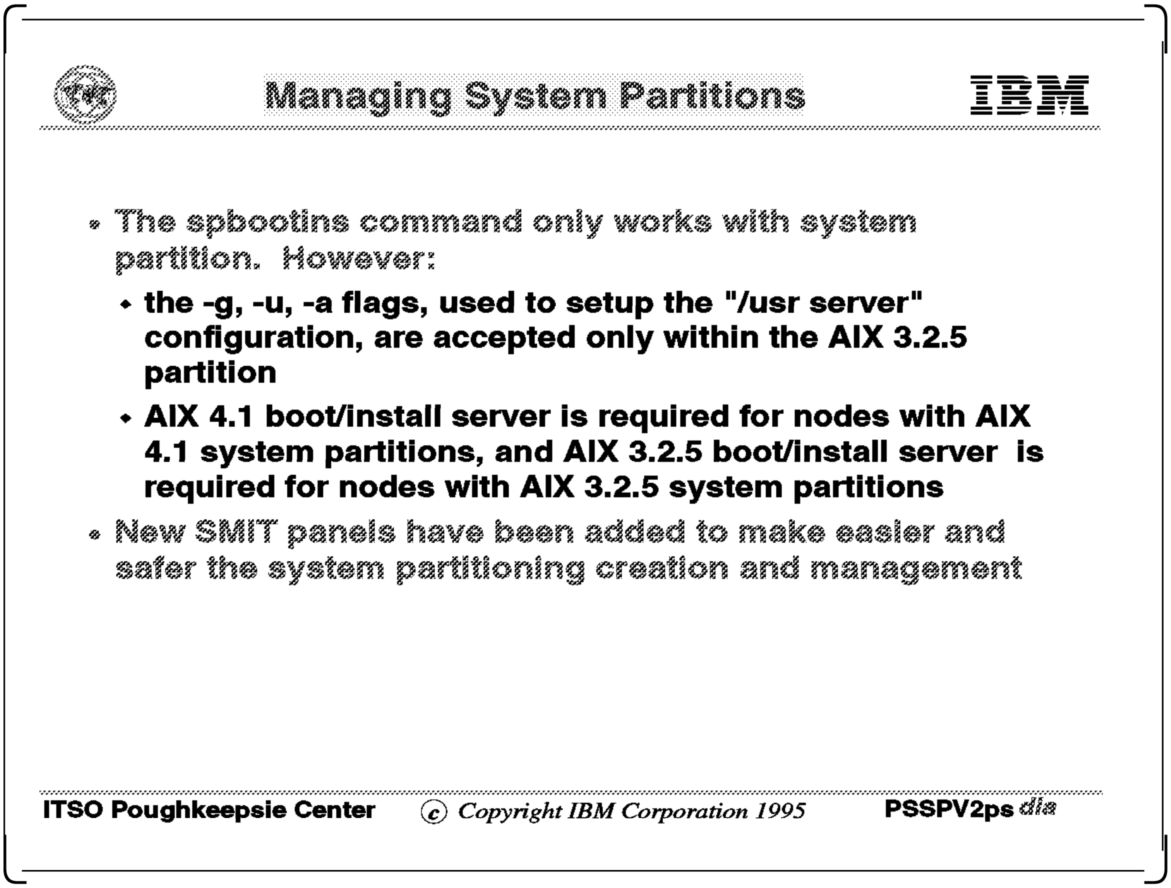
- Many commands work on a "partition basis"
- They accept the -G flag to work on the global system
- They can work on nodes/frames outside the current partition if explicitely addressed
  - **dsh -a**
  - **hostlist -a**
  - **splstdata**
  - **spdelfram**
  - **sphrdwrad**
  - **spadaptrs**
  - **cshutdown**
  - **cstartup**
  -

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* PSSPV2ps *dl*

In general, system management tasks continue to function as they did the previous release. However, many commands (commands that generate or verify lists of nodes) now work, by default, in the current partition only.

However they accept a new flag (**-G**) which means globally to be used when a particular action must be referred to the whole system. Also, these commands can reach each node outside the current partition simply naming the node.

An easy way to control the system for the system administrator is to open *n* windows, where *n* is the number of existing partitions, set the SP_NAME variable in each and use them for monitoring different partitions. Optionally, one additional window can be used for system wide operations.



Managing System Partitions

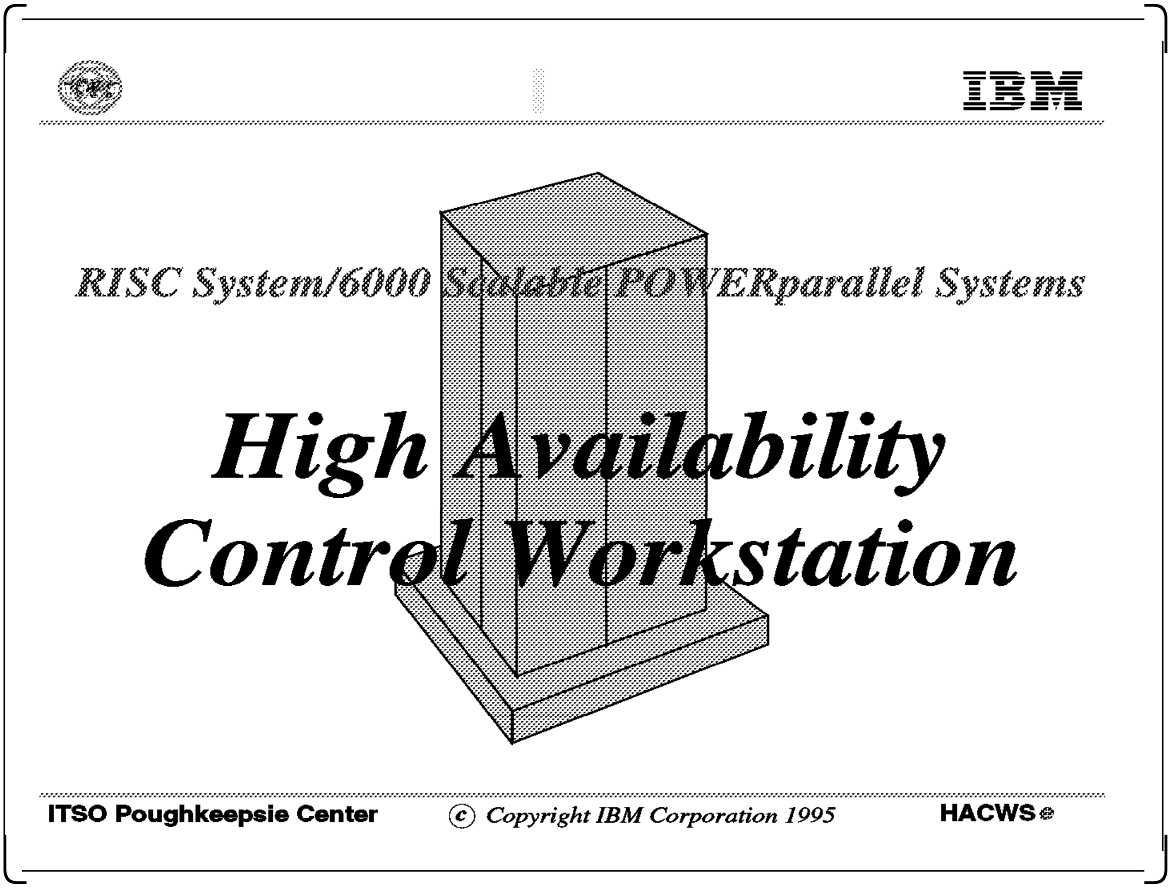
- The spbootins command only works with system partition. However:
  - the -g, -u, -a flags, used to setup the "/usr server" configuration, are accepted only within the AIX 3.2.5 partition
  - AIX 4.1 boot/install server is required for nodes with AIX 4.1 system partitions, and AIX 3.2.5 boot/install server is required for nodes with AIX 3.2.5 system partitions
- New SMIT panels have been added to make easier and safer the system partitioning creation and management

ITSO Poughkeepsie Center PSSPV2ps dia

The *spbootins command* still works regardless of system partitions. However:

- The **-g, -u, -a** flags, used to define the */usr* servers, only apply to any AIX 3.2.5 partition.

Part 4. HACWS



RISC System/6000 Scalable POWERparallel Systems

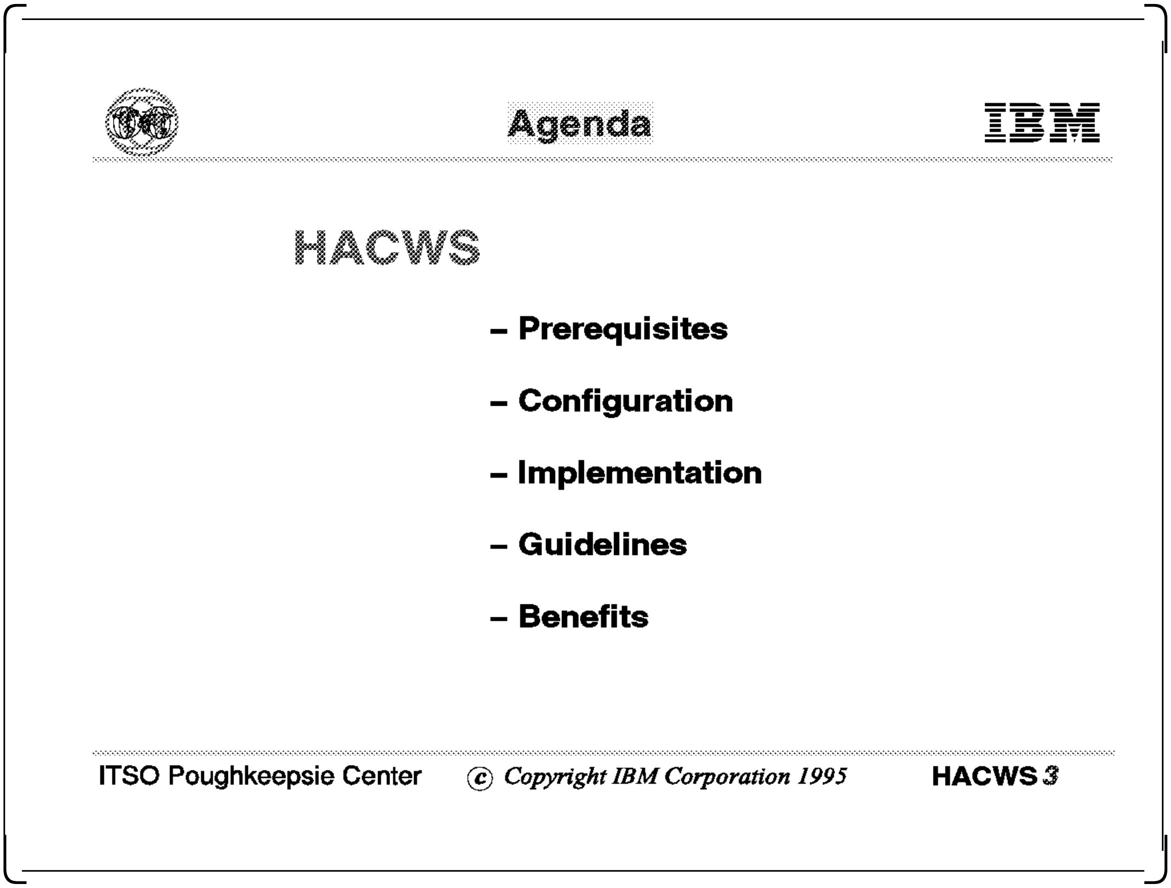
High Availability Control Workstation

ITSO Poughkeepsie Center ⓒ *Copyright IBM Corporation 1995* **HACWS**

The High Availability Control Workstation (HACWS) uses two RS/6000 workstations, a primary control workstation and a backup workstation. The concept of HACWS is modeled on the AIX High Availability Cluster Multi-Processing Licensed Program (HACMP). HACWS provides ways of eliminating the control workstation as a single point of failure.

The RS/6000 SP software with the HACWS lets a system continue to provide services critical to an installation even though a key system component - the control workstation - is no longer available. When the control workstation becomes unavailable, either through a planned event or inadvertent event, the high availability components are able to detect the loss and shift that component′s workload to a backup control workstation.

Chapter 10. HACWS Presentation



Agenda

IBM

HACWS

- Prerequisites
- Configuration
- Implementation
- Guidelines
- Benefits

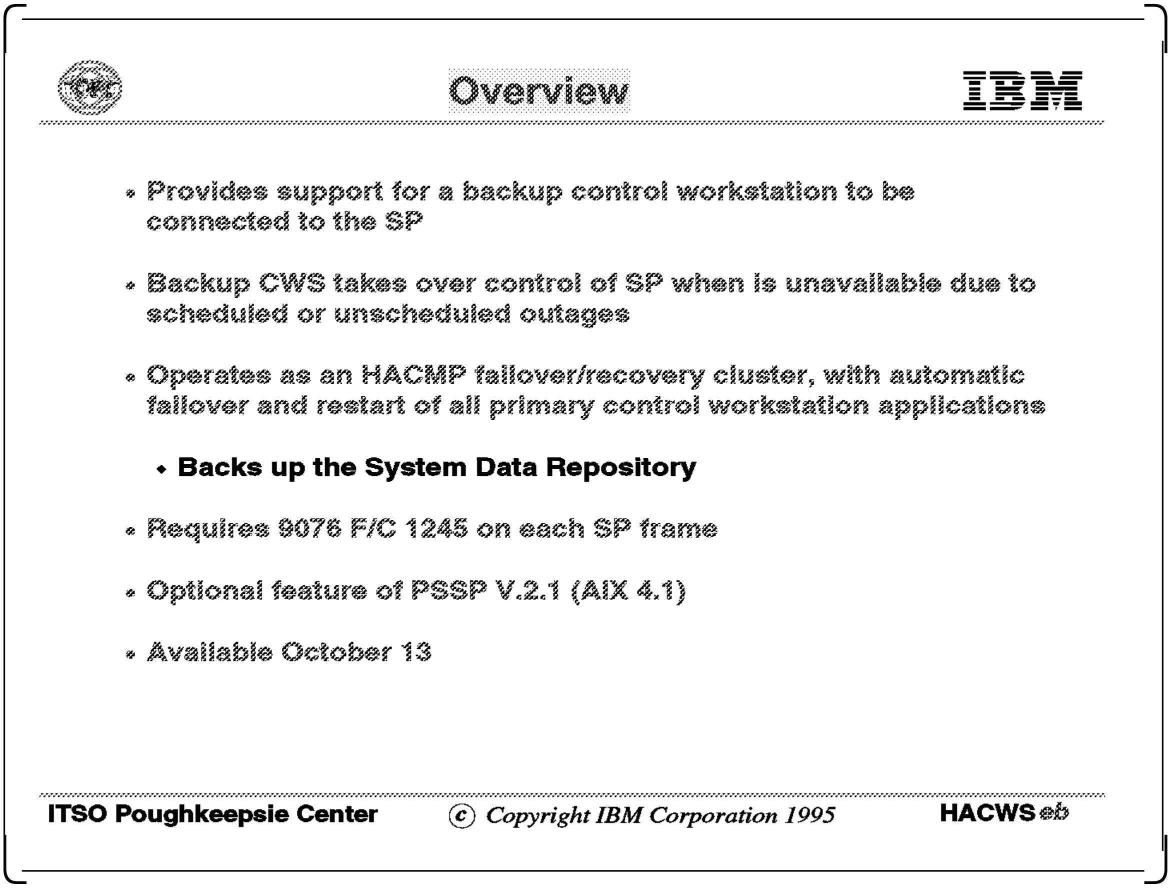
ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* HACWS 3

In this presentation we will talk about the High Availability Control Workstation (HACWS). The topics on the agenda are the following:

- The prerequisites for using HACWS.
- How a HACWS configuration should look.
- How HACWS is implemented.
- Important guidelines that you have to watch when you use HACWS.

  At the end of this chapter we will talk about great benefits for using HACWS.

10.1 Overview



Overview

- Provides support for a backup control workstation to be connected to the SP
- Backup CWS takes over control of SP when is unavailable due to scheduled or unscheduled outages
- Operates as an HACMP failover/recovery cluster, with automatic failover and restart of all primary control workstation applications
  - **Backs up the System Data Repository**
- Requires 9076 F/C 1245 on each SP frame
- Optional feature of PSSP V.2.1 (AIX 4.1)
- Available October 13

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* HACWSeb

The idea to use the backup control workstation is that the software provides script files with the ability to do an automated failover. Then the backup control workstation takes over the external disk storage and changes the IP addresses and does some other tasks, which we will discuss later on.

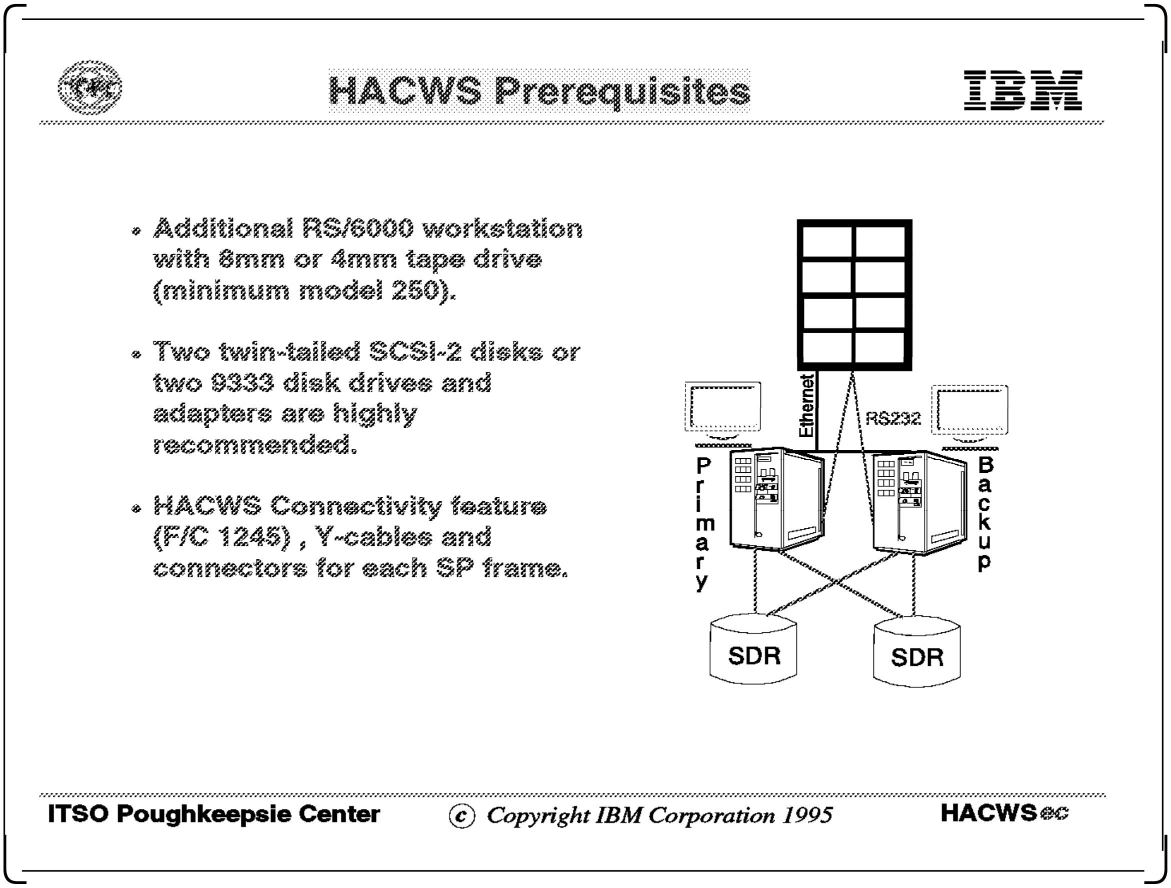
For each High Availability Control Workstation, you need a license of HACMP/6000 V4.1.

The system data for the RS/6000 SP is on an external disk. To improve the reliability of your system, you should include disk mirroring, uninterruptable power supplies (UPS), and dual disk controllers both internal and external. Since the system data repository (SDR) is the most important data to control the SP, you should use disk mirroring. You should also attach (twin tail) the external disk to both control workstations, to be safe in the event of a processor failure.

The High Availability Control Workstation software may be on the installation media, but you are required to get a license to use it.

As prerequisite hardware, you need the dual RS-232 frame supervisor card in the RS/6000 SP system, and for the control workstations you need the external mirrored disks, and a serial TTY link or target mode SCSI.

10.2 HACWS Prerequisites



HACWS Prerequisites

- Additional RS/6000 workstation with 8mm or 4mm tape drive (minimum model 250).
- Two twin-tailed SCSI-2 disks or two 9333 disk drives and adapters are highly recommended.
- HACWS Connectivity feature (F/C 1245) , Y-cables and connectors for each SP frame.

ITSO Poughkeepsie Center

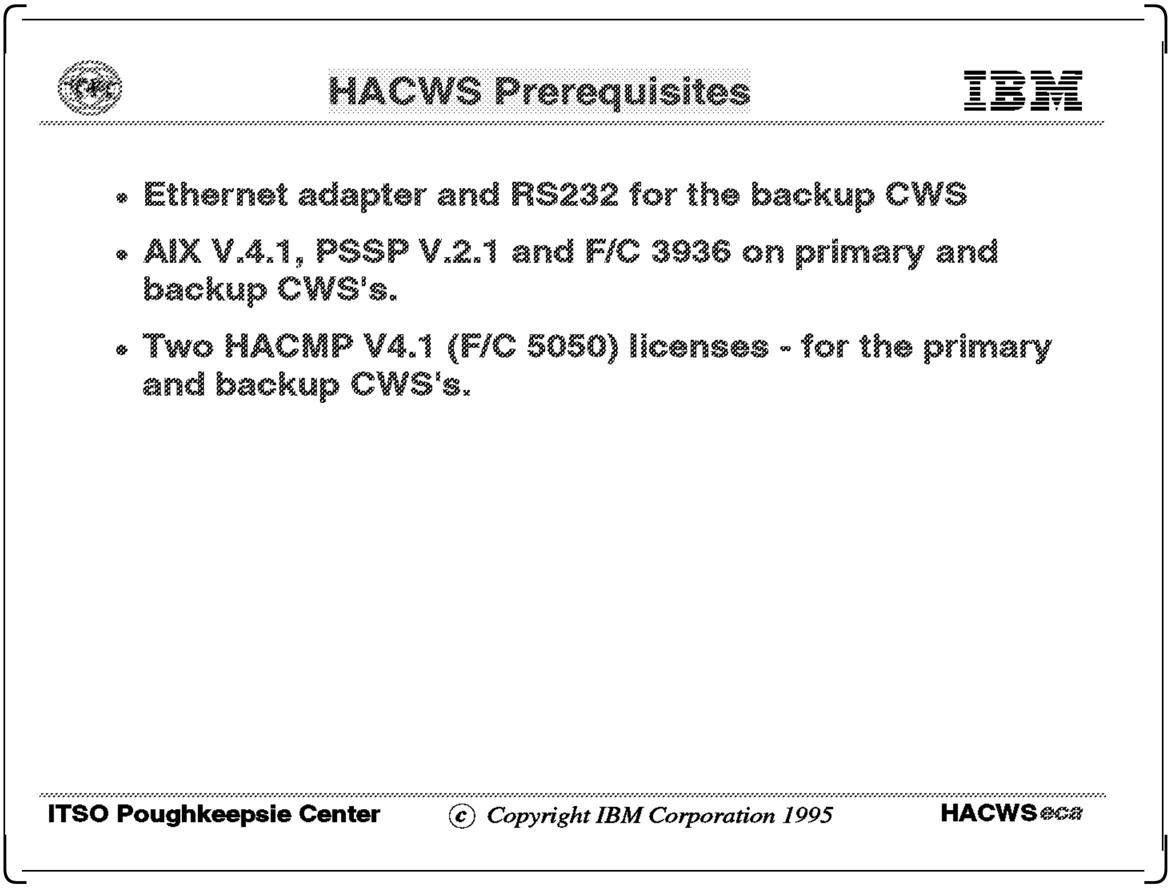
© Copyright IBM Corporation 1995

HACWSec

What are the prerequisites to use HACWS?

For ease of use, you should have the same configuration for each control workstation. The minimum control workstation for a maximum of 16 nodes (one frame) should be an RS/6000 Model C20. You could use a Model 250, but due to slot limitations, a single-point-of-failure is unavoidable in shared disk or shared network resources. Every workstation must have a 4mm or a 8mm tape drive and a minimum of 64MB memory. You need two twin-tailed SCSI-2 Differential disks or two 9333 disk drives and two adapter cards for each workstation. For different configurations, refer to the HACMP/6000 documentation.

On each node you need the HACWS Connectivity feature (feature code 1245), which is the frame supervisor card, a TTY Y cable that connects to both control workstations.



HACWS Prerequisites

- Ethernet adapter and RS232 for the backup CWS
- AIX V.4.1, PSSP V.2.1 and F/C 3936 on primary and backup CWS's.
- Two HACMP V4.1 (F/C 5050) licenses - for the primary and backup CWS's.

ITSO Poughkeepsie Center ⓒ *Copyright IBM Corporation 1995* HACWSeca

HACWS does not require an extra point-to-point TCP/IP network. However, a serial network should be added and identified to HACMP in order to guard against a failure of the TCP/IP subsystem. HACMP supports two types of serial networks;

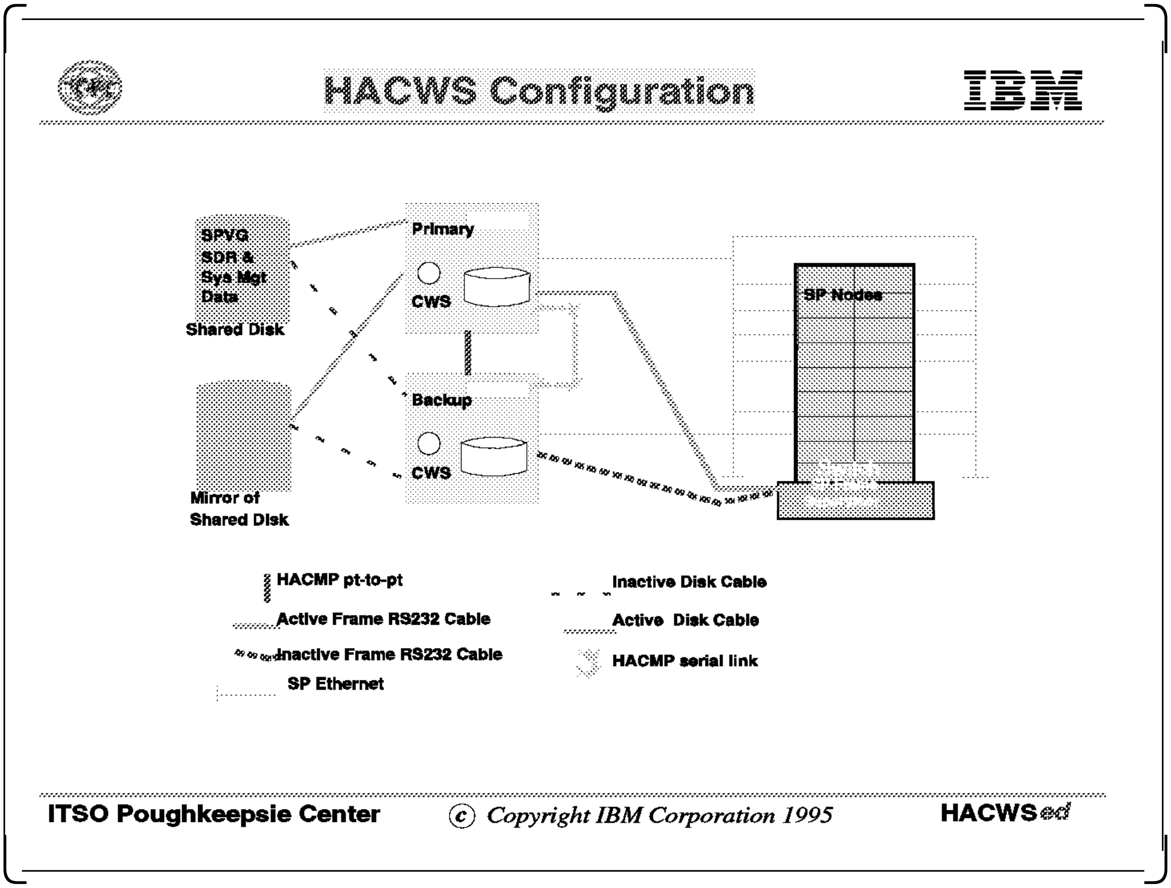
- RS232 serial line connection
- SCSI-2 differential bus using target mode SCSI

Ethernet will be the preferred network because it is cheaper than other network connectivity.

For HACWS, every node and both control workstations have to be on AIX 4.1.3 and have to use PSSP V2.1. The F/C 3936 is the HACWS software part of PSSP V2.1 and has to be ordered for both control workstations.

For both control workstation, you need a license of HACMP/6000 V4.1 (F/C 5050).

10.3 Configuration



HACWS Configuration

- HACMP pt-to-pt
- Active Frame RS232 Cable
- Inactive Frame RS232 Cable
- SP Ethernet
- Inactive Disk Cable
- Active Disk Cable
- HACMP serial link

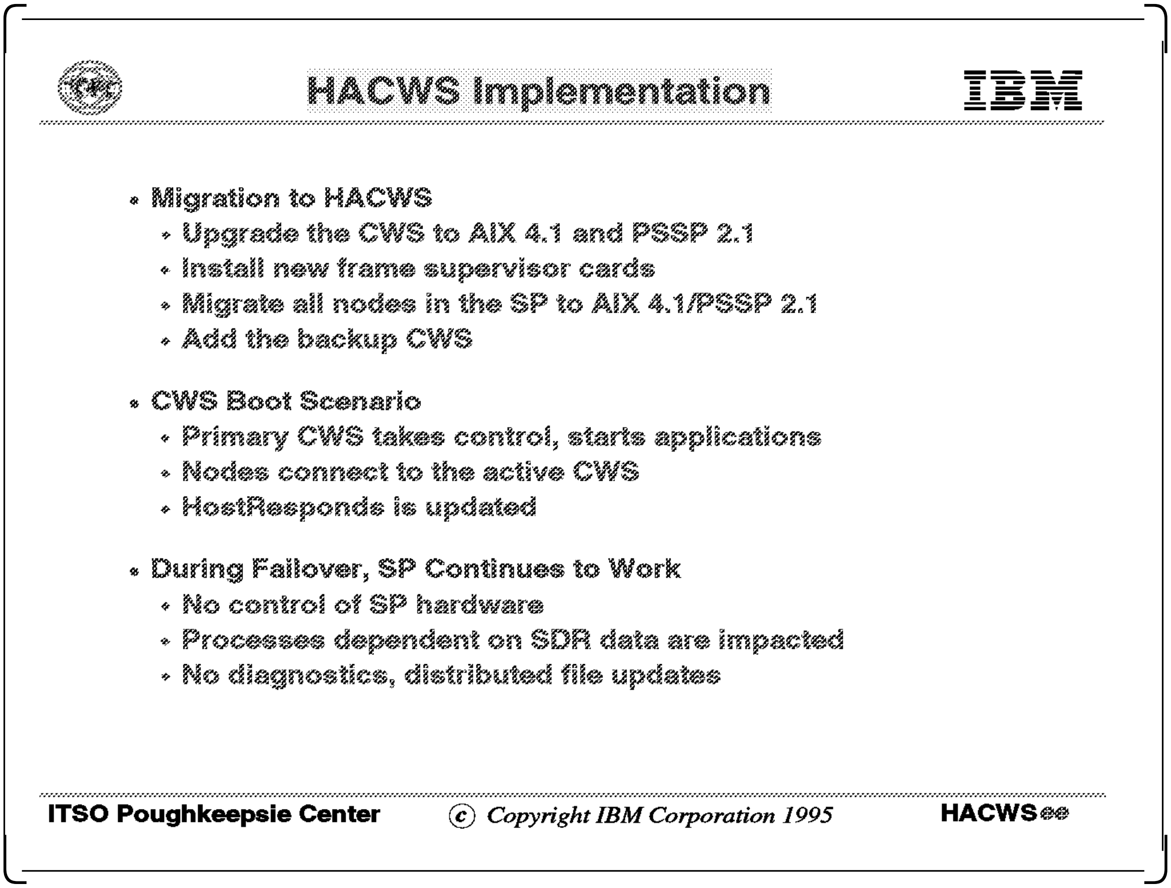
ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* HACWSed

The RS/6000 SP system looks similar except that there are two control workstations connected to the SP Ethernet and TTY network. The frame supervisor TTY network is modified to add a standby link. The second control workstation is the backup. In this picture you see a logical view of a High Availability Control Workstation. You see also disk mirroring, a very important part of high availability planning.

If the primary control workstation fails, there is a disruptive failover that does the following:

- The external disk storage is switched to the backup control workstation.
- The hardware and IP addresses are switched to the backup control workstation.
- The control workstation applications are restarted.
- The file systems are remounted.
- Hardware monitoring is resumed.
- Clients are allowed to reconnect to obtain data or to update the System Data Repository (SDR).

Chapter 11. HACWS Implementation



HACWS Implementation

- Migration to HACWS
  - Upgrade the CWS to AIX 4.1 and PSSP 2.1
  - Install new frame supervisor cards
  - Migrate all nodes in the SP to AIX 4.1/PSSP 2.1
  - Add the backup CWS
- CWS Boot Scenario
  - Primary CWS takes control, starts applications
  - Nodes connect to the active CWS
  - HostResponds is updated
- During Failover, SP Continues to Work
  - No control of SP hardware
  - Processes dependent on SDR data are impacted
  - No diagnostics, distributed file updates

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* HACWSee

If you are migrating to HACWS, then you have to install on your control workstation AIX 4.1.3 and PSSP 2.1. You have to install the new frame supervisor cards on all nodes. Then you have to install or migrate to AIX 4.1.3 on all nodes. Every node also needs the latest PSSP 2.1 software release. Finally you add the backup control workstation to your RS/6000 SP system and configure HACMP/6000.

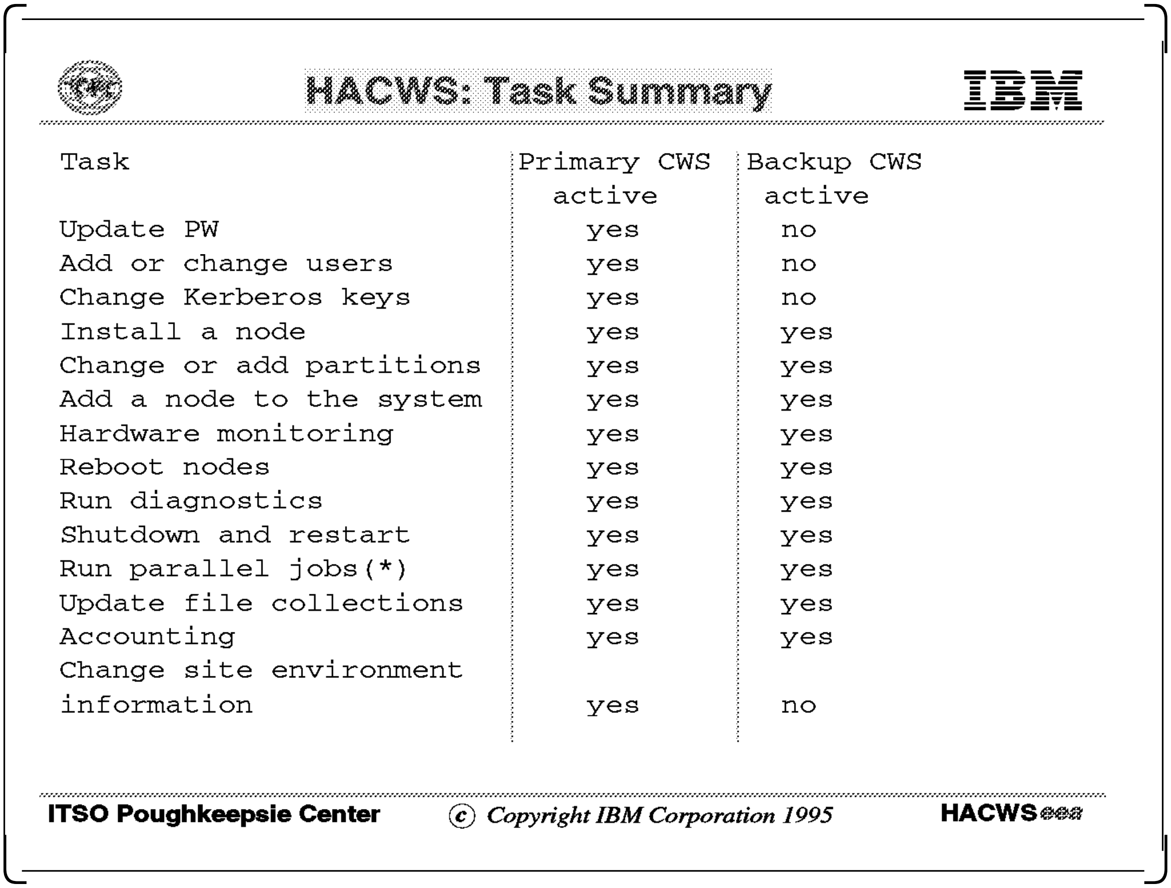
During a normal boot, the primary control workstation takes control and starts the AIX daemons, the PSSP daemons and applications. All nodes connect to the active control workstation. Typical applications that rely on connectivity to the control workstation are the Resource Manager and the Host Response daemon.

The failover is disruptive, but the SP nodes continue to do work. Applications at the control workstation that are interrupted will not resume automatically; they must be restarted. Applications within nodes that require no communication with the control workstation may not notice the failover. In an environment where the Resource Manager allocates nodes, an outage of hardware or software on the control workstation causes the entire system to stop. The Resource Manager will fail when it loses connection to the SDR. New parallel jobs cannot be started and existing parallel jobs cannot be controlled.

The Resource Manager is *not* restarted when control is transferred to a backup control workstation. The operator must decide whether to kill the currently running parallel jobs (suspend will not work) and restart the Resource Manager, or to wait until all running jobs complete and then restart the Resource Manager.

New parallel jobs cannot start until the Resource Manager is restarted.

11.1 HACWS Task Summary



HACWS: Task Summary

Task	Primary CWS active	Backup CWS active
Update PW	yes	no
Add or change users	yes	no
Change Kerberos keys	yes	no
Install a node	yes	yes
Change or add partitions	yes	yes
Add a node to the system	yes	yes
Hardware monitoring	yes	yes
Reboot nodes	yes	yes
Run diagnostics	yes	yes
Shutdown and restart	yes	yes
Run parallel jobs(*)	yes	yes
Update file collections	yes	yes
Accounting	yes	yes
Change site environment information	yes	no

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* HACWS

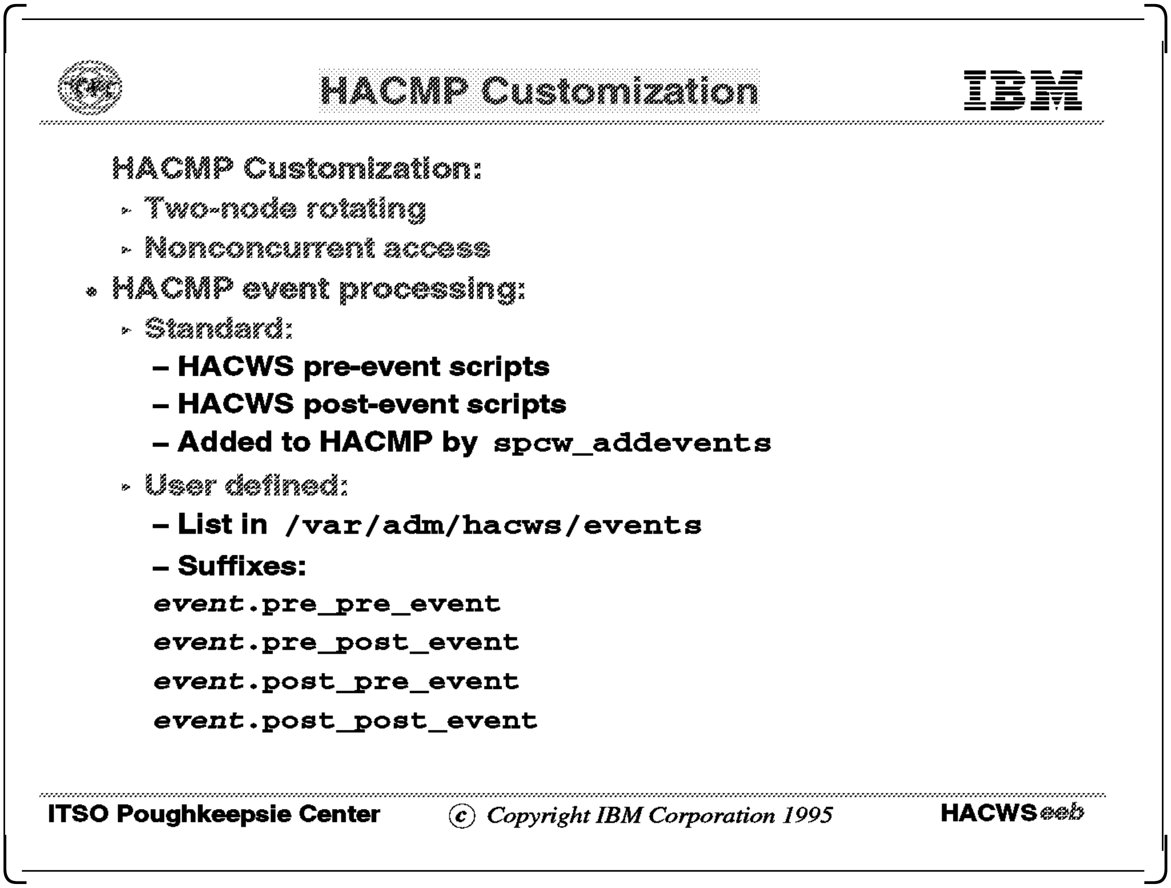
On this foil, are presented the tasks available either on the primary control workstation or on the backup control workstation, and the tasks that are allowed only on the primary control workstation.

The main operations on the hardware, such as hardware monitoring, installing a node, rebooting a node, are available on the primary and backup control workstations, and prevent the computing center from control workstation and monitor function unavailability.

However, several tasks related to the security functions, either AIX or Kerberos, are not available when the backup control workstation is active. Such functions need the primary control workstation to be active.

Also, several directories and databases, which are updated when you change the RS/6000 SP hardware configuration or the partitions definitions may reside on the control workstation local disk. In such cases, the possible changes, if not on shared external disks must be propagated to the primary control workstation to avoid any inconsistency.

11.2 HACMP Customization



HACMP Customization

- HACMP Customization:
  - Two-node rotating
  - Nonconcurrent access
- HACMP event processing:
  - Standard:
    - HACWS pre-event scripts
    - HACWS post-event scripts
    - Added to HACMP by `spcw_addevents`
  - User defined:
    - List in `/var/adm/hacws/events`
    - Suffixes:

      event`.pre_pre_event`

      event`.pre_post_event`

      event`.post_pre_event`

      event`.post_post_event`

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* HACWSeeb

When you customize HACMP, you have to select the following options:

- The control workstation cluster will be defined as a two-node rotating cluster.

  It means that, when you boot your control workstations, the one where the HACMP cluster manager is started first will get the shared resources and become the active control workstation. To avoid the active control workstation be the wrong one, you may manually start the control workstations and favor the one you want it to get the shared resources. Otherwise, operators will have to use the clstart and the clstop commands to swap the active and inactive control workstations.

- The control workstation cluster must be defined with the non concurrent option.

  In fact, this option is only available for raw disks and does not support journaled file systems, which exclude the /spdata file system that contains the SDR data base.

HACMP provides exits before and after the event manager executes the standard event script. HACWS uses this feature to include its own set of event scripts. These scripts are loaded in the /usr/sbin/hacws directory and defined to HACMP by the command *spcw_addevents*.

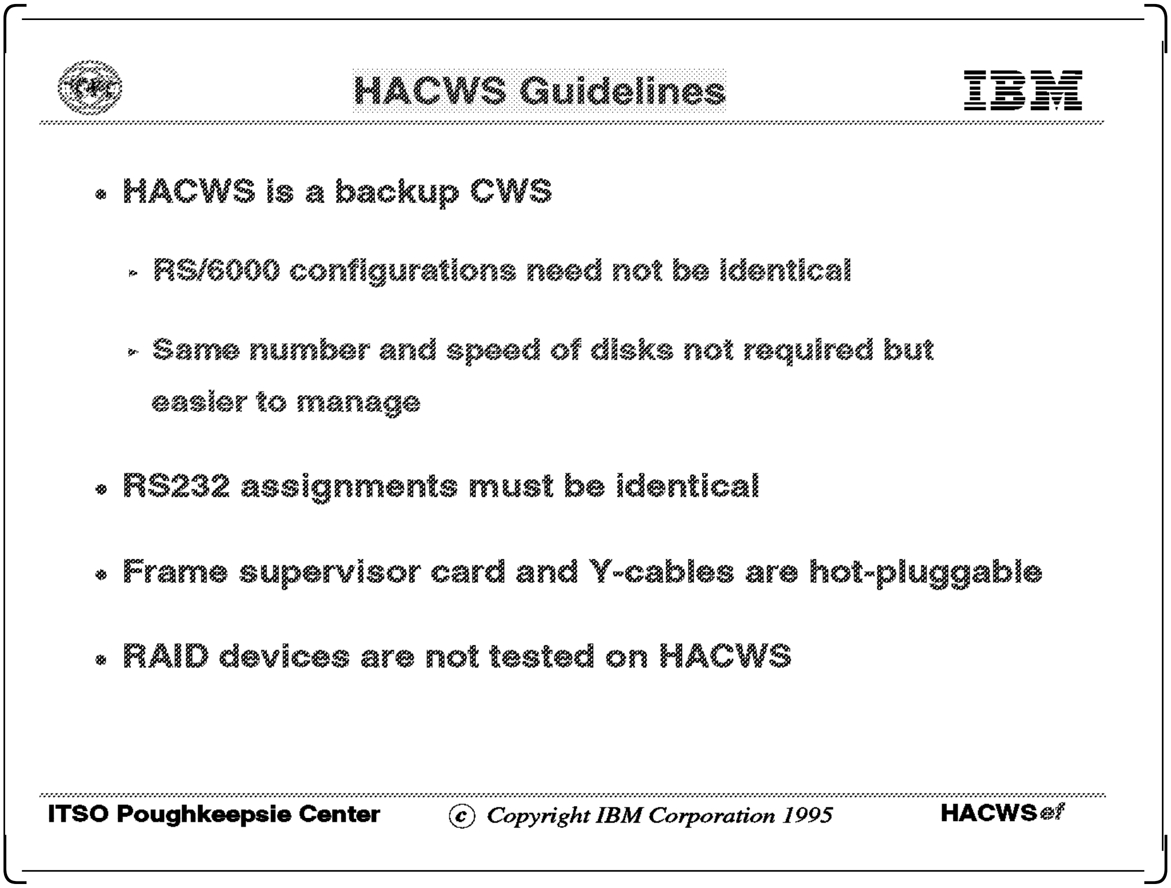
HACWS allows users to develop their own exit scripts. Such scripts are stored in the /var/adm/hacws/events directory and must respect the following name conventions:

*event_name***pre_pre_event**
*event_name***pre_post_event**
*event_name***post_pre_event**
*event_name***post_post_event**

So, an event will start the following execution:

```
HACWS customer supplied pre_pre_event
    HACWS pre_event Script
HACWS customer supplied pre_post_event
    HACMP event script
HACWS customer supplied post_pre_event
    HACWS post_event Script
HACWS customer supplied post_post_event
```

11.3 Guidelines



HACWS Guidelines

- HACWS is a backup CWS
  - RS/6000 configurations need not be identical
  - Same number and speed of disks not required but easier to manage
- RS232 assignments must be identical
- Frame supervisor card and Y-cables are hot-pluggable
- RAID devices are not tested on HACWS

ITSO Poughkeepsie Center ⓒ *Copyright IBM Corporation 1995* HACWS*ef*

Since HACWS is a backup control workstation, it cannot be used as a control workstation (CWS) for another SP system. You can have only one control workstation online. There is a one-to-one relationship, so a single primary and backup control workstation combination can be used to control only one RS/6000 SP system.

The RS/6000 workstations do not have to be identical, but it is highly recommended for ease of use. Some components must be identical, others can be similar. If you have the same number and type of disks on each, your planning and operation will be simpler. Otherwise you might have to plan HACMP recovery scripts that address lv01 on one control workstation and lv04 on the other. TTY assignments on each CWS must be identical and should be configured in the same slot of each.

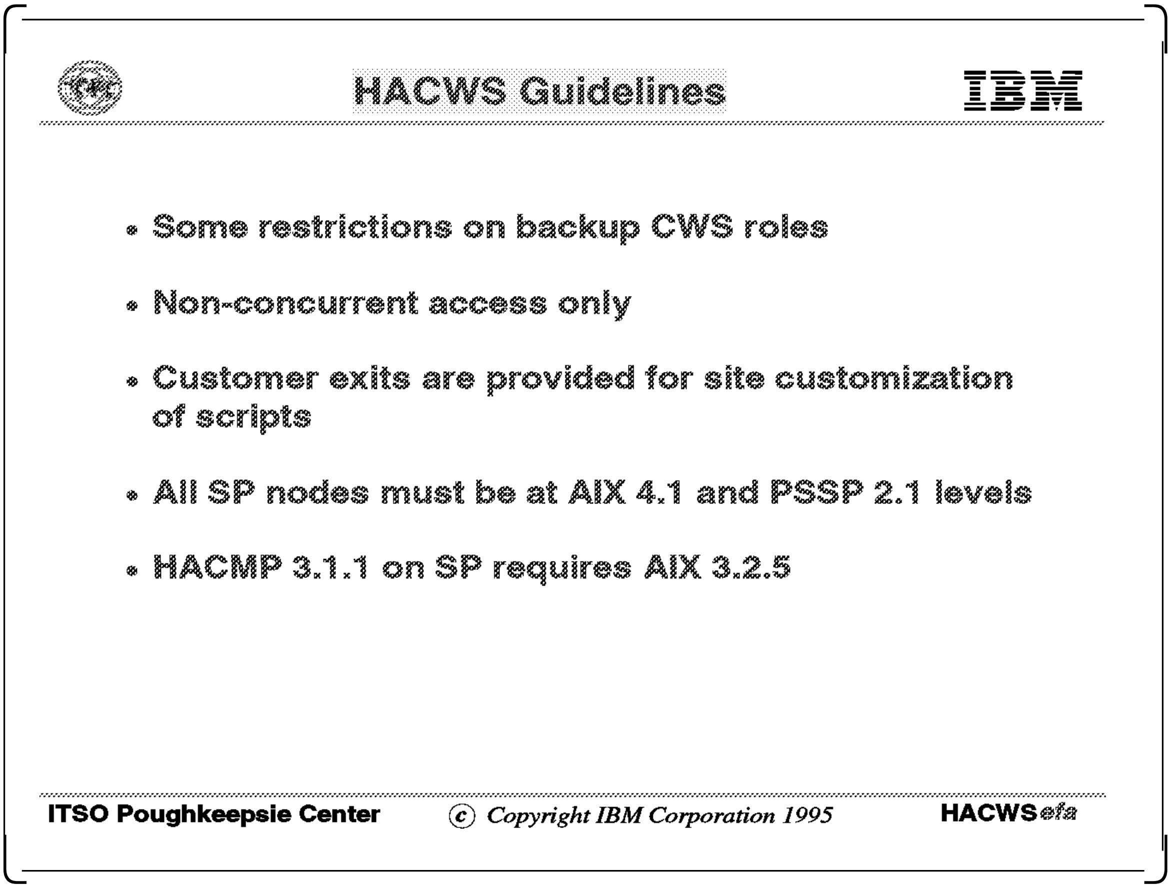
The frame supervisor card and the Y-cables are hot pluggable from the SP nodes.

As you know from the HACMP announcement letter, HACMP for AIX 4.1 supports RAID devices like the:

- IBM 7135 RAIDiant Array Model 210
- IBM 3514 Disk Array Subsystem Models 212 or 213

- IBM 7137 Disk Array Subsystem Models 412, 413, 414, 512, 513, or 514

These devices will be supported in the future.



HACWS Guidelines

- Some restrictions on backup CWS roles
- Non-concurrent access only
- Customer exits are provided for site customization of scripts
- All SP nodes must be at AIX 4.1 and PSSP 2.1 levels
- HACMP 3.1.1 on SP requires AIX 3.2.5

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* HACWS*efa*

The High Availability Control Workstation has the following restrictions:

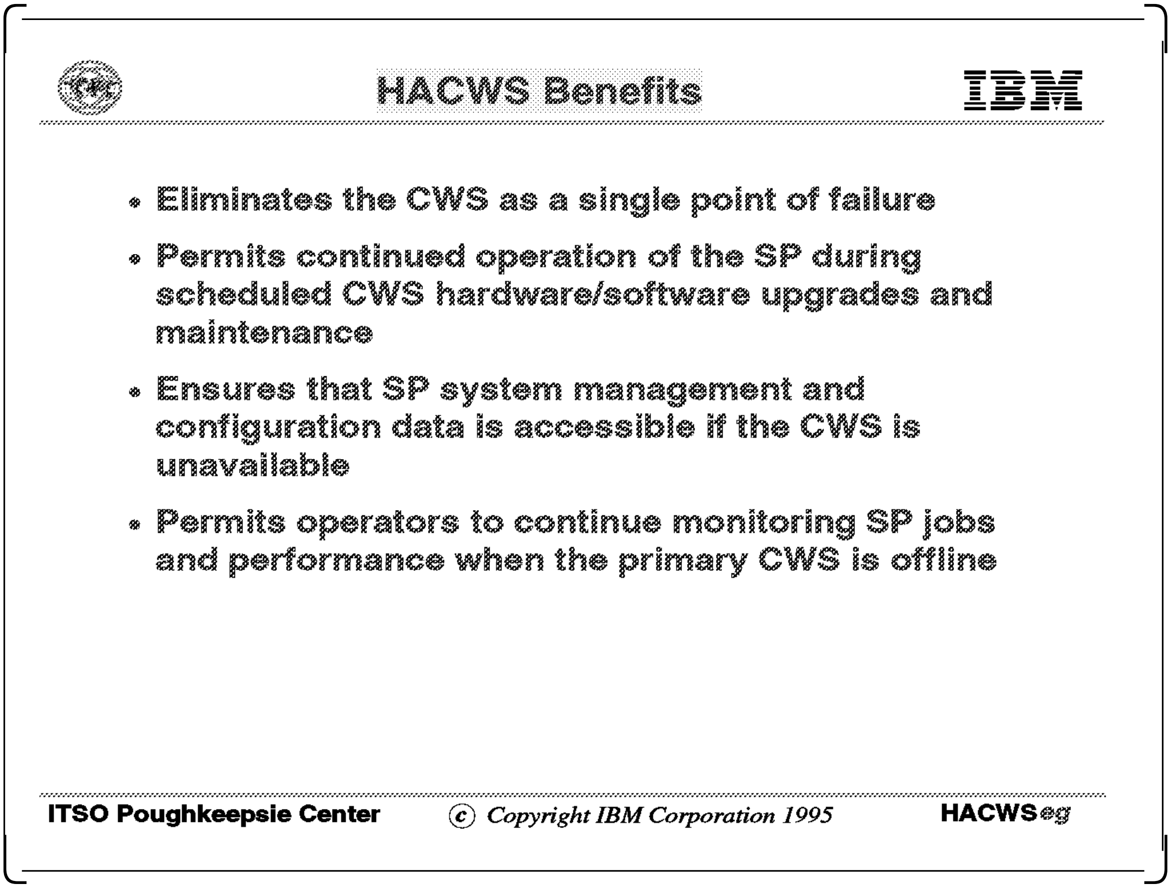
- The primary and backup control workstations must each be a RISC System/6000 workstation.
- The backup control workstation cannot be used as a control workstation for another RS/6000 SP system.
- You cannot split the load across a primary and backup control workstation. Either the primary or the backup provides all the function at one time.
- The backup control workstation cannot be a shared backup of two primary control workstations.

HACMP for AIX 4.1 provides customer exits in the site customization scripts.

HACMP Support on the SP nodes

All SP nodes must be at AIX 4.1.3 and PSSP 2.1 levels, but the only HACMP version for the SP nodes is HACMP 3.1.1 and this version requires AIX 3.2.5.

11.4 Benefits



HACWS Benefits

- Eliminates the CWS as a single point of failure
- Permits continued operation of the SP during scheduled CWS hardware/software upgrades and maintenance
- Ensures that SP system management and configuration data is accessible if the CWS is unavailable
- Permits operators to continue monitoring SP jobs and performance when the primary CWS is offline

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 HACWSeg

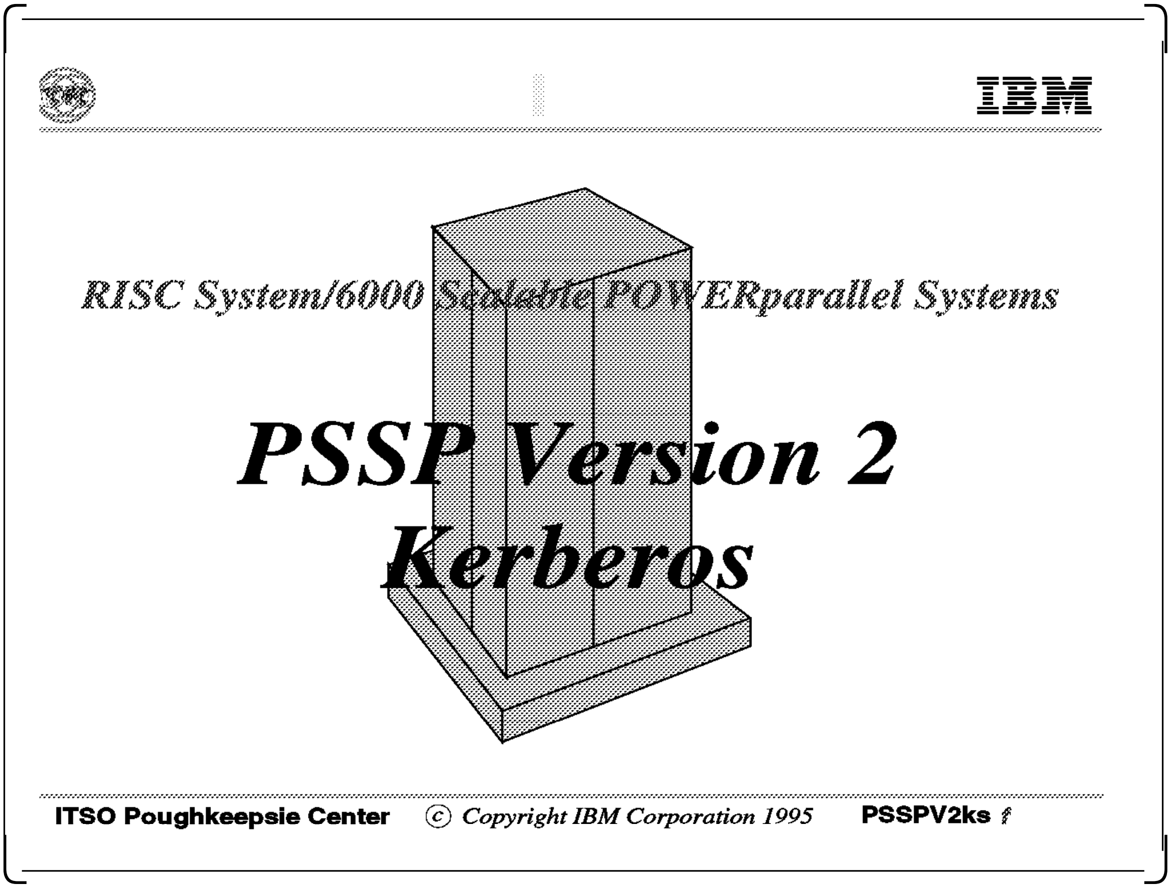
The High Availability Control Workstation is a major component in the effort to reduce the possibility of single point of failure in the RS/6000 SP system. There are many elements of hardware and software that could fail on a control workstation (CWS). In prior releases, a CWS failure usually quickly led to an unusable system.

On the nodes, you have already redundant power supplies and can replace single wide nodes when they break. With the backup control workstation your, SP system will continue to run during scheduled CWS hardware or software upgrades. Now you can power down for maintenance without affecting the entire SP system.

All SP system management and configuration data is accessible on the external mirrored disk. When the primary control workstation is not available, the backup control workstation takes over the disks and has access to the SP data.

Operators can start the resource manager on the backup control workstation and parallel job submission will work. This would not be possible without HACWS, when the primary control workstation is down.

Part 5. Kerberos



RISC System/6000 Scalable POWERparallel Systems

PSSP Version 2
Kerberos

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* **PSSPV2ks**

This section describes Kerberos, the sofware that securely authenticates the users and servers of certain administrative services on the SP. Currently, the PSSP functions protected by Kerberos are the **system monitor**, commands such as **dsh**, **rsh**, **rcp** and the **sysctl** application.

Kerberos is a public domain software package developed at MIT as part of project Athena. It was initially designed by Steve Miller and Clifford Neuman with suggestions from Jeff Schiller and Jerry Saltzer.

11.4.1 References

This presentation is based on the ideas presented in the following papers and manuals:

1. *The Data Encryption Standard (DES) and Its Strength Against Attacks*, D. Coppersmith, IBM J. Res. & Dev. v 38 n 3, May 1994
2. *Kerberos: An Authentication Service for Open Network Systems* , by Jennifer G. Steiner, Clifford Neuman and Jeffrey I. Schiller, Usenix Conference Proceedings, Dallas TX, 1988
3. *Secure Distributed Computing*, by Jeffrey I. Schiller, Scientific American, Nov. 1994

4. *IBM RISC System/6000 Scalable POWERparallel Systems Administration Guide*, IBM (GC23-3897), 1995

5. *IBM RISC System/6000 Scalable POWERparallel Systems Command and Technical Reference*, IBM (GC23-3900), 1995

11.4.2 Bibliography

Of the many publications on the subject of Kerberos and distributed systems security, the following are suggested as additional reading on the subject:

1. *RS/6000 SP System Management: Easy, Lean and Mean*, IBM ITSO (GG24-2563), 1995

2. *National Bureau of Standards "Data Encryption Standard"*, Federal Information Processing Standards Publication 46, Government Printing Office, Washington DC, January 1977

3. *Using Encryption for Authentication in Large Networks of Computers*, R. M. Needham and M. D. Schroeder, Communications of the ACM, Vol. 21, No. 12, 1978

4. *Distributed Computing and Management Strategies*, edited by Raman Khanna, Prentice Hall, 1993

5. *So You Think Your Information Is Secure - Is It?*, Roger Lesser, Defense Electronics v 27 n 5 May 1995

6. *Kerberos. An Authentication Service for Computer Networks*, B. C. Neuman, and T. Ts′o, IEEE Communications Magazine v 32 n 9, Sep 1994

7. *Building a Secure Communications Network*, Timothy Erman, Telecommunications (Americas Edition) v 29 n 2, Feb 1995

8. *Security Architecture for Distributed Systems*, Sead Muftic, and Morris Sloman, Computer Communications v 17 n 7 Jul 1994

9. *Authentication Services in Distributed Systems*, Dieter Gollmann, Thomas Beth and Frank Damm, Computers & Security v 12 n 8 Dec 1993

10. *Authentication and Authorization Techniques in Distributed Systems*, Claude Laferriere and Richard Charland, Proceedings of the 1993 IEEE International Carnahan Conference on Security Technology

11. *Authentication for Distributed Systems*, Thomas Y. C. Woo and Simon S. Lam, Computer v 25 n 1 Jan 1992

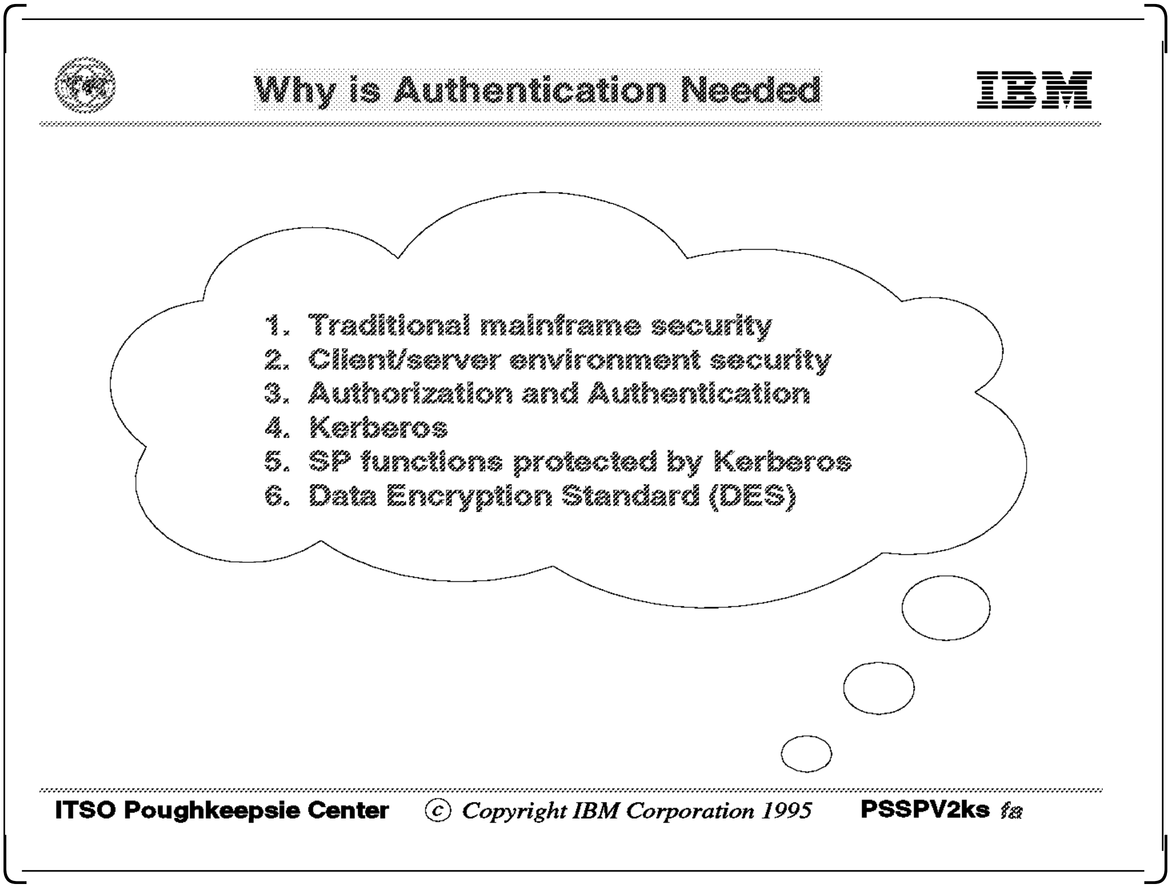
12. *A Modular Family of Secure Protocols for Authentication and Key Distribution*, R. Bird, I. Gopal, A. Herzberg, P. Janson, S. Kutten, R. Molva, and M. Yung, IBM Research Report RZ-2402, November 1992

13. *Distributed System and Security Management with Centralized Control*, C. Tsai, and V. D. Gligor, IBM Technical Report TR-85.0155, August 1992.

14. *Secure and Inexpensive Authentication with Minimalist Smartcards*, R. Molva, G. Tsudik, IBM Research Report RZ-2315, May 1992

15. *Digital Signatures and Authentication*, A. G. Konheim, IBM Research Report RC-8074, January 1980

Chapter 12. Why is Authentication Needed?



Why is Authentication Needed

1. Traditional mainframe security
2. Client/server environment security
3. Authorization and Authentication
4. Kerberos
5. SP functions protected by Kerberos
6. Data Encryption Standard (DES)

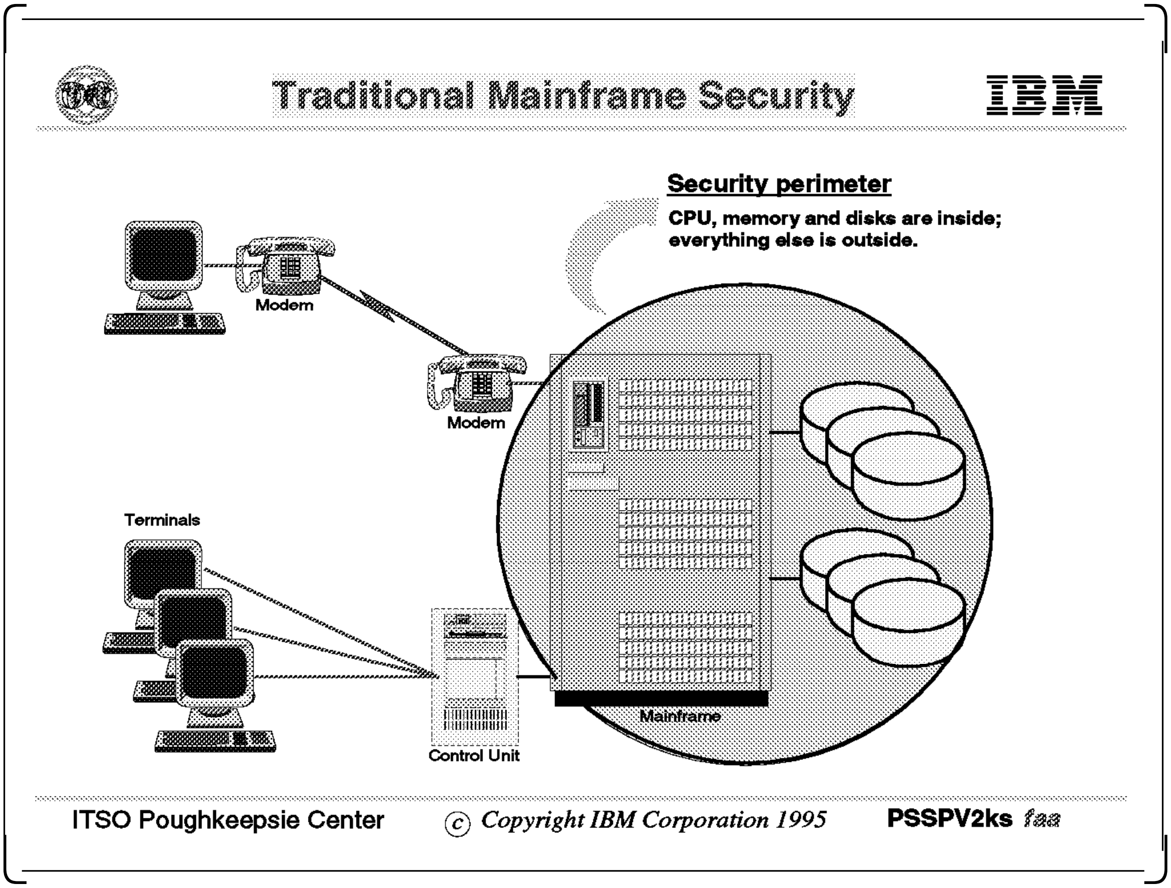
ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ks fa

In the following foils we explain the motivation for secure authentication in client/server systems, also referred to as distributed or networked systems.

The following topics are discussed:

1. Traditional mainframe security
2. Client/server environment security
3. Authorization and authentication
4. Kerberos
5. SP functions protected by Kerberos
6. Data Encryption Standard (DES)

12.1 Traditional Mainframe Security



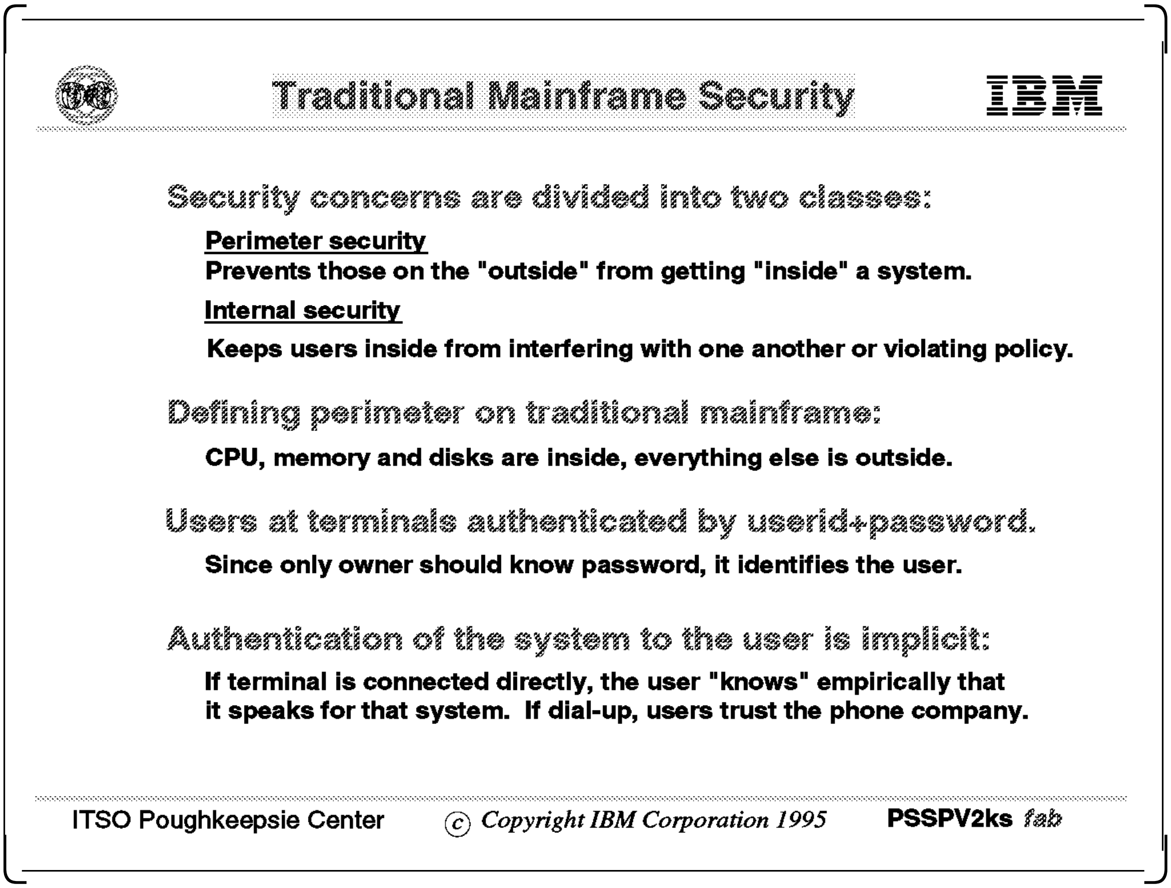
Traditional Mainframe Security

Security perimeter

CPU, memory and disks are inside; everything else is outside.

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ks faa

The foil shows a mainframe configuration. Traditionally, specialists have divided their concerns into two classes: *perimeter security*, which prevents those on the outside from getting inside a system, and *internal security*, which keeps users inside from interfering with one another or otherwise violating the security policy of the system.



Traditional Mainframe Security

Security concerns are divided into two classes:

Perimeter security
Prevents those on the "outside" from getting "inside" a system.

Internal security
Keeps users inside from interfering with one another or violating policy.

Defining perimeter on traditional mainframe:

CPU, memory and disks are inside, everything else is outside.

Users at terminals authenticated by userid+password.

Since only owner should know password, it identifies the user.

Authentication of the system to the user is implicit:

If terminal is connected directly, the user "knows" empirically that it speaks for that system. If dial-up, users trust the phone company.

ITSO Poughkeepsie Center ⓒ *Copyright IBM Corporation 1995* **PSSPV2ks** *fab*

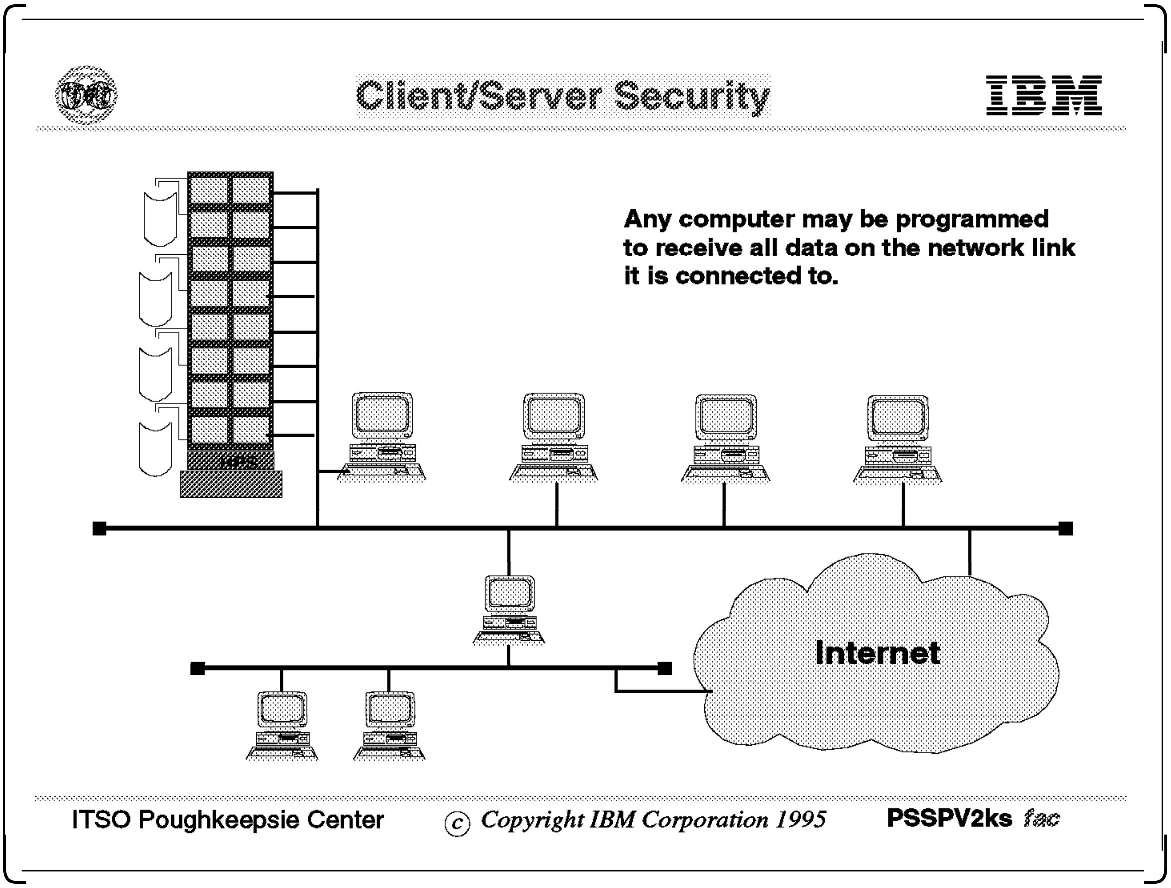
Defining the perimeter of a traditional mainframe system is fairly simple. The central processing unit, memory and disk drives are inside, and everything else is outside. Input/output devices such as tape drives or terminals *are* the perimeter; any information entering the system must pass through them.

When users sit down at a terminal, for example, they authenticate themselves by typing an account name and password. Only the owner of a particular account should know its password, and so the two in conjunction suffice to identify the user to the system.

The authentication of the system to the user is implicit; the user assumes that he or she is communicating with the intended computer. If the terminal is connected directly to a specific computer, the user knows empirically that the terminal *is* that system. If users dial a mainframe through a modem, they trust the telephone company to connect the telephone call to the computer system that corresponds to the dialed number.

As long as a computer′s internal security is not subverted, the mutual trust between user and mainframe remains in force until the user logs out. The system assumes that all the keystrokes it receives have been typed on behalf of the user. Similarly, the user assumes that all information appearing on the screen comes from the appropriate computer.

12.2 Client/Server Security

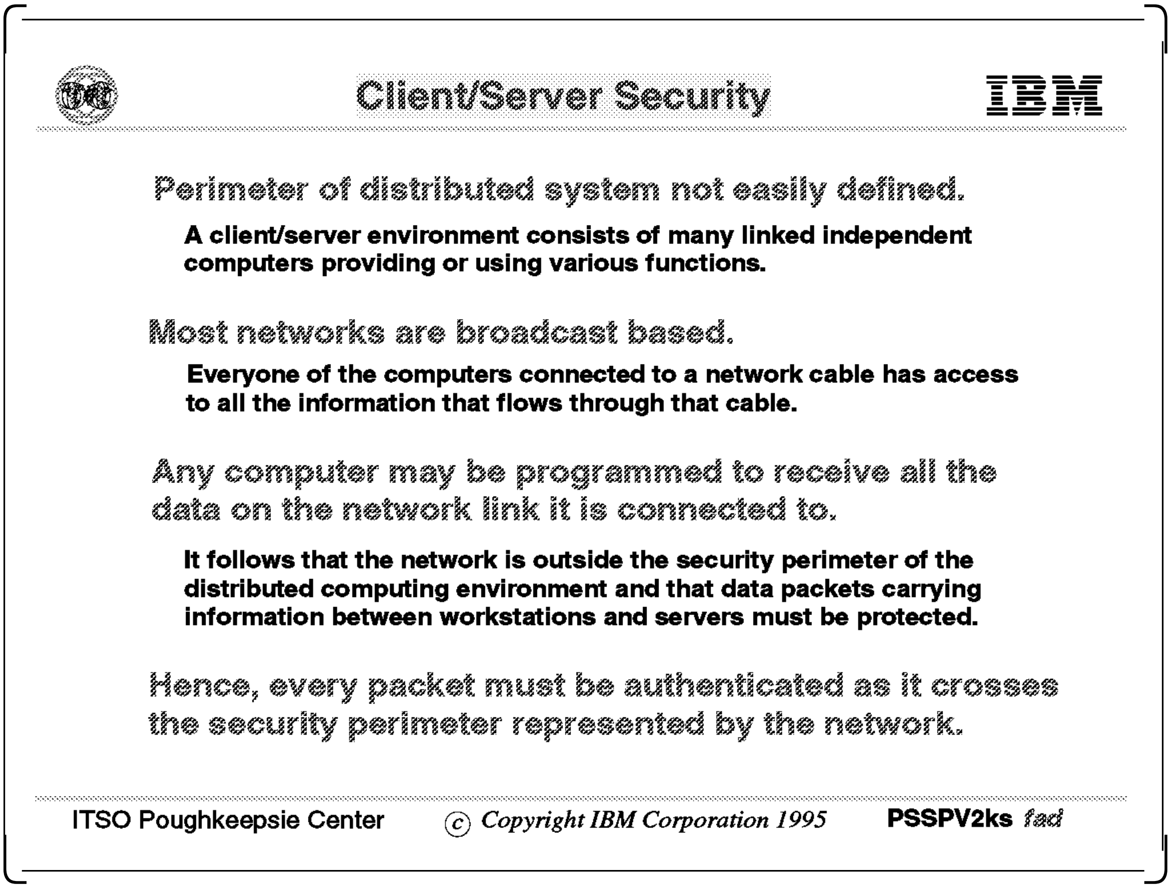


Client/Server Security

Any computer may be programmed to receive all data on the network link it is connected to.

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ks fac

This foil shows a simple networked computing environment as an illustration to the statements that follow.



Client/Server Security

Perimeter of distributed system not easily defined.

A client/server environment consists of many linked independent computers providing or using various functions.

Most networks are broadcast based.

Everyone of the computers connected to a network cable has access to all the information that flows through that cable.

Any computer may be programmed to receive all the data on the network link it is connected to.

It follows that the network is outside the security perimeter of the distributed computing environment and that data packets carrying information between workstations and servers must be protected.

Hence, every packet must be authenticated as it crosses the security perimeter represented by the network.

ITSO Poughkeepsie Center ⓒ *Copyright IBM Corporation 1995* **PSSPV2ks** *fad*

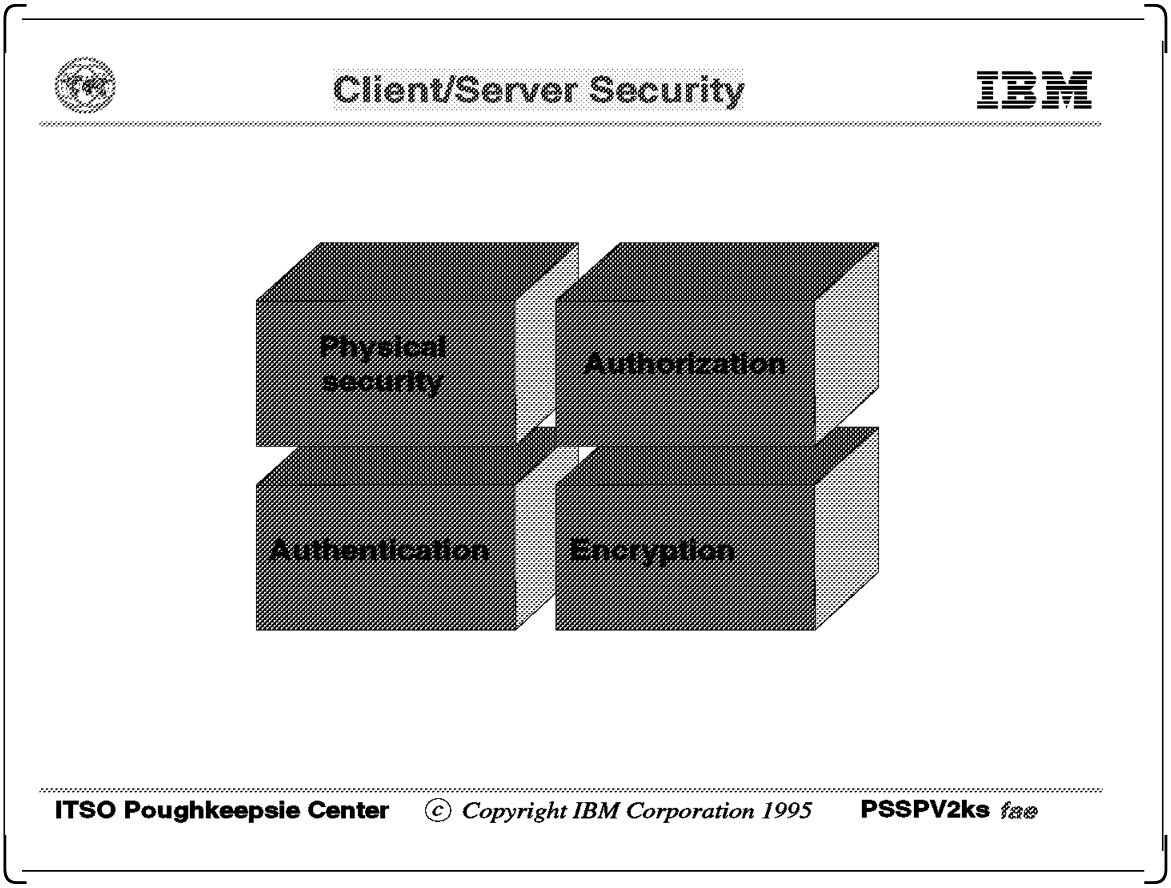
The assumptions of mainframe security fail once workstations and networks come into the picture. First, the perimeter of a networked system is not easily defined. Rather than a single computer system around which a line can be drawn, a distributed computing environment consists of many independent systems connected by links of uncertain trustworthiness.

Most networks tend to be physically large and difficult to secure (think for instance of a university campus). It is simply too expensive to lock all the wires inside unbreachable conduits. The advent of global Internet adds yet another dimension to network size. As the number of global users expands, so does the number of less than honest users.

In addition, most networks are broadcast based; *everyone of the computers connected to a network cable has access to all the information that flows through that cable*. An intruder who has control of a computer can readily program it to receive either all the data on the network link it is connected to or all the data intended for some other computer. An intruder may also be able to send information while making it appear to have come from somewhere else.

The ease with which an intruder can perform such illicit acts means that the network is not, properly speaking, inside the security perimeter of the distributed computing environment. If the network is outside the perimeter, then one must somehow protect data packets as they carry information between workstations and servers. Every packet must be authenticated as it crosses the security perimeter represented by the network.

12.3 Elements of Client/Server Security



Client/Server Security

Physical security

Authorization

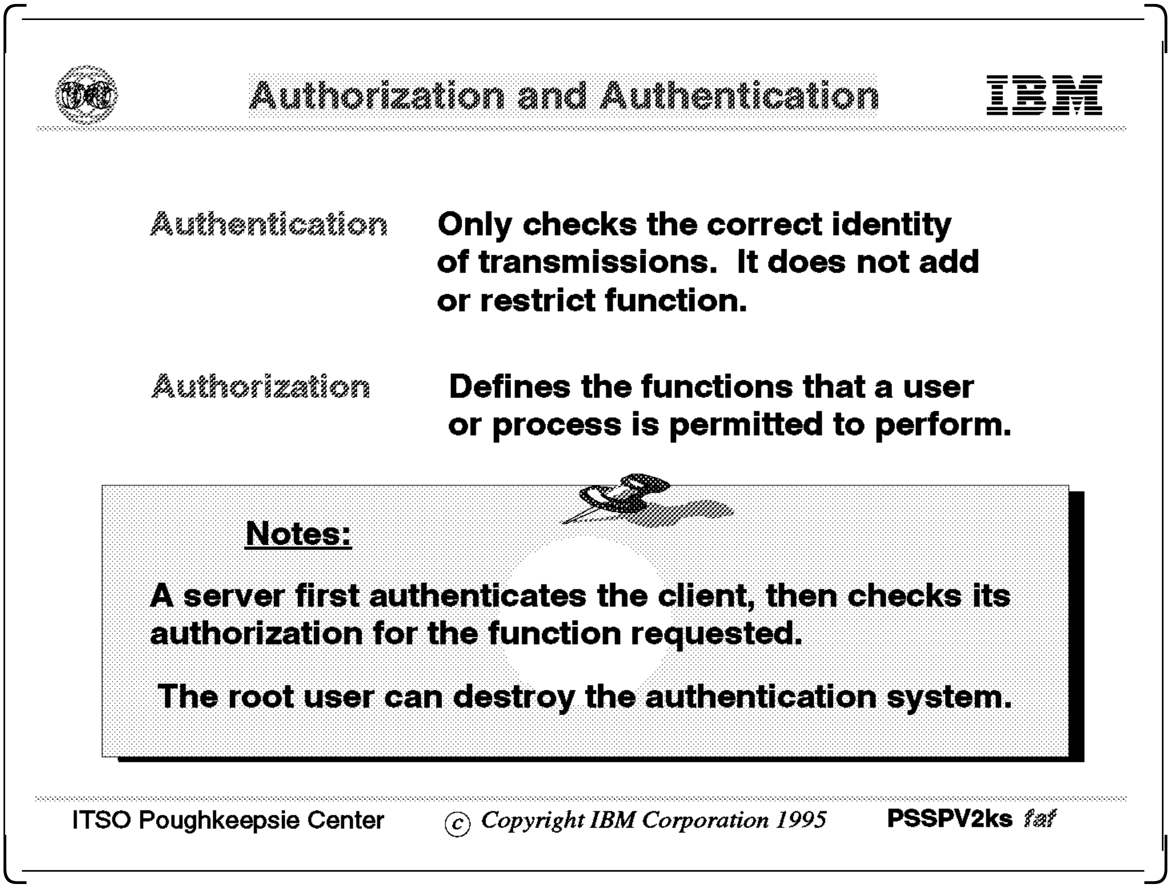
Authentication

Encryption

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ks fae

Using secure authentication (for example, by means of Kerberos and the associated encryption) does not ensure that a distributed computing environment is secure. Other measures are necessary to prevent attackers from making an end run around the security of information in a client/server environment. These measures are clarified in the following foils and summarized in 19.1, “Authentication Server Security Policy” on page 228.

12.4 Authentication and Authorization



Authorization and Authentication

Authentication Only checks the correct identity of transmissions. It does not add or restrict function.

Authorization Defines the functions that a user or process is permitted to perform.

Notes:

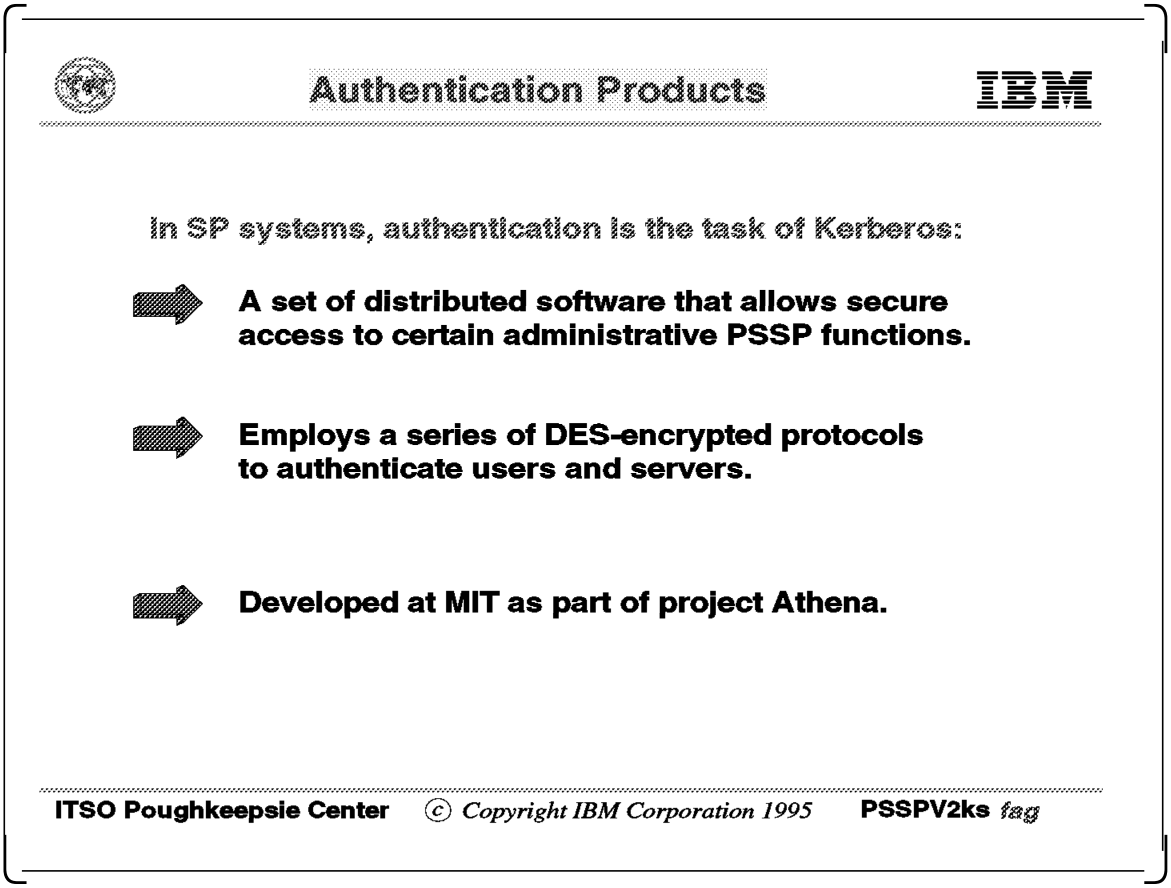
A server first authenticates the client, then checks its authorization for the function requested.

The root user can destroy the authentication system.

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ks faf

In normal usage, a server (such as the *sysctl* server) must first authenticate the requesting user or client, and then check his authorization for the requested service.

12.5 What is Kerberos



Authentication Products

In SP systems, authentication is the task of Kerberos:

- **A set of distributed software that allows secure access to certain administrative PSSP functions.**
- **Employs a series of DES-encrypted protocols to authenticate users and servers.**
- **Developed at MIT as part of project Athena.**

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* **PSSPV2ks** *fag*

In the SP, as in many other systems today, the task of authentication is entrusted to Kerberos.

Kerberos is a set of distributed software that employs a series of encrypted exchanges of information to allow a user access to servers. Kerberos also provides for cryptographic checks to make sure that data passing between workstations and servers is not corrupted either by accident or by tampering.

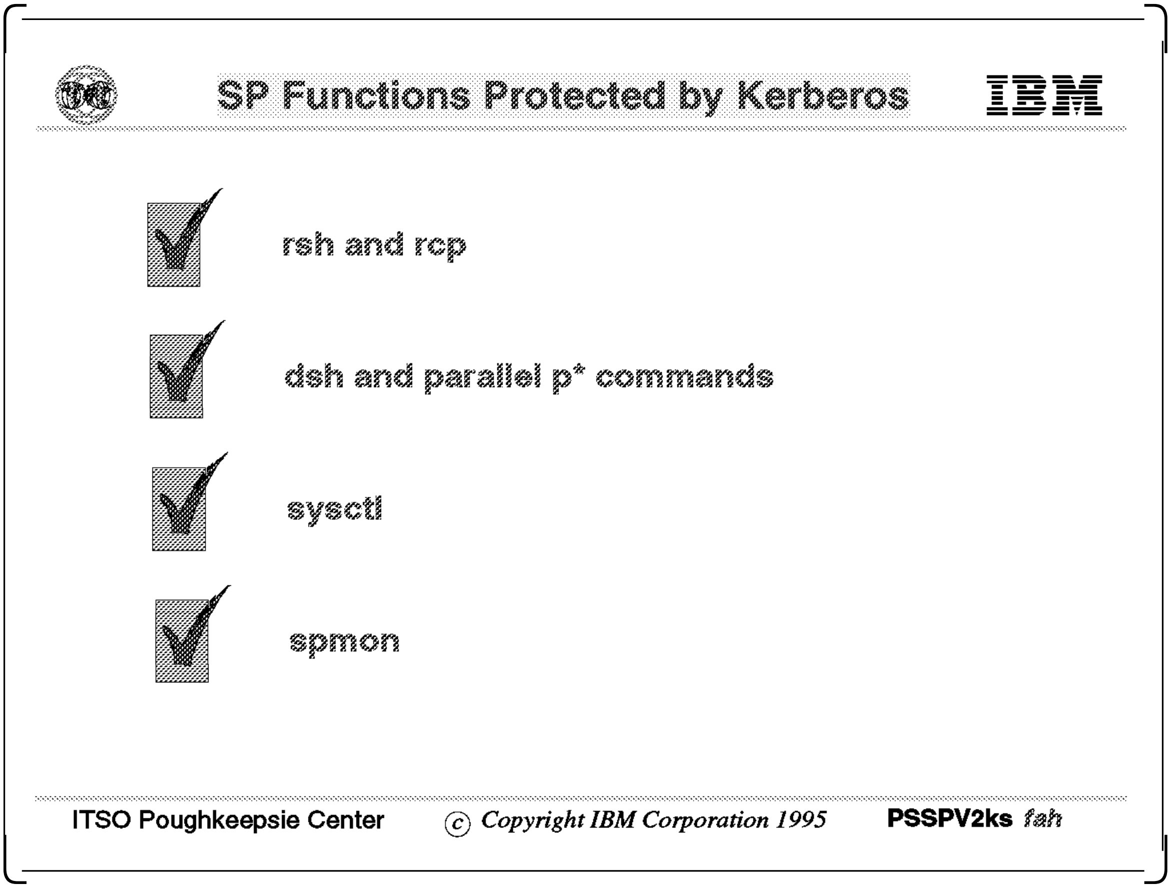
The current version of Kerberos is Version 5. This version has some new features. It is less dependent on the details of the Unix operating system and can therefore be adapted to other computing environments. Version 5 has been adopted by the Open Software Foundation as a component of its Distributed Computing Environment. At MIT, work in progress on Kerberos, such as the implementation of public key cryptography, is carried out on Version 5.

On SP systems, one of the following Kerberos implementations is used:

1. The SP implementation, based on MIT Kerberos Version 4
2. The implementation of Kerberos included in AFS 3.3 or 3.3a
3. Another Kerberos implementation, provided it is compatible with SP Kerberos-authenticated services

In the remainder of this chapter, we discuss only the SP implementation of Kerberos.

12.6 SP Functions Protected by Kerberos



SP Functions Protected by Kerberos

☑ rsh and rcp

☑ dsh and parallel p* commands

☑ sysctl

☑ spmon

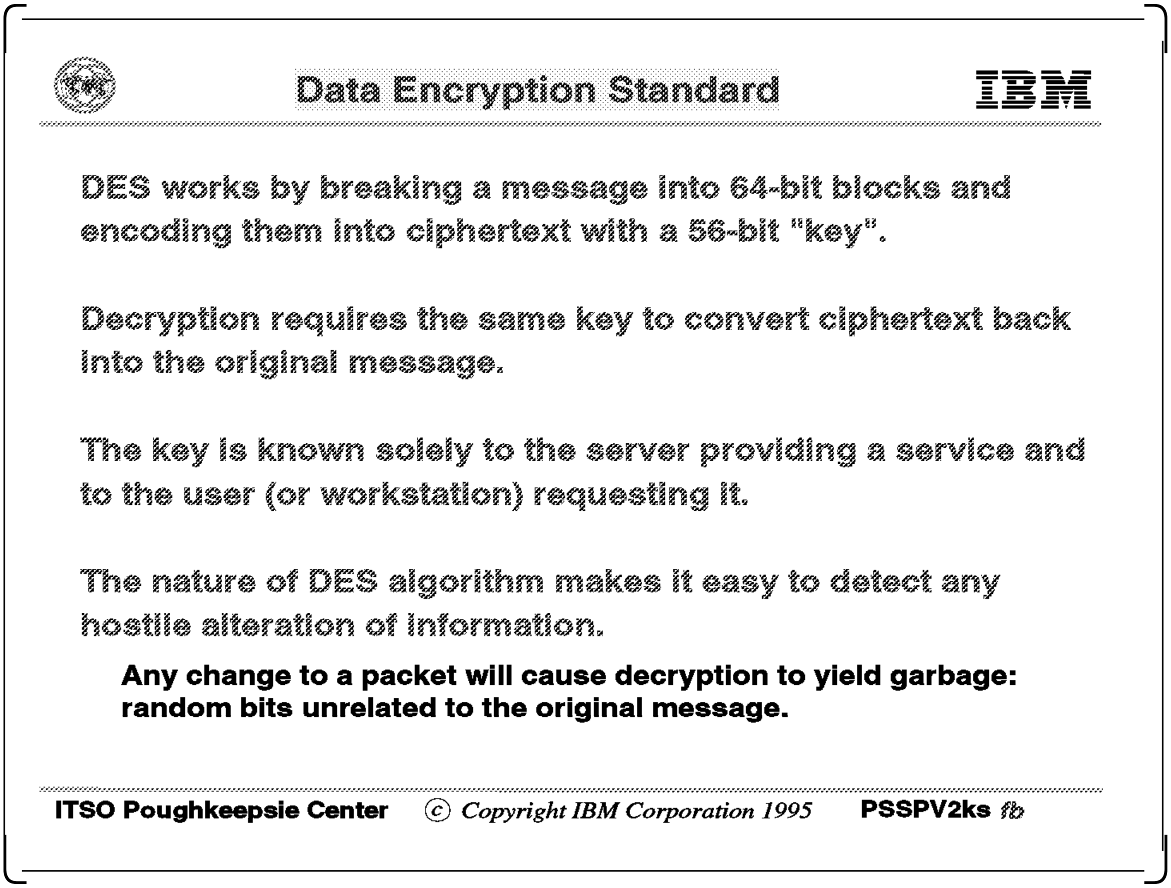
ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* **PSSPV2ks** *fah*

The following SP functions require Kerberos authentication:

- The System Monitor (spmon)
- The sysctl application
- The commands rsh, rcp and dsh
- A number of commands based on dsh or sysctl, such as pcat, pdf, pexec, and so on.

We will come back to this subject in more detail in Chapter 16, “Kerberos Implementation on the SP” on page 187.

Chapter 13. Data Encryption Standard



Data Encryption Standard

DES works by breaking a message into 64-bit blocks and encoding them into ciphertext with a 56-bit "key".

Decryption requires the same key to convert ciphertext back into the original message.

The key is known solely to the server providing a service and to the user (or workstation) requesting it.

The nature of DES algorithm makes it easy to detect any hostile alteration of information.

Any change to a packet will cause decryption to yield garbage: random bits unrelated to the original message.

ITSO Poughkeepsie Center ⓒ *Copyright IBM Corporation 1995* PSSPV2ks *fb*

The version of Kerberos implemented in the SP (Version 4) uses the Data Encryption Standard (DES) to encode its communications.

The DES algorithm belongs to the public domain. The entire algorithm is published in the Federal Register (see publication 2 in 11.4.2, "Bibliography" on page 150).

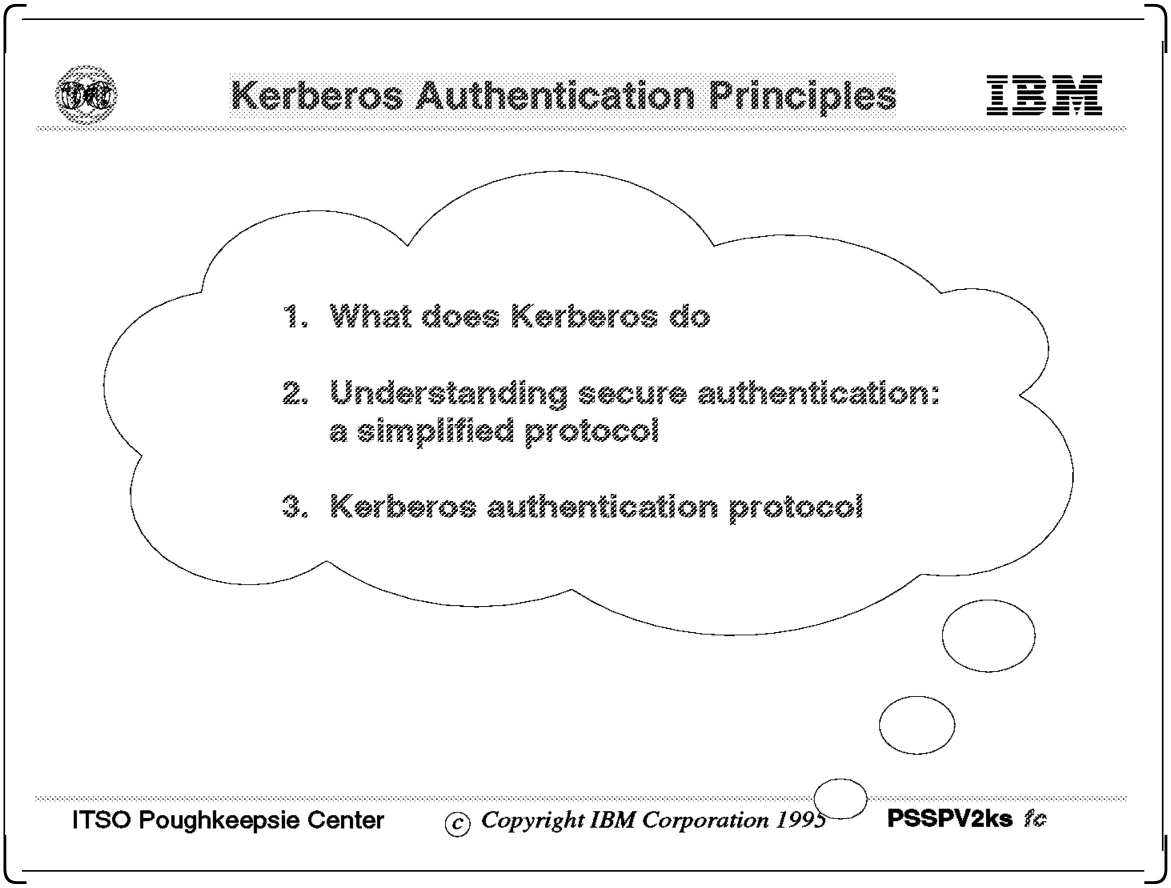
The Data Encryption Standard was developed at IBM during the period of 1973-74. A banking customer asked IBM to develop a system for encrypting ATM data. With this problem as a starting point, a team including Horst Feistel, Alan Kornheim, Bryant Tuckerman, Edna Grossman and Don Coppersmith of the Yorktown Mathematical Research Department developed the algorithm. Bill Noltz and Lynn Smith did much of the implementation. The team was assisted by several outside consultants and benefited from the expertise of the National Security Agency.

DES works by breaking messages into discrete blocks of information (usually 64 bits) and transforming them into blocks of ciphertext according to a 56-bit "key." The input message block is split into two halves, left and right. During the first of sixteen sequential *rounds*, the 32-bit right half, along with 48 of the 56 key bits, is fed into a nonlinear function F. The 32-bit output of this function, added to the left half of the message, becomes the new right half message. In the meantime, the old right half message is funneled forward to become the new left half message. Thus ends one round. The process is repeated sixteen times, using a different selection of 48 key bits each time. The final left half and right half messages become the ciphertext.

Decryption requires the *same* key to convert ciphertext blocks back into the original message. One simply climbs back up the ladder, reversing the effects of one round at a time. At the beginning of a reverse round, one has the new left half message and the new right half, as well as the key. The new left (which is the same as the old right) and the appropriate 48 key bits are fed into F. The 32-bit output of F is subtracted from the new right half to obtain the old left half. Thus one can reverse one round, and by repeated application, one reverses the entire encryption.

Furthermore, the nature of the DES algorithm makes it easy to detect any hostile alteration of the information passing through the network. Virtually any changes to the packet will cause decryption to yield garbage, random characters completely unrelated to the original message. Such corrupted original messages are easy to detect and a workstation or server can simply discard them and request a retransmission.

Chapter 14. Understanding Secure Authentication



Kerberos Authentication Principles

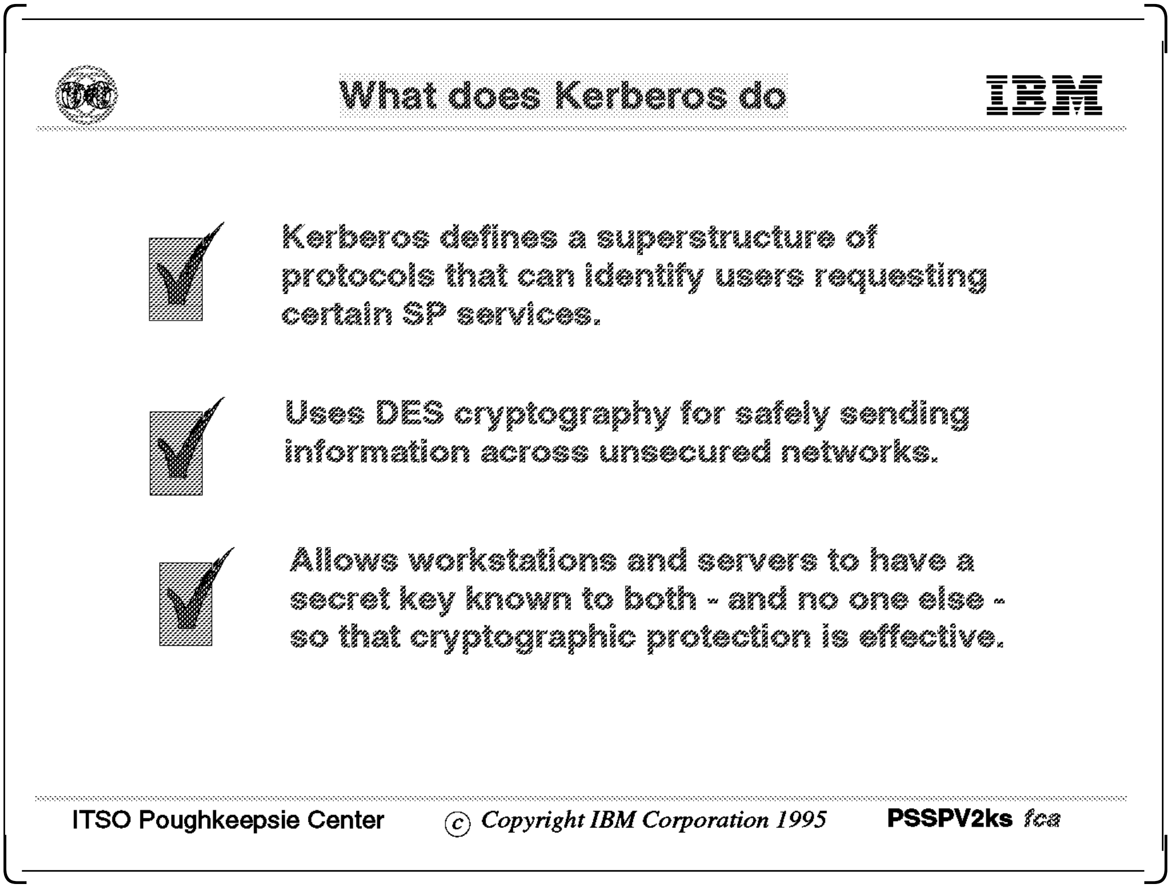
1. What does Kerberos do
2. Understanding secure authentication: a simplified protocol
3. Kerberos authentication protocol

ITSO Poughkeepsie Center ⓒ Copyright IBM Corporation 1995 PSSPV2ks fc

On the way to understanding the Kerberos secure authentication protocol, we go through the following three stages:

1. We begin by a very brief overview and a few definitions.
2. Then we detail a (hypothetical) simplified Kerberos protocol and explain why it is inadequate.
3. Finally, we present the complete Kerberos protocol, that is, the exchanges that take place between a client, a server and the Authentication Server, resulting in authentication.

14.1 What Does Kerberos Do



What does Kerberos do

IBM

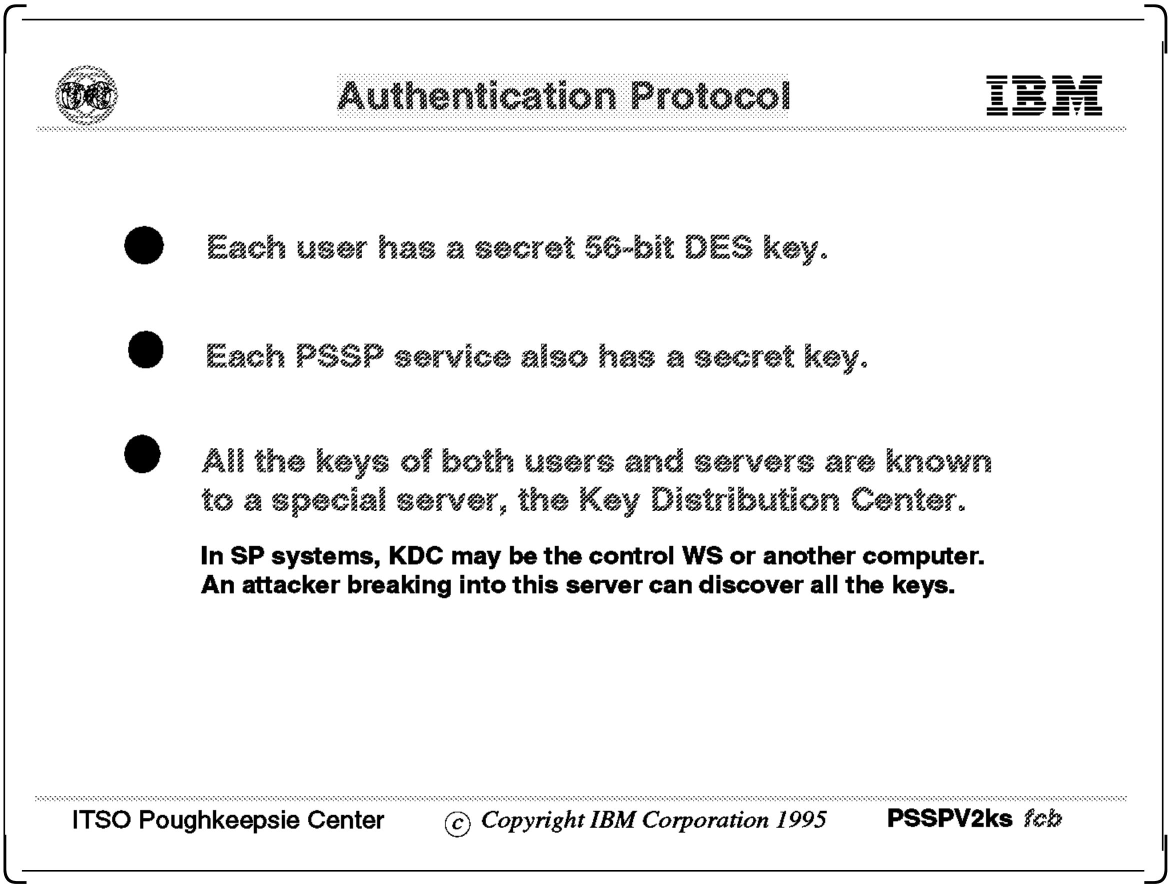
☑ Kerberos defines a superstructure of protocols that can identify users requesting certain SP services.

☑ Uses DES cryptography for safely sending information across unsecured networks.

☑ Allows workstations and servers to have a secret key known to both - and no one else - so that cryptographic protection is effective.

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* **PSSPV2ks** *fca*

Cryptographic methods for sending information safely across unsecured networks are only the foundation of Kerberos. On them has been built a superstructure of protocols that can identify the individuals requesting computing services. It also arranges for workstations and servers to have a secret key (the *session key*) known to both machines and no one else (so that cryptographic protection will in fact be effective).



Authentication Protocol

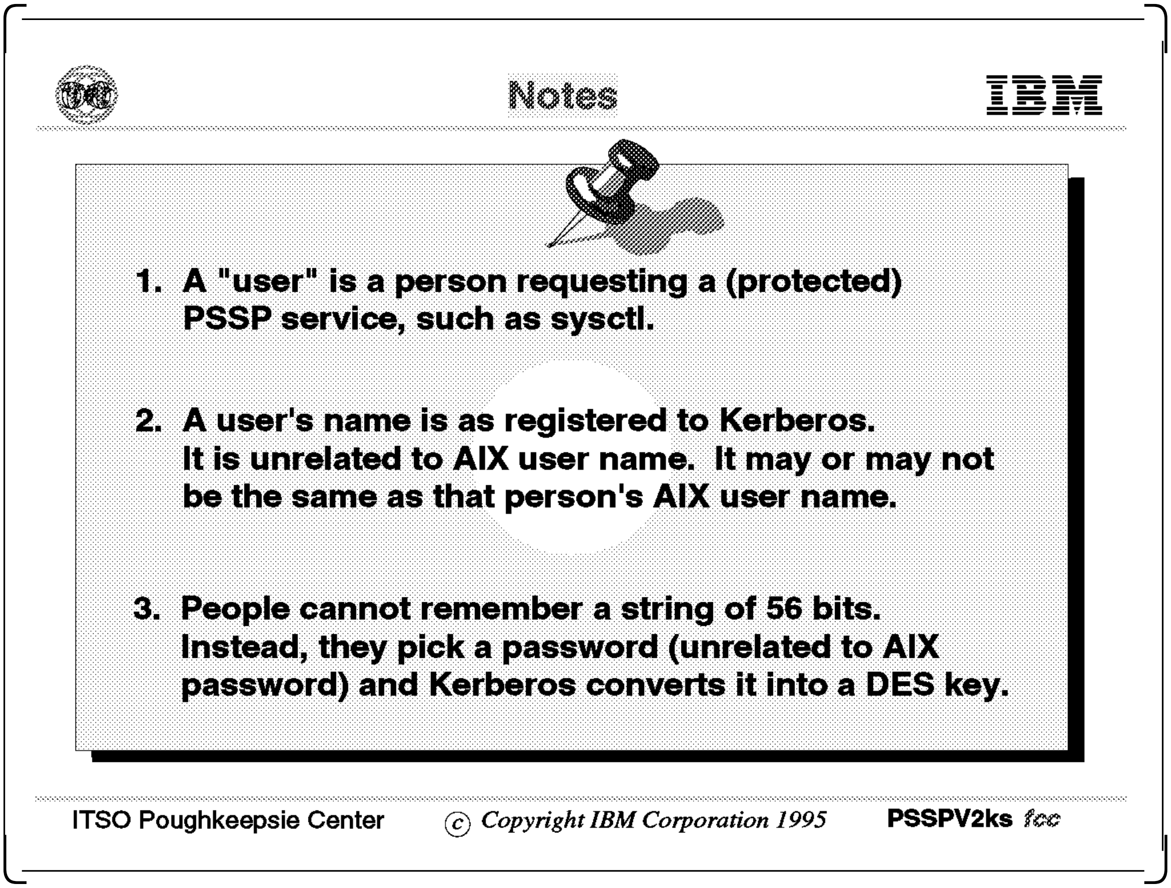
- Each user has a secret 56-bit DES key.
- Each PSSP service also has a secret key.
- All the keys of both users and servers are known to a special server, the Key Distribution Center.

In SP systems, KDC may be the control WS or another computer.
An attacker breaking into this server can discover all the keys.

ITSO Poughkeepsie Center ⓒ *Copyright IBM Corporation 1995* **PSSPV2ks** *fcb*

The protocols start with a secret 56-bit DES key for each user. Each network service also has a secret key. Users and servers are referred to as *principals*. All the principals′ keys are recorded in the database of a third computer, the *authentication server*.

14.2 Notes



Notes

1. A "user" is a person requesting a (protected) PSSP service, such as sysctl.

2. A user's name is as registered to Kerberos. It is unrelated to AIX user name. It may or may not be the same as that person's AIX user name.

3. People cannot remember a string of 56 bits. Instead, they pick a password (unrelated to AIX password) and Kerberos converts it into a DES key.

ITSO Poughkeepsie Center ⓒ Copyright IBM Corporation 1995 PSSPV2ks fcc

An SP user who wishes to use any protected service must first be registered to Kerberos (using the **kadmin** or the **kdb_edit** commands).

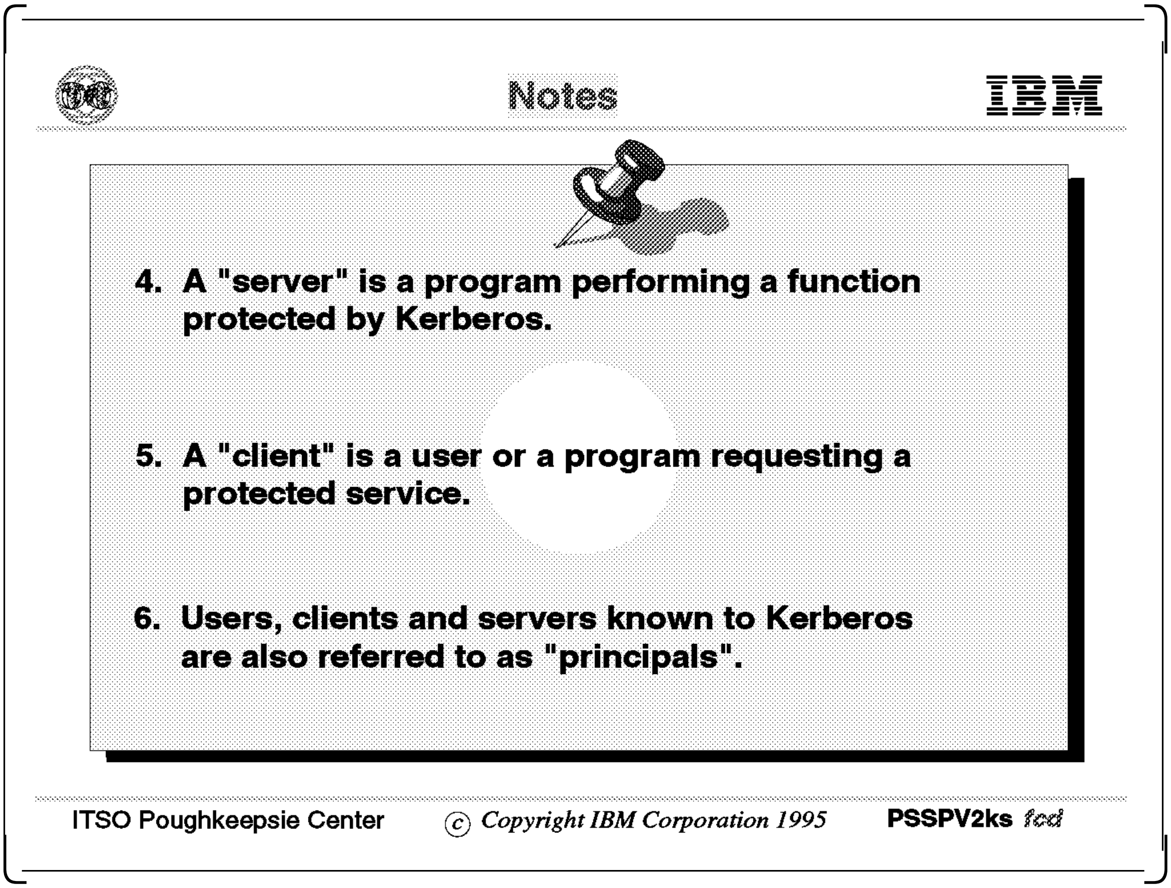
By virtue of this registration, he or she then becomes a Kerberos user, or principal, known by a name of the form:

name{.instance}{@realm}

(In this example, *instance* and *realm* are optional. See 16.4, "Naming Kerberos Principals" on page 196 for more detail about the naming convention.) A private DES key is generated for the user and stored in the Kerberos database.

Note that the Kerberos name space is unrelated to the AIX name space, so that an individual may be known by one name to Kerberos and by another to AIX. However, we have found it more convenient to assign the same name in each space.

People cannot remember a random string of 56 bits (which corresponds to a 20-digit number), and so Kerberos permits them to pick a password of six to 128 characters. An additional encryption step converts the password into a DES key. Again, note that a user′s Kerberos password is distinct from the AIX password.



Notes

IBM

4. A "server" is a program performing a function protected by Kerberos.

5. A "client" is a user or a program requesting a protected service.

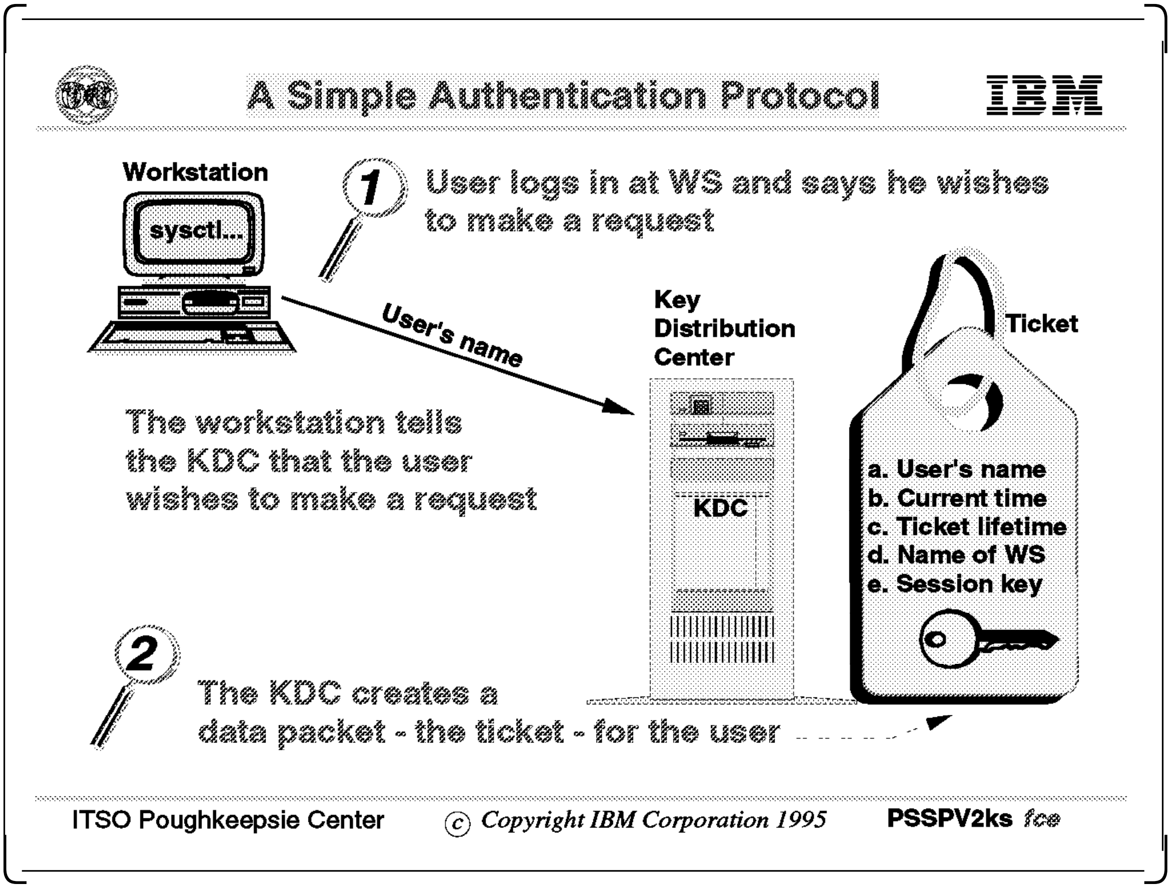
6. Users, clients and servers known to Kerberos are also referred to as "principals".

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* PSSPV2ks fcd

SP servers (such as hardmon) that perform protected functions, must also be registered in the Kerberos database, that is, assigned private keys that are stored in the database. They are also assigned names in the Kerberos name space, and are referred to as *principals* or *service principals*. Similarly to users, a service principal need not be the same as the server′s AIX name.

If a protected service is requested by another program (as opposed to a user), that program must also be registered as a principal. A program may be both a server and a client.

14.3 A Simple Authentication Protocol



A Simple Authentication Protocol

1 User logs in at WS and says he wishes to make a request

The workstation tells the KDC that the user wishes to make a request

2 The KDC creates a data packet - the ticket - for the user

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ks fce

The principle of secure authentication over an untrusted network is best understood through the following *simplified* protocol.

Step 1:

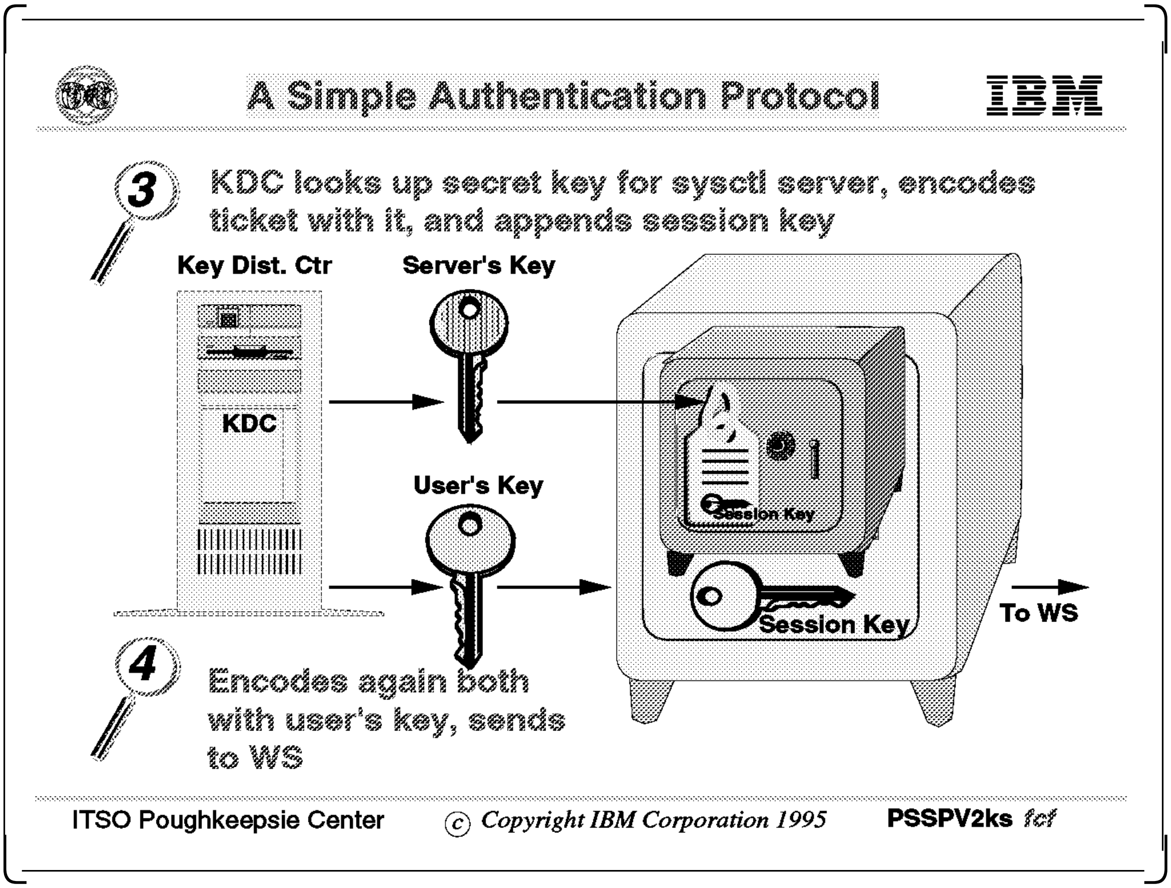
The user enters a request for a protected service, say *sysctl*. The distributed Kerberos code in the workstation sends a message to the **Key Distribution Center** (KDC) telling it that the user wishes to make a request. The message contains the user′s name and the workstation name in clear.

The KDC is the authentication server. It resides in a physically secure and, at the same time, highly available computer. We installed it on the SP control workstation, as probably most installations do, but it may reside in any computer in the network.

Step 2:

The KDC creates a data packet, the ticket, that contains the name of the user, the current time, the length of time the ticket will be valid, the name of the workstation and a randomly generated DES key, called a **session key**.

Note that the session key is not stored in the KDC. It will be known only to the user and to the server.



A Simple Authentication Protocol

3 KDC looks up secret key for sysctl server, encodes ticket with it, and appends session key

4 Encodes again both with user's key, sends to WS

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ks fcf

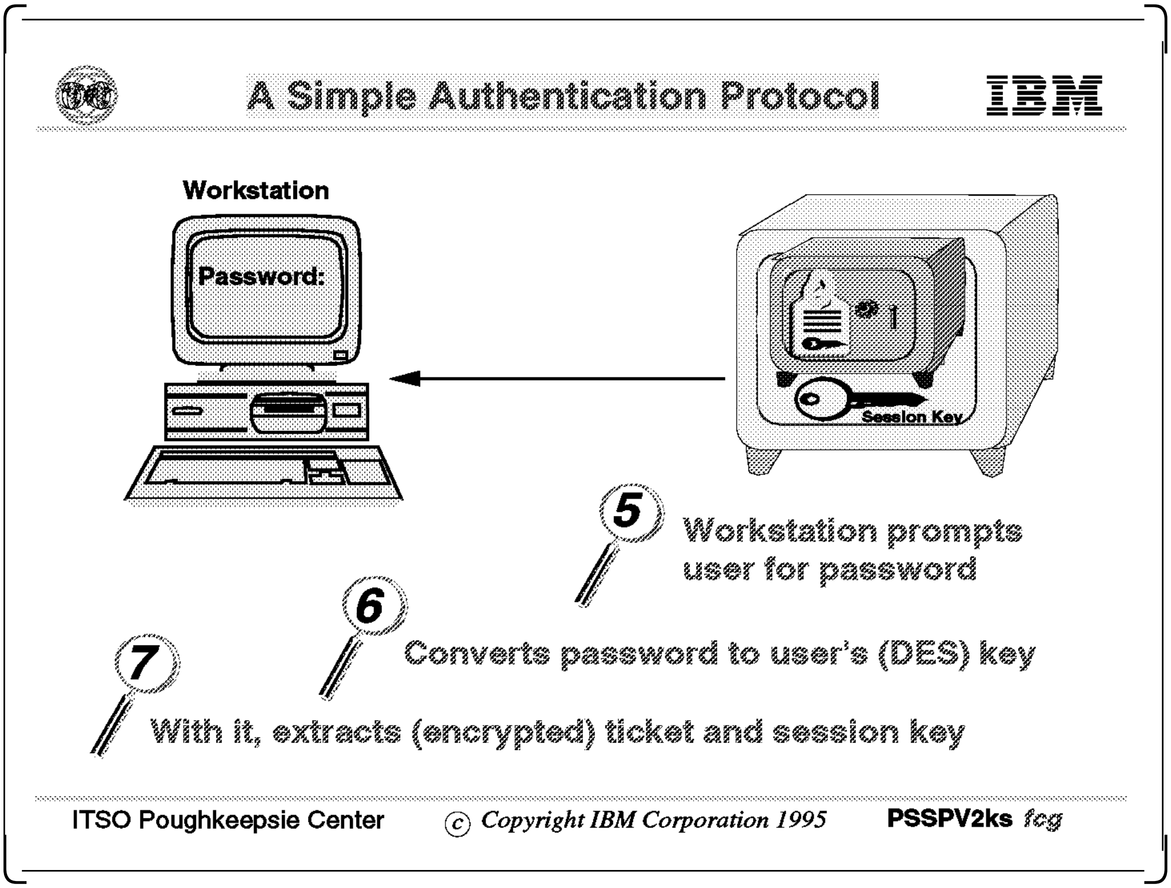
Step 3:

The KDC then looks up the secret key for the sysctl server (for example) and encrypts the ticket with it, so that only the sysctl server can read it.

On the foil, the safe is meant to depict encryption of its contents. The ticket within a safe means the encrypted ticket. The sysctl server′s key that is on the arrow pointing to the safe is the DES key used for encoding.

Step 4:

The KDC appends the session key to the encrypted ticket, encodes both with the user′s key and sends the result (depicted by the outer safe) back to the workstation.



A Simple Authentication Protocol

5 Workstation prompts user for password

6 Converts password to user's (DES) key

7 With it, extracts (encrypted) ticket and session key

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* **PSSPV2ks** *fcg*

Step 5:

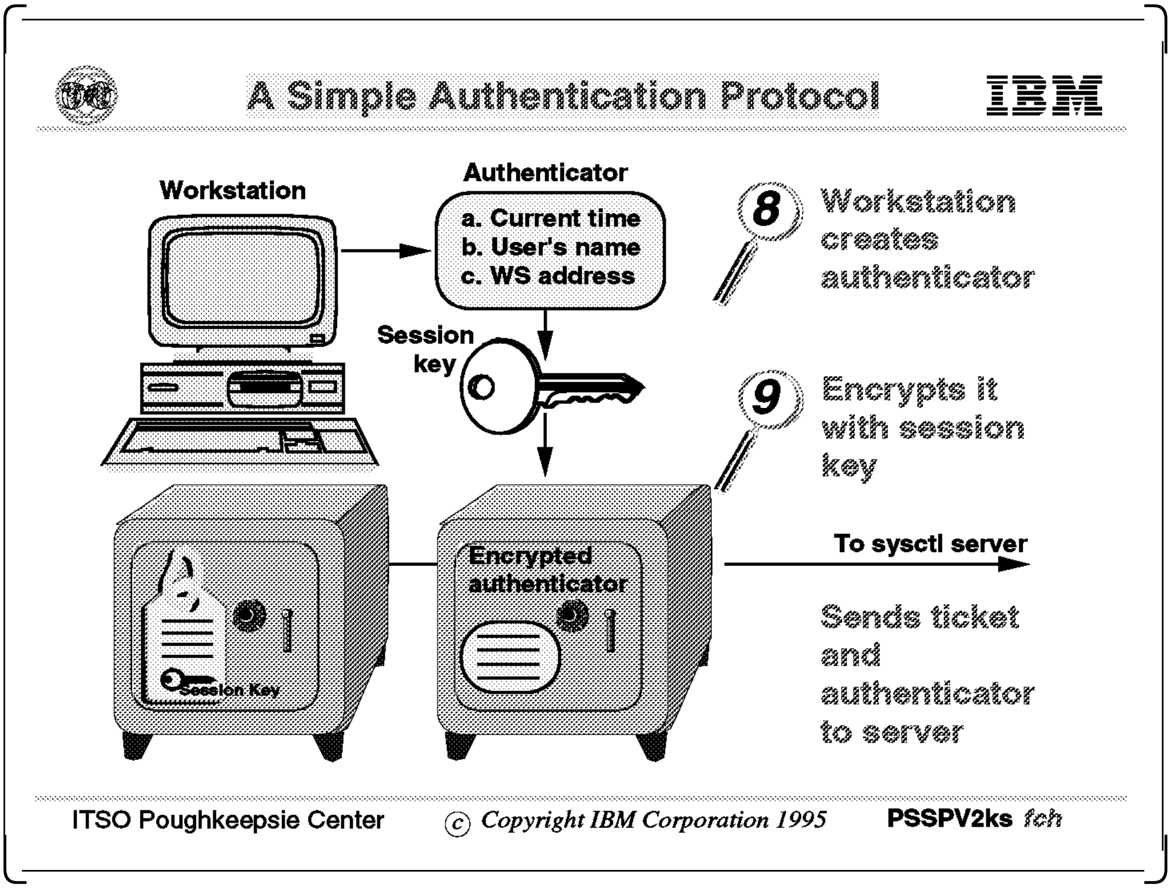
When the workstation receives this information, it prompts the user for a password.

Steps 6 and 7:

It converts the password to a DES key and uses it to decrypt the ticket an accompanying session key.

Only if the user supplies the correct password to the workstation will he be able to decrypt this information properly.

At this point, the workstation has an (encrypted) ticket and a session key. The workstation cannot read the ticket because it is encoded, in our example, with the sysctl server′s key.



A Simple Authentication Protocol

8 Workstation creates authenticator

9 Encrypts it with session key

Sends ticket and authenticator to server

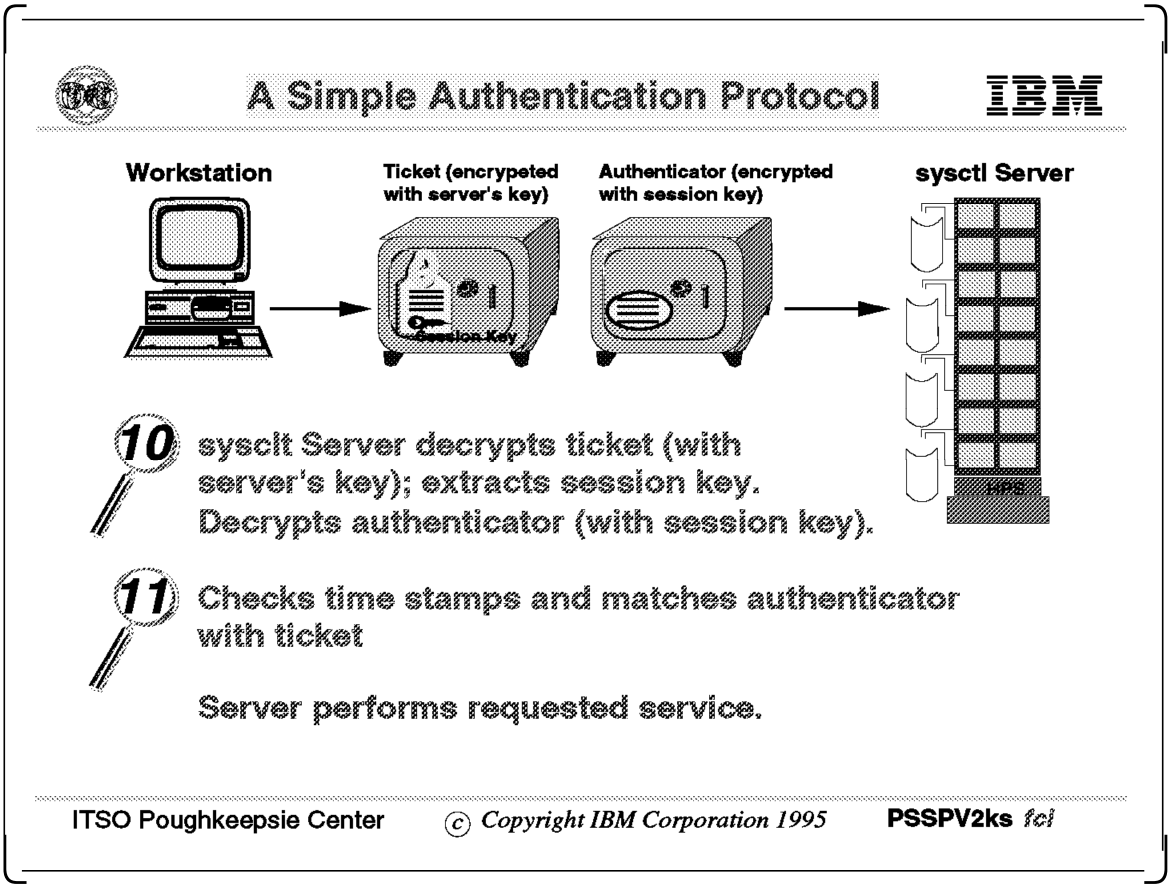
ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* **PSSPV2ks** *fch*

Steps 8 and 9:

So far, the workstation has not communicated with the sysctl server yet.

Now, when the user wants to request a service from, say, the sysctl server, the workstation forwards the (encrypted) ticket and an additional packet of data, called an **authenticator** to the server. The authenticator consists of the following items, all encrypted with the session key:

- The current time
- The user′s name
- The workstation address



Workstation
Ticket (encrypeted with server's key)
Session Key
Authenticator (encrypted with session key)
sysctl Server
HPS

A Simple Authentication Protocol

10 syscit Server decrypts ticket (with server's key); extracts session key. Decrypts authenticator (with session key).

11 Checks time stamps and matches authenticator with ticket

Server performs requested service.

ITSO Poughkeepsie Center ⓒ *Copyright IBM Corporation 1995* **PSSPV2ks** *fcl*

Step 10:

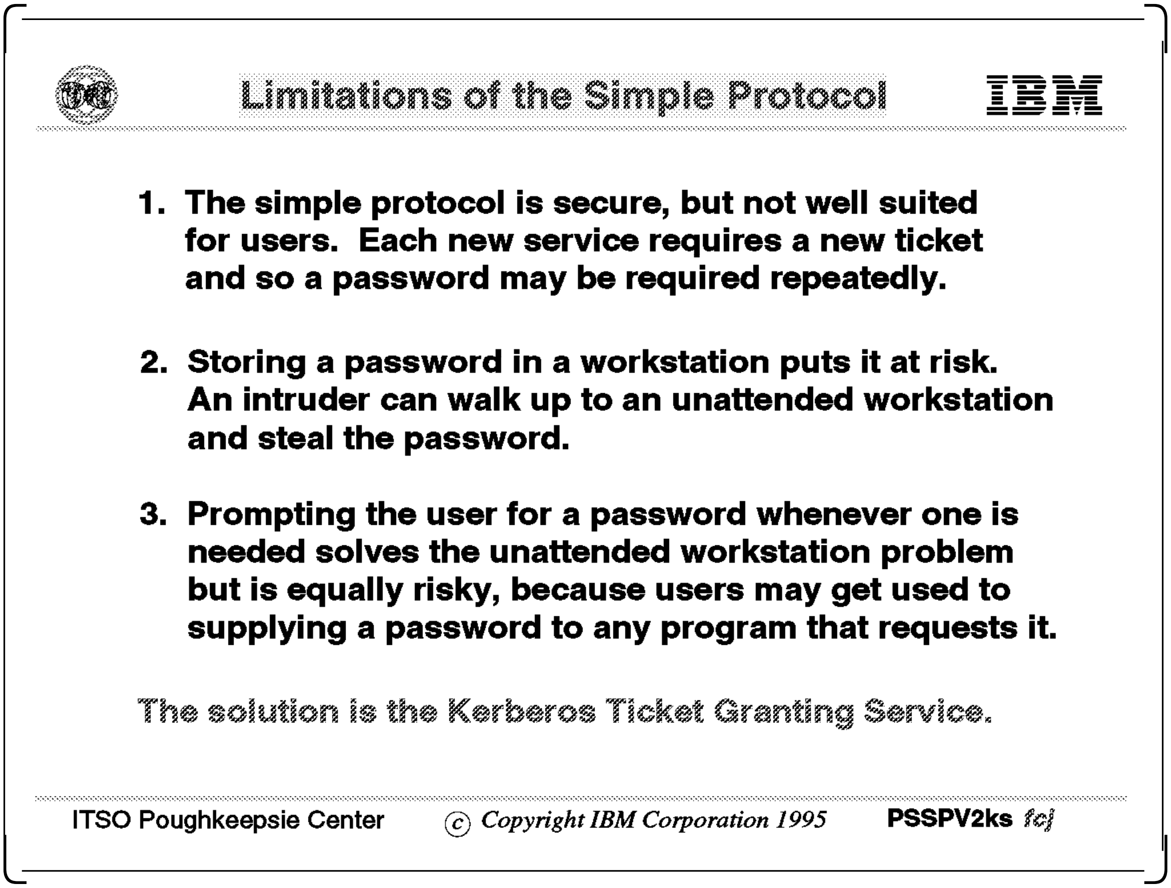
The sysctl server decrypts the ticket and extracts the session key, with which it can then decrypt the authenticator.

Step 11:

The sysctl server makes sure that the user and workstation named in the ticket match those in the authenticator. It also checks that the time stamps are valid.

If all these credentials pass inspection, the sysctl server processes the user′s request.

14.4 Limitations of the Simple Protocol



Limitations of the Simple Protocol

1. The simple protocol is secure, but not well suited for users. Each new service requires a new ticket and so a password may be required repeatedly.
2. Storing a password in a workstation puts it at risk. An intruder can walk up to an unattended workstation and steal the password.
3. Prompting the user for a password whenever one is needed solves the unattended workstation problem but is equally risky, because users may get used to supplying a password to any program that requests it.

The solution is the Kerberos Ticket Granting Service.

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* PSSPV2ks fcj

Although the *simple* protocol described so far is secure, it is not well suited for ordinary users. Each new service requires a new ticket, and so the user could be required to supply a password any number of times during a session. Storing a password on a workstation puts it at risk. A clever intruder could simply walk up to an unattended workstation and steal the password from it.

Prompting the user for a password each time he requests a service solves the unattended workstation problem, but assuming that users hold their temper at being asked repeatedly to prove their identity, in the long run the repeated prompts are equally risky. Users will get used to supplying a password to any program that requests it. They then become easy prey for an intruder who provides a program (called a *Trojan horse*) that prompts for a password, but instead of using it for Kerberos authentication, stores it away for later pickup.

The solution to this dilemma is the **Ticket Granting Service (TGS)**. The TGS runs on the same system as the Key Distribution Center and has access to its database of users, services and keys[1] .

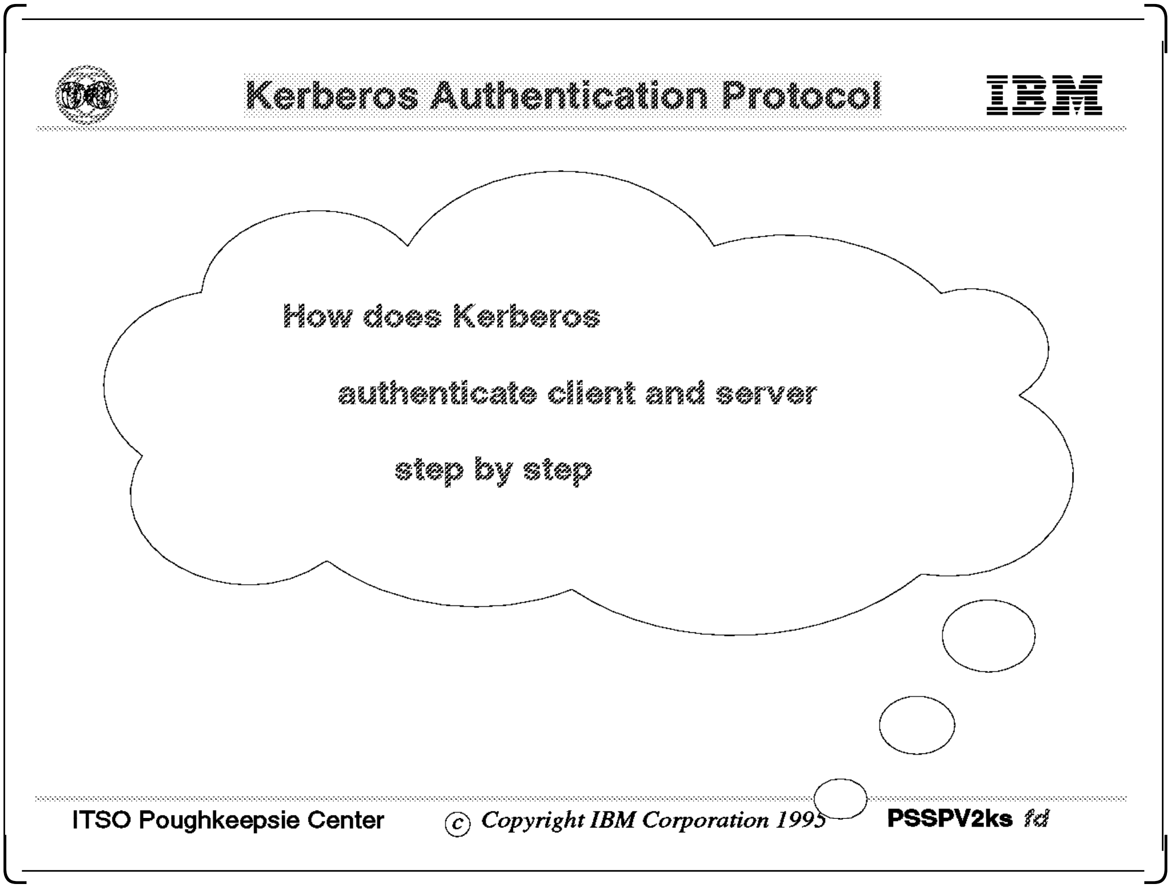
[1] Indeed, in the SP implementation, the TGS and KDC functions are both performed by the same daemon called **kerberos**.

Users provide a password *once* when logging in[2] to fetch a ticket for the TGS from the KDC. Subsequent requests for tickets to other services go to the TGS, which encrypts them not with the user′s password but rather with the session key that accompanied the initial TGS ticket.

Consequently, a user′s password need not be stored in the workstation. It resides in memory just long enough to permit the workstation to decrypt the TGS ticket. If a user leaves his or her workstation unattended, an intruder may be able to obtain tickets and session keys from the workstation. These tickets, however, are usable only from that workstation (because they contain the name of the workstation), and each is valid for a limited time (30 days at most in the SP implementation).

[2] (More precisely, the user enters the **kinit** command, which prompts for the Kerberos password. This need be done once per session or per ticket lifetime, whichever is shorter.)

Chapter 15. Kerberos Authentication Protocol



Kerberos Authentication Protocol

How does Kerberos

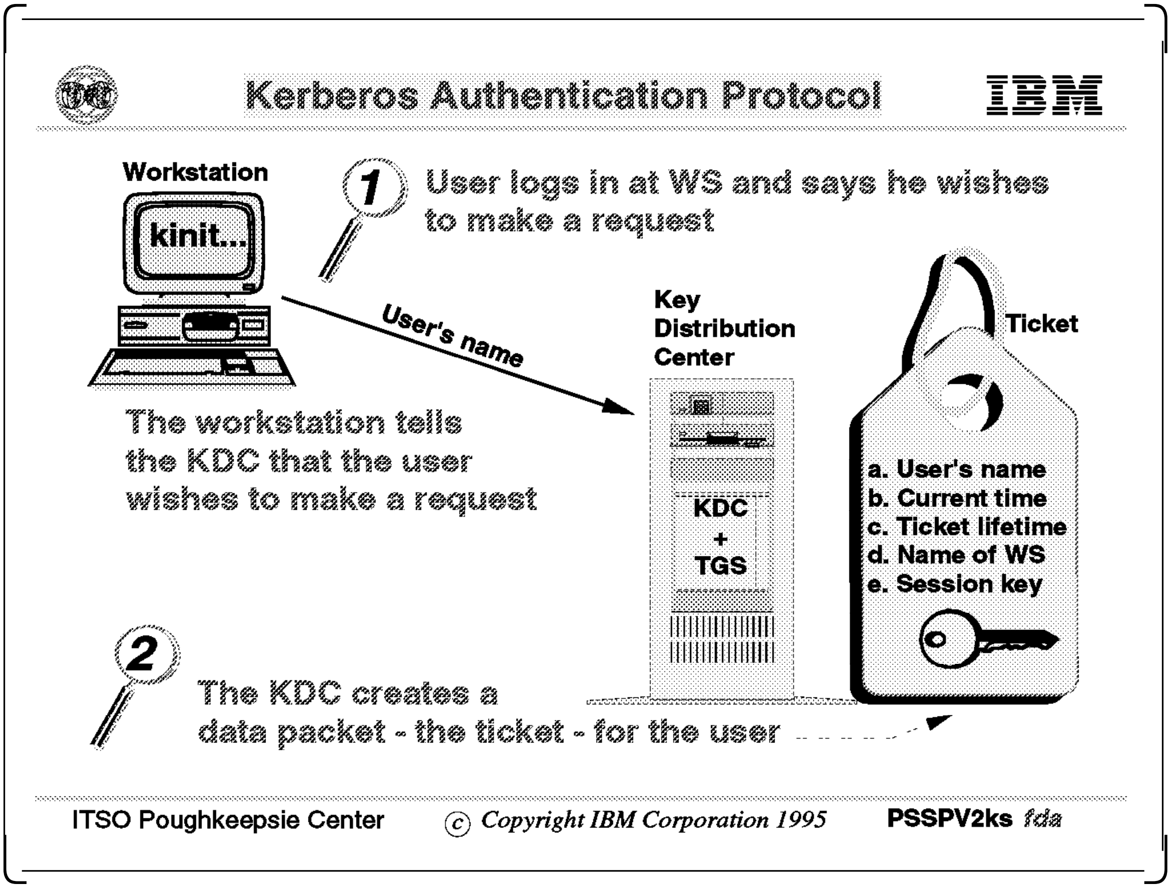
authenticate client and server

step by step

ITSO Poughkeepsie Center ⓒ Copyright IBM Corporation 1995 PSSPV2ks fd

The actual Kerberos authentication protocol is explained step by step in the following foils. Steps 1 through 7 are essentially the same as those of the hyphothetical *simple* protocol outlined in the previous foils, except that users start with the **kinit** command, and then request protected services such as *sysctl*, in our example.

15.1 Kerberos Authentication Protocol



Kerberos Authentication Protocol

Workstation

kinit...

1 User logs in at WS and says he wishes to make a request

User's name

The workstation tells the KDC that the user wishes to make a request

Key Distribution Center

KDC + TGS

Ticket

a. User's name
b. Current time
c. Ticket lifetime
d. Name of WS
e. Session key

2 The KDC creates a data packet - the ticket - for the user

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* PSSPV2ks *fda*

Now that we understand the principle of secure authentication over an untrusted network, we can step through the workings of the actual Kerberos protocol.

Step 1:

The user indicates that he or she intends to use protected services usually by invoking the **kinit** command. You may consider this command as *logging in* to Kerberos. The *kinit* command is typically entered at most once per session (only if the generated ticket expires needs this command be executed again during the same session).

The distributed Kerberos code in the workstation sends a message to the **Key Distribution Center** (KDC) telling it that the user wishes to make a request. The message contains the user′s name and the workstation name in clear.

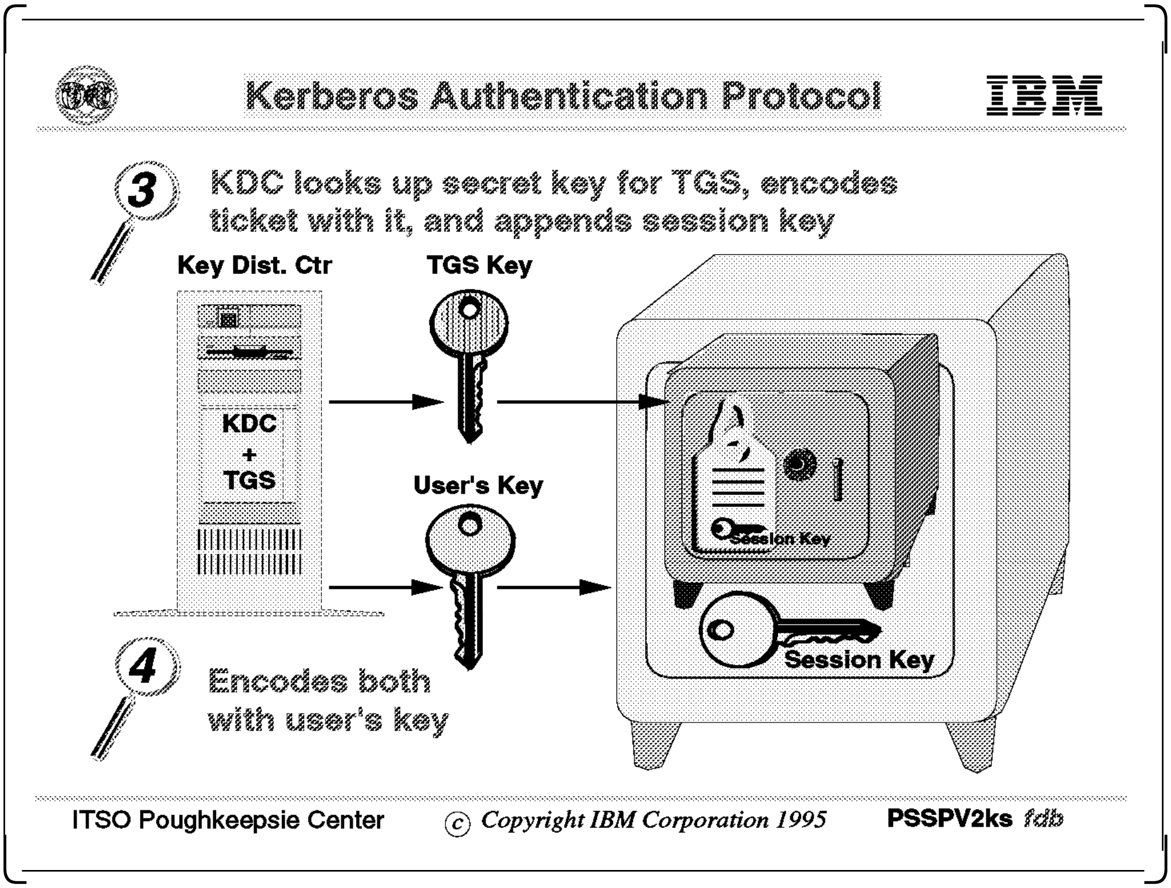
The KDC is the authentication server. It resides in a physically secure and, at the same time, highly available computer. We installed it on the SP control workstation, as probably most installations do, but it may reside in any computer in the network.

Step 2:

The KDC creates a data packet, the ticket, that contains:

- The name of the user
- The current time
- The length of time the ticket will be valid
- The name of the workstation
- a randomly generated DES key, called a **session key**

Note that the session key is not stored in the KDC. It will be known only to the user and to the server.



Kerberos Authentication Protocol

3 KDC looks up secret key for TGS, encodes ticket with it, and appends session key

4 Encodes both with user's key

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ks fdb

Step 3:

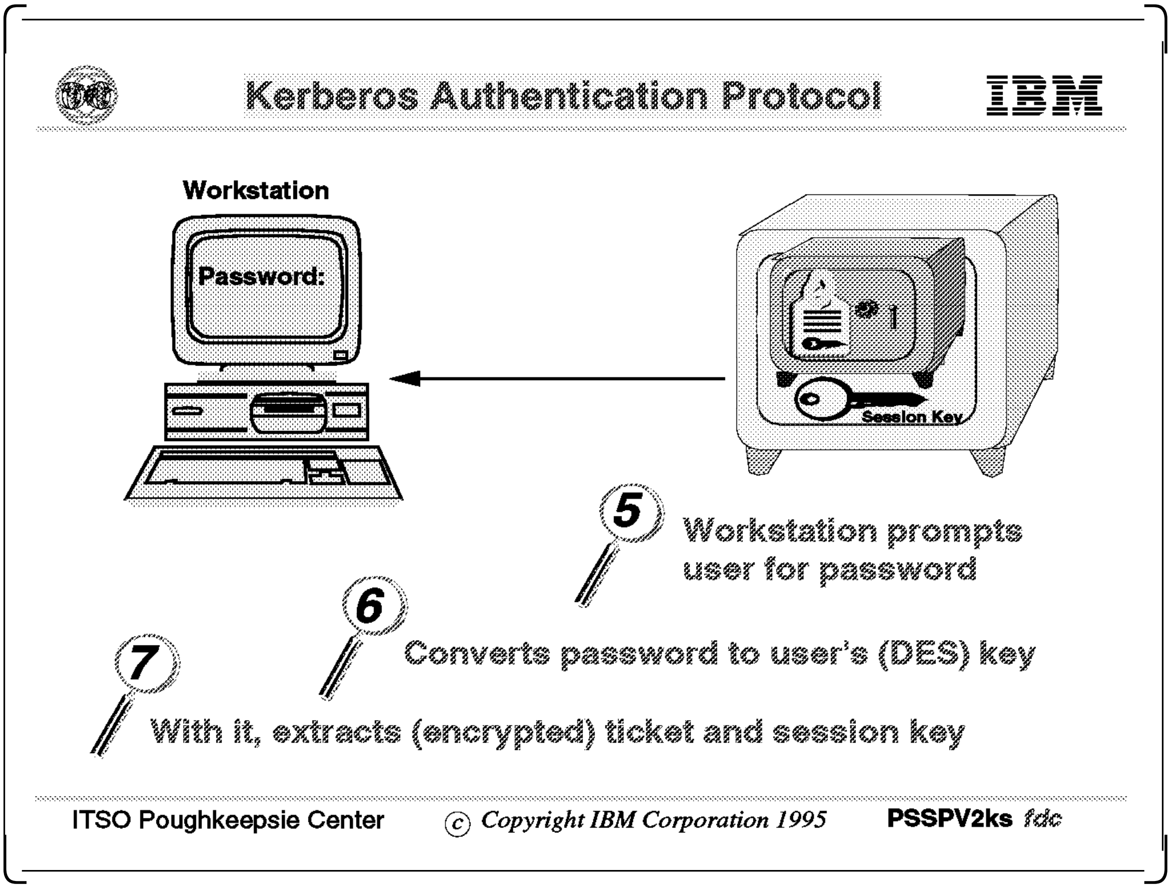
The KDC then looks up the secret key for the Ticket Granting Service (TGS) and encrypts the ticket with it, so that only the TGS can read it.

On the foil, the safe is meant to depict encryption of its contents. The ticket within a safe means the encrypted ticket. The TGS server′s key that is on the arrow pointing to the safe is the DES key used for encoding.

Remember that in the SP implementation, the TGS is always on the same machine as the KDC. In our setup, we used the control workstation.

Step 4:

The KDC appends the session key to the encrypted ticket, encodes both with the user′s key and sends the result (depicted by the outer safe) back to the workstation.



Kerberos Authentication Protocol

5 Workstation prompts user for password

6 Converts password to user's (DES) key

7 With it, extracts (encrypted) ticket and session key

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* **PSSPV2ks** *fdc*

Step 5:

When the workstation receives this information, it prompts the user for a password. The user perceives the prompt as the result of the *kinit* command and replies with his or her (Kerberos) password.

Step 6:

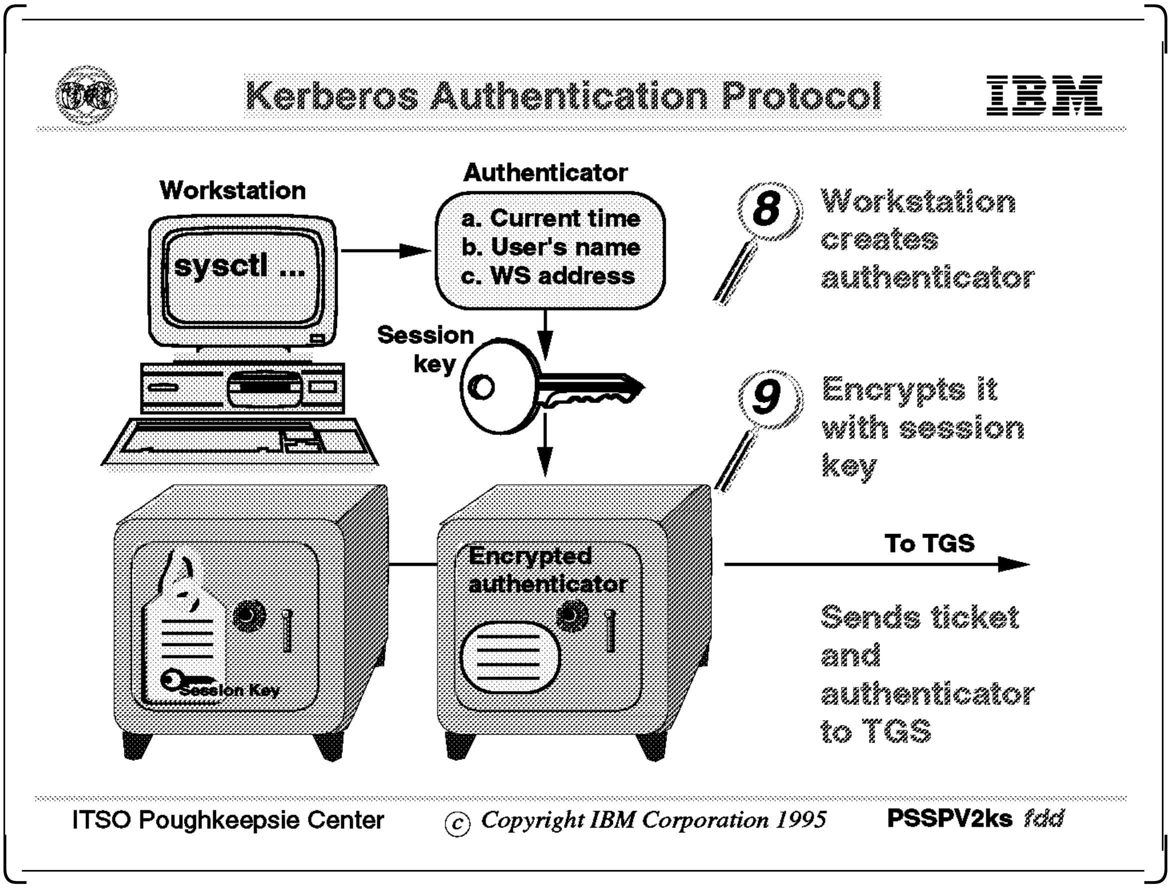
The distributed Kerberos code in the workstation converts the user′s password to a DES key.

Step 7:

The workstation uses the user′s key thus generated to decrypt the ticket and the accompanying session key. This ticket is sometimes referred to as the **ticket granting ticket**.

Only if the user supplies the correct password to the workstation will he be able to decrypt this information properly.

At this point, the workstation has a (ticket granting) ticket and a session key. The workstation cannot read the ticket because it is encrypted with the TGS server′s key.



Kerberos Authentication Protocol

8 Workstation creates authenticator

9 Encrypts it with session key

Sends ticket and authenticator to TGS

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ks fdd

Step 8:

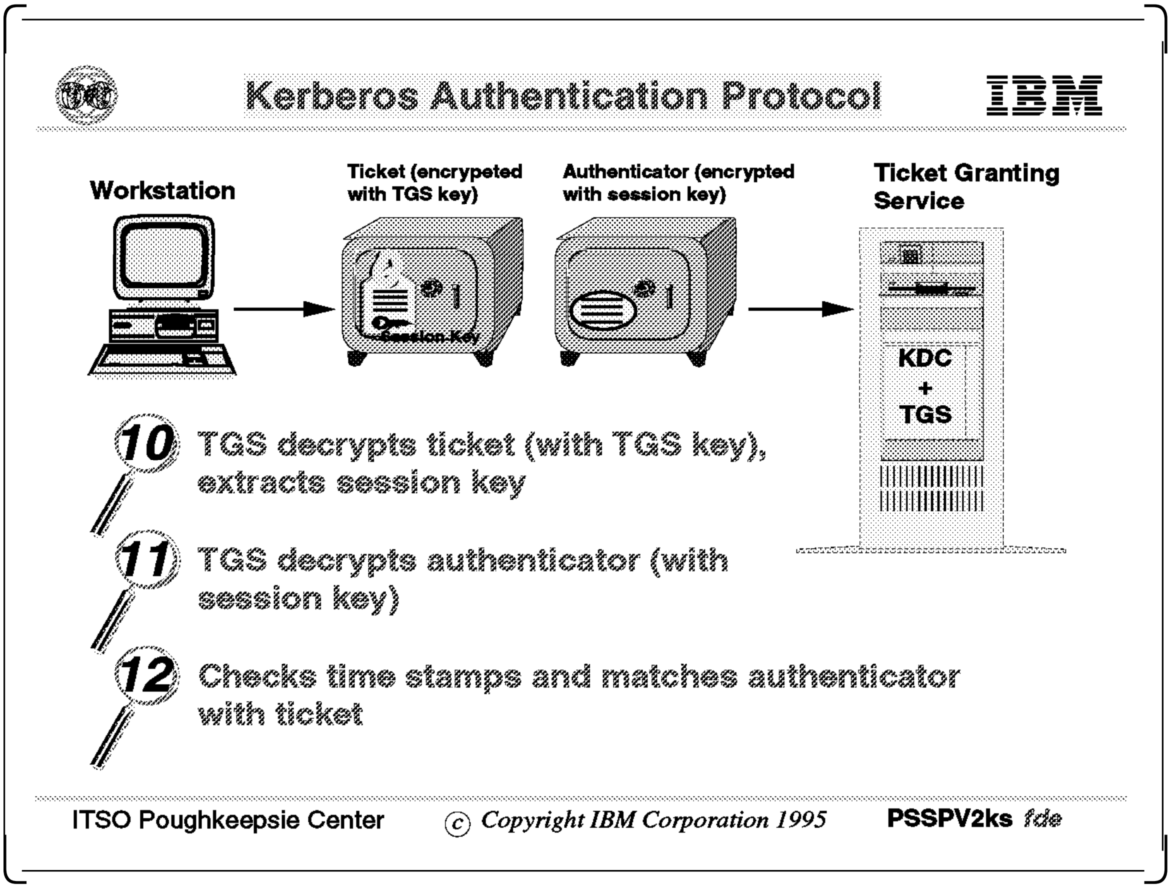
So far, the workstation has not communicated with any protected service yet.

Now, when the user wants to request a service from, say, the *sysctl* server, the workstation creates a packet of data, called an **authenticator**, which consists of the following:

- The current time
- The user′s name
- The workstation address

Step 9:

The workstation encrypts the authenticator with the session key (that it extracted in Step 7). It forwards the authenticator and the ticket to the Ticket Granting Service.



Kerberos Authentication Protocol

Workstation

Ticket (encrypeted with TGS key)

Session Key

Authenticator (encrypted with session key)

Ticket Granting Service

KDC + TGS

10 TGS decrypts ticket (with TGS key), extracts session key

11 TGS decrypts authenticator (with session key)

12 Checks time stamps and matches authenticator with ticket

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ks fde

Step 10:

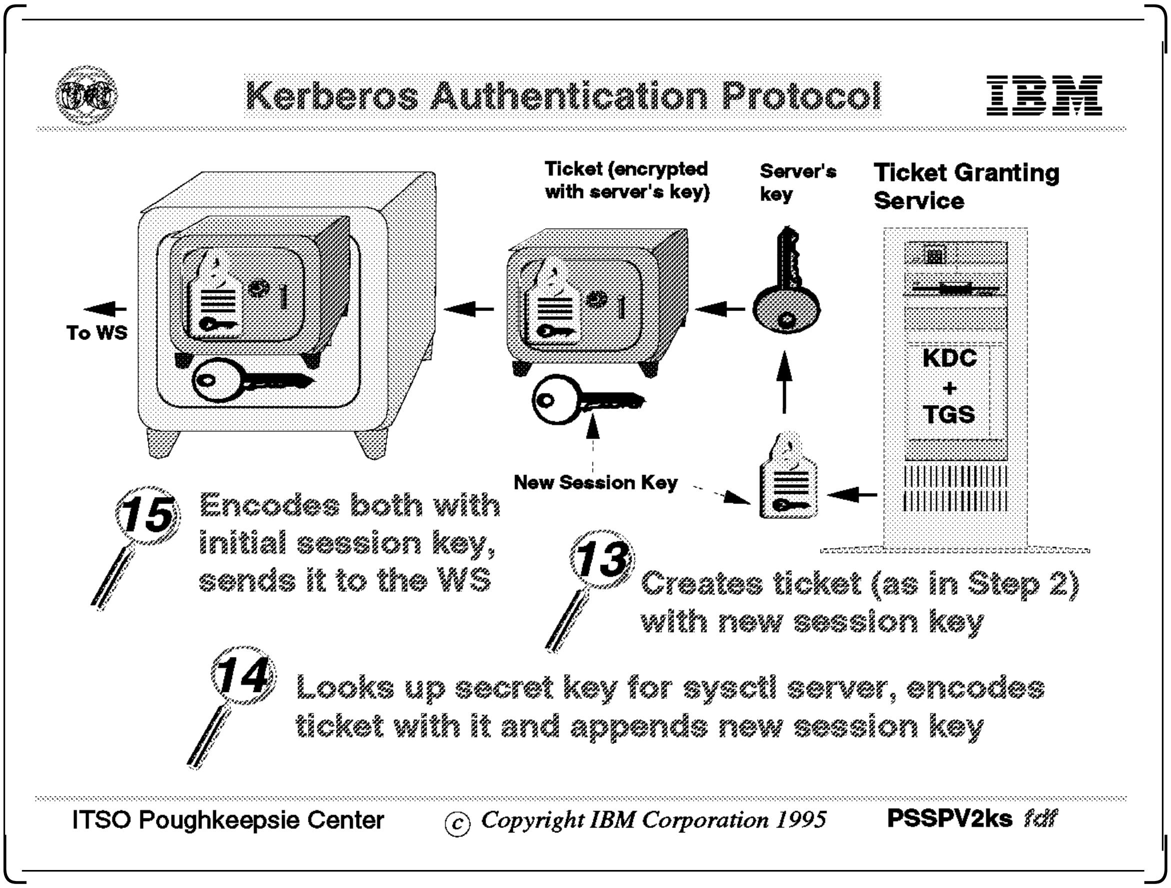
The TGS decrypts the ticket (with the TGS key) and extracts the session key.

Step 11:

The TGS decrypts the authenticator (with the session key obtained in Step 10).

Step 12:

The TGS checks that the time stamps are valid and that the user and the workstation named in the ticket match those in the authenticator.



Kerberos Authentication Protocol

15 Encodes both with initial session key, sends it to the WS

13 Creates ticket (as in Step 2) with new session key

14 Looks up secret key for sysctl server, encodes ticket with it and appends new session key

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* **PSSPV2ks** *fdf*

Step 13:

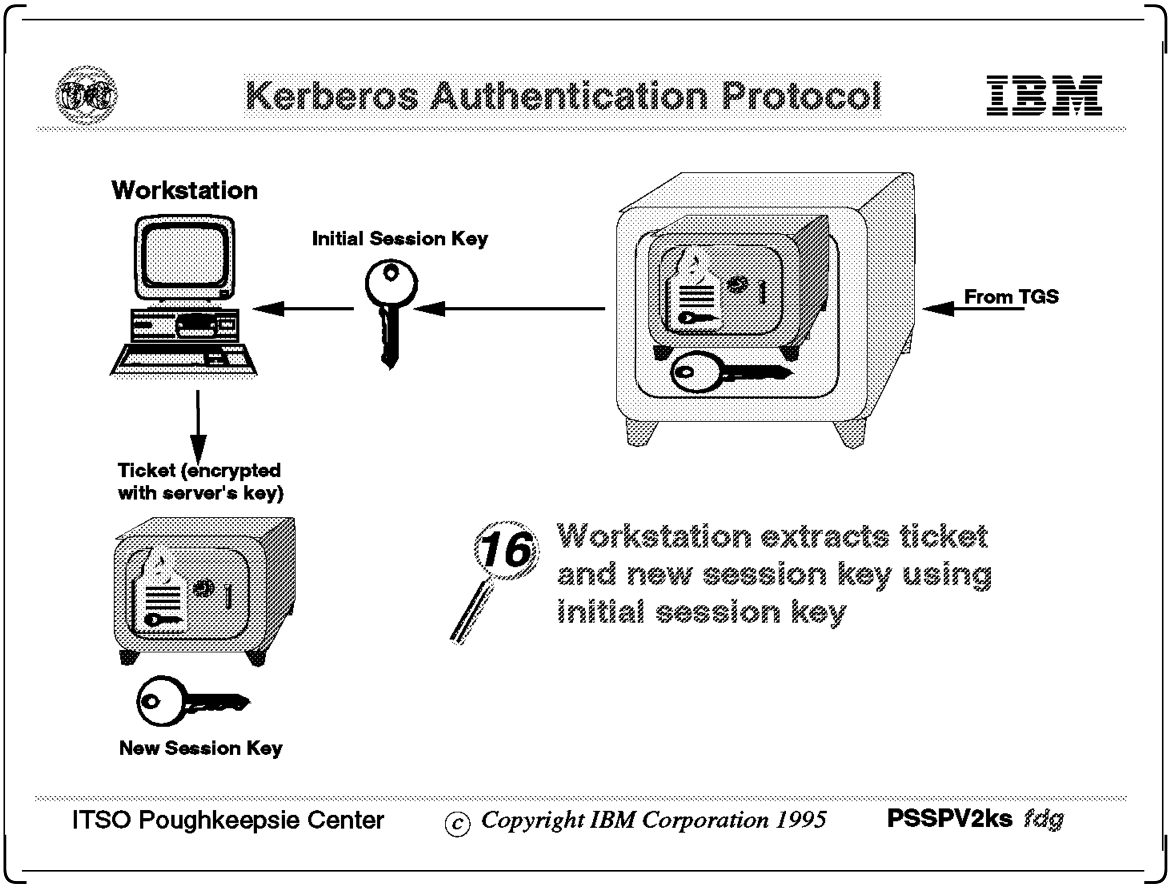
The TGS creates a ticket, with the same content as that created in Step 2, but with a *new* session key.

Step 14:

The TGS looks up the secret key for the *sysctl* server, encrypts the ticket with it, and appends the *new* session key.

Step 15:

The TGS encodes both (the encrypted ticket and the new session key) with the *initial* session key, and sends the encrypted message to the workstation.



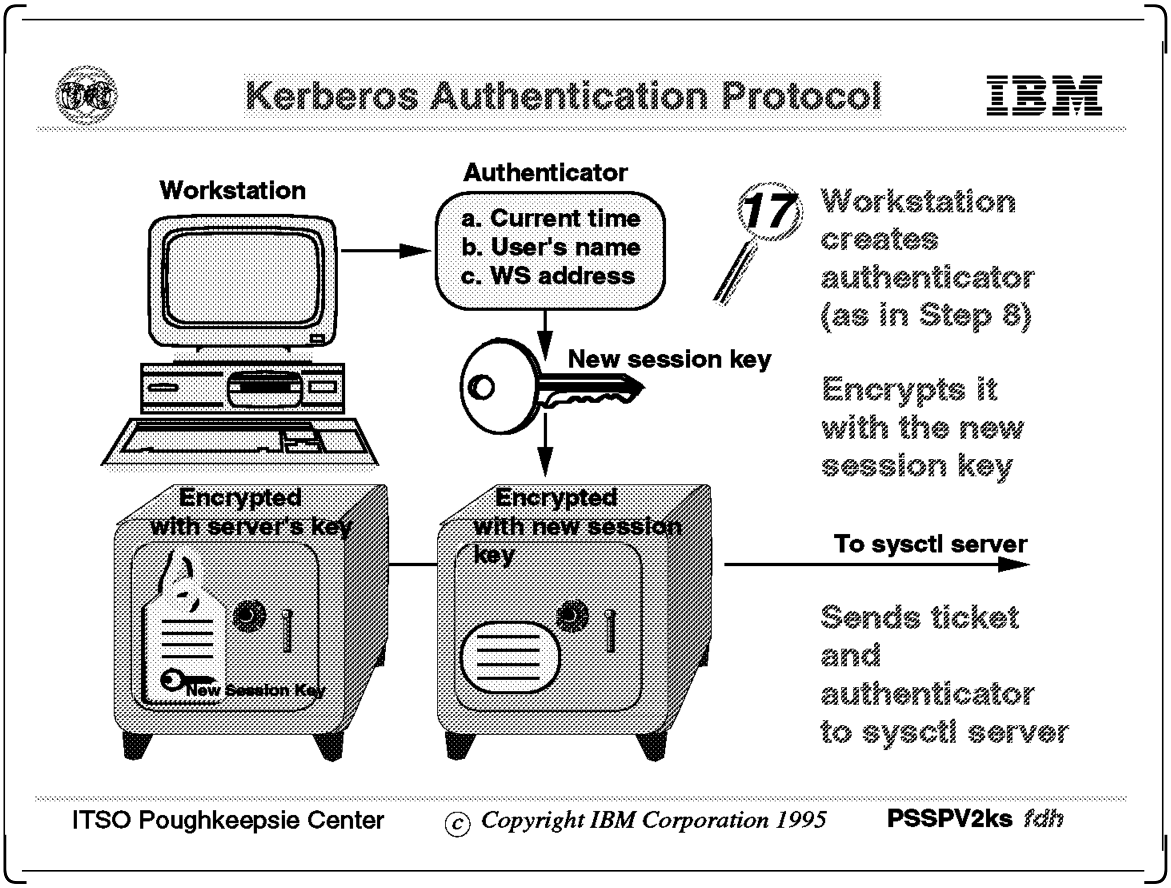
Kerberos Authentication Protocol

16 Workstation extracts ticket and new session key using initial session key

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ks fdg

Step 16:

The workstation uses the initial session key to extract the ticket and the new session key.



Kerberos Authentication Protocol

17 Workstation creates authenticator (as in Step 8)

Encrypts it with the new session key

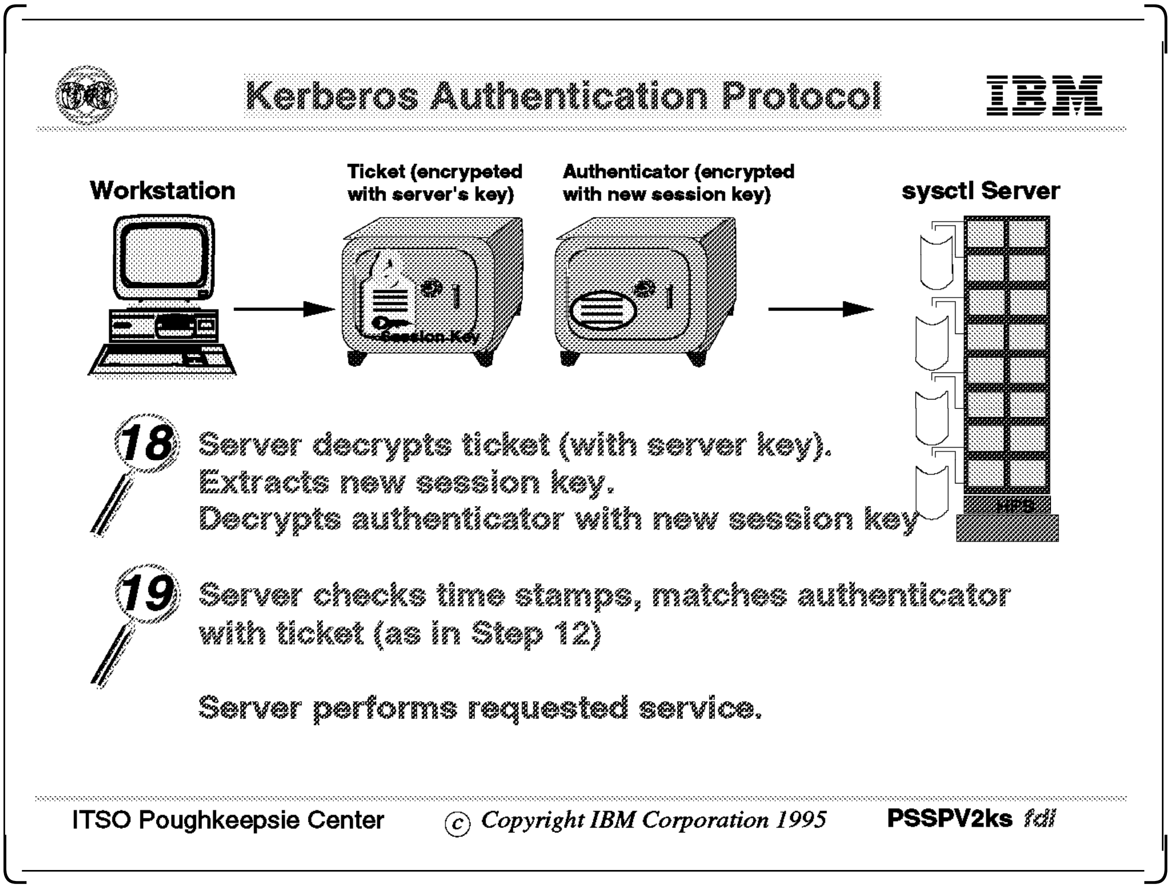
Sends ticket and authenticator to sysctl server

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* **PSSPV2ks** *fdh*

Step 17:

At this point, the workstation has a ticket encrypted with the sysctl server′s key and a new session key. It is thus equipped to send an authenticated request to the (protected) sysctl server.

The workstation creates an authenticator (with the same contents as in Step 8) and encrypts it with the new session key. The workstation then sends the ticket and the authenticator to the sysctl server.



Kerberos Authentication Protocol

Workstation

Ticket (encrypeted with server's key)

Authenticator (encrypted with new session key)

sysctl Server

Session Key
1
1
HPS

18 Server decrypts ticket (with server key).
Extracts new session key.
Decrypts authenticator with new session key

19 Server checks time stamps, matches authenticator with ticket (as in Step 12)

Server performs requested service.

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ks fdl

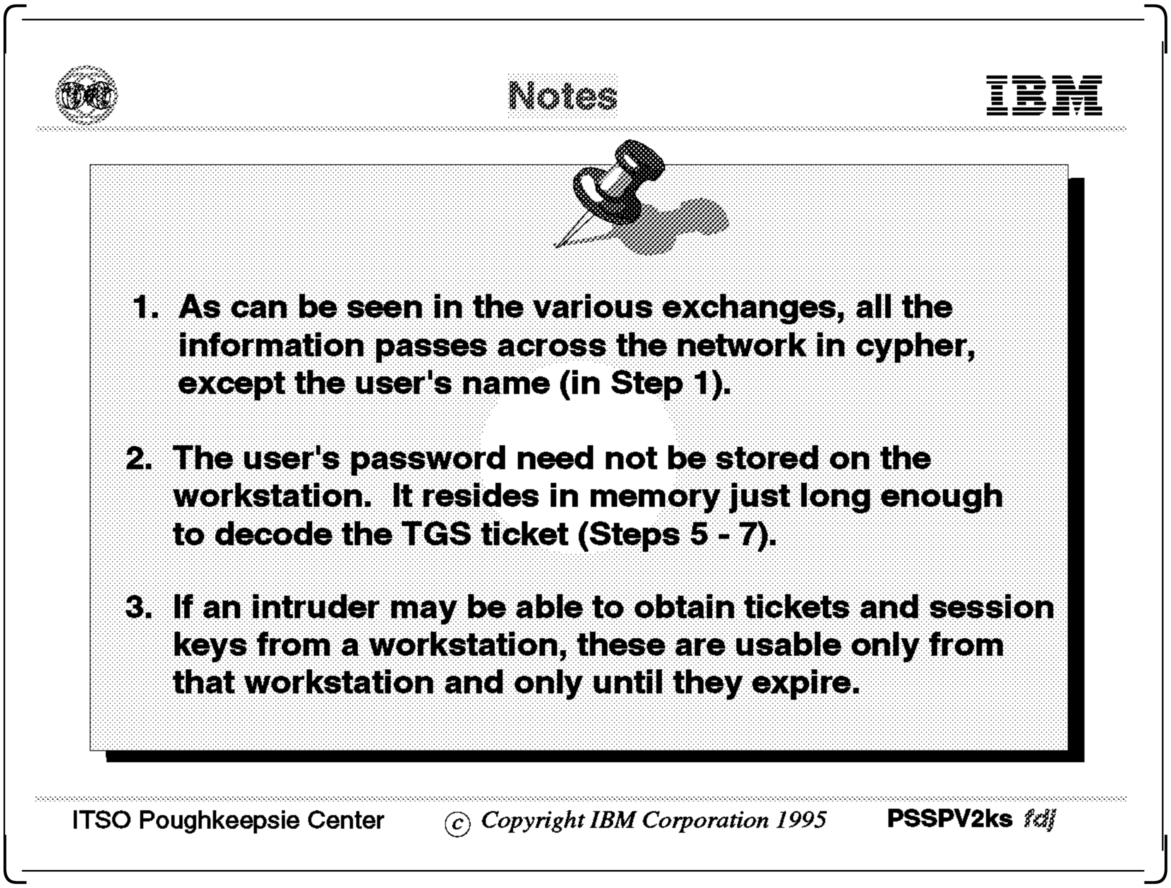
Step 18:

The sysctl server decrypts the ticket, using its own secret key, and extracts the session key. The sysctl server decrypts the authenticator, using the session key just extracted.

Step 19:

The sysctl server checks that the time stamps are valid and that the user and the workstation named in the ticket match those in the authenticator.

If all the checks pass, the sysctl server performs the requested function.



Notes

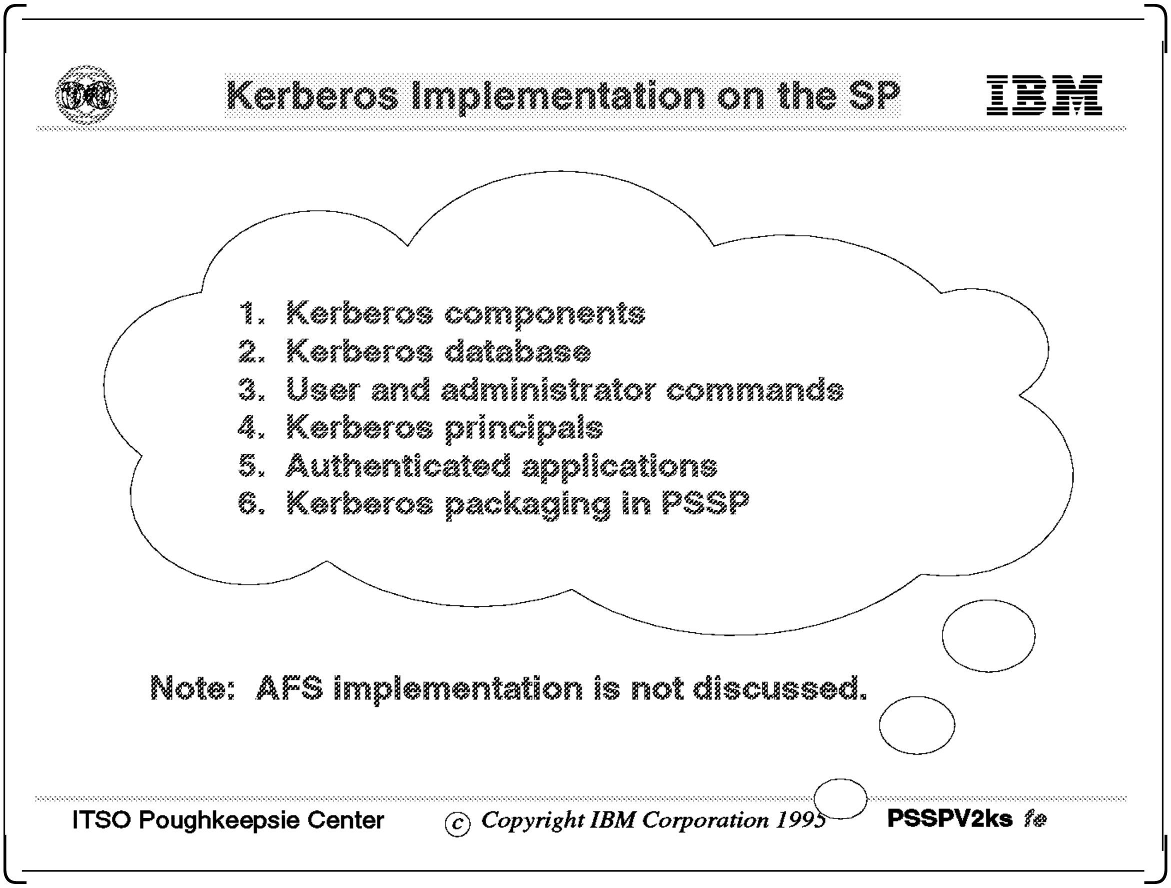
1. As can be seen in the various exchanges, all the information passes across the network in cypher, except the user's name (in Step 1).

2. The user's password need not be stored on the workstation. It resides in memory just long enough to decode the TGS ticket (Steps 5 - 7).

3. If an intruder may be able to obtain tickets and session keys from a workstation, these are usable only from that workstation and only until they expire.

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* **PSSPV2ks** *fdj*

Chapter 16. Kerberos Implementation on the SP



Kerberos Implementation on the SP

1. Kerberos components
2. Kerberos database
3. User and administrator commands
4. Kerberos principals
5. Authenticated applications
6. Kerberos packaging in PSSP

Note: AFS implementation is not discussed.

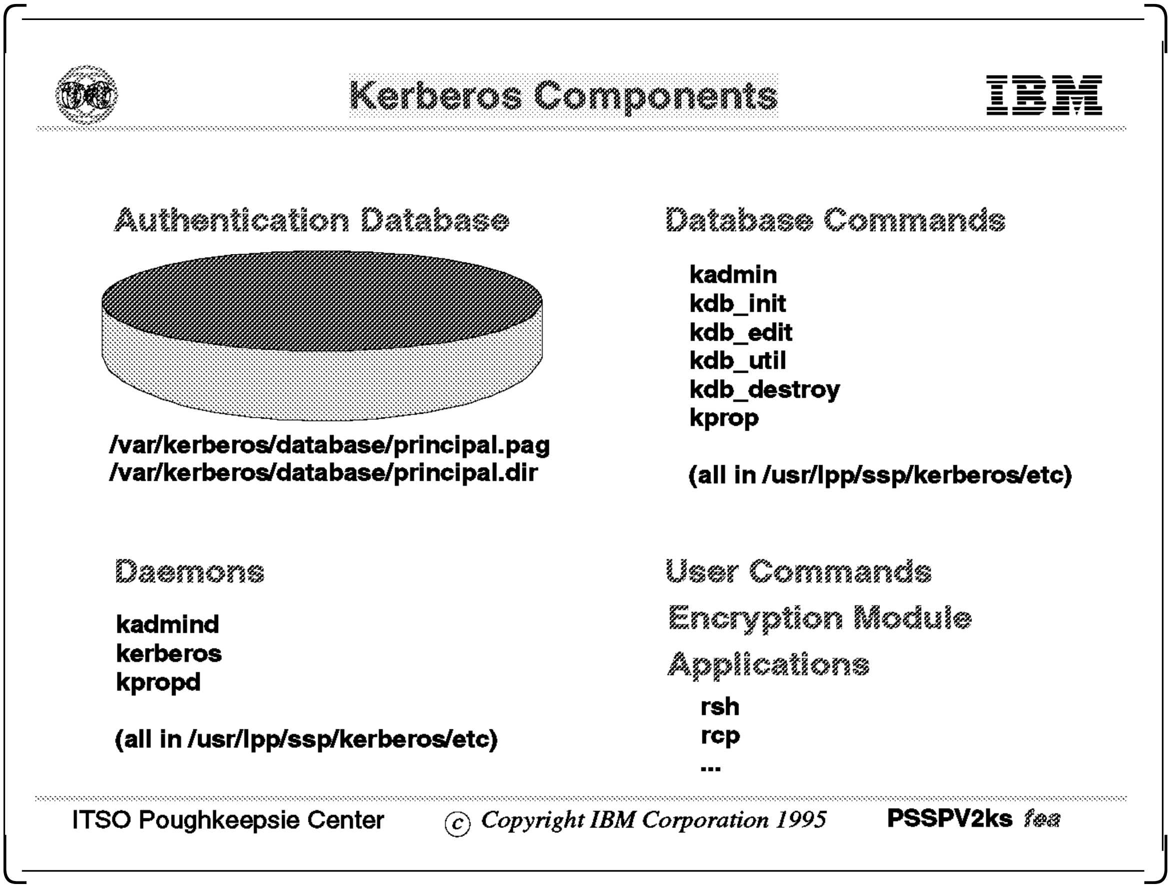
ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ks fe

SP authentication is based on Version 4 of MIT Kerberos.

The following topics are discussed:

1. Kerberos components
2. Kerberos database
3. User and administrator commands
4. Kerberos principals
5. Authenticated applications
6. Kerberos packaging in PSSP

16.1 Kerberos Components



Kerberos Components

Authentication Database

/var/kerberos/database/principal.pag
/var/kerberos/database/principal.dir

Database Commands

kadmin
kdb_init
kdb_edit
kdb_util
kdb_destroy
kprop

(all in /usr/lpp/ssp/kerberos/etc)

Daemons

kadmind
kerberos
kpropd

(all in /usr/lpp/ssp/kerberos/etc)

User Commands

Encryption Module

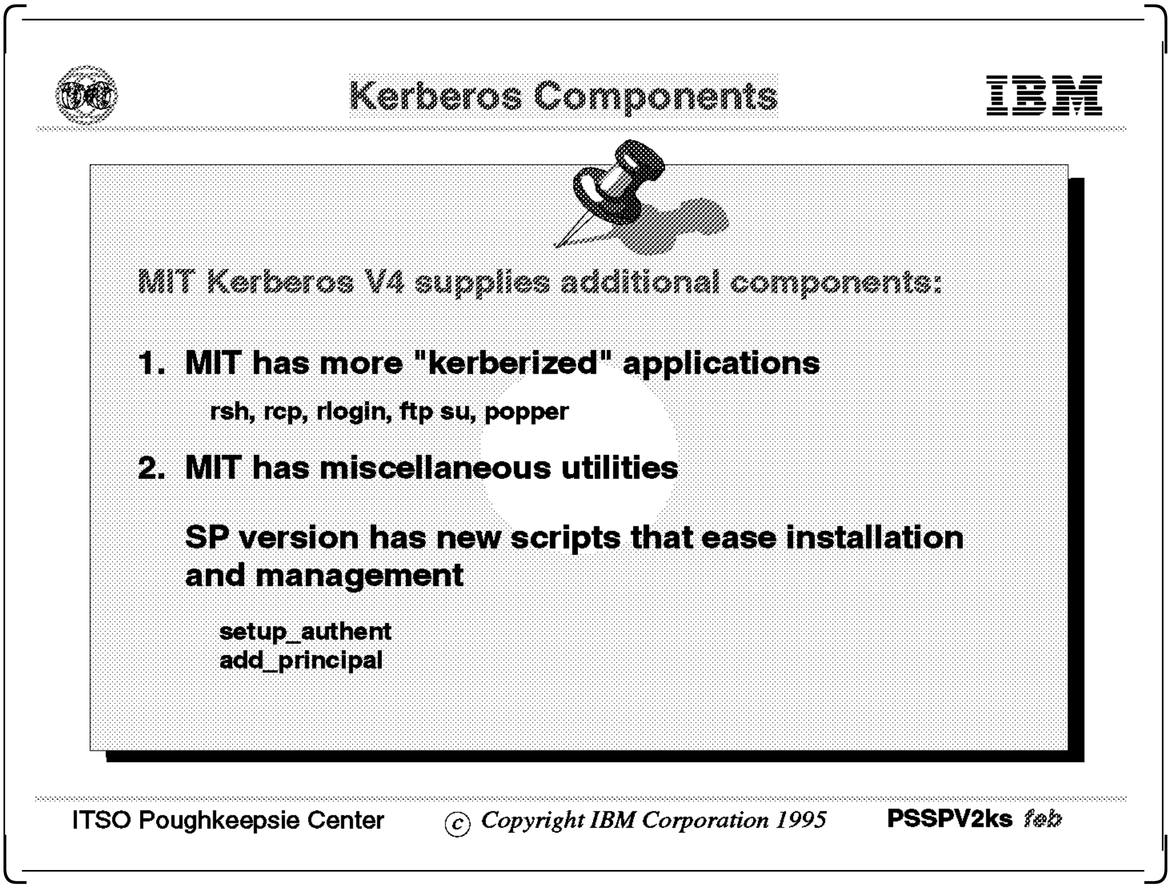
Applications

rsh
rcp
...

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* PSSPV2ks *fea*

Kerberos keeps a **database** of principals and their DES keys, includes **daemons** that perform the functions of authentication server (**kerberos**), database administration server (**kadmind**) and database propagation server (**kpropd**), and provides various administrative commands to maintain the authentication system. In addition, a number of *kerberized* applications, that is, protected services, are supplied with Kerberos.

These components are summarized on the foil and explained in detail in the following sections.



Kerberos Components

MIT Kerberos V4 supplies additional components:

1. **MIT has more "kerberized" applications**

   rsh, rcp, rlogin, ftp su, popper

2. **MIT has miscellaneous utilities**

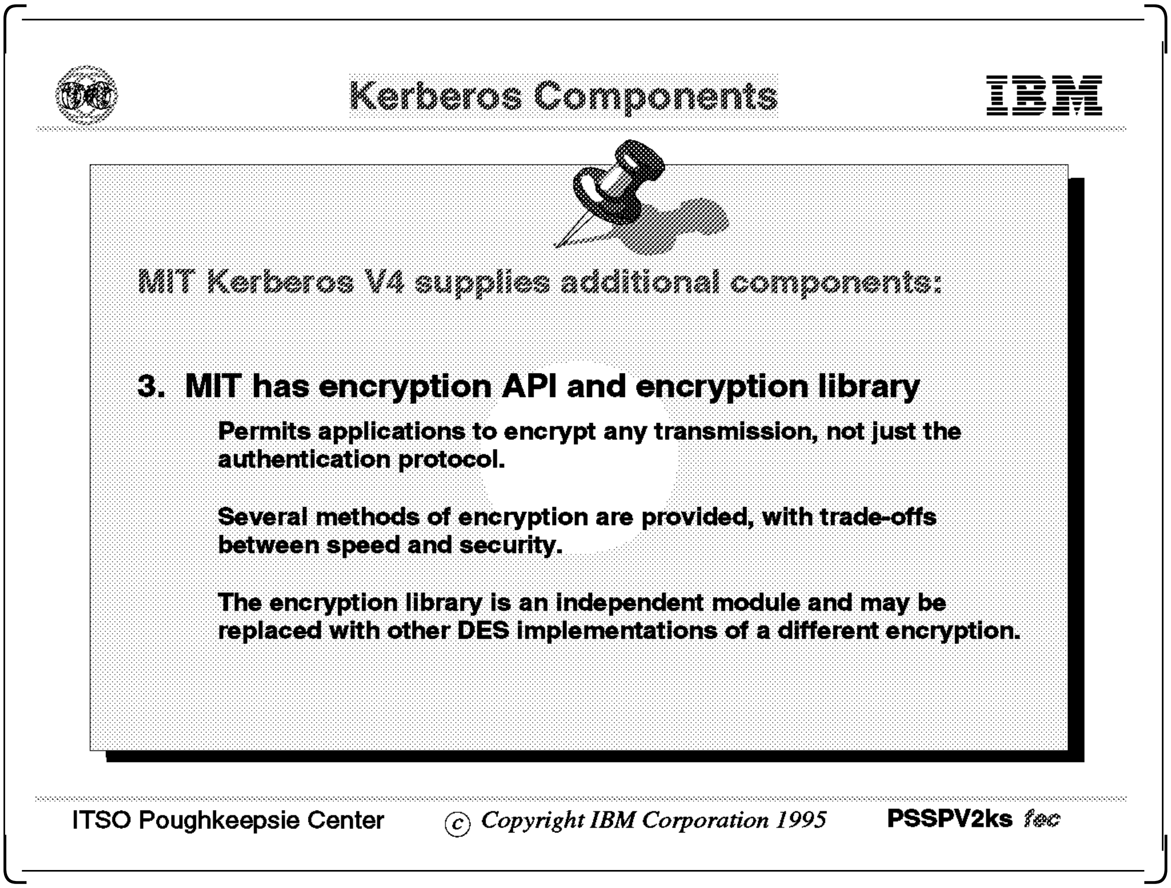
   SP version has new scripts that ease installation and management

   setup_authent
   add_principal

ITSO Poughkeepsie Center **PSSPV2ks** *feb*

Some of the kerberized applications that are part of MIT Kerberos, such as **rlogin**, **ftp**, **su**, and **popper** are not ported to PSSP.

On the other hand, PSSP includes scripts, such as **setup_authent** that transparently perform the initial setup of Kerberos, and thus effectively hide its complexity to the unfamiliar SP user.



Kerberos Components

MIT Kerberos V4 supplies additional components:

3. **MIT has encryption API and encryption library**

   Permits applications to encrypt any transmission, not just the authentication protocol.

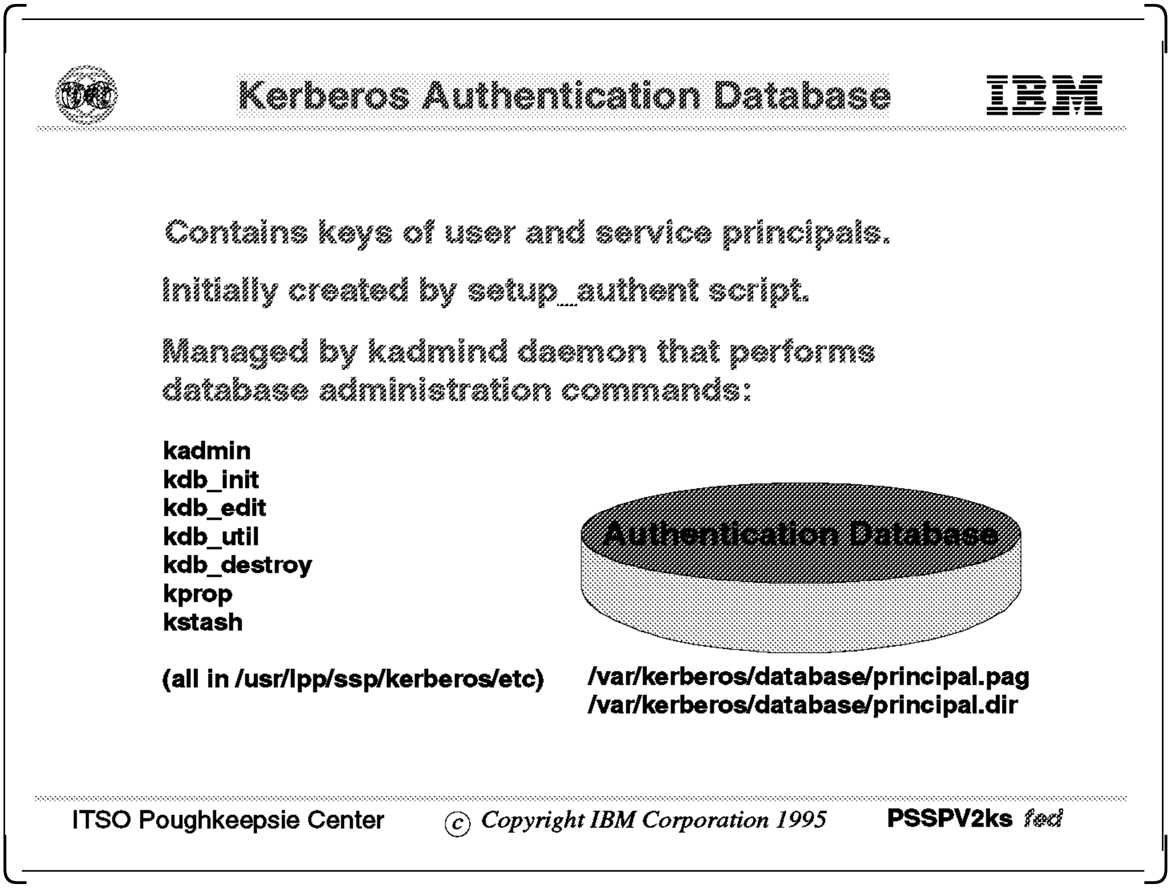
   Several methods of encryption are provided, with trade-offs between speed and security.

   The encryption library is an independent module and may be replaced with other DES implementations of a different encryption.

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* **PSSPV2ks** *fec*

The encryption API, also part of MIT Kerberos, is not supplied with PSSP because of the US export restrictions on any code implementing the Data Encryption Standard algorithm.

16.2 Kerberos Authentication Database



Kerberos Authentication Database

Contains keys of user and service principals.

Initially created by setup_authent script.

Managed by kadmind daemon that performs database administration commands:

kadmin
kdb_init
kdb_edit
kdb_util
kdb_destroy
kprop
kstash

(all in /usr/lpp/ssp/kerberos/etc)

Authentication Database

/var/kerberos/database/principal.pag
/var/kerberos/database/principal.dir

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* **PSSPV2ks** *fed*

The Kerberos database contains the name of the authentication realm and all the principals′ names and their keys. The keys are encrypted with the master key password. The database is made of two binary files **principal.pag** and **principal.dir** in the directory /var/kerberos/database. It may be converted into a viewable ASCII file and back by the commands **kdb_util dump** and **kdb_util load** respectively.

(The Kerberos database′s format is probably similar to that of the standard AIX files DB.pag and DB.dir in /etc/aliasesDB used by Mail, or that of rgb.pag and rgb.dir in /usr/lpp/X11/lib/X11 used by AIXwindows.)

The database is created by the command **kdb_init**. During PSSP installation, this command is invoked by **setup_authent** which also adds the first authentication administrator (often *root.admin*) and the service principals. The authentication administrator may add principals to the database with the commands **kadmin** or **kdb_edit**.

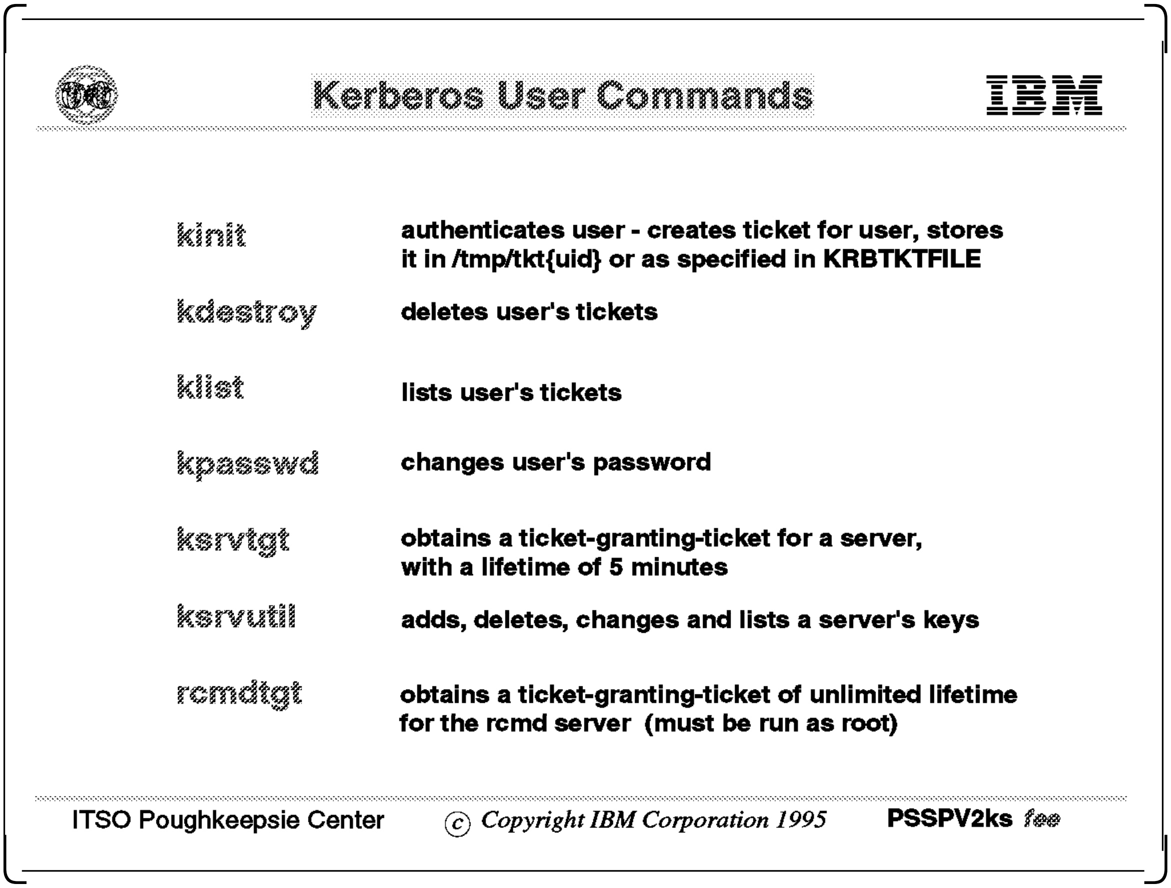
There is no command to directly remove a principal. To delete a principal, you copy the database to an ASCII file with *kdb_util dump*, edit the ASCII file to delete the desired lines, and reload the database from the shortened file with *kdb_util load*. This command can be run by *root* on the primary server only.

Kerberos database commands are performed by the **kadmind** daemon on behalf of the requesting clients.

Note:

The directory /var/kerberos/database also contains the access control lists **admin_acl.add**, **admin_acl.mod** and **admin_acl.get** that define the authentication administrators (see 18.2.7, “Kerberos Access Control Lists” on page 225).

16.3 Kerberos User Commands



Kerberos User Commands

kinit	authenticates user - creates ticket for user, stores it in /tmp/tkt{uid} or as specified in KRBTKTFILE
kdestroy	deletes user's tickets
klist	lists user's tickets
kpasswd	changes user's password
ksrvtgt	obtains a ticket-granting-ticket for a server, with a lifetime of 5 minutes
ksrvutil	adds, deletes, changes and lists a server's keys
rcmdtgt	obtains a ticket-granting-ticket of unlimited lifetime for the rcmd server (must be run as root)

ITSO Poughkeepsie Center ⓒ *Copyright IBM Corporation 1995* **PSSPV2ks** *fee*

16.3.1 kinit

The **kinit** command is used to authenticate a user to the Kerberos authentication services. A ticket granting ticket is obtained and stored in the user′s ticket cache file (see 18.2.3, “Ticket Cache File” on page 221). All the user′s previous tickets are deleted. Invoking the *kinit* command is sometimes referred to as **logging in** to the Kerberos authentication services.

Logging in to the authentication services is a separate step from logging in to AIX. The Kerberos principal name space is separate from the AIX user name space. However, things can be made easier for users by assigning principal names and passwords that are the same as AIX user names and passwords. You could also transparently execute the *kinit* command by including it in the **.profile** (for ksh) or **.login** (for csh) scripts.

The easiest way to invoke the *kinit* command is to specify the principal as the only argument, without flags:

kinit *<principal>*

For example:

```
kinit henry
kinit root.admin
kinit michel@ITSC.POK.IBM.COM
```

If you enter **kinit** without an argument, it prompts you for the various parts of the principal. For example:

```
kinit
Kerberos name: henry
```

or

```
kinit -i
Kerberos name: root
Kerberos instance: admin
```

or

```
kinit -r
Kerberos name: michel
Kerberos realm: ITSC.POK.IBM.COM
```

A common error is to omit the *kinit*argument and reply with more than the name to the prompt. For example:

```
kinit
Kerberos name: root.admin
2503-003 Bad Kerberos name format
```

The default lifetime of the ticket obtained by *kinit* is 30 days or the maximum assigned to your principal by the authentication administrator, whichever is less. For added security, you may request a ticket granting ticket with a lifetime shorter than the default by specifying the **-l** flag. For example, to request a ticket valid for ten minutes:

```
kinit -l
Kerberos name: henry
Kerberos ticket lifetime (minutes): 10
```

Ticket lifetime is discussed in more detail in 19.5, “Ticket Lifetime” on page 243.

A user′s tickets are shared by all processes running under the user′s *uid*. If a user has multiple logins, runs background processes or shares a *uid*, the tickets in use may be destroyed by a subsequent *kinit* or similar command. This can be avoided by storing the tickets in *multiple cache files*. In this case, the KRBTKFILE environment variable must be set to the path name of a unique ticket cache file for each login session or background process, before using authentication services. For example:

```
export KRBTKFILE=/tmp/tickets.$$
```

16.3.2 kdestroy

The **kdestroy** command writes zeros to the user′s *current* ticket cache file and then removes the file. As a result, the user′s all active authentication tickets are deleted. The path to the *current* cache file is discussed in 18.2.3, “Ticket Cache File” on page 221.

For added security, you may wish to destroy your tickets automatically when you log out. C shell users can accomplish this by including the *kdestroy* command in the **.logout** script.

16.3.3 klist

The **klist** command (without arguments) displays, for each ticket held in the user′s *current* cache file, the principal name, issue time and expiration time. The *current cache file* is discussed in 18.2.3, “Ticket Cache File” on page 221.

The command **klist -file** *<filename>* displays the same information in the cache file *<filename>*. The ticket granting ticket is displayed first, followed by the service tickets. The key necessary to decrypt the service tickets is found in the server key table (see below).

The command **klist -srvtab** or **klist -srvtab -file** *<filename>* displays the local instances of services and their private key versions found in the *server key table*. If **-file** is not specified, the default *server key table* is **/etc/krb-srvtab**. Specifying a server key table other than the system default is not supported in the SP system. The server key file is accessible only to the user owning the server daemons, that is, only to *root*.

16.3.4 ksrvutil

The **ksrvutil** command allows you to display, add, change or delete entries in the server key table */etc/krb-srvtab*.

16.3.5 ksrvtgt and rcmdtgt

These commands are intended to be used in shell scripts running with *root* privileges. The command:

ksrvtgt *<principal>*

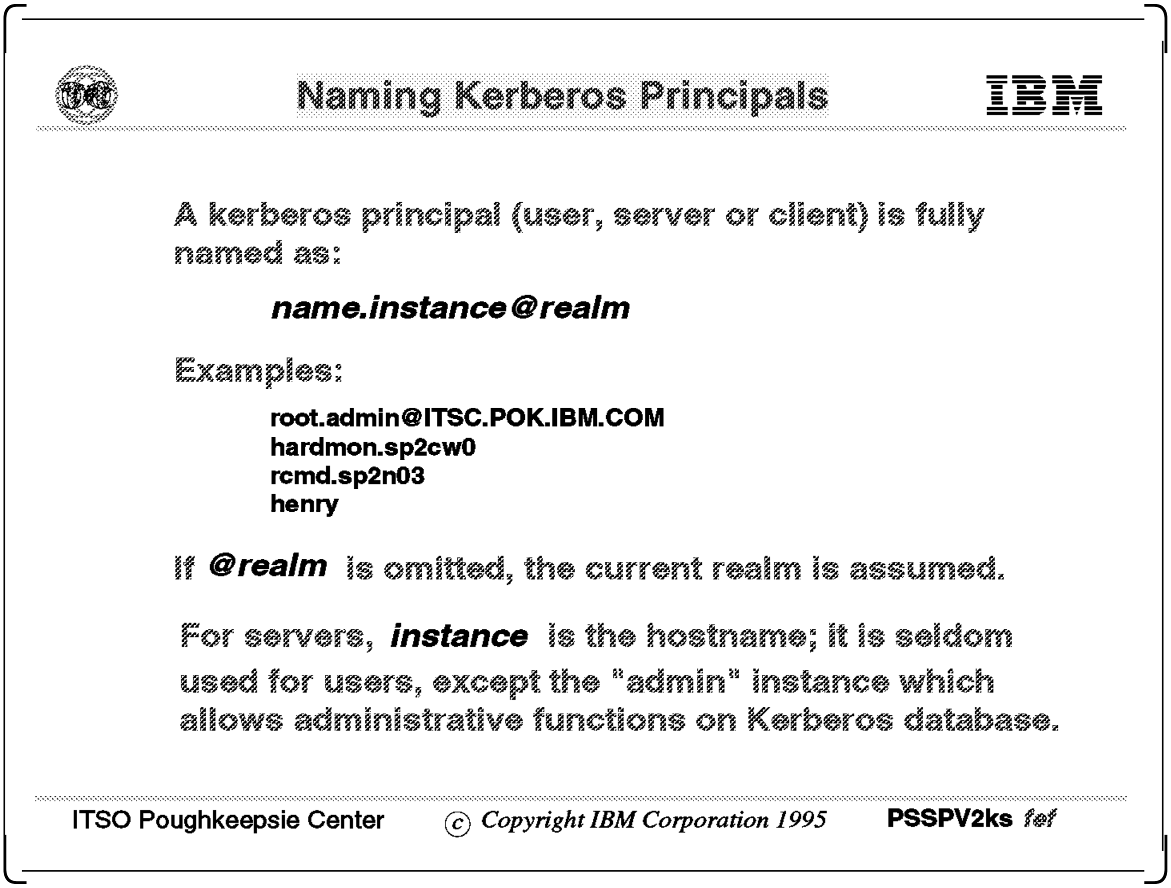
obtains a ticket granting ticket for *<principal>* with a lifetime of five minutes and stores it in the ticket cache file. (The ticket cache file is discussed in 18.2.3, “Ticket Cache File” on page 221). The command:

rcmdtgt

obtains and caches a ticket-granting-ticket for the local realm, with a maximum allowed timelife, using the service key for the instance of rcmd on the local host.

When using the SP implementation of Kerberos, the tickets obtained with *rcmdtgt* never expire. Under AFS authentication, the maximum lifetime is 30 days.

16.4 Naming Kerberos Principals



Naming Kerberos Principals

A kerberos principal (user, server or client) is fully named as:

name.instance@realm

Examples:

root.admin@ITSC.POK.IBM.COM
hardmon.sp2cw0
rcmd.sp2n03
henry

If ***@realm*** is omitted, the current realm is assumed.

For servers, ***instance*** is the hostname; it is seldom used for users, except the "admin" instance which allows administrative functions on Kerberos database.

ITSO Poughkeepsie Center ⓒ *Copyright IBM Corporation 1995* **PSSPV2ks** *fef*

A **principal** represents a protected *service* or the *user* of such a service.

A service principal is assumed by a *server* program for authentication purposes. A server is usually a set of daemon and child processes running on a particular host. Multiple servers may use the same service principal. For example, the **rcmd** service principal is used by the *kshd*, *sysctld* and *kpropd* server daemons. Similarly, the **hardmon** service principal is used by the *hardmon* and *splogd* server daemons.

A user principal represents an AIX user and all processes running under that user's *uid* that request a service protected by Kerberos (in this context, the processes are also referred to as clients). A user principal *name* may or may not be the same as that user's AIX name. From a straight authentication point of view, the Kerberos names are separate from AIX names.

Principals are fully named as:

name.instance@realm

The *name* part is an identifying string unique within the realm. A service principal's *instance* represents a particular occurrence of the server as perceived from the network. It is set to the hostname of the workstation on which the server runs. If a host has multiple interfaces, and hence perceived from the network as multiple hostnames (interface names), there is an *instance* of the service for each interface. For example, if node 1 has four network interfaces, the *rcmd* principal may have the four instances:

rcmd.spn01en
rcmd.spn01tr
rcmd.spn01fi
rcmd.spn01sw

For user principals, the *instance* allows a single user to assume addional or alternate roles with different authority. For example, the Kerberos database administration commands require that the invoking user have the *admin* instance.

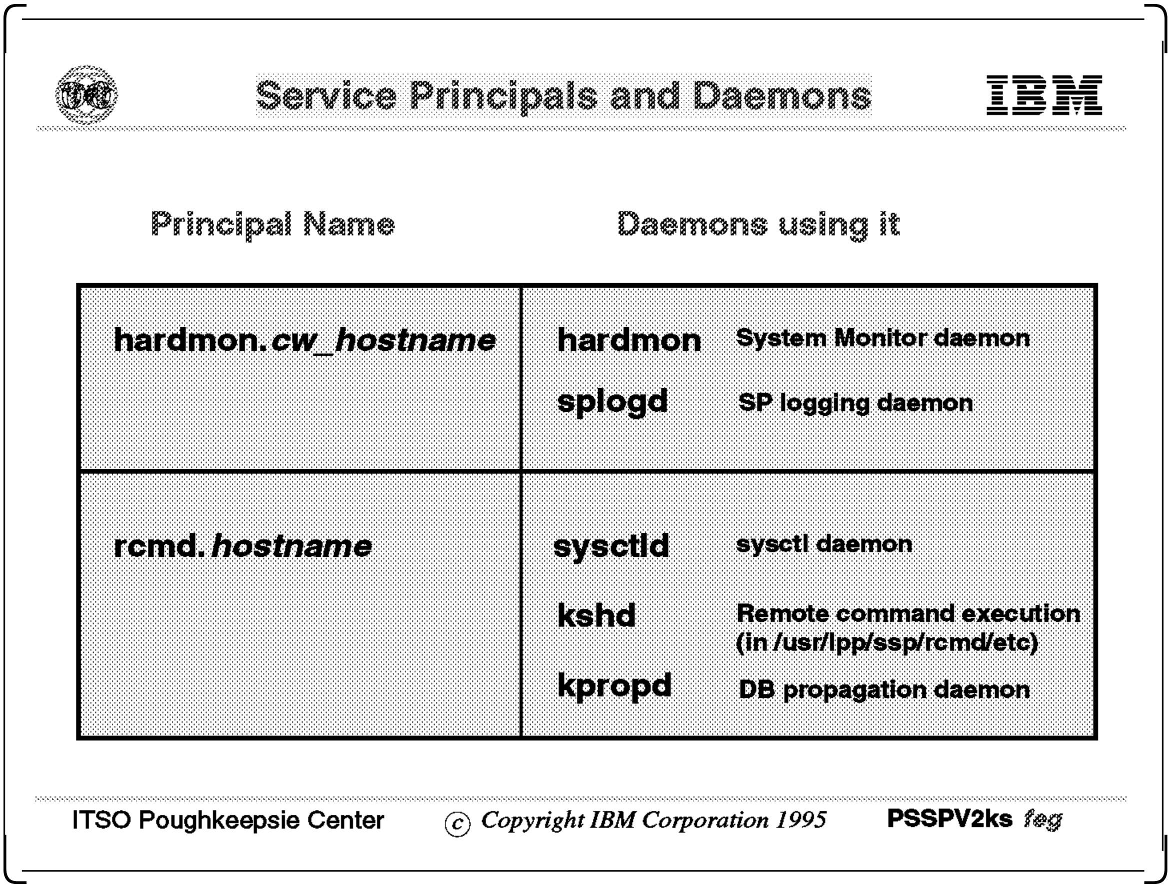
A **realm** is the set of principals sharing the same authentication database and authentication servers. A realm is also the collection of users and (protected) services registered with the same realm name. The realm name is the first line of the **/etc/krb.conf** file. It can be set to any string at the time Kerberos is installed by specifying it in the command:

kdb_init <*realm*>

If <*realm*> is not specified, the realm name is set to the local host′s domain name converted to uppercase. When you use the **setup_authent** script to set up the primary authentication server in an SP system (usually the control workstation), the realm is set by default to that workstation′s domain name converted to uppercase. You must supply the */etc/krb.conf* and the */etc/krb.realms* files if you want to set your own realm name or if the hostname of the primary authentication server has no domain portion.

Principals and their DES keys are stored in the Kerberos authentication database. The keys are encrypted with the database′s master password.

16.5 Service Principals and Daemons



Service Principals and Daemons

Principal Name	Daemons using it	
hardmon.*cw_hostname*	hardmon	System Monitor daemon
	splogd	SP logging daemon
rcmd.*hostname*	sysctld	sysctl daemon
	kshd	Remote command execution (in /usr/lpp/ssp/rcmd/etc)
	kpropd	DB propagation daemon

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* PSSPV2ks *feg*

In SP systems, only two principal names are used by all the Kerberos-authenticated services:

1. The principal name **hardmon** is used by the system monitor daemon (also called **hardmon**) on the control workstation and by the logging daemon (**splogd**).

2. The principal name **rcmd** is used by the **kshd** daemon that serves the authenticated version of the remote commands **rsh** and **rcp**, as well as the **sysctld** and **kpropd** daemons.

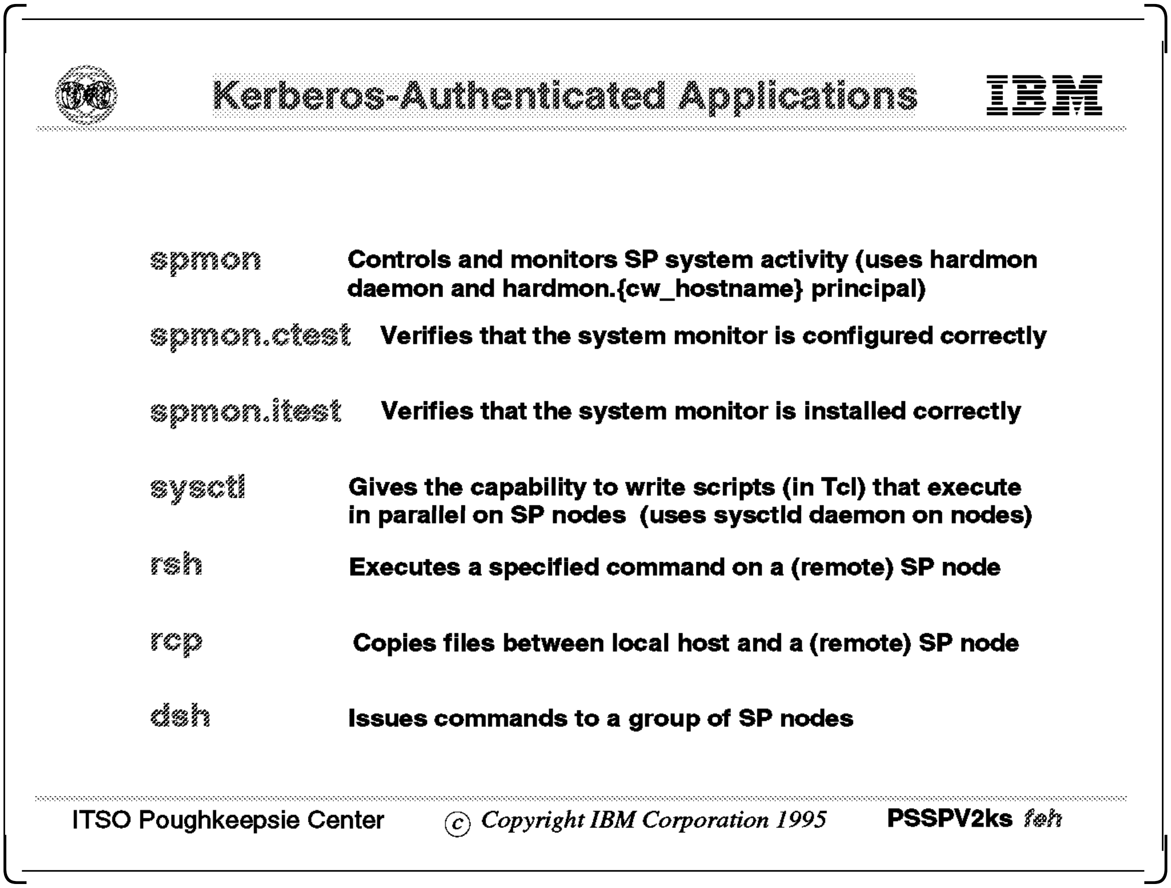
The *instance* for these service principals is the short form of the *hostname* of the local host. On SP nodes with more than one network interface, as well as the control workstation and other hosts where authenticated services are installed, there is an instance of the principal for each interface rather than just the standard hostname. Kerberos always works with the resolved network names rather than any local aliases.

The *hardmon* daemon runs only on the control workstation. The *splogd* daemon usually runs on the control workstation, but may be set up to run on another RS/6000 host (only) or on both the control workstation and other hosts. (To run the *splogd* daemon on a host other than the control workstation, install the **ssp.clients** component and run the **setup_logd** command on that host. You may wish to do this, for example, to offload logging from the control workstation or to have your own scripts called when a state change occurs.) If the host where *splogd* runs has multiple network interfaces, there should be a service principal named **hardmon.***<interface_name>* for each interface on that host.

The remote commands may be run from or to the nodes, the control workstation and any RS/6000 host on which the SP authenticated client services are installed. Therefore, there should be a service principal named **rcmd.***<interface_name>* for each interface defined on the nodes, the control workstation and the other hosts, if any.

The **setup_authent** script creates service principals for each network interface defined on the control workstation. The **setup_server** script creates service principals for each network interface defined on each node. On other hosts where authenticated services are installed, service principals must be manually maintained (see 19.4, "Multi-Homed Host Support" on page 241).

16.6 Kerberos-Authenticated Applications



Kerberos-Authenticated Applications

spmon Controls and monitors SP system activity (uses hardmon daemon and hardmon.{cw_hostname} principal)

spmon.ctest Verifies that the system monitor is configured correctly

spmon.itest Verifies that the system monitor is installed correctly

sysctl Gives the capability to write scripts (in Tcl) that execute in parallel on SP nodes (uses sysctld daemon on nodes)

rsh Executes a specified command on a (remote) SP node

rcp Copies files between local host and a (remote) SP node

dsh Issues commands to a group of SP nodes

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* **PSSPV2ks** *feh*

Some of the Kerberos-authenticated commands on SP systems are briefly summarized below. Other protected commands not discussed here include: **cstartup**, **cshutdown**, **sphrdwrad**, **nodecond**, **penotify**, **psyslclr**, and others.

For more detailed information, see publications 4 and 5 in 11.4.1, “References” on page 149.

16.6.1 splm

The **splm** command performs various log management functions on a single host or on a set of hosts in parallel. It may be used to view, archive, gather or collect log and system data. The command is driven by a table file that specifies the target nodes and associated commands or files.

The *view* function may be performed by any authenticated user principal. The other functions require, in addition, that the principal be included in the **/etc/logmgt.acl** file. In addition, the file **/etc/splm.allow** may restrict the table commands that can be executed.

16.6.2 spmon

The **spmon** command executes system monitor functions or opens the monitor′s graphical user interface. System monitor functions include monitoring SP system activity and operating system controls.

This command requires that the user be a Kerberos principal and that the principal name be included in the **/spdata/sys1/spmon/hmacls** file with the appropriate authorization.

16.6.3 spmon_ctest, spmon_itest

The **spmon_ctest** and **spmon_itest** commands are used during system installation to verify that the system monitor is installed, configured correctly and operational.

These commands require that the user be a Kerberos principal and that the principal name be included in the **/spdata/sys1/spmon/hmacls** file with the appropriate authorization.

16.6.4 nodecond, sphrdwrad

The **nodecond** and **sphrdwrad** commands are used during system installation to obtain the ethernet hardware address of a node′s en0 interface or to initiate a network boot of a node.

These commands require that the user be a Kerberos principal and that the principal name be included in the **/spdata/sys1/spmon/hmacls** file with the appropriate authorization.

16.6.5 sysctl

The **sysctl** command, together with the **sysctld** server daemon, provide the monitoring and execution abilities to remotely manage SP nodes and other hosts in a large distributed computing environment. Typically, one instance of sysctld runs on every node, the control workstation and other hosts on the authentication realm.

The sysctld server is augmented with application specific commands that may vary between servers, as specified by a configuration file (by default **/etc/sysctl.conf**). The expression passed by sysctl to the sysctld server for execution may be anything from a single command to an entire *tool command language (Tcl)* script. The server includes an embedded Tcl interpreter.

Any sysctl user must be a Kerberos principal. The sysctld server also includes a built-in authorization mechanism based on *callbacks* to control the set of commands available to the user. A *callback* is a Tcl routine paired with a sysctl command that determines whether the client is authorized to run the command. A callback may require a user to be listed in an acl file (by default **/etc/sysctl.acl**) for the request to succeed.

16.6.6 rsh

The command **rsh** *<remote_host> <command>* executes *<command>* at *<remote_host*. Standard output and standard error are sent to the local host (where rsh is executed).

The local user (executing rsh) is used at the remote host (to execute *<command>*). The user must be a Kerberos principal. A **$HOME/.klogin** file is not required on the remote host, but if it exists there, it must include the user principal name (see 18.2.2, ".klogin File" on page 220).

For example:

```
rsh sp2n03 df
```

The command df is executed on sp2n03. The result is displayed on the local host.

The path name /usr/lpp/ssp/rcmd/bin must be in the local host′s PATH environment variable before /usr/bin. Otherwise use the form:

```
/usr/lpp/ssp/rcmd/bin/rsh sp2n03 df
```

You may run the specified command on behalf of another user at the remote host by including the -l flag. For example, user principal *giulia* may enter on some host in the realm ITSC.POK.IBM.COM:

```
rsh sp2n03 -l henry df
```

The command df is executed on sp2n03. In this case, a **$HOME/.klogin** file must exist on *henry*′s home directory in sp2n03, and include the line:

```
...
giulia@ITSC.POK.IBM.COM
...
```

Shell metacharacters (such as |, >, and the like) in the input command string are interpreted by the local shell. To have shell metacharacters interpreted on the remote host, place them inside quotes (for example ″|″, ″>″).

The default path used by rsh on the target node is is **/usr/ucb:/bin:/usr/bin** (unless of course *<command>* is specified by a full path name).

If the originating user′s authentication fails, the /usr/lpp/ssp/rcmd/bin/rsh command issues an error message and passes its arguments to /usr/bin/rsh. In this case, the originating user needs normal rsh access to the remote host (through /etc/hosts.equiv or $HOME/.rhosts).

16.6.7 rcp

The **rcp** command copies files or directories between any two hosts in the authentication realm (that is, from the local host to a remote host, from a remote host to the local host, between two remote hosts or within the same remote host).

For example, to copy the local file *mytext* from the local host to the remote host sp2n03:

```
rcp mytext sp2n03:/home/henry
```

The path name /usr/lpp/ssp/rcmd/bin must be in the local host′s PATH environment variable before /usr/bin. Otherwise use the form:

```
/usr/lpp/ssp/rcmd/bin/rcp mytext sp2n03:/home/henry
```

In this example, the local user (executing rcp) is used at the remote host to set ownership of the file copied and to determine the file access privileges at the remote host. The user must be a Kerberos principal. A **$HOME/.klogin** file is not required on the remote host, but if it exists there, it must include the user principal name (see 18.2.2, ".klogin File" on page 220).

You may copy to or from another user at the remote host. For example, user principal *giulia* may enter on some host in the realm ITSC.POK.IBM.COM:

```
rcp -r private henry@sp2n03:private
```

The flag **-r** indicates that *private* is a directory to be recursively copied. The prefix **henry@** indicates that user ID henry determines file access privileges and sets ownership of the transferred files and directories at the remote host. Because the destination is not specified as a full path name, it is interpreted as beginning at the home directory of user henry at the remote host. The directory *private* at the local host is recursively copied to the home directory of user henry at remote host sp2n03.

In this case, a **$HOME/.klogin** file must exist on *henry*′s home directory in sp2n03, and include the line:

```
...
giulia@ITSC.POK.IBM.COM
...
```

If the originating user′s authentication fails, the /usr/lpp/ssp/rcmd/bin/rcp command issues an error message and passes its arguments to /usr/bin/rcp. In this case, the originating user needs normal rcp access to the remote host (through /etc/hosts.equiv or $HOME/.rhosts).

16.6.8 dsh

The **dsh** command uses rsh to execute a specified command on any group of nodes or other remote RS/6000 hosts within the authentication realm, in parallel.

The group of target hosts is called the *working collective*. The working collective is specified in the command line or by setting the WCOLL environment variable to the name of a file including the target host names, one hostname per line.

For example:

1. Issue the *ps* command on all the hosts in the working collective

   ```
   dsh ps
   ```

2. Issue the *ps* command on each host listed in the file *myhosts*

```
WCOLL=./myhosts dsh ps
```

3. Set the current collective to all the SP nodes (**-a** and **-G** flags) and issue the *ps* command in parallel

   ```
   dsh -aG ps
   ```

4. Set the current collective to the nodes in the current partition (**-a** flag) plus the hosts indicated after the **-w** flag,, issue the *cat* command in parallel and format the output on the local host (**dshbak** command)

   ```
   dsh -w halifax,verdi,augustus -a cat /etc/passwd | dshbak -c
   ```

5. Run the *ps* command on the working collective hosts and filter the result on the local host

   ```
   dsh ps -ef | grep root
   ```

   As for the *rsh* command, shell metacharacters (such as |, >, and the like) in the input command string are interpreted by the local shell. To have shell metacharacters interpreted on the remote host, place them inside quotes (for example ″|″, ″>″).

6. Run the *ps* command and filter the results on the working collective hosts

   ```
   dsh ps -ef ″|″ grep root
   ```

   or

   ```
   dsh ′ps -ef | grep root′
   ```

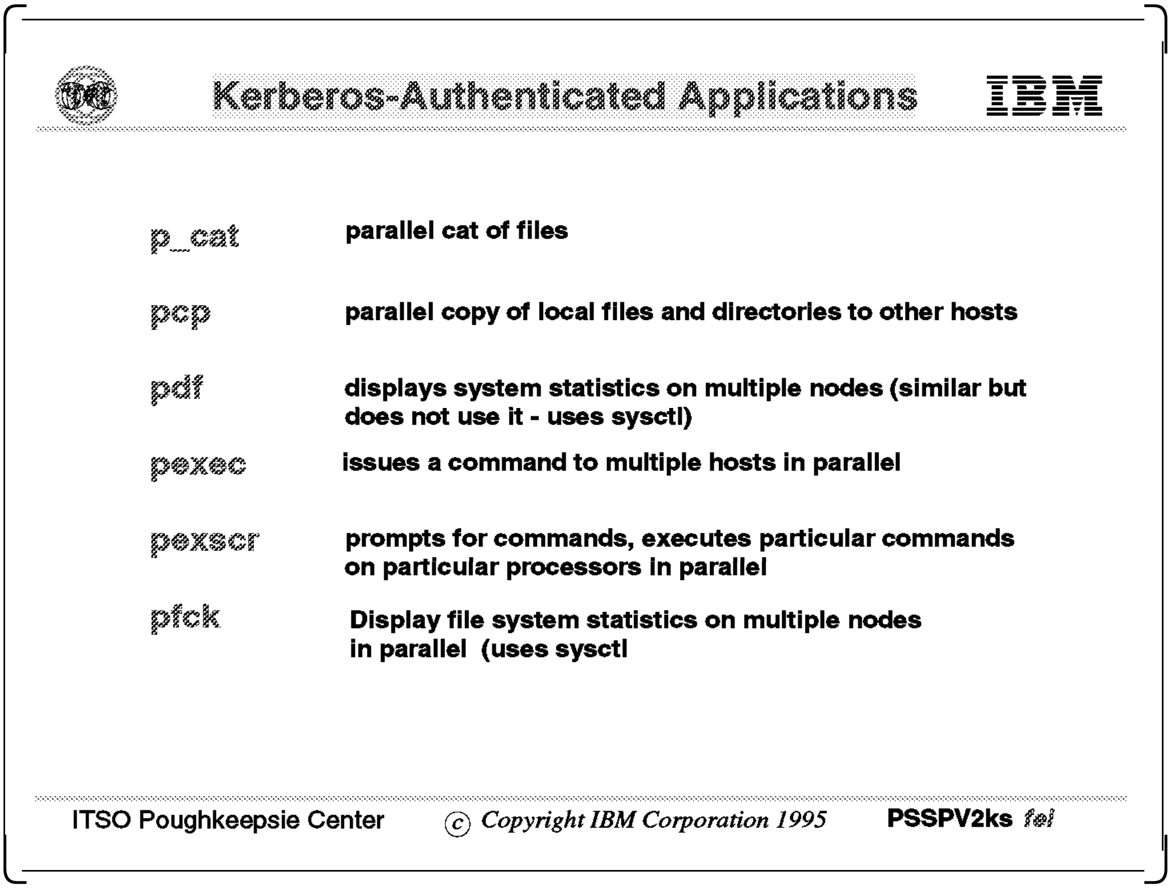
   Compared to the previous example, this can improve performance significantly.

7. Run the *ps* command on the nodes occupying slot 1 of the first four frames in parallel

   ```
   hostlist -s 1-4:1 | dsh -w - ps
   ```

8. Run the *ps* command on the nodes 1 through 16 and 33 through 35 as well as on host verdi

   ```
   hostlist -n 1-16,33-35 -w verdi | dsh -w - ps
   ```



Kerberos-Authenticated Applications

p_cat **parallel cat of files**

pcp **parallel copy of local files and directories to other hosts**

pdf **displays system statistics on multiple nodes (similar but does not use it - uses sysctl)**

pexec **issues a command to multiple hosts in parallel**

pexscr **prompts for commands, executes particular commands on particular processors in parallel**

pfck **Display file system statistics on multiple nodes in parallel (uses sysctl**

ITSO Poughkeepsie Center ⓒ *Copyright IBM Corporation 1995* **PSSPV2ks** *fol*

The **p*** commands have generally similar functions to **dsh**, with the added convenience that the working collective may be specified as a *node range* (for SP nodes only) or in any of the forms acceptable to the **hostlist** command.

16.6.9 p_cat

The **p_cat** command invokes the AIX **cat** command on multiple hosts in parallel. It uses the dsh command which, in turn, uses the rsh command.

For example:

1. Copy /etc/hosts from each of nodes: frame 1 slots 1-3, frame 2 slots 7-13, frame 4 slot 2 and frame 1 slot 10, to the local file /tmp/hosts.

   p_cat 1-3,23-29,50,10 /etc/hosts >> /tmp/hosts

   Remember that the metacharacter >> is interpreted by the local shell on the local host.

2. Copy /etc/hosts from each of nodes: sp2n01, sp2n02, sp2n03 to the local file /tmp/hosts.

   p_cat -w sp2n01,sp2n02,sp2n03 /etc/hosts >> /tmp/hosts

3. Copy /etc/hosts from each of nodes in the current system partition that are responding, to the local file /tmp/hosts.

   p_cat ′-av′ /etc/hosts >> /tmp/hosts

16.6.10 pcp

The **pcp** command copies files from the local host to one or more others in parallel. It uses the secure version of **rcp** and **dsh**.

16.6.11 pdf

The **pdf** command displays file systems statistics on one or more others in parallel. It is similar to **df** but does not use it. It uses **sysctl** and provides more information than *df*,

16.6.12 pexec

The **pexec** command uses **dsh** to issue a command on multiple hosts in parallel. The output is formatted so that duplicate output is displayed only once.

The commands **pls**, **prm**, **pmv**, **pfind** and **pps** are simply links to **pexec**. If any of the commands **pls**, **prm**, **pmv**, **pfind** or **pps** are renames, they do not work properly.

16.6.13 pexscr

The **pexscr** command executes particular commands on particular processors in parallel. It reads lines of the following format from stdin:

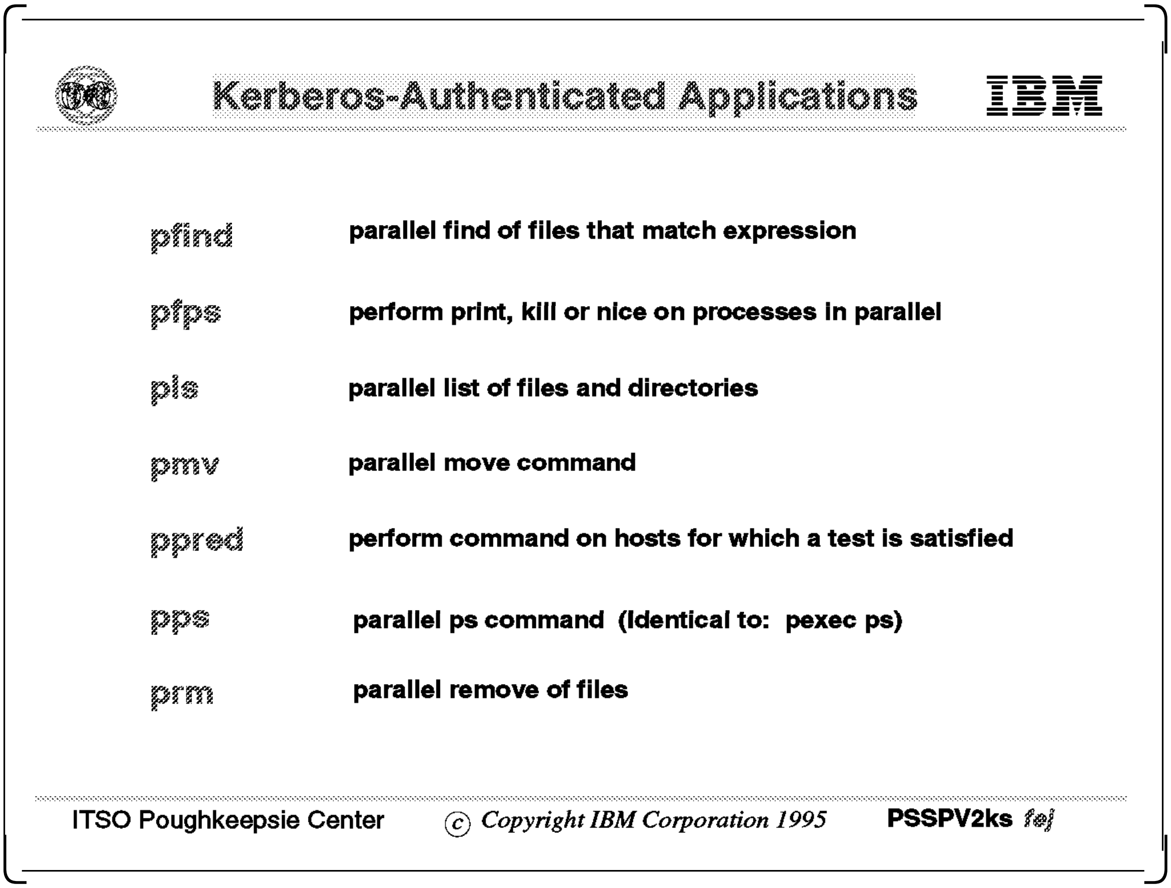
<host_name>:*<command>*

and executes each *<command>* on the specified host. All commands are run in parallel.

pexscr uses **rsh** to run remote commands; local commands are run directly.

16.6.14 pfck

The **pfck** command display file system statistics on multiple hosts in parallel. It uses **sysctl**.



Kerberos-Authenticated Applications

pfind	parallel find of files that match expression
pfps	perform print, kill or nice on processes in parallel
pls	parallel list of files and directories
pmv	parallel move command
ppred	perform command on hosts for which a test is satisfied
pps	parallel ps command (Identical to: pexec ps)
prm	parallel remove of files

ITSO Poughkeepsie Center ⓒ *Copyright IBM Corporation 1995* **PSSPV2ks** *foil*

16.6.15 pfind

The **pfind** command issues the AIX **find** command on multiple hosts in parallel, using **dsh**. This command is identical to **pexec find**.

16.6.16 pfps

The **pfps** command performs operations on processes on multiple hosts in parallel, using **sysctl**. The operations include displaying information about processes (**-print** flag), sending a signal to processes (**-kill** flag), and changing the priority of processes (**-kill** flag).

To use the *-kill* option on a process it does not own or the *-nice* option to raise a process to a higher priority, a principal must further be authorized by being included in the **/etc/sysctl.pfps.acl** file.

16.6.17 pls

The **pls** command issues the AIX **ls** command on multiple hosts in parallel, using **dsh**. Output is written to stdout and formatted so that duplicate output is displayed only once. This command is identical to **pexec ls**.

16.6.18 pmv

The **pmv** command issues the AIX **mv** command on multiple hosts in parallel, using **dsh**. This command is identical to **pexec mv**.

16.6.19 ppred

The **ppred** uses **dsh** to perform a command in parallel on those hosts for which a test is satisfied (and optionally an additional command if the test fails).

The syntax of *ppred* is as follows:

ppred [host_list] *′ksh_test′ ′true_command′* [*′false_command′*]

host_list may be in any of the forms acceptable to the **hostlist**command, as in other **p*** commands.

The second argument *′ksh_test′* is passed to the remote hosts and evaluated there using the **ksh test** command.

The third argument *′true_command′* is executed on the remote hosts for which the test is true.

The optional fourth argument *′false_command* is executed on the remote hosts for which the test is false.

The following example verifies that a file exists on the nodes occupying the first slot in each of four frames:

```
ppred ′-s 1-4:1′ ′-f /etc/passwd′ ′echo \′hostname\′′
```

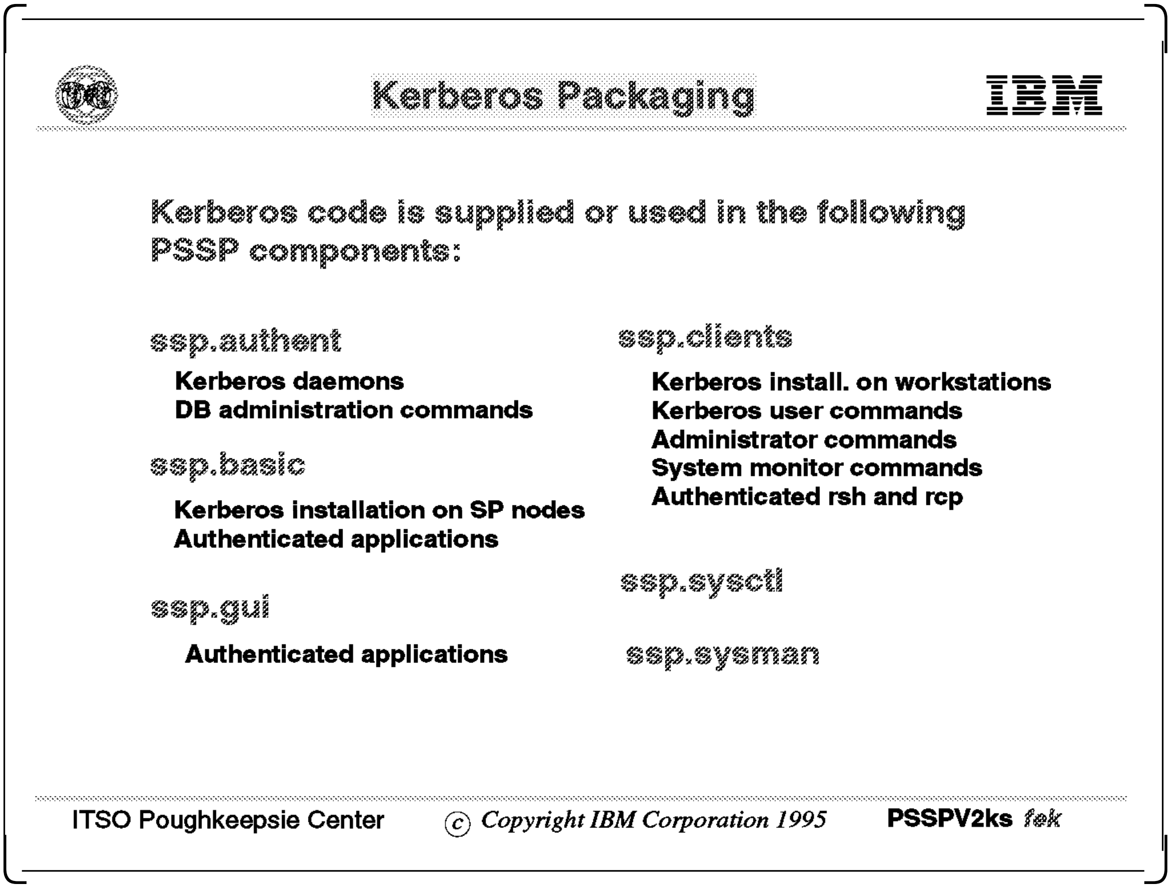
16.6.20 pps

The **pps** command issues the AIX **ps** command on multiple hosts in parallel, using **dsh**. Output is written to stdout and formatted so that duplicate output is displayed only once. This command is identical to **pexec ps**.

16.6.21 prm

The **prm** command issues the AIX **rm** command on multiple hosts in parallel, using **dsh**. This command is identical to **pexec rm**.

16.7 Kerberos Packaging



Kerberos Packaging

Kerberos code is supplied or used in the following PSSP components:

ssp.authent

Kerberos daemons
DB administration commands

ssp.basic

Kerberos installation on SP nodes
Authenticated applications

ssp.gui

Authenticated applications

ssp.clients

Kerberos install. on workstations
Kerberos user commands
Administrator commands
System monitor commands
Authenticated rsh and rcp

ssp.sysctl

ssp.sysman

ITSO Poughkeepsie Center ⓒ *Copyright IBM Corporation 1995* PSSPV2ks fek

The following components include the Kerberos code and the Kerberos-authenticated applications supplied with PSSP:

16.7.1 ssp.authent

This component contains the authentication server code and the authentication administrator commands. It is installed only on the primary authentication server (typically the control workstation) or the secondary authentication servers. It may not be installed if the system already has another MIT Kerberos or AFS authentication implementation installed. This component is not installed on the nodes.

16.7.2 ssp.basic

This component contains code for installing and monitoring the SP, including:

- System monitor
- Installation and configuration commands
- System Data Repository (SDR)
- Centralized Management Interface (CMI), that is, the SMIT panels

The first two items include Kerberos-authenticated applications.

16.7.3 ssp.clients

This component is installed on the control workstation, the SP nodes and other RS/6000 hosts where Kerberos-authenticated applications are used. It includes:

- All user authentication commands
- kshell services
- System monitor command line interfaces
- Logging daemon

16.7.4 ssp.gui

This component includes the system monitor graphical user interface. It is installed on the control workstation only.

16.7.5 ssp.sysctl

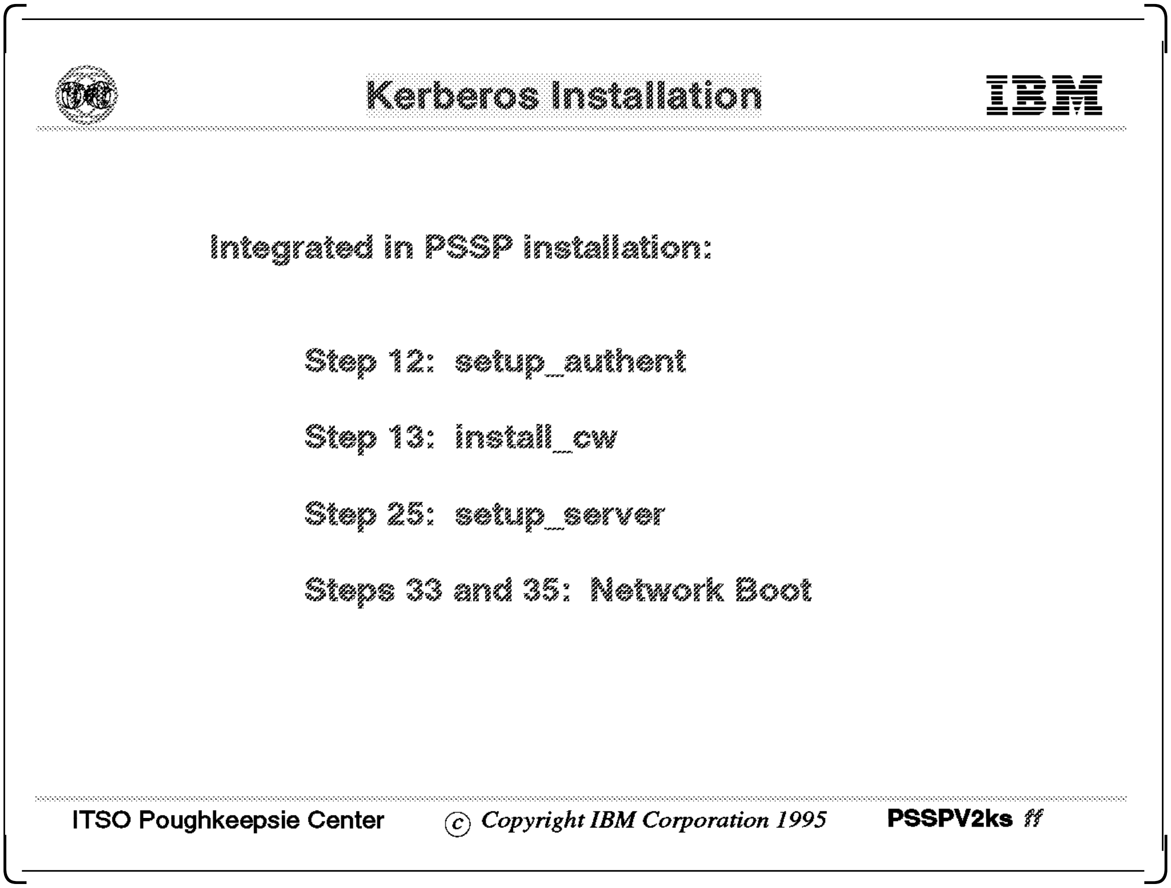
Contains the sysctl application.

16.7.6 ssp.sysman

Contains the SP system management tools, including:

- User management support
- Print support
- File collections
- Login control
- Accounting support
- Network Time Protocol (NTP)

Chapter 17. Kerberos Installation



Kerberos Installation

Integrated in PSSP installation:

Step 12: setup_authent

Step 13: install_cw

Step 25: setup_server

Steps 33 and 35: Network Boot

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ks ff

The installation of Kerberos is integrated in the PSSP installation steps below:

Step 12: setup_authent

Step 13: install_cw

Step 25: setup_server

Steps 33 and 35: network boot

The step numbers refer to the *SP Installation Guide* (GC23-3898).

The integration of Kerberos into PSSP scripts is almost transparent to the installation process. This has the significant advantage that SP installation can be carried out with no more than a token knowledge of Kerberos.

When installing authentication services on your SP system, you must choose one of the following Kerberos implementations:

1. The SP implementation, based on MIT Kerberos Version 4
2. The implementation of Kerberos included in AFS 3.3 or 3.3a
3. Another Kerberos implementation, provided it is compatible with SP Kerberos-authenticated services

We discuss here the installation of the SP implementation only.

A number of files are modified or created during the installation of SP authentication services, including:

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

For the role of these and other files, see Chapter 18, "Kerberos Files" on page 219 and 19.3, "Kerberos Port Assignments" on page 239.

Other files may also influence Kerberos installation. For example, when network boot occurs during install or customization of a node, **tftp** is used to transfer the file **/tftpboot/**<*nodename*>**-new-srvtab** from the node′s boot server and write it to the node as **/etc/krb-srvtab**. This transfer fails if the **/etc/tftpaccess.ctl** file on the node does not allow access to the appropriate directories.

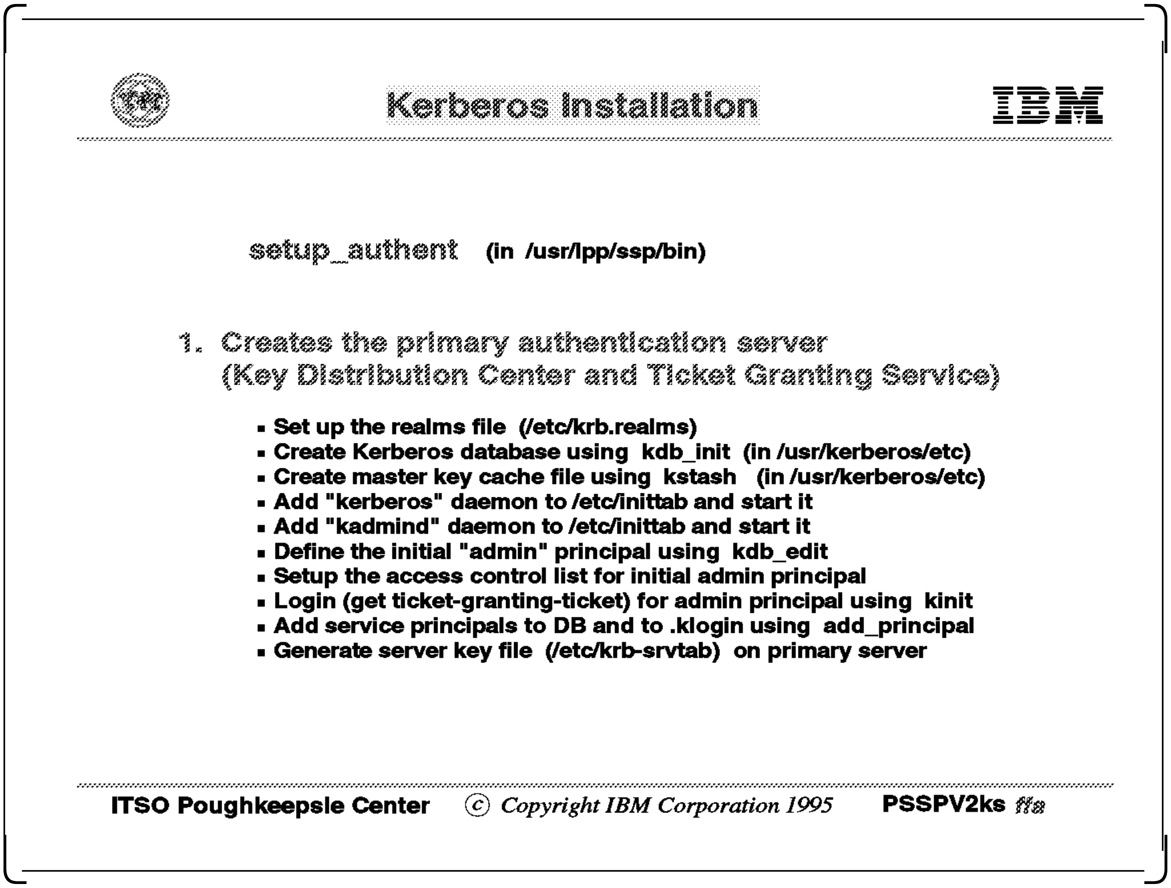
By default, the */etc/tftpaccess.ctl* file does not exist, allowing universal tftp access. However, if a customer restricts tftp access at his installation by creating an */etc/tftpaccess.ctl* file (a model is supplied in /usr/samples/tcpip/tftpaccess.ctl), and uses his AIX image to install the SP nodes, Kerberos installation may fail. In this case, you must edit /etc/tftpaccess.ctl before netboot.

Note that tftp may fail for other reasons, such as if it is not configured in /etc/inetd.conf or if TCP/IP security features are enabled or auditing is in effect, etc.

In the following pages, we discuss in sequence:

1. Setting up the primary authentication server
2. Setting up secondary authentication servers
3. Setting up authentication clients

17.1 Setting Up the Primary Authentication Server



Kerberos Installation

setup_authent (in /usr/lpp/ssp/bin)

1. Creates the primary authentication server
   (Key Distribution Center and Ticket Granting Service)
   - Set up the realms file (/etc/krb.realms)
   - Create Kerberos database using kdb_init (in /usr/kerberos/etc)
   - Create master key cache file using kstash (in /usr/kerberos/etc)
   - Add "kerberos" daemon to /etc/inittab and start it
   - Add "kadmind" daemon to /etc/inittab and start it
   - Define the initial "admin" principal using kdb_edit
   - Setup the access control list for initial admin principal
   - Login (get ticket-granting-ticket) for admin principal using kinit
   - Add service principals to DB and to .klogin using add_principal
   - Generate server key file (/etc/krb-srvtab) on primary server

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ks ffa

Within an authentication realm, there must be at least one authentication server, but you may choose to have more than one. When you configure your realm (through *ssp.authent*), you designate one authentication server as the *primary* server. All others are *secondary* servers.

Only the primary server has the **kadmind** daemon that manages the (primary) authentication database. The databases that reside in the secondary authentication servers are copies and are updated periodically from the primary server (see 17.2, “Setting Up Secondary Authentication Servers” on page 215). The addition of principals and changing of passwords takes place in the primary database and is propagated to secondary databases through these periodic updates only.

Multiple authentication servers can provide greater reliability, security or performance:

1. If the primary server is unavailable or if there are network problems, authentication requests can be handled as long as any one of the configured servers is accessible. Kerberos tries all the servers listed in the configuration file **/ect/krb.conf** (in the order listed there) before failing an authenticated service request.

   Providing secondary servers is particularly relevant if your SP system has to provide high availability with no single point of failure.

2. You may want to have your authentication server(s) entirely off the SP system, for example, to provide greater physical security for the servers or because you may want to allow logins to the control workstation that are not appropriate for the authentication servers.

3. However, only when there is an authentication server running on the control workstation can the high performance switch be used for authentication protocol traffic to and from the SP nodes.

   If you are integrating the SP system into a realm where the primary authentication server is already on another workstation, you can make the nodes′ protocol traffic flow over the switch by setting up a *secondary authentication server* on the control workstation.

To set up the primary authentication server, on the control workstation or another RS/6000 host, perform the following steps:

1. Install **ssp.authent** and **ssp.clients**.

2. Create a **/etc/krb.conf** file if necessary, for example, if your hostnames do not follow Kerberos convention.

3. Run the **ssp_authent** script and answer the prompts.

When you install the **ssp.authent** fileset before running the script, it assumes that the host will be an SP authentication server. If you provide your own **/etc/krb.conf** file, the script looks for an entry for the local hostname. If it cannot find one, it allows you to configure the host without any server, that is, as an authentication client.

setup_authent then proceeds to:

1. Create or update the /etc/krb.realms file as needed

2. Add the **kerberos** daemon to /etc/inittab and start it

3. Add the **kadmind** daemon to /etc/inittab and start it

4. Create the authentication database using /usr/kerberos/etc/kdb_init

5. Create the master key cache file /.k using /usr/kerberos/etc/kdb_init

6. Define the initial authentication administrator principal using kdb_edit

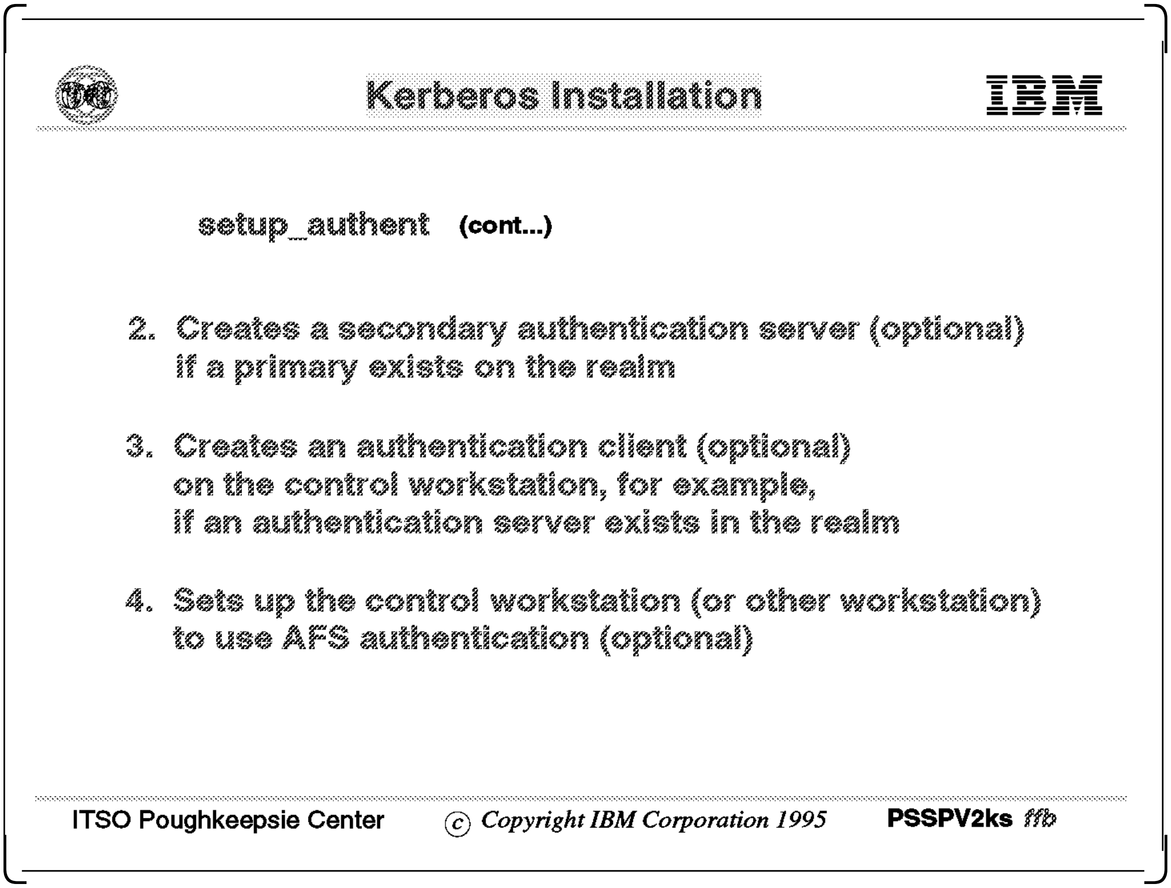
7. Setup the Kerberos access control lists for the initial administrator (see 18.2.7, “Kerberos Access Control Lists” on page 225)

8. Create the root user′s .klogin file to authorize the administrator principal to use remote commands

9. Define the service principals for local service instances (in the authentication database and .klogin)

10. Create the server key file containing the local service instances

17.2 Setting Up Secondary Authentication Servers



Kerberos Installation

setup_authent **(cont...)**

2. Creates a secondary authentication server (optional) if a primary exists on the realm

3. Creates an authentication client (optional) on the control workstation, for example, if an authentication server exists in the realm

4. Sets up the control workstation (or other workstation) to use AFS authentication (optional)

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* **PSSPV2ks** *ffb*

To set up a secondary authentication server, on the control workstation or another RS/6000 host, proceed as follows:

1. Install **ssp.authent** and **ssp.clients**.
2. Copy the **/etc/krb.conf** file from the primary authentication server.
3. Add a line to /etc/krb.conf listing this host as a secondary server for the local realm.
4. Copy the **/etc/krb.realms** file from the primary authentication server.
5. Run the **setup_authent** script, answer the prompts.
6. Add an entry for the new secondary server to the **/etc/krb.conf** file on other servers, if any.
7. On the primary server, if this is the first secondary server, create a root **crontab** entry that invokes the script **/usr/kerberos/etc/push-kprop** to periodically propagate database changes.

Secondary databases are maintained by the **kpropd** daemon that runs only on secondary servers. It receives the database content encrypted with the master password from the **kprop** command that you run on the primary server.

PSSP supplies a script, **/usr/kerberos/etc/push-kprop**, that you can schedule for execution daily or at some other interval from the root user′s **cron** file

(/var/spool/cron/crontabs/root) to keep the secondary authentication database(s) up-to-date. The push-kprop script:

1. Invokes the command

   kdb_util slave_dump slavesave

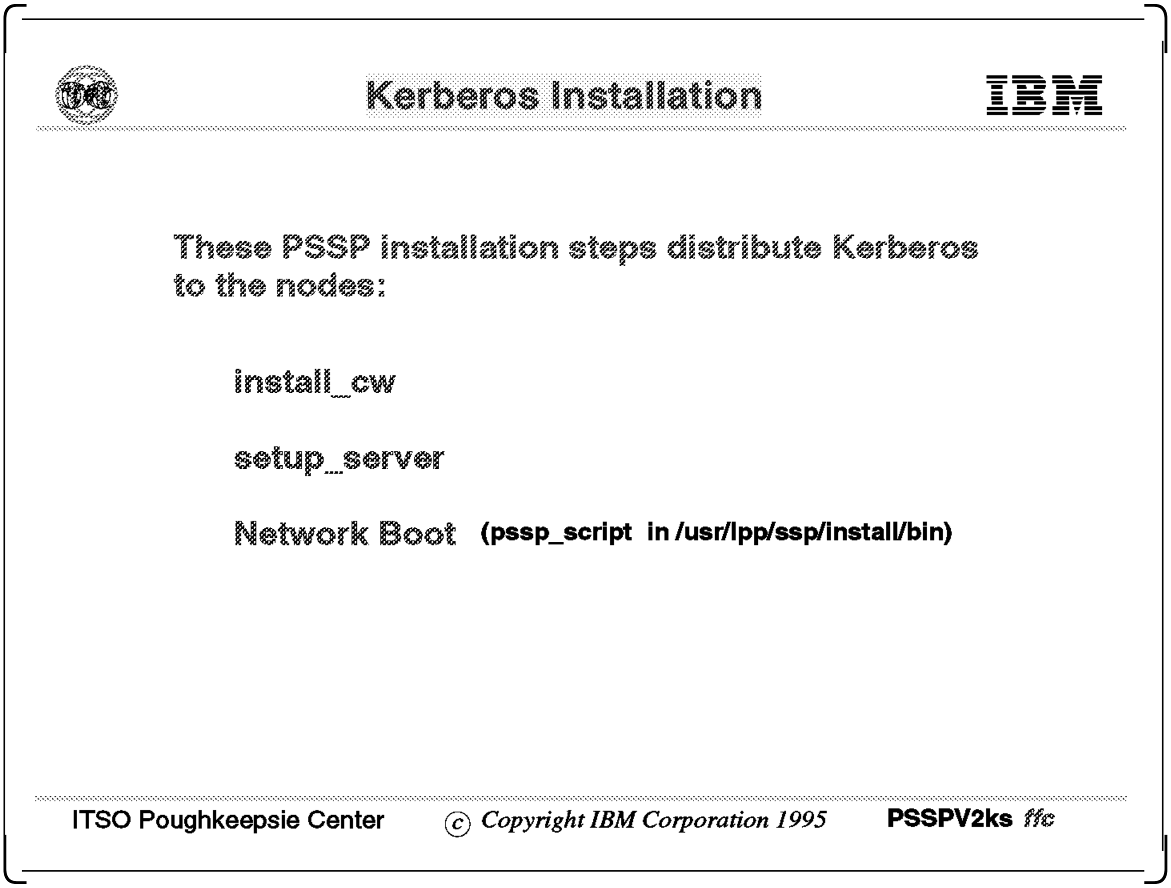
   to copy the primary database to an ASCII text representation /var/kerberos/database/slavesave (in the same directory as the database). This command also creates a semaphore file slavesave.ok and sends a mail message to root.

2. Executes the command

   kprop slavesave *hosts_list*

   where *hosts_list* is a file containing the list of secondary servers, derived from the /etc/krb.conf file. The **kprop** command connects to each host named in *hosts_file* in turn, using the network service provided by the **kpropd** daemon on that host. The service name used for mutual authentication is **rcmd**. The ASCII file **slavesave** is transferred if it has been modified since it was last sent.

17.3 Setting Up Authentication Clients



Kerberos Installation

These PSSP installation steps distribute Kerberos to the nodes:

install_cw

setup_server

Network Boot (pssp_script in /usr/lpp/ssp/install/bin)

ITSO Poughkeepsie Center ⓒ *Copyright IBM Corporation 1995* PSSPV2ks ffc

To set up an RS/6000 host or the control workstation as an authentication client, proceed as follows:

1. Install **ssp.clients**.
2. Copy the **/etc/krb.conf** and the **/etc/krb.realms** files from the primary authentication server to this host.
3. On the primary authentication server, add a line for the new host to **/etc/krb.realms** file, if the new host is outside primary server′s realm.
4. Copy the **/etc/krb.realms** file from the primary authentication server to this host.
5. Run **setup_authent**, answer the prompts.
6. Copy **/etc/krb.realms** file to other hosts if modified by **setup_authent**.

17.4 Kerberos Installation Following setup_authent

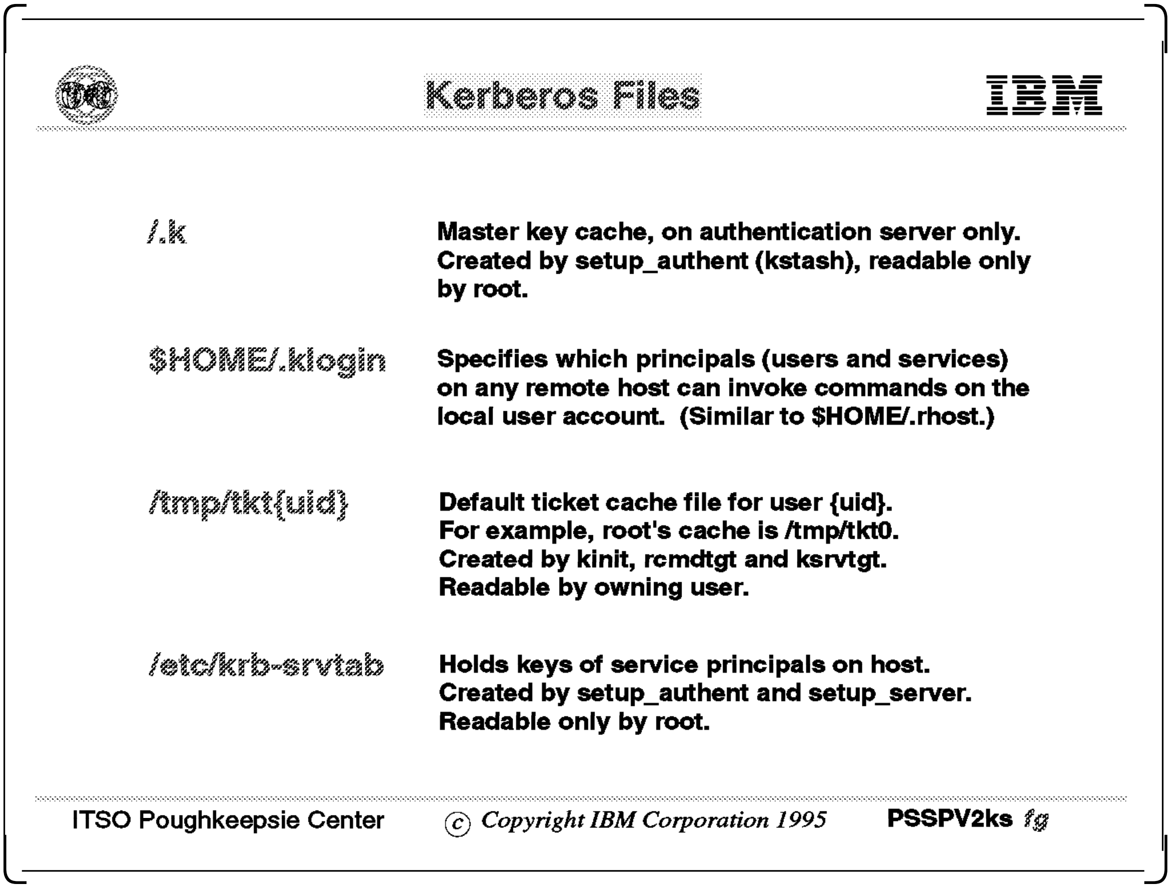
The following PSSP installation steps complete the installation of Kerberos on the control workstation and the nodes:

1. **install_cw** creates the hardware monitor access control list **/spdata/sys1/spmon/hmacls** and includes the initial authentication administrator.

2. **setup_server** configures the control workstation as a NIM server and includes the Kerberos files intented for the nodes in the **/tftpboot** directory.

3. **network boot** installs Kerberos on the nodes.

Chapter 18. Kerberos Files



Kerberos Files

/.k — Master key cache, on authentication server only. Created by setup_authent (kstash), readable only by root.

$HOME/.klogin — Specifies which principals (users and services) on any remote host can invoke commands on the local user account. (Similar to $HOME/.rhost.)

/tmp/tkt{uid} — Default ticket cache file for user {uid}. For example, root's cache is /tmp/tkt0. Created by kinit, rcmdtgt and ksrvtgt. Readable by owning user.

/etc/krb-srvtab — Holds keys of service principals on host. Created by setup_authent and setup_server. Readable only by root.

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* PSSPV2ks *fg*

18.1 Pathnames for SP Authentication Services

Before describing the individual files used by Kerberos, we summarize below the directories where the various Kerberos commands and files can be found. You may add some of these to your PATH environment variable, as appropriate.

Directory	Contents
/usr/kerberos/bin	Symbolic link to **/usr/lpp/ssp/kerberos/bin**.
/usr/kerberos/etc	Symbolic link to **/usr/lpp/ssp/kerberos/etc**.
/usr/lpp/ssp/bin	**setup_authent**.
/usr/lpp/ssp/rcmd/bin	Kerberos authenticated versions of **rsh** and **rcp**.
/usr/lpp/ssp/kerberos/bin	Commands for Kerberos users, such as **kinit**, **klist** and others.
/usr/lpp/ssp/kerberos/etc	Commands and daemons for authentication administrators, such as **kadmin**, **kdb_util**, and so on.

/etc	The files **krb-srvtab**, **krb.conf** and **krb.realms**.
/var/kerberos/database	Kerberos database on the authentication server.
/var/adm/SPlogs/kerberos	Error logs for the authentication servers.
/usr/afsws/etc	AFS executables.
/usr/vice/etc	AFS cell information and utilities.

Users of authentication commands may find it convenient to have in their PATH environment variable:

/usr/kerberos/bin:/usr/lpp/ssp/rcmd/bin:/usr/lpp/ssp/bin:/etc

Authentication database administrators may have in their PATH:

/usr/kerberos/etc:/usr/kerberos/bin:/usr/lpp/ssp/rcmd/bin:/usr/lpp/ssp/bin:/etc

18.2 Kerberos Files

We summarize below the files specific to Kerberos.

18.2.1 Master Cache File

The *master key cache file* **/.k** contains the DES key derived from the master password. The master password is supplied initially by the administrator when the primary authentication server is created, using the **kdb_init** command. The corresponding DES key is saved in */.k* using the **kstash** command. The Kerberos daemon **kadmind** and the database utility commands read the master key from this file instead of prompting for the master password.

18.2.2 .klogin File

The **$HOME/.klogin** file specifies remote principals *authorized* to invoke commands on the local user account.

If the originating remote user is authenticated to one of the principals named in the *.klogin* file, access is granted to the account. This is an instance where Kerberos provides an *authorization* function (as opposed to just authentication).

Remote principals are users or services on any computer on the network within an authentication realm. The *local account* is the user on whose home directory the *.klogin* file resides. Remote principals must be defined to Kerberos, either on the same realm as the local user or on a different realm.

The *.klogin* file has similar function to the TCP/IP **$HOME/.rhosts** file, with added security. It must be owned by the local user or root and must have permission 600. If the *.klogin* file does not exist in a account′s home directory, the owner of the account is the only one who may access it from a remote host. If a *.klogin* file is present, the owner must also be listed in it in order to access his or her account from a remote host.

The *.klogin* file contains a list of principals in the form:

```
name.instance@realm
```

A typical *.klogin* file may look like the following:

```
root.admin@ITSC.POK.IBM.COM
henry@ITSC.POK.IBM.COM
franz.root@ITSC.POK.IBM.COM
rcmd.spn01@ITSC.POK.IBM.COM
```

For service principals, the instance indicates the workstation the service is running on. In the SP, service entries are found only on the root directory */.klogin* file. For user principals, the instance may indicate special privileges such as root access.

18.2.3 Ticket Cache File

The *ticket cache file* **/tmp/tkt**<*uid*> contains the tickets owned by a client.

A user may choose to have more than one ticket cache file by setting the KRBTKFILE environment variable to the full path name of the **current** cache file, If KRBTKFILE is not set, then the tickets are cached in the default file **/tmp/tkt**<*uid*>, where <*uid*> is the user′s AIX ID number.

A client′s tickets are stored in the *current* cache file as they are created. The first ticket in the file is the ticket granting ticket. It is obtained by the user using one of the commands **kinit**, **ksrvtgt** or **rcmdtgt**. These commands delete the *current* cache file (named as indicated above), if any, and create a new one. Hence, when the TGT is stored, all previously issued tickets (in the *current* cache file) are deleted.

In addition to the TGT, the cache file contains any number of service tickets for application services such **rcmd** or **hardmon**. These tickets are obtained by the client commands such as **rsh** and **rcp**, and indirectly by SP administration tools that invoke them.

A user′s tickets are shared by all processes running under the user′s *uid*. If a user has multiple logins, runs background processes or shares a *uid*, the tickets in use may be destroyed by a subsequent *kinit* or similar command. This can be avoided by storing the tickets in *multiple cache files*. In this case, the KRBTKFILE environment variable must be set to the path name of a unique ticket cache file for each login session or background process, before using authentication services. For example:

```
export KRBTKFILE=/tmp/tickets.$$
```

The **klist** command displays the contents of the current cache file and the **kdestroy** command deletes the current cache file.

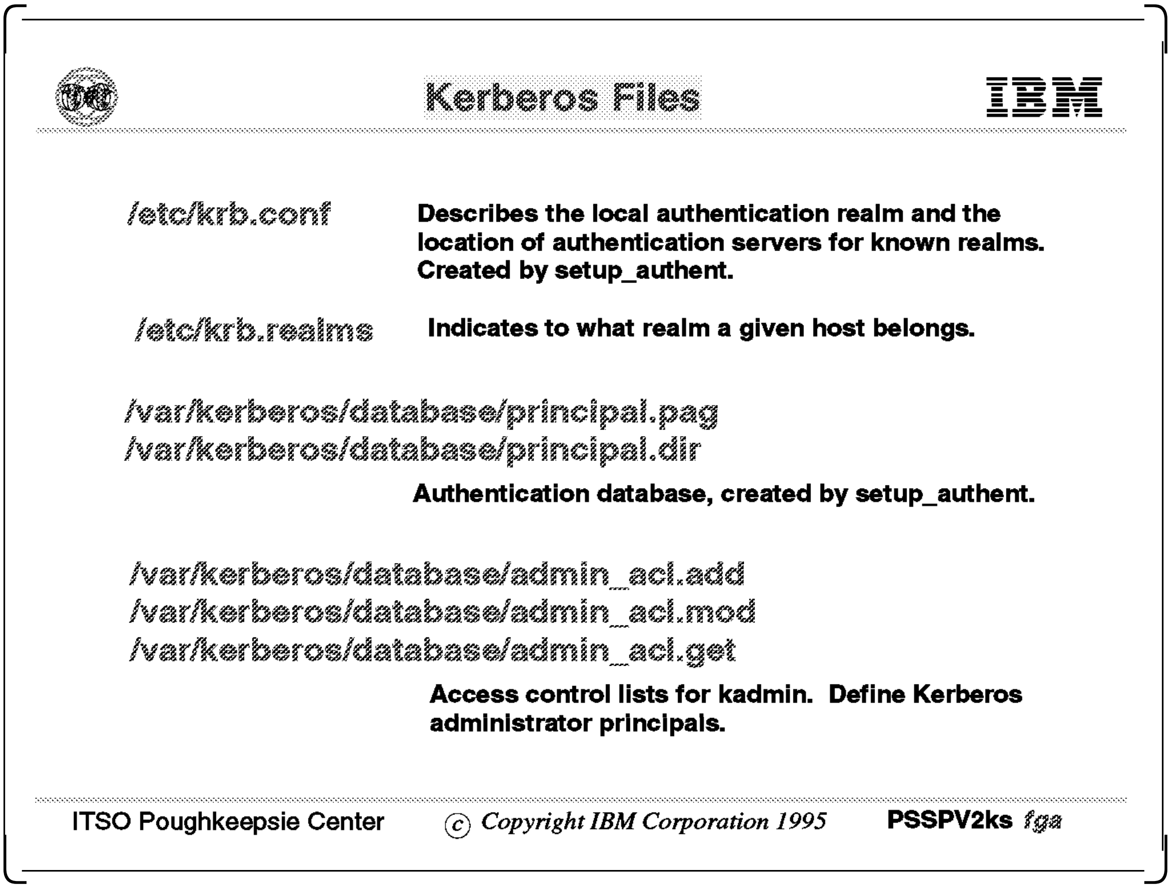
18.2.4 Server Key File

The *server key file* **/etc/krb-srvtab** contains the names and private keys of the local instances of protected services. During setup of the control workstation or the nodes, the keys for service principals are stored in the authentication database (for use by the authentication server) and in the file */etc/krb-srvtab* (for use by the services themselves. Therefore, every SP node and the control workstation includes an */etc/krb-srvtab* file that contains the keys for the services provided on that host. On the control workstation, the **hardmon** and **rcmd** service principals are in the file; on the nodes, the **rcmd** service principals are in the file, with their appropriate instances, as explained below.

The **setup_authent** script creates service principals for each network interface defined on the control workstation. The **setup_server** script creates service principals for each network interface defined on each node. In addition, **setup_server** automatically recognizes when interfaces are added to the control workstation or nodes (the network interfaces are recorded in the SDR). It follows that whenever you add a network interface to the control workstation, you should run setup_server; whenever you add an interface to a node, you should run setup_server on the control workstation and customize the node.

As a result, in SP nodes, the server key table contains an instance of the *rcmd* principal for each network interface defined on that node. It follows that the *kshd* and *sysctld* servers accept requests using any interface name. This is referred to as *multi-homed host support*.

Note that the user invoking *setup_server* must have Kerberos *database administrator* credentials (see 18.2.7, “Kerberos Access Control Lists” on page 225).



Kerberos Files

/etc/krb.conf — Describes the local authentication realm and the location of authentication servers for known realms. Created by setup_authent.

/etc/krb.realms — Indicates to what realm a given host belongs.

/var/kerberos/database/principal.pag
/var/kerberos/database/principal.dir

Authentication database, created by setup_authent.

/var/kerberos/database/admin_acl.add
/var/kerberos/database/admin_acl.mod
/var/kerberos/database/admin_acl.get

Access control lists for kadmin. Define Kerberos administrator principals.

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* PSSPV2ks *fga*

18.2.5 Configuration File

The **/etc/krb.conf** file defines the local authentication realm and the location of authentication servers for known realms.

A simple */etc/krb.conf* file may look like:

```
PRODUCTS.GROUP
PRODUCTS.GROUP  augustus.pok.ibm.com  admin server
PRODUCTS.GROUP  verdi.pok.ibm.com
```

The first line of the */etc/krb.conf* file contains the name of the local authentication realm. Each additional line specifies an authentication server for a realm.

(The PRODUCTS.GROUP realm is defined as the set of service and user principals registered to Kerberos with PRODUCTS.GROUP as their realm name.)

In the second line of */etc/krb.conf*, the suffix admin server identifies the host *augustus* as the primary authentication server that administers the master Kerberos database. Host *verdi* is a secondary authentication server. There may be multiple secondary authentication servers. The order of entries in the */etc/krb.conf* file determines the order in which the servers are contacted for tickets. The file must contain at least one entry for each realm used by the local system.

During SP installation, */etc/krb.conf* is created by **setup_authent** on the host (usually the control workstation) that you set up as primary authentication server. It is copied from the control workstation to the nodes at network boot. You *may* supply your own */etc/krb.conf* file on the primary authentication server before running *setup_authent*, for example if you wish to set a non-default realm name (the default realm name is the domain portion of the primary authentication server′s host name converted to upper case). You *must* supply a */etc/krb.conf* file when you define the primary authentication server if its host name does not have a domain portion.

When you define an SP secondary authentication server, you *must* supply a */etc/krb.conf* file (and the */etc/krb.realms* file) before running *setup_authent* on the secondary server. In this case, you add an entry for the secondary server to the primary server′s */etc/krb.conf* file on the primary server, then copy it to the secondary server.

You must also supply a */etc/krb.conf* file when you define an SP client authentication system. In this case, you copy the file from any authentication server before running *setup_authent* on the client.

18.2.6 Realms File

The **/etc/krb.realms** file maps a host name to an authentication realm for the services provided by that host.

The lines of the file are in one of the following forms:

host_name REALM_NAME

or

.domain_name REALM_NAME

For example:

augustus.pok.ibm.com PRODUCTS.GROUP
.pok.ibm.com PRODUCTS.GROUP

The first line maps a specific host to the realm PRODUCTS.GROUP. The second entry maps all host names whose domain portion is pok.ibm.com to the same realm.

If no entry matches the host, the host′s realm is considered to be the host name′s domain portion converted to upper case. If the host name has no domain portion, the host′s realm is considered to be the host name converted to upper case.

The following default mapping is always assumed, even if the */etc/krb.realms* file is empty.

.products.group PRODUCTS.GROUP

Entries for all network interface names (all hostnames) that require mapping on the control workstation and on the SP nodes are automatically added either initially by **setup_authent** or subsequently or by **setup_server** when it finds new interfaces. (The interface names are recorded in the SDR.) The */etc/krb.realms* file is kept identical on the control workstation and the nodes by the node customization process. You must manually update the file on other RS/6000 hosts.

18.2.7 Kerberos Access Control Lists

admin_acl.add
admin_acl.get
admin_acl.mod

These access control lists authorize users to administer the Kerberos authentication database. A Kerberos database administrator is a principal with an **admin** instance whose name appears in one or more access control lists, as follows:

/var/kerberos/database/admin_acl.add: Principals authorized to *add* entries to the authentication database.

/var/kerberos/database/admin_acl.get: Principals authorized to *retrieve* entries from the authentication database.

/var/kerberos/database/admin_acl.mod: Principals authorized to *modify* entries in the authentication database.

For example, to authorize user ID *henry* to perform authentication administration functions:

1. Define the principal **henry.admin** using one of the commands **kadmin**, **kdb_edit** or **add_principal**.
2. Include this principal′s name in one or more of the acl files indicated above.

18.2.8 Hardware Monitor Access Control List

The hardware monitor access control list **/spdata/sys1/spmon/hmacls** is used by the hardware monitor daemon **hardmon** and its associated commands, in addition to Kerberos authentication.

A user is authorized to use the hardware monitor commands by having his or her name in the */spdata/sys1/spmon/hmacls* file *and* having been authenticated (for example, by issuing *kinit*) with that name. The affected commands are:

hardmon,
hmadm,
hmcmds,
hmmon,
nodecond,
spmon,
spmon_ctest and
s1term.

The **hmacls** file consists of lines of the form:

object principal permissions

The follwing is an example of an *hmacls* file:

```
sp2cw0.itsc.pok.ibm.com  root.admin  a
1  root.admin  vsm
2  root.admin  vsm
3  root.admin  vsm
4  root.admin  vsm
1  henry  m
2  henry  m
3  henry  m
4  henry  m
```

The first field is either the hostname of the control workstation (known as the Monitor and Control Node, or MACN, by the hardware monitor), or a frame number.

The second field is a user principal in the form *name* or *name.instance*.

The *permissions* field specify the operations that the user is allowed to execute on the object indicated by the first field.

If the *object* is the MACN hostname, the *permissions* must be the character **a** which specifies the authorization to use the **hmadm** command. If the *object* is a frame number, *permissions* is one or more of the following characters:

v Specifies *Virtual Front Operator Panel (VFOP)* permission. It is required, for example, by the **hmcmds** command and for certain operations of the **spmon -g** command.

s Specifies *S1* permission. It is required, for example, by the **s1term** command.

m Specifies *Monitor* permission. It is required, for example, by the **hmmon** and **spmon** commands.

VFOP permission implies monitor permission.

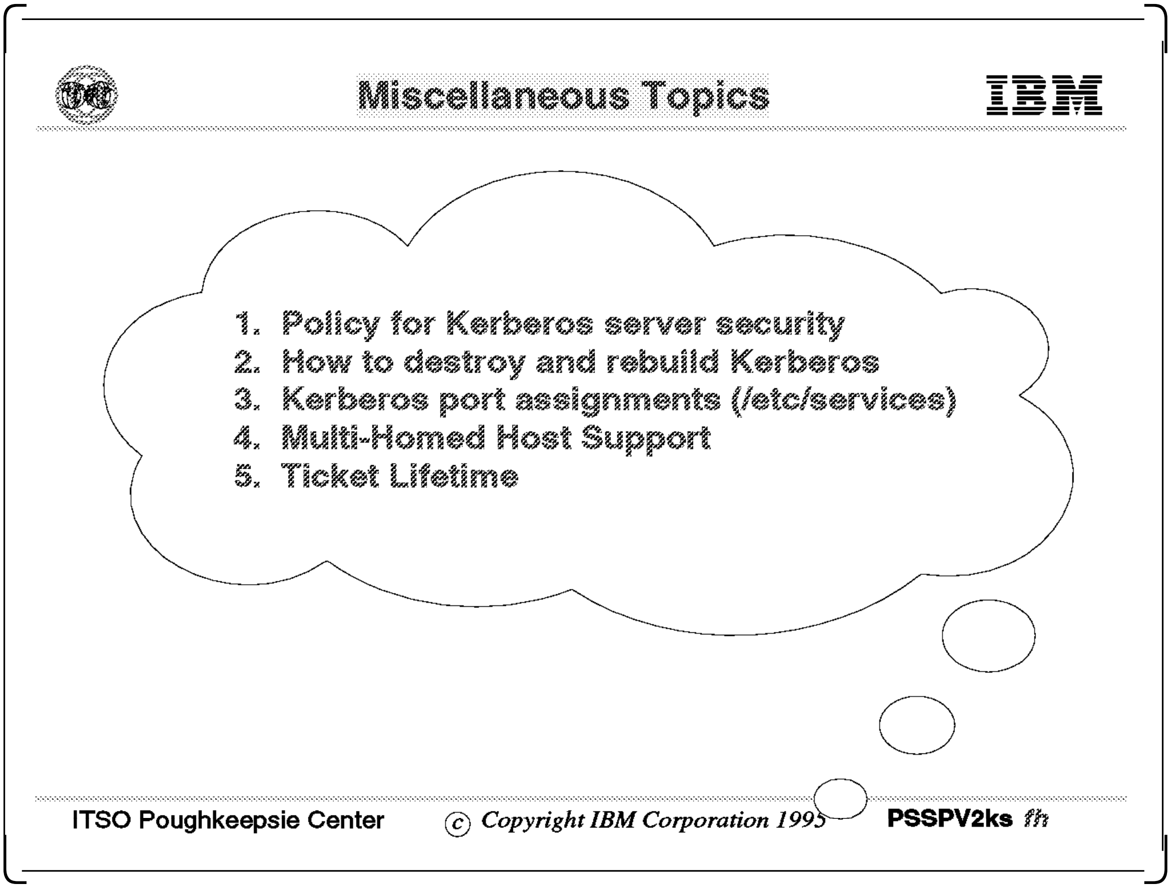
18.2.9 Process Operations Access Control List

The access control list **/etc/sysctl.pfps.acl** authorizes listed principals to perform the **pfps** command′s **-kill** and **-nice** options on processes they do not own.

18.2.10 Log Management Access Control List

The access control list **/etc/logmgt.acl** authorizes listed principals to perform log management commands such as **penotify** and **psyslclr**.

Chapter 19. Miscellaneous Topics



Miscellaneous Topics

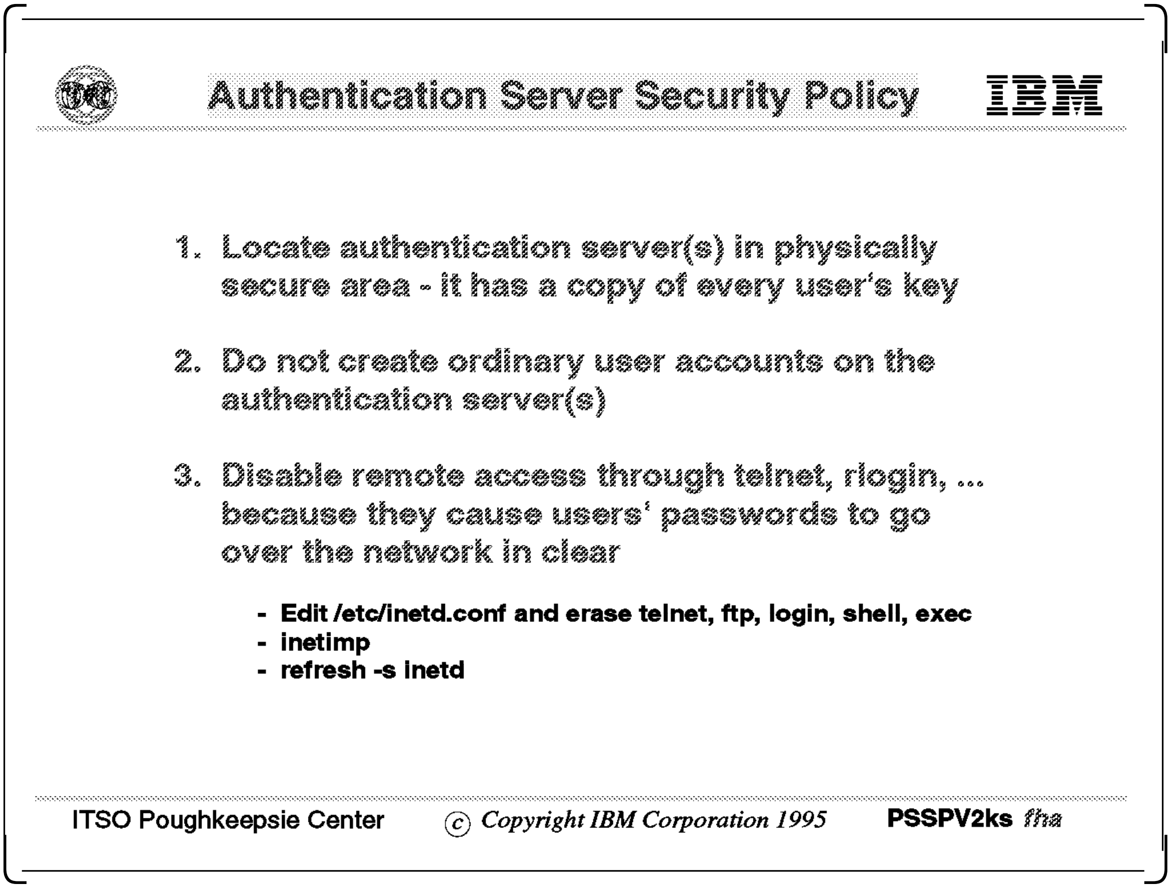
1. Policy for Kerberos server security
2. How to destroy and rebuild Kerberos
3. Kerberos port assignments (/etc/services)
4. Multi-Homed Host Support
5. Ticket Lifetime

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ks *fh*

In this section, the following miscellaneous topics are covered:

1. Policy for authentication server security
2. How to destroy and rebuild Kerberos
3. Kerberos port assignments in /etc/services
4. Multi-homed host support
5. Ticket lifetime

19.1 Authentication Server Security Policy



Authentication Server Security Policy

1. Locate authentication server(s) in physically secure area - it has a copy of every user's key
2. Do not create ordinary user accounts on the authentication server(s)
3. Disable remote access through telnet, rlogin, ... because they cause users' passwords to go over the network in clear
   - Edit /etc/inetd.conf and erase telnet, ftp, login, shell, exec
   - inetimp
   - refresh -s inetd

ITSO Poughkeepsie Center ⓒ Copyright IBM Corporation 1995 PSSPV2ks fha

As mentioned in 12.3, “Elements of Client/Server Security” on page 156, using Kerberos and the associated encryption does not ensure that a distributed computing environment is secure. Other measures are necessary to prevent attackers from making an end run around the security of information in a client/server environment. These measures are clarified in the following foils and summarized in 19.1, “Authentication Server Security Policy.”

First, the servers must be secure in their own right. Secure authentication, such as provided by Kerberos, provides a means for workstation or servers to confirm individual identities. After that, however, a server must still make the proper decision about whom it will *authorize* to access a particular resource.

Even more important, since the Kerberos authentication system relies on an *authentication server*, that server itself must be physically safe, that is, located in a physically secure area with entry limited to authorized personnel. It has a copy of every user′s key, and so an intruder who gains access to the authentication server and reads this information in essence gets everyone′s password.

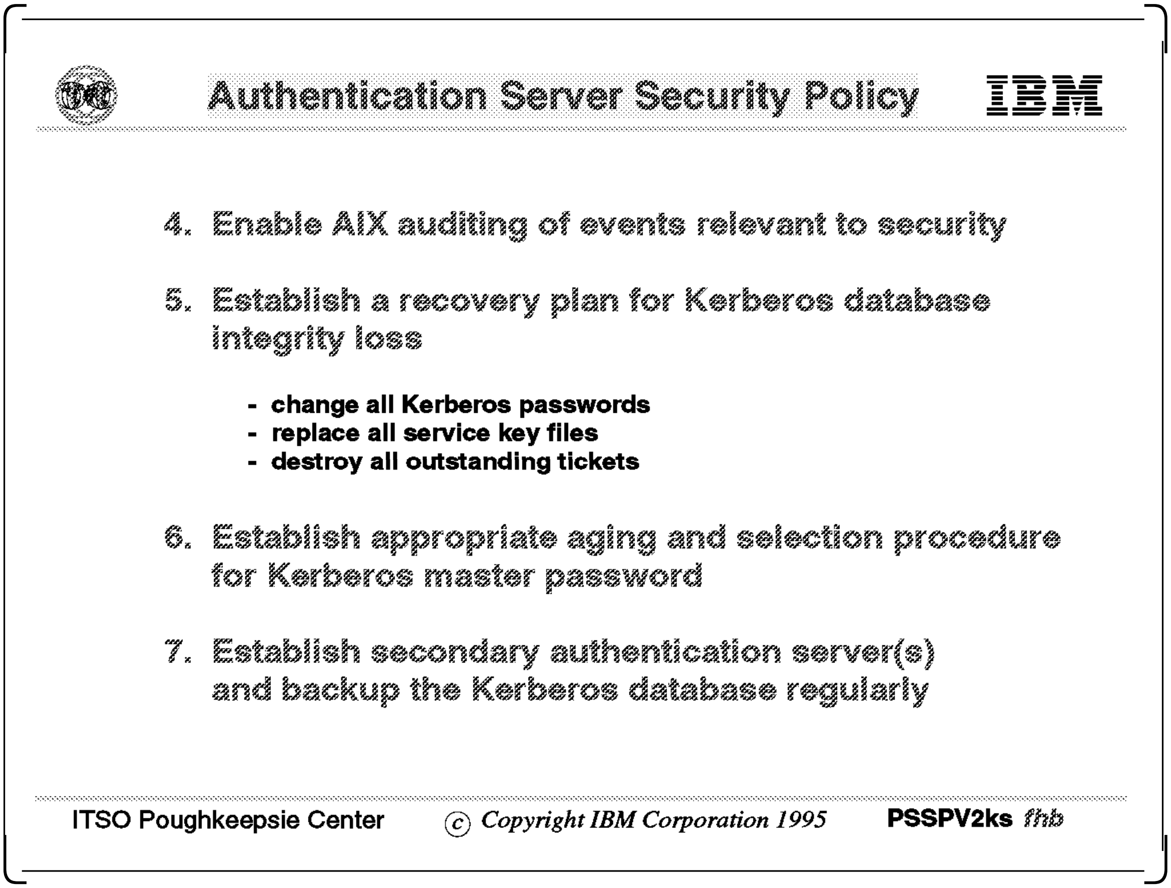
The authentication server must be secure, but it must also be always running so that users can use the protected services. The traditional approach to improving availability in a distributed computing environment is to replicate critical services on multiple machines as a hedge against power failure or other problems. The authentication server may be replicated, yet each additional authentication server must also be physically protected. An increase in the number of authentication servers improves availability, but it also amplifies the risk that the system will be compromised.

Kerberos does not protect systems from breaches such as the guessing of poorly chosen passwords, or abuse of privileges by trusted individuals. The first can be alleviated by installing a bad-password filter that does not permit people to choose an easily guessed password such as their name or a common word.

For added security, you may also consider the following actions.

- Disable remote access through **telnet**, **rlogin**, **ftp** and similar commands by commenting out the corresponding entries in */etc/inetd.conf*.
- Disable services such as **finger**, **systat**, **netstat** and similar services by commenting out the corresponding entries in */etc/inetd.conf*.
- Do not create ordinary user accounts on authentication servers.
- Enable AIX auditing of events relevant to security.
- Establish an appropriate password aging and selection procedure for the Kerberos master password (see 19.1.2, “Changing the Kerberos Master Password” on page 231). All keys in the Kerberos database are encrypted with this password.
- Establish a recovery plan for loss of Kerberos database integrity. Such a plan may include changing all Kerberos passwords, replacement of all server key files and destruction of all outstanding tickets.
- Back up the authentication database regularly (see 19.1.1, “Backing Up the Authentication Database” on page 230).

19.1.1 Backing Up the Authentication Database



Authentication Server Security Policy

4. Enable AIX auditing of events relevant to security

5. Establish a recovery plan for Kerberos database integrity loss

   - **change all Kerberos passwords**
   - **replace all service key files**
   - **destroy all outstanding tickets**

6. Establish appropriate aging and selection procedure for Kerberos master password

7. Establish secondary authentication server(s) and backup the Kerberos database regularly

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* **PSSPV2ks** *fhb*

The authentication database is contained in two files:

/var/kerberos/database/principal.pag
/var/kerberos/database/principal.dir

The database may be dumped (copied into a ASCII text representation) by the command:

kdb_util dump <*filename*>

where <*filename*> is the full path name of a file (it is recommended to use the same directory as the database which is easy to remember and accessible only to root). This file may be viewed or printed to examine the database contents. Note that the DES keys in the database (as well as in <*filename*>) are encrypted with the master password.

The database may be recovered with the command:

kdb_util load <*filename*>

where <*filename*> is the name of a file created with the *dump* option. Any existing database is overwritten. The *kdb_util*command requires root authority.

To keep a current backup of the authentication database, just add an entry to the root user′s **cron** file to invoke the following command with the desired frequency:

kdb_util dump /var/kerberos/database/slavesave

If you have one or more secondary authentication servers, a backup file is created as a by-product of the process of propagating the primary database to the secondary servers. In this case, the **push-kprop** script is invoked daily from the root user′s cron file to create a backup file named */var/kerberos/database/slavesave*.

19.1.2 Changing the Kerberos Master Password

The master password may be changed by a database administrator (see 18.2.7, “Kerberos Access Control Lists” on page 225) who also has root authorization, such as *root.admin*, on the primary authentication server.

The sequence below may be executed:

1. Login to Kerberos as authentication administrator.

   kinit root.admin

2. Change the master password.

   ```
   kdb_util new_master_key /var/kerberos/database/newdb.$$
   kdb_util load           /var/kerberos/database/newdb.$$
   rm                      /var/kerberos/database/newdb.$$
   ```

   The **new_master_key** option prompts for the old and new master passwords and then dumps the database into the file **newdb.$$** (notice that **kdb_util** prompts only *once* for the new password). The principal keys in *newdb.$$* are encrypted with the new master key.

   The **load** option re-initializes the database with the records from *newdb.$$* The previous database is overwritten. The net effect is to re-encrypt the principals′ keys with the new master password. However, the principals′ keys (passwords) themselves are not changed.

3. Replace the master key cache file **/.k**.

   kstash

4. Kill and respawn the primary authentication server daemons.

   kill ′ps -e | egrep ′kerberos|kadmind′ | cut -d′-′ -f1′

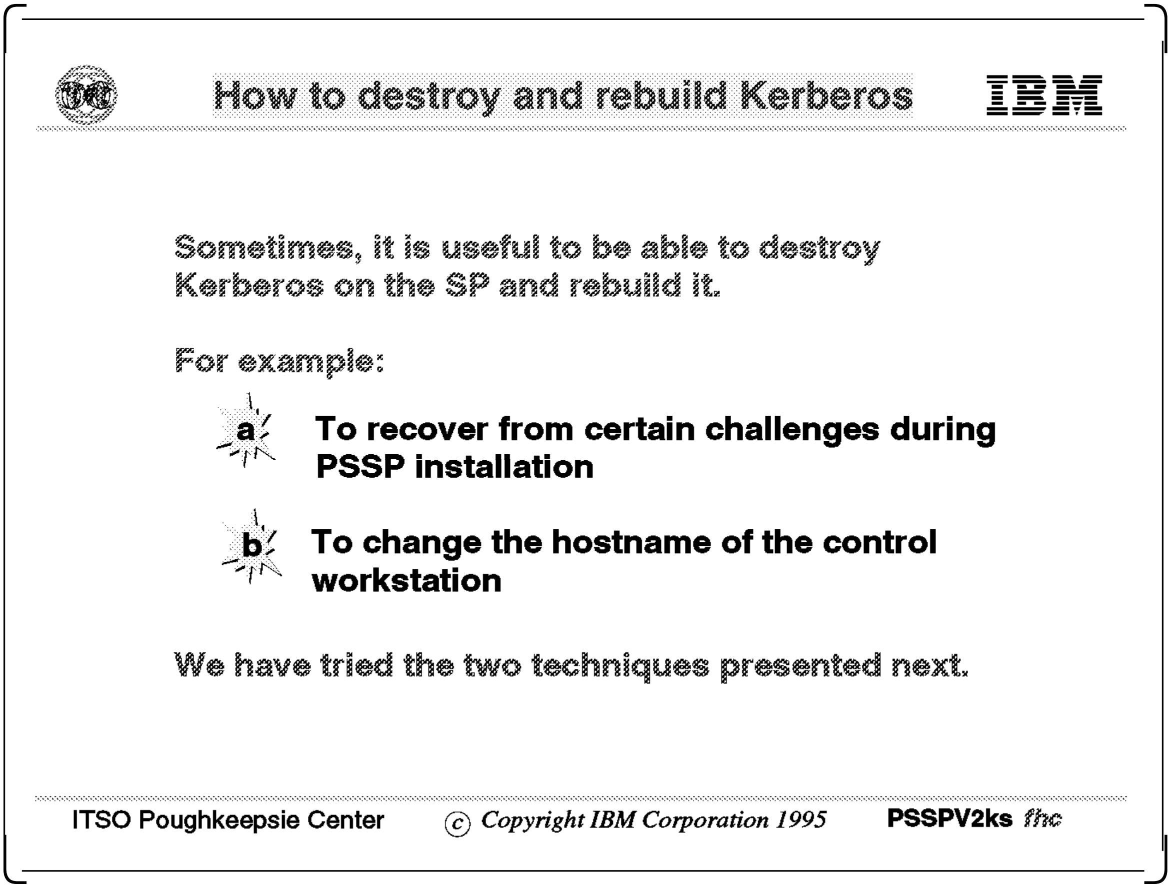
5. If there are secondary authentication servers:

   Propagate the new database
   Copy the master key cache to the secondary servers
   Kill and respawn the secondary server daemons

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

   The file **slavelist** contains the names of the secondary authentication servers.

19.2 How to Destroy and Rebuild Kerberos



How to destroy and rebuild Kerberos

Sometimes, it is useful to be able to destroy Kerberos on the SP and rebuild it.

For example:

- a To recover from certain challenges during PSSP installation
- b To change the hostname of the control workstation

We have tried the two techniques presented next.

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ks fhc

Sometimes, it is useful to be able to destroy Kerberos on the SP and rebuild it. This may happen, for example, if you entered an improper principal name or somehow managed to confuse *setup_authent* during installation.

It may become necessary to rebuild Kerberos if the latter becomes corrupted by accident during operation. Examples of unplanned actions that can corrupt the Kerberos environment include:

- Changing the hostname or IP address of the control workstation
- Changing the hostnames or IP addresses of the nodes
- Changing the DNS or /etc/host in such a way that hostnames or IP addresses cannot be resolved
- Changing the service keys in /etc/krb-srvtab on the nodes (using the **ksrvutil** command) when the primary authentication database is not on the control workstation
- Changing the Kerberos realm on the control workstation (in /etc/krb.conf or /etc/krb.realms) without propagating it to the nodes
- Removing the /.k file in the control workstation (assuming / is the home directory for root)
- Removing the /.klogin, /etc/krb-srvtab, /etc/krb.conf or /etc/krb.realms files on the control workstation or on the nodes

You may also want to destroy and rebuild Kerberos if you wish to change the control workstation′s hostname or the hostname by which the nodes are known to the customer network, in a planned fashion. For example, if the nodes have an FDDI adapter, in addition to the ethernet adapter, the customer′s application may require the hostname of the node to correspond to the *fi0* interface. Or, a customer application that runs on the HPS between the nodes may require that the node′s hostname be that of the *css0* interface.

Kerberos authenticates a node by the node′s hostname. In this context, we distinguish between the node′s *initial hostname* and *reliable hostname*.

- The **reliable hostname** of a node is the interface name on that node that corresponds to the ethernet adapter used to install PSSP, that is, the name of the *en0* interface.
- The **initial hostname** is (paradoxically) the actual hostname of the node.

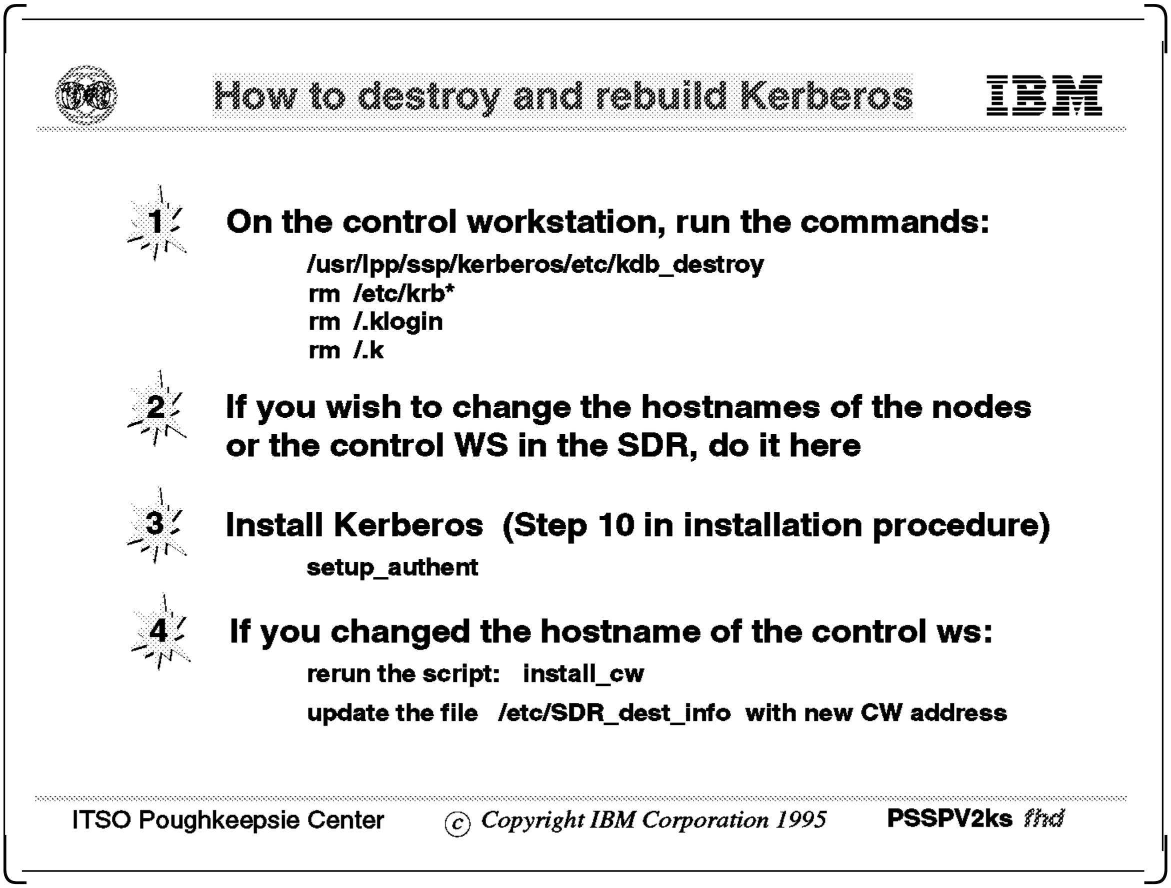
The initial hostname may be the same as the reliable hostname, but in many configurations, the initial hostname is different than the reliable hostname for the reasons outlined above.

Kerberos authenticates a node by its *initial* (actual) hostname. However, some PSSP commands update the SDR with the *reliable* hostname and *not* the initial hostname. If the control workstation or an application runs a remote command such as rsh or rcp, Kerberos will try to validate the node through the reliable hostname. Since Kerberos only knows the node by its initial hostname, it will cause the remote command to fail.

Note that if the initial hostname is different than the name of the en0 interface, there must be a way for the control workstation to communicate with that adapter - the adapter corresponding to the *initial* interface. For example, if you can resolve an adapter′s interface name but cannot *ping* that adapter and other systems can ping that adapter, something is lacking in your network routing.

The following two procedures outline how to destroy Kerberos and then rebuild it. After you do this, you must customize the nodes so that they get the updated information.

19.2.1 Procedure 1



How to destroy and rebuild Kerberos

1 On the control workstation, run the commands:

```
/usr/lpp/ssp/kerberos/etc/kdb_destroy
rm /etc/krb*
rm /.klogin
rm /.k
```

2 If you wish to change the hostnames of the nodes or the control WS in the SDR, do it here

3 Install Kerberos (Step 10 in installation procedure)

```
setup_authent
```

4 If you changed the hostname of the control ws:

rerun the script: install_cw

update the file /etc/SDR_dest_info with new CW address

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* PSSPV2ks *fhd*

This procedure requires a network boot.

Step 1: On the control workstation, run the following commands:

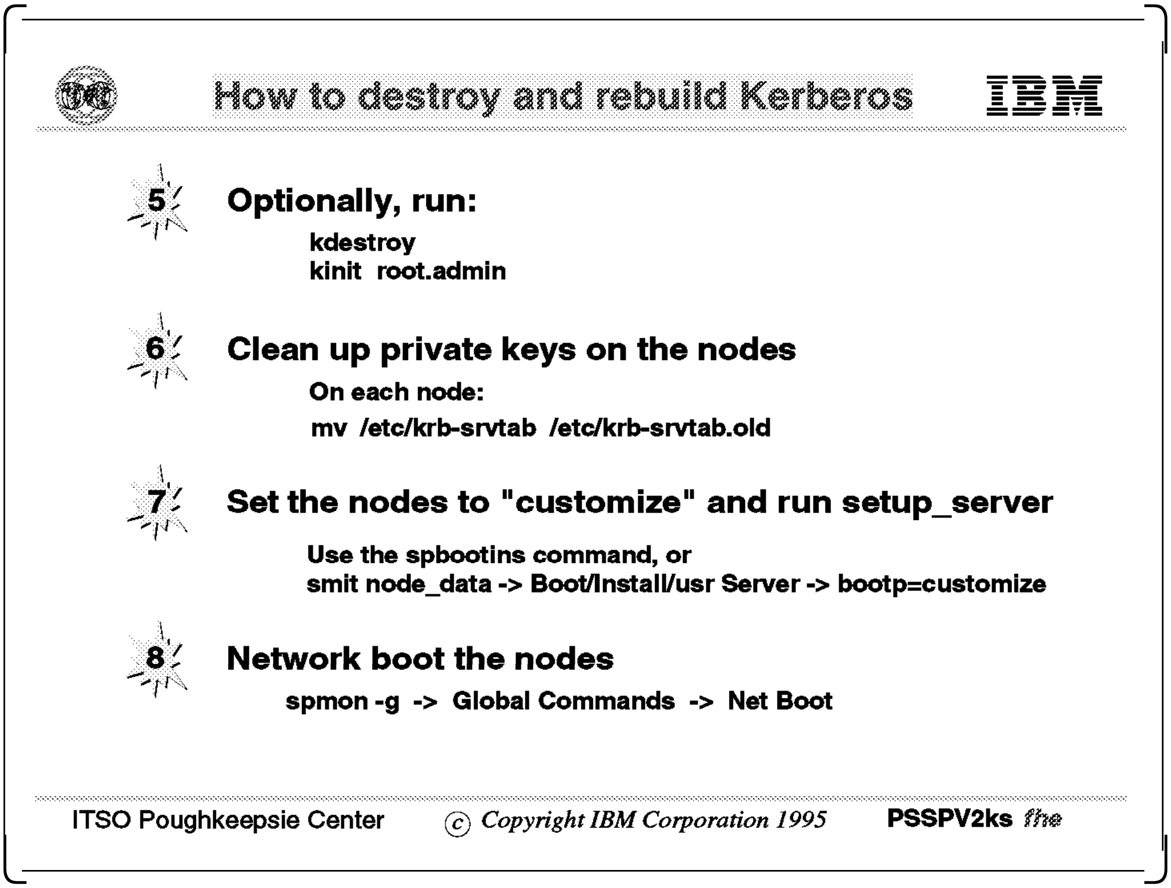
```
/usr/lpp/ssp/kerberos/etc/kdb_destroy
rm /etc/krb*
rm /.klogin
rm /.k
```

Step 2: Change the nodes′ hostnames in the SDR.

Step 3: Run **setup_authent**.

Step 4: If you changed the hostname of the control workstation:

- Verify that the SDR has the correct hostname and IP address.
- Also verify that the file /etc/SDR_dest_info contains the new information.
- Re-run **install_cw**. This will update the **/etc/hmacls** file.



How to destroy and rebuild Kerberos

5 **Optionally, run:**

kdestroy
kinit root.admin

6 **Clean up private keys on the nodes**

On each node:
mv /etc/krb-srvtab /etc/krb-srvtab.old

7 **Set the nodes to "customize" and run setup_server**

Use the spbootins command, or
smit node_data -> Boot/Install/usr Server -> bootp=customize

8 **Network boot the nodes**

spmon -g -> Global Commands -> Net Boot

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* **PSSPV2ks** *fhe*

Step 5: Optionally, run:

kdestroy
kinit root.admin, where root.admin is the principal named during setup_authent.

Step 6: Move the file /etc/krb-srvtab:

mv /etc/krb-srvtab /etc/krb-srvtab.old

If you don′t move it before customizing, the updated /etc/krb-srvtab file will not be pulled over from the control workstation. As a result, the file will not be in sync with the Kerberos database and you will not be able to communicate with that node using Kerberos.

Step 7: Change the nodes′ bootp response to customize:

spbootins -r customize 1 1 16 (for example).

or

smit node_data
Select **Boot/Install/usr Server**
Set **bootp=customize**

and then

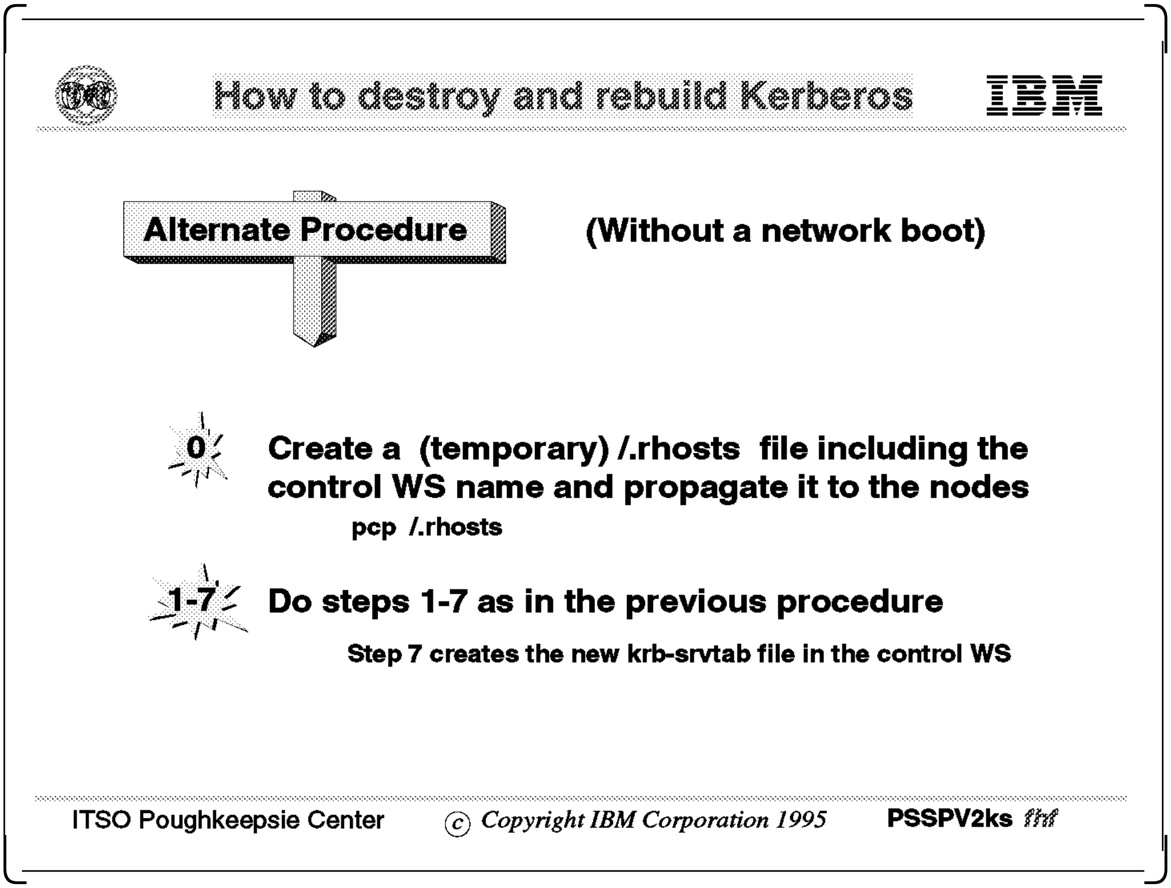
Run **setup_server**

Step 8: Do a netboot:

spmon -g
Select **Global Commands**
Do **Net Boot**

This will propagate the Kerberos files out to the nodes.

19.2.2 Procedure 2



How to destroy and rebuild Kerberos

Alternate Procedure

(Without a network boot)

0 Create a (temporary) /.rhosts file including the control WS name and propagate it to the nodes

pcp /.rhosts

1-7 Do steps 1-7 as in the previous procedure

Step 7 creates the new krb-srvtab file in the control WS

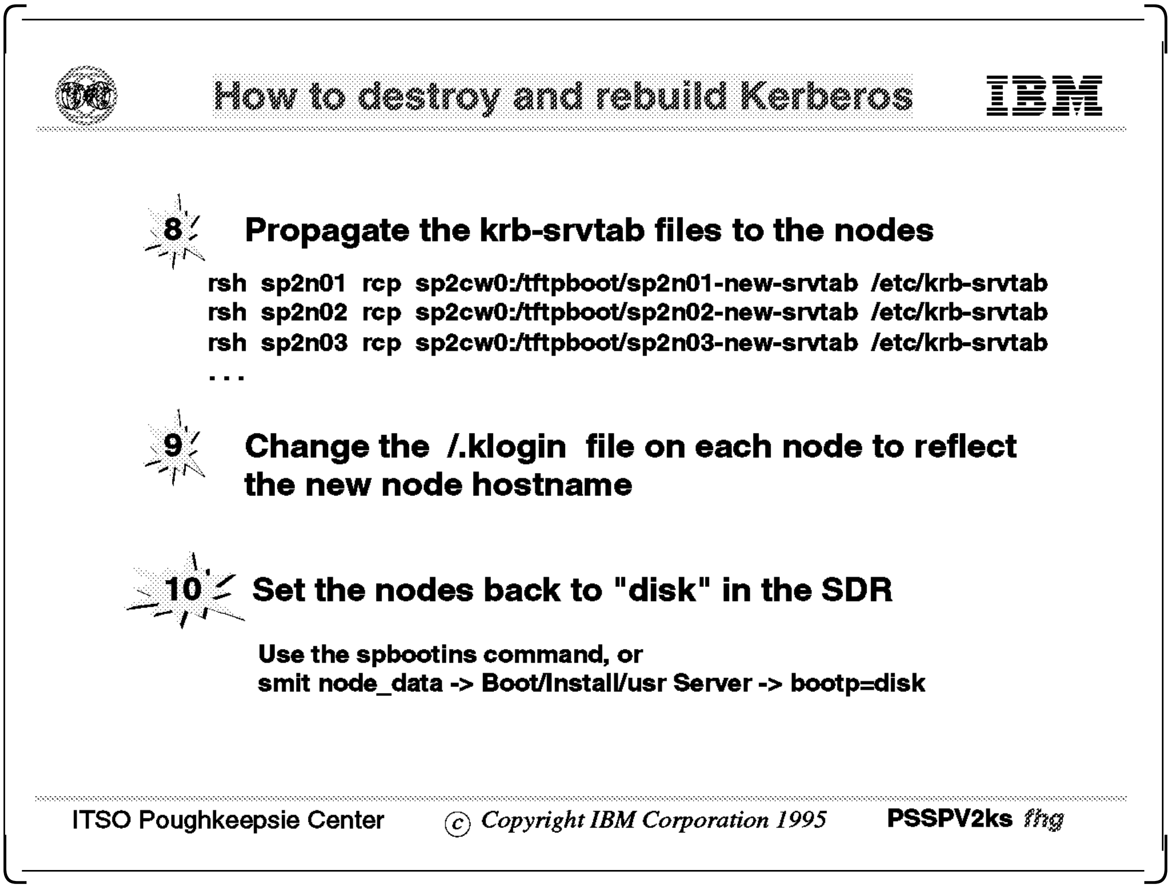
ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* **PSSPV2ks** fhf

This procedure does not require a network boot. On the other hand, it requires a (temporary) **/.rhosts** file containing the name of the control workstation in each node. So begin this procedure by creating such a file and propagating it to the nodes. These /.rhosts files will let us use rsh and remote authorization until Kerberos is rebuilt.

Steps 1-7: Do steps 1 through 7 as described in Procedure 1.

Step 7 creates on the control workstation the new **krb-srvtab** files intended for the nodes. These files exist in the **tftpboot** directory on the control workstation. For example, the krb-srvtab file for node *sp2n01* is called **/tftpboot/sp2n01-new-srvtab**. When it reaches its intended destination sp2n01, it will be called there /etc/krb-srvtab.

If desired, change the hostnames of the nodes (manually or using dsh) now.



How to destroy and rebuild Kerberos

8 Propagate the krb-srvtab files to the nodes

```
rsh  sp2n01  rcp  sp2cw0:/tftpboot/sp2n01-new-srvtab  /etc/krb-srvtab
rsh  sp2n02  rcp  sp2cw0:/tftpboot/sp2n02-new-srvtab  /etc/krb-srvtab
rsh  sp2n03  rcp  sp2cw0:/tftpboot/sp2n03-new-srvtab  /etc/krb-srvtab
...
```

9 Change the /.klogin file on each node to reflect the new node hostname

10 Set the nodes back to "disk" in the SDR

Use the spbootins command, or
smit node_data -> Boot/Install/usr Server -> bootp=disk

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ks fhg

Step 8: Propagate the krb-srvtab files out to the nodes.

```
rsh sp2n01 rcp sp2cw0:/tftpboot/sp2n01-new-srvtab /etc/krb-srvtab
rsh sp2n02 rcp sp2cw0:/tftpboot/sp2n02-new-srvtab /etc/krb-srvtab
...
```

where sp2cw0 is the control workstation′s hostname.

Step 9: Change the **/.klogin** file on each node to reflect the new node hostname.

Step 10: Set the bootp response of the nodes back to disk in the SDR.

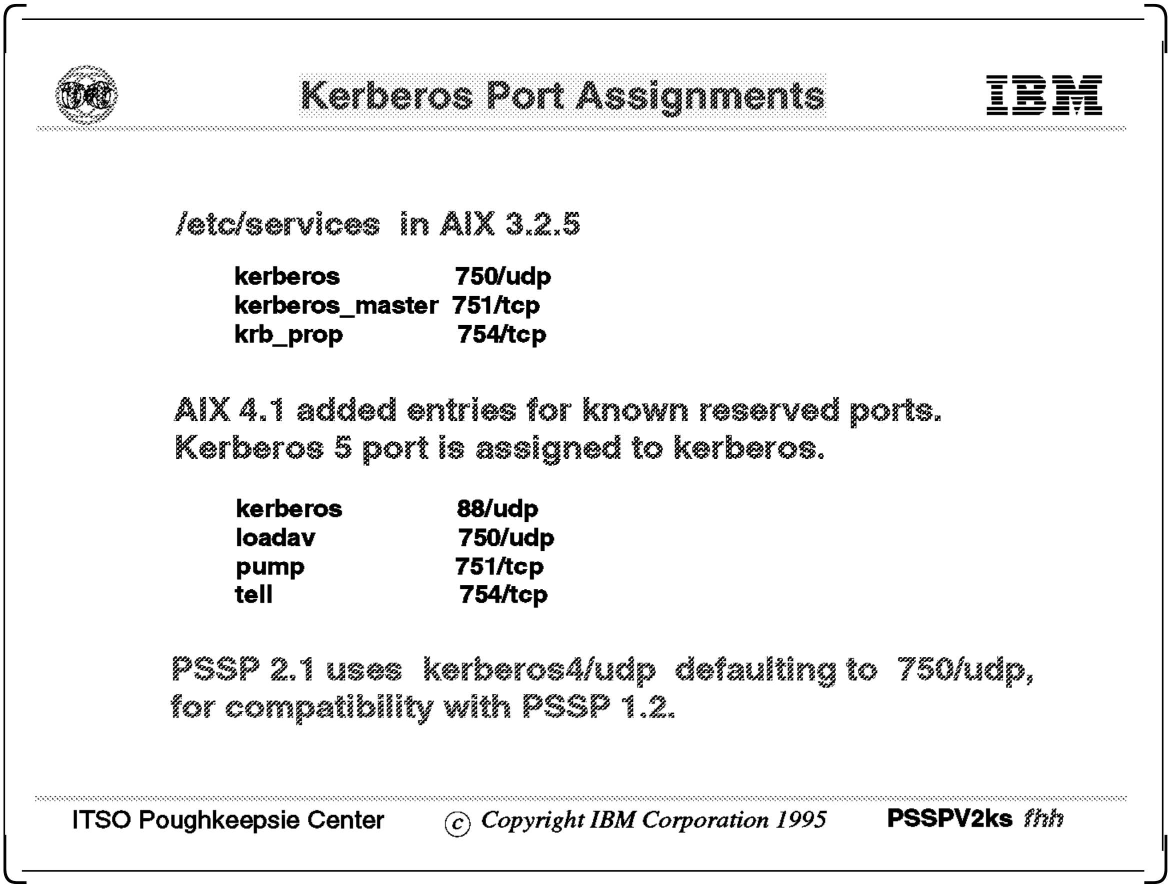
spbootins -r disk 1 1 16 (for example).

or

smit node_data
Select **Boot/Install/usr Server**
Set **bootp=disk**

This operation also removes the new-srvtab files in /tftpboot.

19.3 Kerberos Port Assignments



Kerberos Port Assignments

/etc/services in AIX 3.2.5

```
kerberos          750/udp
kerberos_master   751/tcp
krb_prop          754/tcp
```

AIX 4.1 added entries for known reserved ports.
Kerberos 5 port is assigned to kerberos.

```
kerberos          88/udp
loadav            750/udp
pump              751/tcp
tell              754/tcp
```

PSSP 2.1 uses kerberos4/udp defaulting to 750/udp, for compatibility with PSSP 1.2.

ITSO Poughkeepsie Center © *Copyright IBM Corporation 1995* **PSSPV2ks** *fhh*

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

The **/etc/services** file maps the names of network services to the well-known ports that they use to receive requests from their clients (every network service is known by the port number it uses on the network, hence the term well-known port). As a convenience to programmers, the services may be referred to by names as well as port numbers.

To understand current Kerberos port assignments on the SP, refer to Figure 1 on page 240 which compares excerpts from the */etc/services* file in AIX 3.2.5 and AIX 4.1 respectively.

The well-known port used by standard MIT Kerberos 4 is 750 and its service name is kerberos. PSSP 1.2 used the same port numbers traditionally used (but not formally reserved) by MIT Kerberos 4, shown on the left hand side of Figure 1 on page 240.

In AIX 4.1, entries for well-known ports used by Kerberos 5 and OSF Distributed Computing Environment (DCE), shown on the right hand side of Figure 1 on page 240 were added to */etc/services*. Kerberos 5 uses port 88 for authentication services. Its service name is “kerberos,” the same name used by standard MIT Kerberos 4.

```
      AIX 3.2.5                               AIX 4.1
(with PSSP 1.2 installed)             (with or without PSSP V2)
-------------------------             -------------------------
                                       kerberos         88/tcp
                                       kerberos         88/udp
                                       kshell           544/tcp
                                       kerberos-adm     749/tcp
                                       kerberos-adm     749/udp
                                       rfile            750/tcp
 kerberos         750/udp              loadav           750/udp
 kerberos_master  751/tcp              pump             751/tcp
                                       pump             751/udp
 krb_prop         754/tcp              tell             754/tcp
                                       tell             754/udp
```

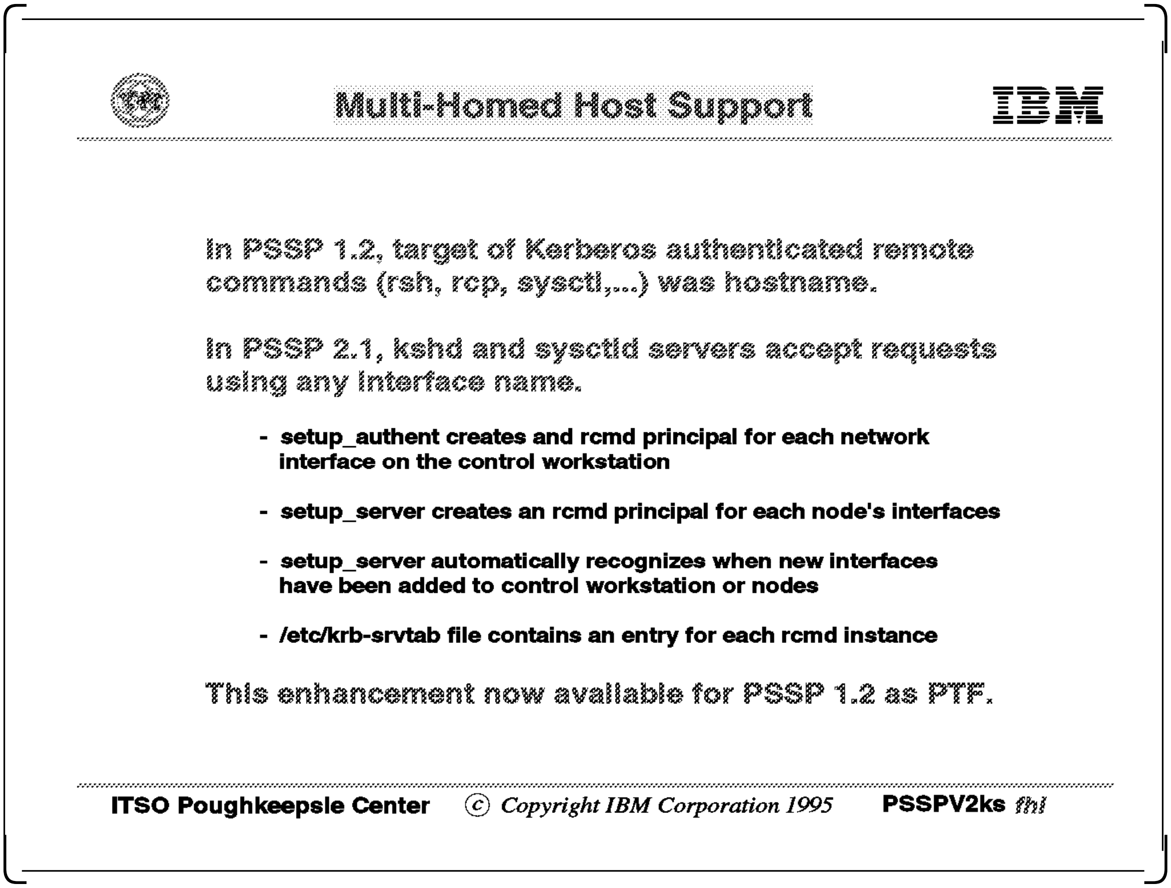
Figure 1. Kerberos Port Assignments in /etc/services

PSSP 2.1 is still based on MIT Kerberos 4, and therefore uses port 750, the port number usually assigned to Kerberos 4 authentication services. It is thus consistent and interoperable with AIX 3.2.5 systems running PSSP 1.2.

However, in order to avoid conflict with the AIX 4.1 definition of the “kerberos” service name and still use port 750 (by default), PSSP 2.1 authentication uses the service name **kerberos4**. It is not necessary for you to create an entry in the */etc/services* file for “kerberos4” because the default port 750/udp will be used if no “kerberos4” entry is found. You should only have to make modifications if your site uses some other service that requires the ports:

```
kerberos4        750/udp
kerberos_admin   751/tcp
krb_prop         754/tcp
```

19.4 Multi-Homed Host Support



Multi-Homed Host Support

In PSSP 1.2, target of Kerberos authenticated remote commands (rsh, rcp, sysctl,...) was hostname.

In PSSP 2.1, kshd and sysctld servers accept requests using any interface name.

- setup_authent creates and rcmd principal for each network interface on the control workstation
- setup_server creates an rcmd principal for each node's interfaces
- setup_server automatically recognizes when new interfaces have been added to control workstation or nodes
- /etc/krb-srvtab file contains an entry for each rcmd instance

This enhancement now available for PSSP 1.2 as PTF.

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ks fhl

In PSSP 1.2, the *server key table* **/etc/krb-srvtab** contained only one instance of the **rcmd.**<*hostname*> principal, where <*hostname*> was the name of the **en0** interface.

In PSSP V2, the *server key table* in SP nodes contains the names and private keys of all the local instances of *rcmd*. In other words, in SP nodes, the server key table contains an instance of the *rcmd* principal for each network interface defined on that node. For example, if node spn01 has four network interfaces, the *rcmd* principal may have the four instances:

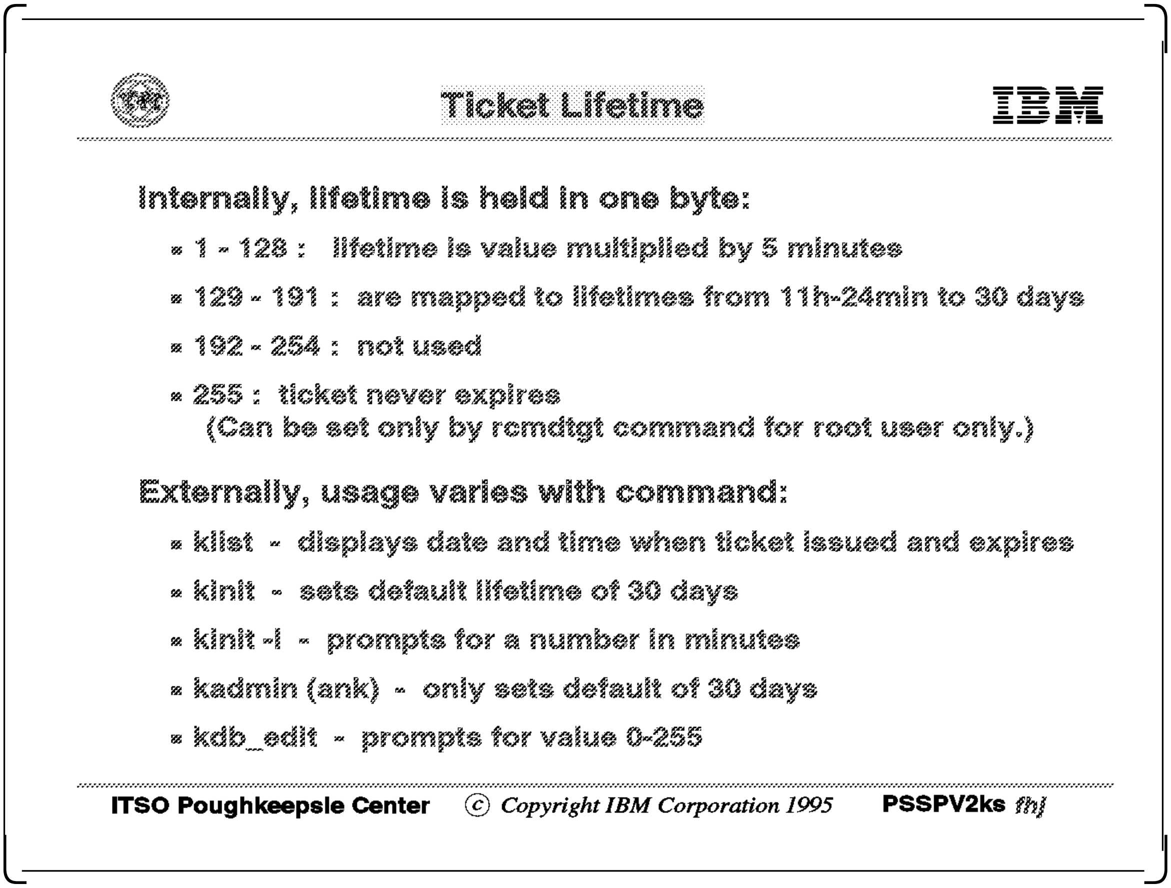
rcmd.spn01en
rcmd.spn01tr
rcmd.spn01fi
rcmd.spn01sw

The **setup_authent** script creates service principals for each network interface defined on the control workstation. The **setup_server** script creates service principals for each network interface defined on each node. In addition, **setup_server** automatically recognizes when interfaces are added to the control workstation or nodes (the network interfaces are recorded in the SDR). The **/etc/krb-srvtab** file is created by concatenating the files <*instance*>**-new-srvtab** generated by the script **ext_srvtab**.

It follows that the *kshd* and *sysctld* servers accept requests using any interface name (since these servers obtain their private key from the server key table when they need it to decrypt a service ticket sent by a client). This is referred to as *multi-homed host support*. This feature is available in PSSP 1.2 as the fix to APAR IX49179.

Note that although the multiple service instances are automatically added on the control workstation and the nodes, these must be manually maintained on other hosts where authenticated services are installed. Note also that the user invoking *setup_server* must have Kerberos *database administrator* credentials (see 18.2.7, “Kerberos Access Control Lists” on page 225). They are initially created by **setup_authent** (see 3.2.3, “Initialize RS/6000 SP Authentication Services” on page 47).

19.5 Ticket Lifetime



Ticket Lifetime

Internally, lifetime is held in one byte:

- 1 - 128 : lifetime is value multiplied by 5 minutes
- 129 - 191 : are mapped to lifetimes from 11h-24min to 30 days
- 192 - 254 : not used
- 255 : ticket never expires
  (Can be set only by rcmdtgt command for root user only.)

Externally, usage varies with command:

- klist - displays date and time when ticket issued and expires
- kinit - sets default lifetime of 30 days
- kinit -l - prompts for a number in minutes
- kadmin (ank) - only sets default of 30 days
- kdb_edit - prompts for value 0-255

ITSO Poughkeepsie Center © Copyright IBM Corporation 1995 PSSPV2ks fhj

The ticket granting ticket obtained when a user issues the *kinit* command has a default lifetime of **30 days**, or the value assigned to that user by the Kerberos administrator. This is the first ticket stored in the user′s current ticket cache file (the current cache file is the file named in the KRBTKFILE environment variable, or **/tmp/tkt**<*uid*> if the variable is not set).

The service tickets obtained thereafter always have the same lifetime as the ticket granting ticket used to obtain them. Service tickets are obtained when the user invokes a protected service such as *rcmd* or *hardmon* for the first time. The Kerberos authentication protocol uses the TGT from the user′s current ticket cache file to obtain the appropriate service ticket (see Step 8 in 15.1, "Kerberos Authentication Protocol" on page 176).

Internally, ticket lifetime is represented by one byte values, between 0 and 255. Byte values from 0 to 128 represent multiples of five minutes. Values from 129 to 191 represent lifetimes up to 30 days. Examples of internal representation are shown in Figure 2 on page 244.

```
Byte Value     Lifetime in minutes      Equivalent duration

  0 or 1          5 x 1 =    5
       2          5 x 2 =   10
       3          5 x 3 =   15
     ...
     ...
     128          5 x 128 = 640           10 hours 40 minutes

     129                                  11 hours 24 minutes
     130                                  12 hours 11 minutes
     131                                  13 hours  2 minutes
     ...
     ...
     191                                  30 days
```

Figure 2. Internal Representation of Ticket Lifetime

Values 192 to 254 are not used. The special value 255 represents a never expiring ticket. This value can be set only by the **rcmttgt** command and only by the *root* user.

It is possible to request a ticket with a specific lifetime using the command:

kinit -l

You are prompted to enter a value in *minutes* (not a multiple of five minutes). The value you enter is rounded up to the next higher discrete value from the internal representation table above. Hence, the minimum ticket lifetime is **five minutes**. The maximum lifetime that you can set is the value assigned to your principal by the Kerberos administrator in the authentication database or 30 days, whichever is less. Figure 3 shows some sample ticket lifetimes that you can request with *kinit*.

```
Response to kinit prompt   Approximate ticket lifetime

            1500                  1 day
            3000                  2 days
           10000                  1 week
           20000                  2 weeks
           43000                  1 month
```

Figure 3. Sample Ticket Lifetime Requests with kinit

A principal′s maximum (and default) ticket lifetime may be set by the administrator using the **kdb_edit** command. When creating or modifying a principal, the *kdb_edit* commands prompts for a ticket lifetime. The response must be a number between 0 and 255, representing a lifetime of five minutes to 30 days, the same as for the internal representation of ticket lifetime. Figure 4 on page 245 shows some examples of settings with *kdb_edit*.

```
Response to kdb_edit prompt   Approximate ticket lifetime

            141                     1 day
            151                     2 days
            170                     1 week
            180                     2 weeks
            191-255                 1 month
```

Figure 4. Sample Ticket Lifetime Settings with kdb_edit

The **klist** command displays the ticket expiration time (along with other information) as a standard time stamp.

List of Abbreviations

AIX	advanced interactive executive (IBM′s flavor of UNIX)
CD-ROM	(optically read) compact disk - read only memory
CDE	customer data extensions
CWS	control workstation
DAEMON	distribution and electronic maintenance over network
FOIL	file oriented interpretive language
FORTRAN	formula translation (programming language)
GB	gigabyte (1024*1024*1024 bytes)
Gb	gigabit (1024*1024*1024 bits)
HACMP	high availability cluster multi-processing (AIX)
HPS	high-performance switch
HSD	hashed shared disk
I/O	input/output
IBM	International Business Machines Corporation
IP	internet protocol (ISO)
ITSO	International Technical Support Organization
LPP	licensed program product
MB	megabyte (1024*1024 bytes)
Mb	megabit (1024*1024 bits)
MPI	message passing interface
NIM	network interface manager
ODM	object data manager (AIX)
PC	Personal Computer (IBM)
PING	packet internet groper
POWER	performance optimization with enhanced RISC (architecture)
PPD	POWERparallel Division (IBM)
PSSP	AIX Parallel System Support Programs (IBM program product for scalable POWERparallel systems)
PTF	program temporary fix
PVM	parallel virtual machine (developed by Oak Ridge National Laboratory)
RAID	Redundant Array of Independent Disks
RAS	reliability, availability, serviceability
RISC	reduced instruction set computer/cycles
ROM	read only memory
SCSI	small computer system interface
SDR	software data repository (IBM PSSP for AIX)
SMIT	System Management Interface Tool (see also DSMIT)
SP	Scalable POWERparallel
SRC	system resource controller
TCP/IP	Transmission Control Protocol/Internet Protocol (USA, DoD, ARPA
TTY	teletypewriter
UNIX	an operating system developed at Bell Laboratories (trademark of X/OPEN)
UPS	uninterruptible power supply/system
VSD	virtual shared disk

Index

ITSO Technical Bulletin Evaluation RED000

International Technical Support Organization
RS/6000 Scalable POWERparallel Systems:
PSSP Version 2 Technical Presentation
December 1995

Publication No. SG24-4542-00

Your feedback is very important to help us maintain the quality of ITSO Bulletins. **Please fill out this questionnaire and return it using one of the following methods:**

- Mail it to the address on the back (postage paid in U.S. only)
- Give it to an IBM marketing representative for mailing
- Fax it to: Your International Access Code + 1 914 432 8246
- Send a note to REDBOOK@VNET.IBM.COM

Please rate on a scale of 1 to 5 the subjects below.
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction ____

Organization of the book	____	Grammar/punctuation/spelling	____
Accuracy of the information	____	Ease of reading and understanding	____
Relevance of the information	____	Ease of finding information	____
Completeness of the information	____	Level of technical detail	____
Value of illustrations	____	Print quality	____

Please answer the following questions:

a) If you are an employee of IBM or its subsidiaries:

Do you provide billable services for 20% or more of your time? Yes____ No____

Are you in a Services Organization? Yes____ No____

b) Are you working in the USA? Yes____ No____

c) Was the Bulletin published in time for your needs? Yes____ No____

d) Did this Bulletin meet your needs? Yes____ No____

If no, please explain:

____

____

What other topics would you like to see in this Bulletin?

____

____

What other Technical Bulletins would you like to see published?

____

Comments/Suggestions: **(THANK YOU FOR YOUR FEEDBACK!)**

____
Name

____
Address

____
Company or Organization

____

____
Phone No.

____

ITSO Technical Bulletin Evaluation **RED000**
SG24-4542-00

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM International Technical Support Organization
Mail Station P099
522 SOUTH ROAD
POUGHKEEPSIE NY
USA 12601-5400

Fold and Tape **Please do not staple** Fold and Tape

Cut or Fold
Along Line

Printed in U.S.A.

SG24-4542-00