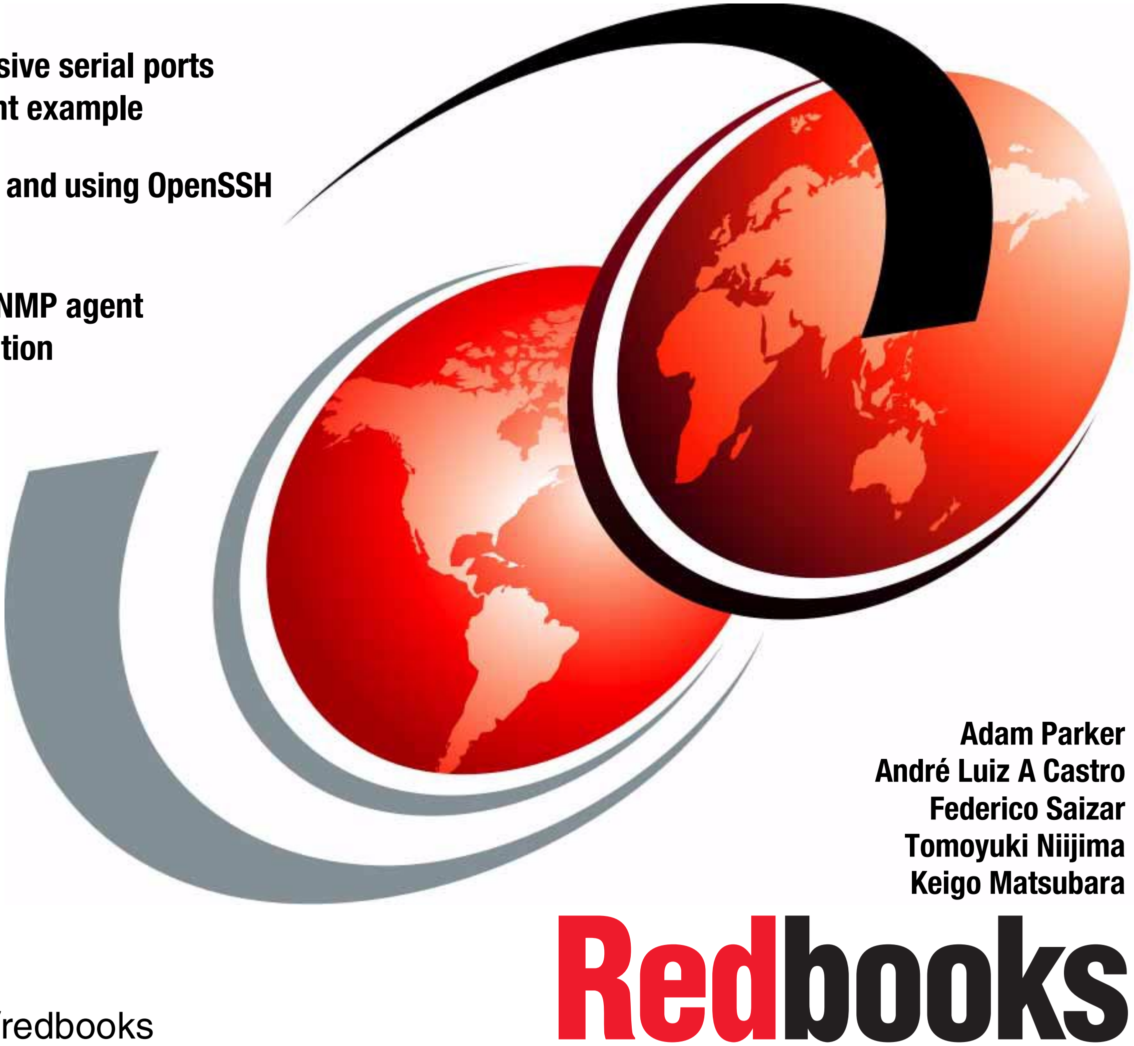
Managing AIX Server Farms

- Comprehensive serial ports management example
- Configuring and using OpenSSH on AIX
- Exploring SNMP agent implementation



Adam Parker
André Luiz A Castro
Federico Saizar
Tomoyuki Niijima
Keigo Matsubara

Redbooks

ibm.com/redbooks

International Technical Support Organization

Managing AIX Server Farms

June 2002

SG24-6606-00

Take Note! Before using this information and the product it supports, be sure to read the general information in “Notices” on page xiii.

First Edition (June 2002)

This edition applies to AIX 5L Version 5.1, Program Number 5765-E61.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. JN9B Building 003 Internal Zip 2834
11400 Burnet Road
Austin, Texas 78758-3493

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

Contents

Figures

Tables

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX®
AIX 5L™
DB2®
Electronic Service Agent™
IBM®
NetView®
OS/400®
Perform™
PowerPC®
pSeries™
Redbooks™
Redbooks(logo)™
RS/6000®
SP™
Tivoli®
WebSphere®

The following terms are trademarks of other companies:

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

C-bus is a trademark of Corollary, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.

Preface

Today's e-business relies heavily on secure, robust, scalable, and cost-effective server management, which is not an easy task to accomplish. The focus of this redbook is to explain practical methodology for administering AIX systems in a server farm environment, providing configuration and usage examples of several AIX operating system functions and free software tools, which include:

- Planning AIX server farms
- Understanding serial connections
- Practical use of serial connections
- Secure network connection on AIX
- Remote monitoring using SNMP
- Packaging your software tools
- Day-to-day tasks

This redbook is an ideal desk side reference for IBM employees, Business Partners, and customer system administrators or technical specialists who manage IBM @server pSeries servers running AIX 5L Version 5.1 in a server farm environment.

The team that wrote this redbook

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

Keigo Matsubara is an advisory IT specialist at the International Technical Support Organization (ITSO), Austin Center. Before joining the ITSO, he worked in the System and Web Solution Center in Japan as a Field Technical Support Specialist (FTSS) for pSeries. He has been working for IBM for ten years.

Adam Parker is an advisory IT specialist in e-business Hosting Delivery Development in Warwick, England. He has over eight years of experience in AIX. He holds an Honours degree in Computer Science from Wolverhampton University. His areas of expertise include AIX, Linux, IBM HTTP Server, and OpenSSH. He has been working for IBM for three years.

André Luiz A Castro is an advisory IT specialist for IBM Brazil in the pSeries division. He holds a degree in Electrical Engineering and has more than ten years of experience in the IT related environment. He has worked at IBM for four years, and has worked as an FTSS for pSeries for two years.

Federico Saizar is an IT specialist in IBM Uruguay. He provides level 2 support AIX, HACMP, SP and Linux for Latin America. He has an Informatic Analyst degree from Catholic University, Uruguay. He has been working for IBM for four years.

Tomoyuki Niijima is an advisory IT architect in ERM service/BIS/IGS in IBM Japan. He has more than 10 years experience in AIX.

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

International Technical Support Organization, Austin Center

Scott Vetter and Wade Wallace

IBM Austin

Andy Wong, Denise Genty, Jim Partridge, Jorge Rafael Nogueras, Julie Haugh, Julie Craft, Lauren T Vo, Ron Goering, Sajan Lukose, Shouchun Li, Steven Molis, Ufuk Celikkan, Zhi-wei Dai

IBM Japan

Kenji Matsuo

IBM United Kingdom

Bob Dyne

Eclipse Consulting Company

Terry Murray

Notice

This publication is intended to help designers, administrators, and support personnel who manage IBM @server pSeries servers that are running AIX 5L Version 5.1 in a server farm environment. The information in this publication is not intended as the specification of any programming interfaces that are provided by AIX 5L Version 5.1. See the PUBLICATIONS section of the IBM Programming Announcement for AIX 5L Version 5.1 for more information about what publications are considered to be product documentation.

Comments welcome

Your comments are important to us!

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

- Use the online **Contact us** review redbook form found at:

  ibm.com/redbooks

- Send your comments in an Internet note to:

  redbook@us.ibm.com

- Mail your comments to the address on page ii.

1

Planning AIX server farms

Today's e-business heavily relies on secure, robust, scalable, and cost-effective server management, which is not an easy task to accomplish. This chapter contains the following five sections, which outline basic planning tasks for administering AIX systems in a server farm environment:

- Section 1.1, "What is a server farm" on page 2
- Section 1.2, "Network design" on page 4
- Section 1.3, "Server planning" on page 8
- Section 1.4, "Storage planning" on page 9
- Section 1.5, "Security planning" on page 14

1.1 What is a server farm

Today's business demands that every company be connected to the Internet and conduct e-business with a Web site. These web sites are often deployed within a server farm. A server farm is a collection of servers or clusters in a secure, scalable environment, serving several difference services for either intranet or extranet users. A cluster is generally referred to as multiple servers of almost the same hardware resources, serving the same service, such as HTTP. All the servers in a server farm share the same well-designed infrastructure. For example, all the equipment in a server farm should be protected against power failures by uninterrupted power supplies (UPS).

A server farm offers many advantages, which include:

- Fast/reliable delivery of data
- High capacity
- Scalability and flexibility
- Simplified and cost effective physical placement
- Secure remote management
- Redundancy–no single point of failure

Figure 1-1 represents the typical physical layout of a server farm within a data center.

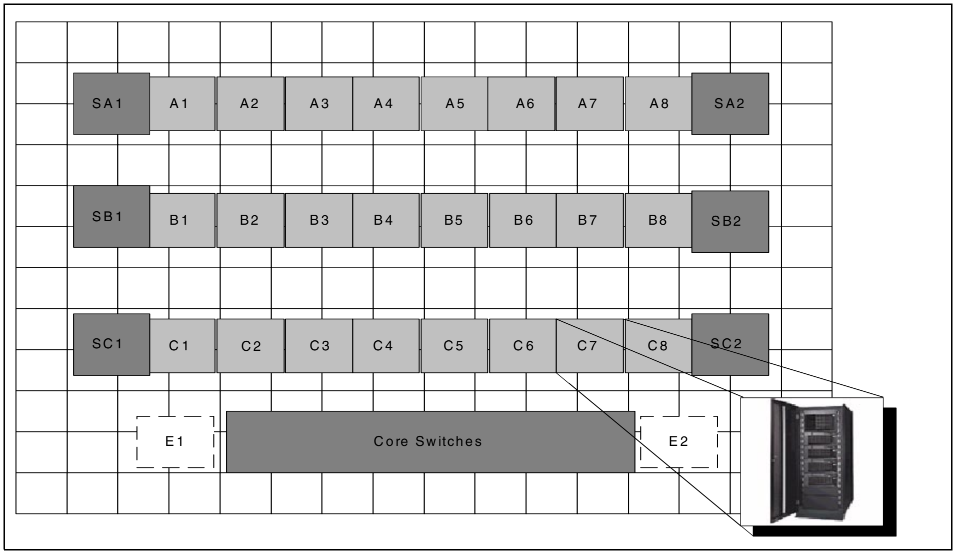


Figure 1-1 Physical racks layout for a large server farm

As shown in Figure 1-1 on page 2, a server farm is best made up of rack-mountable units. With floor space at a premium, this is an ideal solution. It makes physical access much easier for both hardware engineers and installation staff, as the systems are usually mounted on drawers within the racks.

The boxes labeled *A*1-8, B1-8, and C1-8 are rack identifiers. These typically contain rack mountable servers, for example, the IBM @server pSeries 610 Model 6C1 (referred hereafter to as Model 6C1). The darker grey boxes labeled SC*n* are usually locked communications racks, containing all the UTP cable trunking and distribution switches. Each physical device should be carefully labeled on the front *and* back. This will make identification of an individual component easier.

Racks are also labeled for easier identification of individual units. Spaces must be left between the rack for air movement and physical access, such as hardware maintenance and media mounts.

The large grey box at the bottom labeled core switches will contain the farms core switches, firewalls and the routers. Spaces to either side, labeled E1 and E2, could be used by stand-alone enterprise servers, such as the IBM @server pSeries 690.

You can deploy smaller server farm, as shown in Figure 1-2. All methodologies discussed within this publication are applicable in any size farm.

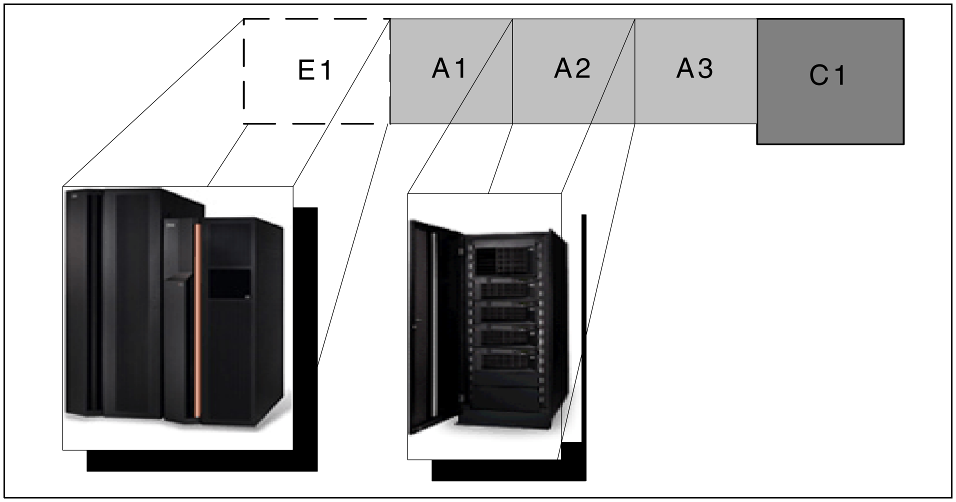


Figure 1-2 Smaller Server farm configuration

1.2 Network design

When designing a network, many aspects must be taken into consideration. Throughout this redbook, we only refer to IP networks. At all stages, careful documentation that includes the considerations explained in this section should be written, so everyone understands the design and can support the infrastructure.

1.2.1 Network technologies

IBM @server pSeries servers and AIX 5L Version 5.1 support varies network technologies, such as Ethernet, Token Ring, ATM (asynchronous transfer mode), and FDDI (Fibre Distributed Data Interface). You should carefully select these network technologies to meet your network requirements.

Ethernet is the most popular LAN technology because it is easy to implement and manage, and it is inexpensive. Therefore, we simply refer to Ethernet as the main network technology throughout this redbook.

Ethernet technology has evolved over many years from 10 Mbps to 100 Mbps (Fast Ethernet) and the recent 1Gbps (Gigabit Ethernet). Currently, AIX also supports the EtherChannel function, which provides us with the capacity and redundancy by aggregating multiple Ethernet network adapters. For further information about EtherChannel, see Section 7.3.1, “EtherChannel” on page 253.

Table 1-1 lists PCI bus Ethernet adapters supported by AIX 5L Version 5.1.

Table 1-1 PCI bus Ethernet adapters

Card Type	Feature code	Speed
Ethernet T2 PCI	2985	10 Mbps
Ethernet T5 PCI	2987	10 Mbps
10/100 Ethernet Tx PCI Adapter	2968	10/100 Mbps
Gigabit Ethernet-SX PCI Adapter	2969	1000 Mbps
4-Port 10/100 Base-TX Ethernet PCI Adapter	4951	10/100 Mbps per port
10/100/1000 Base-T Ethernet PCI adapter	2975	10/100/1000 Mbps
4-Port 10/100 Base-TX Ethernet 64-bit/66MHz PCI Adapter	4961	10/100 Mbps per port

Card Type	Feature code	Speed
10/100 Mbps Ethernet PCI Adapter II	4962	10/100 Mbps per port

Entry and midrange pSeries server models have at least one integrated Ethernet port. For example, Model 6C1 has two integrated 10/100 Mbps Ethernet ports in order to be more usable in a server farm environment.

1.2.2 Physical devices

There is a vast array of different hardware you can install in a server farm. This section outlines some of the most common appliances.

Ethernet switches

An Ethernet switch[1] is the “glue” that binds the server farm together. All devices within the server farm (servers, firewalls, or routers) are all connected to the switches. The switch forwards packets from one port to another. There are many different types of switches that work in different TCP/IP protocol stack layers. A level 2 (L2) switch has limited intelligence and is often used for connecting routers and firewalls. A level 3 (L3) switch has enforcements and broadcast constraints, allowing the creation of VLANs (Virtual LANs). A VLAN is a collection of Ethernet ports on the switches that comprises an virtual separated sub network. Typically, a VLAN forms a broadcast domain and is used to connect servers that share the same services.

Routers

Routers provide the ability to connect logically different networks or sub-networks together. Each incoming network packet entering the router is examined, and predefined rules on the router instruct the router engine where to forward the packet.

Firewalls

A firewall is network device that is designed to keep certain users in and certain users out by filtering network packets. It allows only certain packets that meet a certain pre-defined criteria (sometimes referred to as *rule*) to get through. There are many different types of firewalls, including dedicated appliances and software products that are installed on operating systems.

[1] You should not confuse an Ethernet switch with a traditional Ethernet hub. A hub typically has little or no intelligence; therefore, the network throughput on a switch is much higher than that of a hub.

Terminal Servers

Terminal servers are devices that allowed serially connected devices, such as data terminals and modems, to access the network. They can also be connected to serial ports on servers to provide enhanced security and contingency against network failure. We explain about terminal servers in detail in Section 3.1, “Using terminal server” on page 58.

1.2.3 Logical planning

Logical planning involves the creation of the several different zones within the server farm. Figure 1-3 illustrates the basic concept of zones in a server farm.

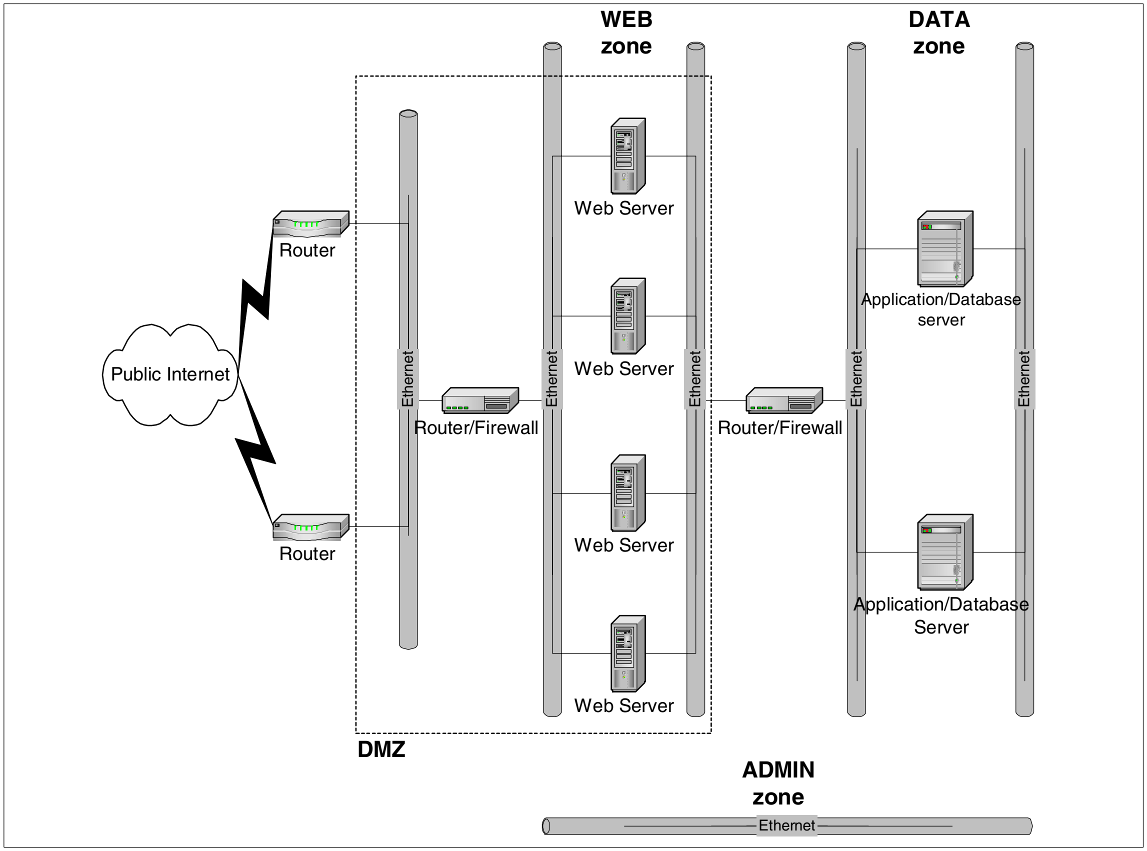


Figure 1-3 Logical network design with DMZ

Logical zones

When designing networks in a server farm, each element should be broken down into several logical zones. Table 1-2 provides you a good starting point how to plan logical zones.

Table 1-2 Logical zones in server farms

Logical zones	Description
Internet zone	The very front of the server farm without firewall protection. Normally, only the front-end routers will sit here. You also may wish to install a Web server monitor tool here too.
Web zone	Normal place for Web servers, network dispatcher devices, SMTP e-mail servers, and DNS servers. This zone is protected by the front-end firewalls.
Data zone	Place for database or application servers.
Admin zone	Location of servers that are used to administer all other systems within the server farm.
Build and test zone	Testing environment where new servers are built and applications are tested.

Serial connectivity

Figure 1-4 on page 8 illustrates two methods to implement serial connectivity in a server farm. The first method that connects servers using a terminal server uses a physically secure private network, because typical terminal server appliance products do not support a secure network connection. The second method that connects servers uses a multiport serial adapter attached to the administration server.

We explain in detail the terminal server implementation and secure network connection in the following two chapters:

- Chapter 3, “Practical use of serial connections” on page 57
- Chapter 4, “Secure network connection on AIX” on page 115

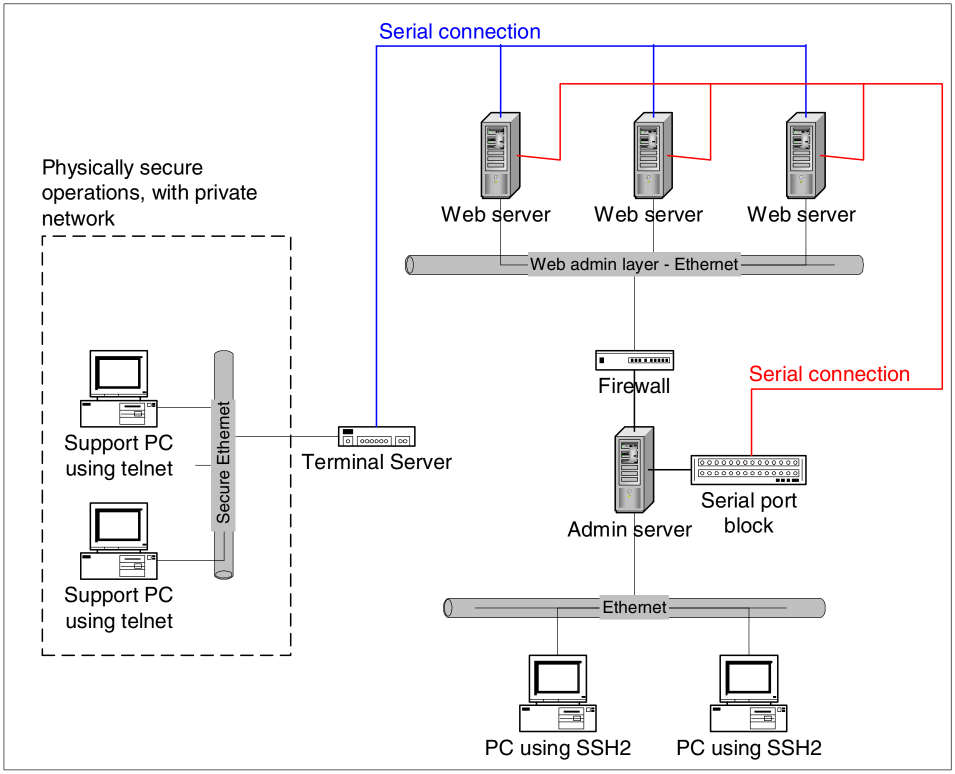


Figure 1-4 Serial connected devices

1.3 Server planning

A server farm is best made up of rack mountable units. You have to consider the following issues to select the best models from the IBM @server pSeries server product line, in addition to the performance and sizing requirement:

- Resource scalability, such as CPU, memory, disk storage, and PCI slots
- Physical size and weight
- Power consumption
- Air flow and heat
- Server sizing based on estimated performance is a complex and wide topic that can take up an entire book. Therefore, we recommend you to refer to the redbook *Understanding IBM @server pSeries Performance and Sizing*, SG24-4810, for more on this topic:

1.4 Storage planning

You have to carefully plan your storage needs to meet your business requirements. Although there are many options available, from locally attached SCSI or SSA disk drives to RAID storage attached using Storage Area Networks (SANs), we refer to internal SCSI disk drives as our storage devices to keep this redbook simple. However, you can also adopt the concept (explained in this section) of other storage devices, because the AIX logical volume manager provides uniformed access to any storage devices through the *physical volume* (PV) device.

1.4.1 Volume groups and logical volumes

The mandatory volume group where the AIX operating system is installed is called rootvg[2]. We recommend that you store only AIX operating systems and application software products in the rootvg. You can also use additional volume groups to store all the non-operating system, customer loaded data, such as static and dynamic Web content. Throughout this redbook, we use the name *datavg* to express an additional volume group in our environment.

You can mirror or stripe logical volumes within a volume group with multiple physical volumes for redundancy or performance reasons.

For further information about logical volume management, please refer to *AIX Logical Volume Manager from A to Z: Introduction and Concepts*, SG24-5432.

1.4.2 Paging and dump devices

Page spaces and dump devices can be dynamically defined and destroyed in AIX 5L Version 5.1.

Paging space

Paging space stores dirty (modified) virtual memory pages that are not mapped in physical memory pages. The size of your paging space depends upon the virtual memory usage, especially what application is running on the system. Many administrators have historically created paging space with a size of twice the physical memory size; however, this could be a waste of disk space on systems equipped with much physical memory. You can create paging spaces other than rootvg. Although there is no established rule for the size of paging space, we offer the following guidelines.

- For typical virtual memory usage systems:

  Total size of Page Space = (Physical Memory) * 1.25

[2] You cannot change this name.

- For huge virtual memory usage systems, such as database servers:

  Total size of Page Space = (Physical Memory) * 2

You should monitor the required paging space size using the **lsps -a** command on your system. If your system has too small size paging space, you might see warning messages sent from the kernel. This prompts you to either add more paging space or increase the size of the existing paging spaces.

If you have too much paging space, it is possible to reduce the size of paging spaces or remove them all together.

Reducing paging spaces

You can easily reduce the size of a paging space in AIX 5L Version 5.1, as shown in the following example:

```
# lsps -a
Page Space  Physical Volume   Volume Group    Size   %Used  Active  Auto  Type
paging00    hdisk2            datavg        1152MB       1     yes   yes    lv
hd6         hdisk0            rootvg         512MB       1     yes   yes    lv
# chps -d 20 paging00
shrinkps: Temporary paging space paging01 created.
shrinkps: Paging space paging00 removed.
shrinkps: Paging space paging00 recreated with new size.
# lsps -a
Page Space  Physical Volume   Volume Group    Size   %Used  Active  Auto  Type
paging00    hdisk2            datavg         512MB       1     yes   yes    lv
hd6         hdisk0            rootvg         512MB       1     yes   yes    lv
```

The -d option of the **chps** command instructs the operating system to reduce the paging space size. In this example, the paging size of paging00 is reduced from the original size of 1152 MB to 512 MB. This process is disk intensive, and may create a temporary performance degradation on your system, but no reboot is required.

Removing paging spaces

You can easily remove a paging space in AIX 5L Version 5.1, as shown in the following example:

```
# lsps -a
Page Space  Physical Volume   Volume Group    Size   %Used  Active  Auto  Type
paging00    hdisk2            datavg         512MB       1     yes   yes    lv
hd6         hdisk0            rootvg         512MB       1     yes   yes    lv
# swapoff /dev/paging00
# lsps -a
Page Space  Physical Volume   Volume Group    Size   %Used  Active  Auto  Type
paging00    hdisk2            datavg         512MB       0      no   yes    lv
hd6         hdisk0            rootvg         512MB       1     yes   yes    lv
```

```
# rmps paging00
rmlv: Logical volume paging00 is removed.
# lsps -a
Page Space  Physical Volume   Volume Group    Size   %Used  Active  Auto  Type
hd6         hdisk0            rootvg         512MB       1     yes   yes    lv
```

The **swapoff** command instructs the operating system to sweep out the existing virtual pages on the target paging space. The **rmps** command can remove the swapped-off paging space.

Dump device

If an unexpected system halt occurs, the operating system attempts to copy key memory areas (called a system dump) to the dump device. The system dump is designed to aid problem determination for the cause of the halt. On systems with less than 4 GB size of physical memory, the default dump device is assigned to the primary paging space. Otherwise, the default dump device lg_dump is created upon AIX installation.

Note: The dump device must be placed in the rootvg.

We recommend that you create a separate primary dump device, because if a system dump is found on reboot, the system will attempt to copy the system dump to the /var file system. If there is not enough room in /var, it will prompt you to select either of copying the dump to media or discarding the dump; therefore, manual intervention is required, which is unacceptable in a server farm environment. To avoid this situation, you should create a separate primary dump device, as shown in the following example:

```
# sysdumpdev -l
primary              /dev/hd6
secondary            /dev/sysdumpnull
copy directory       /var/adm/ras
forced copy flag     TRUE
always allow dump    FALSE
dump compression     OFF
# sysdumpdev -e
0453-041 Estimated dump size in bytes: 113246208
# mklv -y pdumplv -a e -c 2 -w n rootvg 7 hdisk0 hdisk1
pdumplv
# sysdumpdev -p /dev/pdumplv -P
# sysdumpdev -C -d /var/adm/ras
# sysdumpdev -l
primary              /dev/pdumplv
secondary            /dev/sysdumpnull
copy directory       /var/adm/ras
forced copy flag     FALSE
```

```
always allow dump    FALSE
dump compression     ON
# /usr/lib/ras/dumpcheck
```

The **dumpcheck** command is run by root user’s cron entry every day at 15:00. If your dump space is too small, you will see the following error log entry in your system:

```
LABEL:          DMPCHK_TOOSMALL
IDENTIFIER:     E87EF1BE

Date/Time:       Fri Mar 15 15:05:12 CST
Sequence Number: 72
Machine Id:      0001616F4C00
Node Id:         svr04
Class:           O
Type:            PEND
Resource Name:   dumpcheck

Description
The largest dump device is too small.

Probable Causes
Neither dump device is large enough to accommodate a system dump at this time.

        Recommended Actions
        Increase the size of one or both dump devices.

Detail Data
Largest dump device
pdump

Largest dump device size in kb
       16384
Current estimated dump size in kb
      111616
```

1.4.3 File systems

Although the size of your file systems obviously varies depending upon what software you have installed, we show a recommended file system layout and usage in Table 1-3, used throughout this redbook.

Table 1-3 File system descriptions

File system	Volume group	Example size	Description
/	rootvg	64 MB	Root file system.

File system	Volume group	Example size	Description
/usr	rootvg	1.5 GB	/usr file system should only be used to store files provided by AIX and other IBM software products.
/var	rootvg	200 MB	/var spool files, some temporary files.
/tmp	rootvg	500 MB	Temporary files.
/home	rootvg	64 MB	User’s home directories.
/opt	rootvg	64 MB	/opt file system should only be used for storing files provided by AIX toolbox for Linux applications (see Section 6.1.6, “RPM (Red Hat Package Manager)” on page 218).
/stats	rootvg	64 MB	Performance statistics data (see Section 7.5.2, “Long term trend analysis” on page 270).
/backup	datavg	5 GB	DVD-RAM/CD backup file system (see Section 7.1.2, “mksysb on DVD or CD” on page 236).
/usr/local	rootvg	300 MB	User installed software (typically free software tools).
/www	datavg	500 MB	Web server contents.
/var/adm/dump	rootvg	650 MB	System dump copied off the dump device.
/usr/HTTPServer[a]	rootvg	300 MB	/usr/HTTPServer file system should only be used for only storing files provided by IBM HTTP Server (IHS).
/logs	datavg	5 GB	General application logs.[b]
/swdist	datavg	650 MB	Used for distributing software packages (see Section 7.2.1, “Distributing software” on page 252).

[a] We assume that you create this file system before installing IHS.

[b] You should create a symbolic link in this file system to point to /usr/HTTPServer/logs.

The size of /usr will vary considerably depending upon how much software you install. One option you should carefully consider is whether to install X11 desktop environments, such as GNOME, KDE, or CDE. These packages take up a lot of room and may waste a lot of disk space. You should install the X11 and Motif libraries for any software that requires an X window system, for example, the ikeyman key management software that comes with IBM HTTP Server or the IBM WebSphere Application Server console.

The /logs file system can vary considerably in size, depending on how busy your application is. The size of the log also depends upon what type of logging you perform; combined logging on a busy site can produce copious amounts of data.

The two file systems, /usr/HTTPServer and /www, are only required on Web server systems.

1.5 Security planning

Security is critical when designing a server farm. Careful consideration must be given to what you are protecting and restricting access to. Every aspect of security must be considered, from employees walking on to the site, to remote users dialing in from home. The security policy should be defined as you design the network, and presented to all users before any server is set up in the farm.

1.5.1 Physical access

Physical access to the server farm should be restricted to authorized users only. These users include certified hardware engineers, technical support personnel, and customer system operators. Unauthorized indivuals may interfere with the smooth running of your farm and would be unaware of the security rules within the farm. Your server farm should be in an air conditioned data center, possibly even in a locked cage within a computer room. Access should be through controlled badged access (with auditing and logging) or coded entry. If visitors are allowed access, they should be recorded and supervised at all times.

1.5.2 Logical access

Logical access defines who is granted access to the farm over the network. Table 1-4 on page 15 outlines several logical access rules you should consider.

Table 1-4 Server farm logical access rules

Sample rule	Description
Port Filtering	On each firewall, all ports should be blocked, unless specifically needed. This reduces the exposure of insecure services being compromised.
Use VLANs	VLANs introduce an extra level of security; systems which need to communicate with each other can only see each other and not the rest of the servers in the farm.
Separation of zones	Separate each zone in the farm with a firewall or router.
No traffic initiate from within the farm to trusted network	No traffic initiated from within the farm should be allowed out onto your trusted private network. However, traffic initiated from your network should be allowed into the farm.
No UDP across zones (except SNMP)	No UDP protocols should be allowed across zones in the farm. The exception to this will be SNMP.
Physically separated network for building environment	New servers should be built on a physically separated network to isolate your building environment, because newly installed machines may not be hardened while being installed. Once built and hardened, they should be moved into their correct layer.
TCP wrapper on all services	Each AIX service being used should be wrapped to restrict access to authorized hosts (see Section 4.1.3, “TCP wrapper” on page 127).
No compilers allowed on any production server	You should not install any compilers software that can produce or change object codes on any production servers.
No direct login as root	No user should be allowed to log in as root. They should log in using their own individual and unique account and `su` to root.
SMTP should be configured to send only	If you have a dedicated SMTP relay, SMTP or sendmail should be configured on each server to send mail only–not receive.
No TFTP and, where possible, no FTP	The TFTP and FTP services are generally considered weak from the network security perspective (see Section 4.1, “Securing network services on AIX” on page 116).
Disable unused TCP services	Disable any TCP based service you do not plan to use (see Section 4.1, “Securing network services on AIX” on page 116).

Sample rule	Description
No telnet access to server farm	The TFTP and FTP services are generally considered weak from the network security perspective (see Section 4.1, “Securing network services on AIX” on page 116).
Use keys as primary authentication method	Use encrypted public and private keys for login access, not passwords (Section 4.3.4, “Using a passphrase” on page 146).
Public access daemons run as non-root user	Services accessed by anonymous users, such as a HTTP server, should run as a registered non-root user (this excludes the user nobody).
Permissions should be monitored	Monitor the permissions on key AIX system files to ensure they are not modified (see Section 7.6.2, “Security checks” on page 274).
Log retention for 60 days	All logs recording who has accessed what in the server farm should be retained for a minimum of 60 days.
Check security rules every week	You should perform checks to ensure your security policy is being used (see Section 7.6.2, “Security checks” on page 274).

In a server farm environment, all the passwords should be strictly enforced. The root passwords should only be given to very trusted employees. Set up a new UNIX group, for example, called suroot, so that only users in this group will be allowed to **su** to root. The root passwords should be kept in one central location (outside the server farm) in an encrypted database. Access to this database should be controlled and auditable. Table 1-5 indicates some password changing rules you should employ. These can apply to user accounts and to the root account.

Table 1-5 Password changing rule

Rule	AIX user parameter
Passwords expire every 60 days.	`maxage=9`
Password length should be longer than or equal to eight characters.	`minlen=8`
Password should contain two or more numerical characters.	`minother=2`
New password should not contain more than three characters from the old password.	`mindiff=3`
No character should be in the password more than twice.	`maxrepeats=2`

Rule	AIX user parameter
The last 26 old passwords should not be used again.	`histsize=26`
Accounts which have five or more failed login attempts should be locked.	`loginretries=5`
Once a password has been changed, it should not be possible to change it again for two weeks.	`minage=2`
Only suroot group can **su** to root.	`sugroups=suroot (on root account)`

When employees leave the company, their access accounts should be removed from all the servers. If they had access to any infrastructure server, such as a firewall or router, these passwords should be changed.

For further information about user management, see Section 7.6.3, "User maintenance" on page 281.

1.5.3 Network security

Network security is defined as the proper safeguarding of all components associated with a network, including data, media, and infrastructure. A comprehensive approach to network security involves three essential elements, namely, accurate threat assessment, use of the best cryptographic tools available, and deployment of effective network access control products, such as firewalls. Perhaps most importantly, network security may only be achieved by ensuring that all network resources are used in compliance with a prescribed corporate policy and only by authorized personnel.

There are many ways to achieve varying levels of network security; however, these methods can be extremely expensive or may not completely protect users from the many hazards that crop up on a daily basis. Proper implementation of network security is neither trivial nor cheap, and requires expertise that encompasses most areas of network science.

A challenge inherent to network security is determining the right level of security required for proper control of system and network assets. This concept, known as threat assessment, identifies the assets you have and who may attempt to access them. Organizations can best assess corporate network threats by using a structured approach.

The network security consists of the following three major components:

- User authentication is provided at the remote host by a user name and password.

- Connection authentication is provided to ensure that the remote host has the expected Internet Protocol (IP) address and host name. This prevents a remote host from masquerading as another remote host.
- Data import and export security permits data at a specified security level to flow to and from network interface adapters at the same security and authority levels.

The main things you want to protect in the network are:

- *Data* that remains on computers connected to the network transmitted on the network. On behalf of data protection, you may need the following aspects be provided by network protocols:
  - Secrecy
  - Authentication
  - Non repudiation
  - Authorization
  - Integrity check
- You should be concerned about the availability of network resources.

For further information about network security, please consult with the following publications:

- *A Practical Guide To Network Security* Whitepaper, found at (IBM internal):

  `http://w3-1.ibm.com/services/so/e-business_hosting/ftp_files/pdf/exodussecuritywp.pdf`
- *TCP/IP Tutorial and Technical Overview,* GG24-3376

2

Understanding serial connections

This chapter discusses the basic concept of serial connections and how to administer TTY devices on AIX. If you are familiar with these contents, you can skip to Chapter 3, “Practical use of serial connections” on page 57.

This chapter contains the following sections:

- Section 2.1, “Serial connections overview” on page 20
- Section 2.2, “Managing serial ports” on page 32

For further information about serial communications, please refer to the following publications:

- *AIX 5L Version 5.1 System Management Guide: Communications and Networks*
- *AIX 5L Version 5.1 Asynchronous Communication Guide*

2.1 Serial connections overview

To administer serial connections on AIX, you have to understand the following concepts explained in this section:

- Section 2.1.1, “Serial and parallel data transmission” on page 20
- Section 2.1.2, “Synchronous and asynchronous communication” on page 22
- Section 2.1.3, “Serial communications terminology” on page 23
- Section 2.1.4, “EIA-232 Standard” on page 25
- Section 2.1.5, “Serial port cabling” on page 28

2.1.1 Serial and parallel data transmission

The name *serial* of serial communication came from the concept of data transmission methods explained in this section.

Parallel

In Figure 2-1, we show the transmission of the letter *a* from the host (computer) and a printer. The data bits corresponding to the ASCII character “a” are transmitted through the eight pins and wires simultaneously.

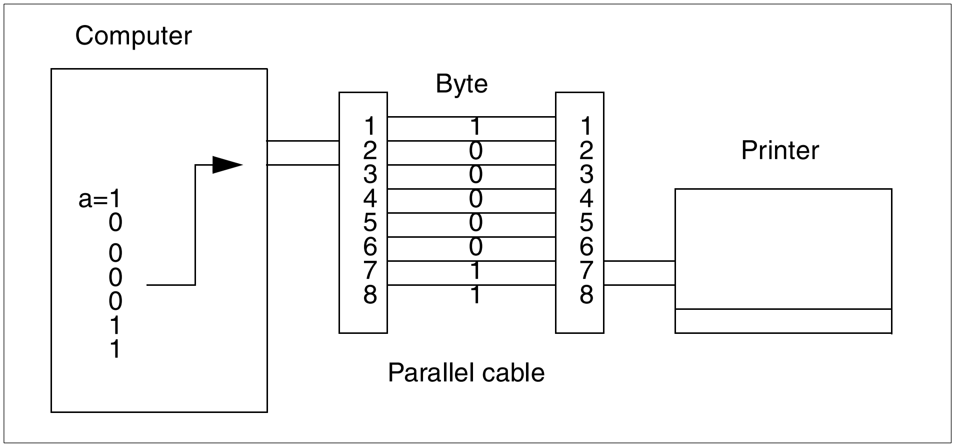


Figure 2-1 Parallel data transmission

A parallel transmission is done on a byte (8 bit) transfer basis, which means a data transmission can be accomplished at extremely high speeds, since all eight bits arrive at their destination at the same instant. This makes it the preferred method for transferring data whenever possible. When distances are short, it may be both feasible and economic to use parallel communications, but some electrical problems arise (caused by capacitance and inductance of the cables) when the distance between source and destination increases, causing data corruption. Also, the number of pins required to transfer data (8 bits + 1 for ground) and all other control signals, could make the wiring become too expensive and complex to deal with for greater distances. This limits this technique to a distance no longer than 3 - 4.5 meters, which restricts its use to devices (such as printers) close to the computer.

Note: The latest standard, known as IEEE-1284, describes an improved interface and cable that can go up to 10 meters; however, the most widely used computer parallel ports (usually called CENTRONICS) follow the distance restrictions described in this section.

Serial

A serial communication requires only a single pin or wire to send the same data character to the device (in this case, a printer). To accomplish this task, the data is converted from a parallel form (sent by the computer) to a sequential form, where bits are organized one after the other in a series. The data is then transmitted to the device with the least significant bit (or zero-bit) sent first. Once received by the remote device, the data is converted back into parallel form. The Figure 2-2 shows the serial transmission of the letter “a”.

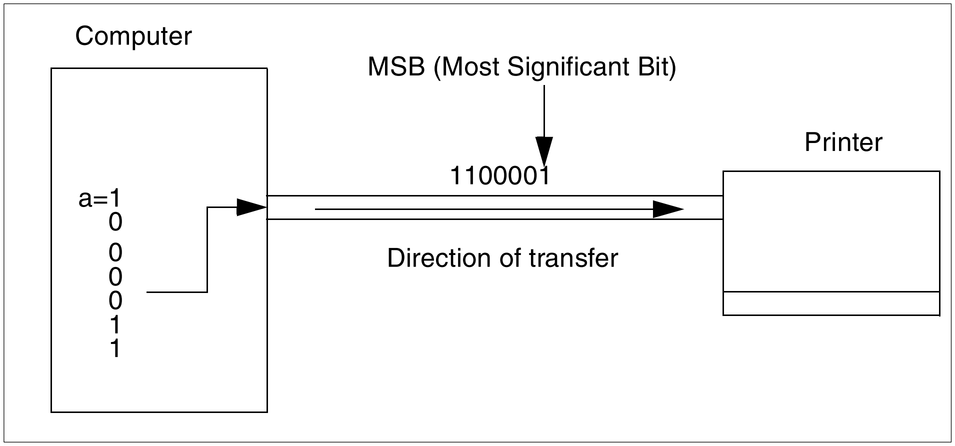


Figure 2-2 Serial data transmission

Serial transmissions of a single character are simple and straightforward; however, complications arise when a large number of characters are transmitted in series. The receiving system does not know where one character ends and the other begins. To solve this problem, both ends of the communication link must be synchronized or timed.

2.1.2 Synchronous and asynchronous communication

Serial data packets usually are *not* sent in uniform rates. The most common situation is a burst of regular spaced serial packets, followed by a pause, then another stream of serial data packets, and so on. Synchronization is the name used for the techniques required to make the receiving end knows exactly when to start and stop reading the bits considered as real data. When a failure in synchronization happens, the received data is corrupted or lost. There are two basic methods used to ensure correct synchronization.

Synchronous

Synchronous systems use separate physical channels (connector pins and cable wires most of the time) to transmit data and synchronization information. The timing channel transmits clock pulses to the receiver end. The receiver only reads data after receiving this clock pulses identified as *syn* pulses. So the entire communication is controlled entirely by the transmitter which originates both data and syn.

Figure 2-3 shows an example of a synchronous transmission.

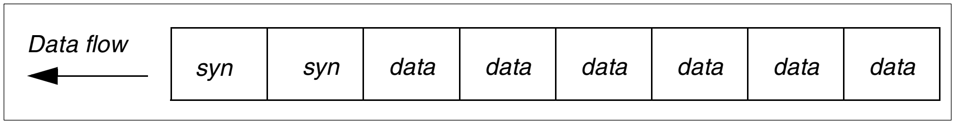


Figure 2-3 Synchronous communication

Asynchronous

Asynchronous systems do *not* use a separate channel for transmitting the synchronization information. Because of that, both transmitter and receiver must set up, before actual communication happens, at what speed data communication will be established and also the necessary parameter configuration for correctly identifying the beginning and ending of serial data packets.

In the most common serial protocol, data is sent in small packets of 10 bits, eight of which are real data information. The signal voltage corresponding to logic *1* indicates the channel is idle. Data packets always starts with logic *0* (the start bit). The start bit triggers an internal timer in the receiver that generates the needed clock pulses. Following the start bit, eight bits of message data are sent bit by bit at the agreed speed rate. The packet is concluded with a parity bit and stop bit. Figure 2-4 on page 23 illustrates a complete example packet.

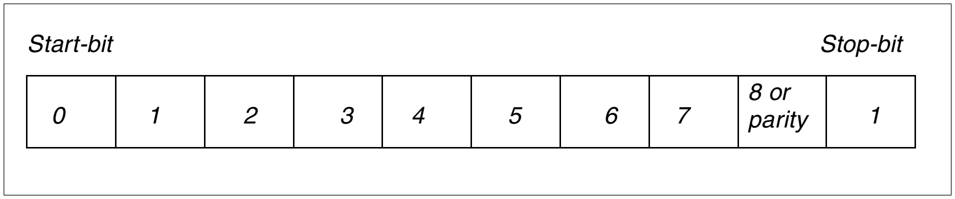


Figure 2-4 Synchronous communication

2.1.3 Serial communications terminology

The following sections explain the terminology used in serial communications.

Bits per character

The number of bits per character (BPC) indicates the number of bits used to represent a single data character during serial communication. This number does not reflect the total amount of parity, stop, or start bits included with the character. Two possible settings for BPC are 7 and 8.

When using the seven bits-per-character setting, it is possible to only send the first 128 characters (0-127) of the standard ASCII character set. Each of these characters is represented by seven data bits. The eight bits-per-character setting must be used to send the ASCII extended character set (128-255). Each of these characters may only be represented using eight data bits.

Bits per second

Bits per second (BPS) is the number of data bits (binary 1s and 0s) that are transmitted per second over the communication line.

Baud rate

The baud rate is the number of times per second a serial communication signal changes states; a state being either a voltage level, a frequency, or a frequency phase angle. If the signal changes once for each data bit, then one bps is equal to one baud. For example, a 300 baud modem changes its states 300 times a second.

Parity

The parity bit, unlike the start and stop bits, is an optional parameter, used in serial communications to determine if the data character being transmitted is correctly received by the remote device.

The parity bit can have one of the following five values:

none Specifies that the local system must not create a parity bit for data characters being transmitted. It also indicates that the local system does not check for a parity bit in data received from a remote host.

even Specifies that the total number of binary 1s, in a single character, adds up to an even number. If they do not, the parity bit must be a 1 to ensure that the total number of binary 1s is even.

odd Operates under the same guidelines as even parity except that the total number of binary 1s must be an odd number.

space Specifies that the parity bit will always be a binary zero. When used for error detection, space will indicate a problem only if the parity bit is not a zero. Another term used for space parity is bit trimming, which is derived from its use as a filler for seven-bit data being transmitted to a device which can only accept eight bit data. Such devices see the space parity bit as an additional data bit for the transmitted character.

mark Operates under the same guidelines as space parity except that the parity bit is always a binary 1.The mark parity bit acts only as a filler.

Start, stop, and mark bits

The start and stop bits are used in asynchronous communication as a means of timing or synchronizing the data characters being transmitted. Without the use of these bits, the sending and receiving systems will not know where one character ends and another begins.

Another bit used to separate data characters during transmission is the mark (or idle) RS bit. This bit, a binary 1, is transmitted when the communication line is idle and no characters are being sent or received. When a start bit (binary 0) is received by the system, it knows that character data bits will follow until a stop bit (binary 1) is received.

Table 2-1 shows explains the relationship among these bits:

Table 2-1 Start, stop, and mark bits

Example 1	Example 2	Example 3	Definition
1	1	1	Start bit
7	7	8	Data character
1	0	0	Parity setting
1	2	1	Stop bit

Most modern serial communication devices such as modem and terminals use a 10-bit transmission character.

A widely used configuration for serial communications is known as *9600 8N1*, which stands for:

- 9600 is the baud rate between serial devices.
- 8 is the number of data bits.
- N indicates no parity.
- 1 indicates 1 stop bit.

2.1.4 EIA-232 Standard

In 1969, the EIA (Electronic Industry Association), Bell Labs and others, established a standard for interfacing terminals and data communication equipment through public shared telephony networks (PSTNs)–*RS-232-C* (Recommended Standard number 232 - revision C). The purpose was simplifying the interconnections of equipment from different manufacturers. This standard defined electrical, mechanical, and functional characteristics, such as voltage levels, cable impedance, pin number assignments and functions of the different electrical signals to be used.

Since the date of the standard introduction, the EIA has been publishing modifications: the latest version is the TIA/EIA-232-F. Besides changing the name from RS-232, some signal lines were renamed and various others were defined.

Note: You can purchase a full copy of the updated standard, found at:

`http://www.tiaonline.org/standards/`

In 1973, three new standards were proposed in order to overcome some limitations of RS-232-C in facing the accelerating rate of technological change. They are *RS-422*, *RS-423*, and *RS-449*.

These new standards had to achieve the following goals:

- Maintain compatibility with RS-232-C.
- Support a higher signaling rate over longer distances.
- Add an interface circuit to perform functions like loop-back testing.
- Solve mechanical interface problems caused from the lack of connector specifications in the old standard.
- Improve electrical characteristics by providing for balanced circuits.

Even after these improved standards publications, the TIA/EIA-232 standard continued to be widely used on the most different data communications applications, because of its simplicity, low cost, and easy implementation.

The major idea of the EIA-232 interface is described by its own title: *Interface Between Data Terminal Equipment and Data Communications Equipment Employing Serial Binary Data Interchange.*

DTE and DCE

The two terms DTE (data terminal equipment) and DCE (data communication equipment) are derived from the terminology used by the telephone company, as shown in Figure 2-5.

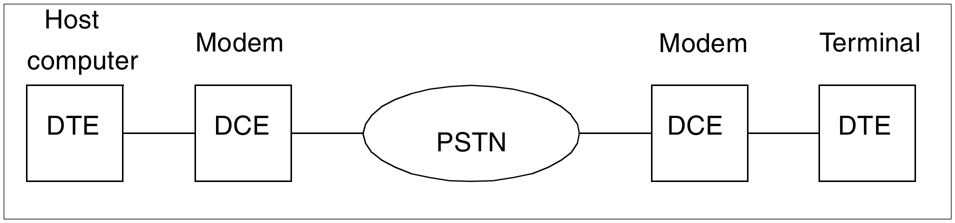


Figure 2-5 DTE and DCE

A DTE is a device that sends data from the transmission data signal (TD[1]). Host computers, terminals, and printers are classified as DTE devices.

A DCE is a device that sends data from the receive data signal (RD[2]). The most common DCE devices are modems and multiplexers.

DTE and DCE pin assignments

Figure 2-6 on page 27 explains the signal pin name and its usage between DCE and DTE defined in the EIA-232 standard.

.

Note: Although EIA232 standards define 25 pins for serial communication, most devices have only 22 pins. In typical usages, serial devices use only 8 pins, therefore many serial devices are equipped with 9 pin serial ports instead of 25 pin.

[1] TD signal uses the second pin on 25 pin serial ports (see Figure 2-6 on page 27).

[2] RD signal uses the third pin on 25 pin serial ports (see Figure 2-6 on page 27).

					Circuit	
PIN	NAME	TO DTE	TO DCE	FUNCTION	CCITT	EIA
01	FG			FRAME GROUND	101	(AA)
02	TD		→	TRANSMITTED DATA	103	(BA)
03	RD	←		RECEIVED DATA	104	(BB)
04	RTS		→	REQUEST TO SEND	105	(CA)
05	CTS	←		CLEAR TO SEND	106	(CB)
06	DSR	←		DCE READY	107	(CC)
07	SG			SIGNAL GROUND	102	(AB)
08	DCD	←		DATA CARRIER DETECT	109	(CF)
09		←		POSITIVE DC TEST VOLTAGE		
10		←		NEGATIVE DC TEST VOLTAGE		
11	QM	←		EQUALIZER MODE	BELL	208A
12	(S)DCD	←		SEC. DATA CARRIER DETECT		
				DATA RATE SELECTION	122	(SCF/CI)
13	(S)CTS	←		SEC. CLEAR TO SEND	121	(SCB)
14	(S)TD		→	SEC. TRANSMITTED DATA	118	(SBA)
	NS		→	NEW SYNC	BELL	208A
15	TC	←		TRANSMITTER CLOCK (DCE)	114	(DB)
16	(S)RD	←		SEC. RECEIVED DATA	119	(SBB)
	DCT	←		DIVIDED CLOCK TRANSMITTER	BELL	208A
17	RC	←		RECEIVER CLOCK (DCE)	115	(DD)
18	DCR	←		DIVIDED CLOCK RECEIVER	BELL	208A
			→	LOCAL LOOPBACK	141	(LL)
19	(S)RTS		→	SEC. REQUEST TO SEND	120	(SCA)
20	DTR		→	DTE READY	108.2	(CD)
21	SQ	←		SIGNAL QUALITY DETECT	110	(CG)
			→	REMOTE LOOPBACK	140	(RL)
22	RI	←		RING INDICATOR	125	(CE)
23			→	DATA RATE SELECTOR	111	(CH)
		←		DATA RATE SELECTOR	112	(CI)
24	(TC)		→	EXT. TRANSMITTER CLOCK (DTE)	113	(DA)
25			→	BUSY ON DIAL MODEMS	BELL	113B
		←		TEST MODE	142	(TM)

Figure 2-6 DTE and DCE pin assignments for 25 pin connectors

Flow control

Serial devices, such as printers and modems, do not process data as quickly or efficiently as the computers they are connected to. Some type of data flow control is needed by the serial device to limit the amount of data transmitted by the system. The term *flow control* is used to describe the method in which a serial device controls the amount of data being transmitted to itself. The three types of flow control discussed in this section are:

- XON/XOFF (software flow control)

  Transmitter on and transmitter off (XON/XOFF) flow controls involves the sending of data transmission control characters along the data stream (TD and RD). For this reason, it is referred to as software flow control.

- RTS/CTS (hardware flow control)

  Ready to send/clear to send (RTS/CTS) is sometimes called pacing or hardware handshaking instead of flow control. The term hardware handshaking comes from the use of cabling and voltages as a method of data transmission control. Unlike XON/XOFF, which sends control characters in the data stream, RTS/CTS uses dedicated pins. A positive voltage means data transmission is allowed, and a negative voltage signifies that data transmission should be suspended.

- DTR/DSR (hardware flow control)

  Data terminal ready (DTR), another form of hardware flow control, is normally generated by the devices, such as printers to indicate that they are ready to communicate with the system. This signal is used in conjunction with data set ready (DSR) generated by the system to control data flow. A positive voltage means data transmission is allowed while a negative voltage signifies that data transmission should be suspended.

  > **Note:** Most operating systems, including AIX, do not support the flow control using DTR/DSR.

2.1.5 Serial port cabling

Depending on the purpose of connecting the serial devices, there are several types of signal cabling (physical wiring) on EIA-232 serial communications. We explains the following common two types of cabling in this section:

- “DTE to DCE communication using a straight cable” on page 29
- “DTE to DTE communication using a null-modem cable” on page 30

DTE to DCE communication using a straight cable

This section explains how a straight cable is commonly used when you connect a modem (DCE) to a serial port of an AIX host (DTE). There are seven or more LEDs on the modem, which are typically labeled as signal pin names, as shown in the following example:

```
DTR DSR RTS CTS DCD TXD RXD
```

A “straight cable” means that all the signal pins are directly wired between DTE and DCE, as shown in Figure 2-6 on page 27. Although EIA-232 standards defines 22 pins, usually only eight pins (including the ground signal pin) are required, as shown in Figure 2-7. Therefore, if you see an LED named DSR on the modem, you can assume that the signal line is also wired to the DTE side as a DSR signal line.

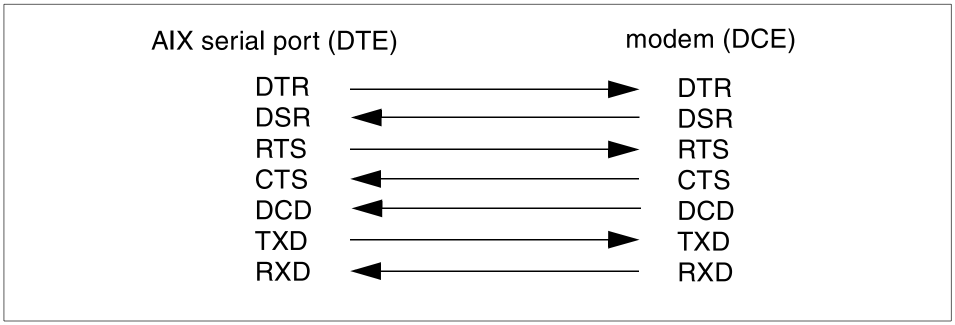


Figure 2-7 Straight serial cable connection

We explain these signal pins’ usage as follows:

- DTR (Data Terminal Ready) is asserted by a serial port when an application program opens the serial port.
- DSR (Data Set Ready) is asserted by a modem when the modem is powered on and ready to work. To send data to the TTY device from the application program, DSR must be asserted by the modem first.
- DCD (Data Carrier Detect) is asserted by the modem when it establishes a connection with remote modem. By default, the assertion of DCD is also required to send data to the serial port, unless you set the clocal option on the TTY device. The application program dials by sending an AT command to the modem, which usually sets the clocal option dynamically. However, an application program that accepts an incoming connection, such as getty, will not set clocal and will wait until DCD is asserted by the modem. Once DCE is asserted, it means a new inbound connection is coming in, and then the getty process sends a login prompt to the serial port.

- RTS is asserted by a serial port when it is ready to receive data. CTS is asserted by a modem when it is ready to receive data. These two signal lines are only used for hardware flow control, which is explained in “Flow control” on page 28.
- A serial port uses TXD to send data to a modem. A modem uses RXD to send data back to a serial port.

DTE to DTE communication using a null-modem cable

When connecting a DTE device to another DTE device, you must make a circuit that replaces the modems and phone network. This connection is performed by a cable that provides required signal transmission without using modems (DCE devices). Therefore, this type of cable is called a *null-modem cable*[3]. Figure 2-8 illustrates the basic concept of the null-modem connection.

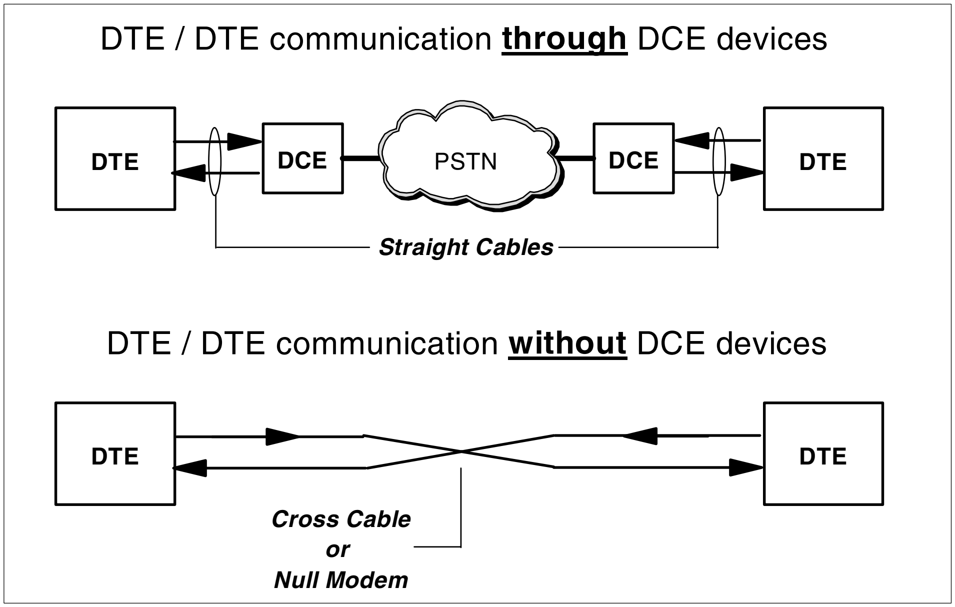

DTE / DTE communication through DCE devices
DTE
DCE
PSTN
DCE
DTE
Straight Cables
DTE / DTE communication without DCE devices
DTE
DTE
Cross Cable
or
Null Modem

Figure 2-8 Basic concept of null-modem connection

Unfortunately, EIA-232 standards do not define the signal cabling for the null-modem connection, therefore the required signal cabling methods depend on what DTE devices are connected.

To simplify this issue, we start to explain an example that connects two AIX systems with modems over PSTN. In this case, the required straight cabling is shown in Figure 2-9 on page 31.

[3] A null-modem cable is sometimes referred to as a cross cable.

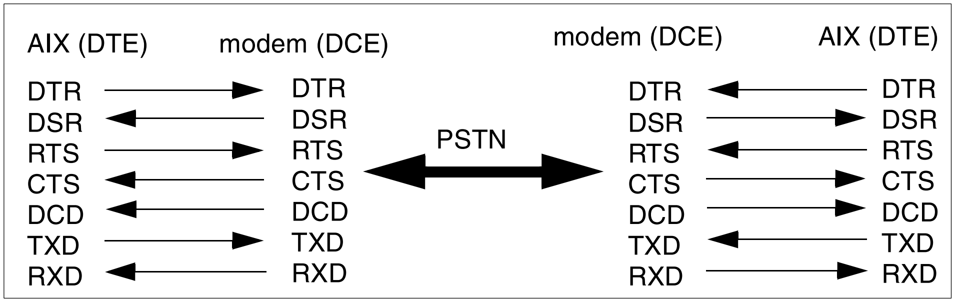


Figure 2-9 AIX to AIX connection over modems

As you can see in Figure 2-9, both AIX systems assert DTR to tell peer (modem) their readiness and expect that DSR is asserted by peer (modem) to know peer's readiness.

However, if you connect two AIX systems directly, you cannot wire DTR to DTR or DSR to DSR directly. You have to connect the DTR to peer's DSR and the DSR to peer's DTR, so that the assertion of DTR is accepted by peer as a DSR signal, and vice versa. Same wiring is required for the RTS/CTS pair and the TXD/RXD pair. But what should manage DCD?

In the sense that the null-modem connection emulates a firmly established modem connection, DCD should be always asserted during connection, but by what? One idea is that to jump from DSR to DCD, so that DCD is always asserted by peer while peer is asserting its DTR.

Figure 2-10 illustrates the required null-modem signal wiring explained in this section.

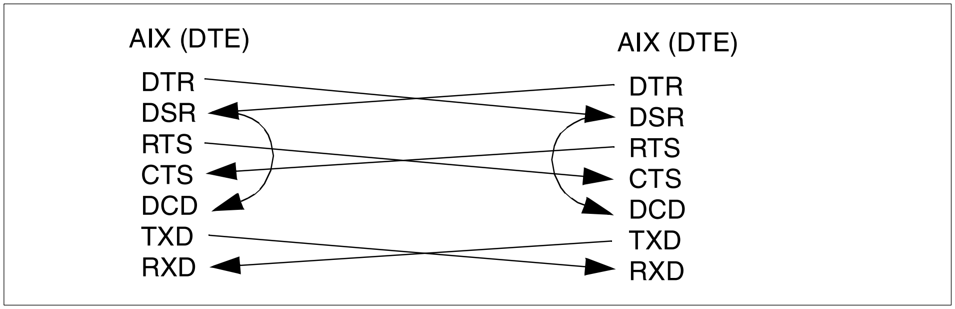


Figure 2-10 Example of null-modem cable connection

Note: The null-modem cabling has several variations, depending on the connected DTE devices. We recommend you label each null-modem cable to identify the environment that you confirm will work properly.

Verifying the null-modem connection

You can use the **stty** command to test the null-modem connection using the following steps:

1. Connect the serial ports between two AIX systems. In this example, we assume that you connect tty1 on svr01 and tty1 of host svr03 using a null-modem cable.
2. Invoke the **stty** command on svr01:

   ```
   root@svr01:/ # stty < /dev/tty1
   ```

   The command hangs waiting for an input from tty1.
3. Then invoke the **stty** command on svr03:

   ```
   root@svr03:/ # stty < /dev/tty1
   ```

   As soon as you have entered this command, the hanged **stty** command on svr01 displays the TTY attributes, as shown in the following example, as well as the **stty** command on svr03:

   ```
   root@svr01:/ # stty < /dev/tty1
   speed 19200 baud; -parity hupcl
   eol2 = ^?
   brkint -inpck -istrip icrnl -ixany ixoff onlcr tab3
   echo echoe echok
   ```

2.2 Managing serial ports

This section explains how to manage serial ports on AIX. Figure 2-11 on page 33 provides a big picture that illustrates the relationship among the various software components used in serial connections on AIX.

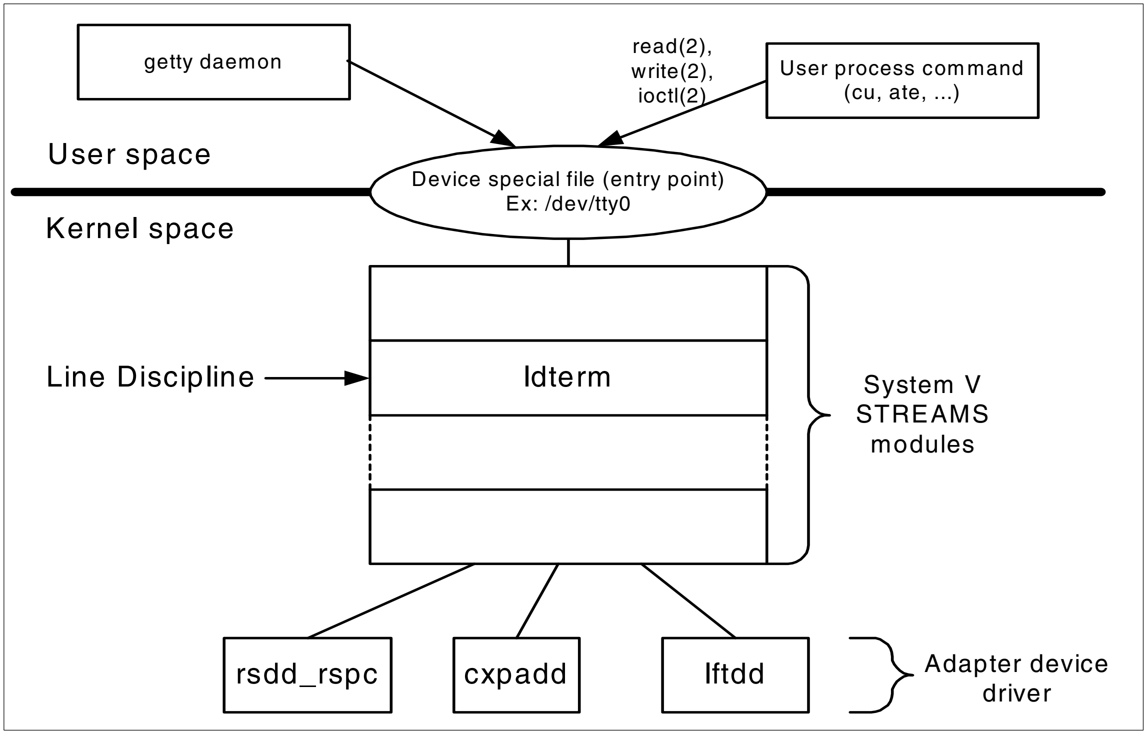


Figure 2-11 Software components used in serial connections on AIX

2.2.1 Components

Here we explain the major concepts used in the serial connections environment from an AIX perspective.

Serial adapter devices

The physical serial devices are the actual *pieces of hardware* used by the host computer for serial connections. The currently available IBM @server pSeries server models have at least one integrated serial port (native serial port). If you have to add serial ports, you can use asynchronous PCI adapters.

The currently available asynchronous adapters are:

- 8-Port Async Adapter, EIA-232/422 (PCI) - FC 2943
- 128-Port Async Controller (PCI) - FC 2944

Once configured, native serial ports and asynchronous PCI adapters are shown as a *sa device* (/dev/sa*X*, where X is the instance number of device) on AIX, as shown in Example 2-1 on page 34.

Example 2-1 Listing serial port adapters

```
# lsdev -Cc adapter | grep sa
sa0     Available 01-S1    Standard I/O Serial Port
sa1     Available 01-S2    Standard I/O Serial Port
sa2     Available 01-S3    Standard I/O Serial Port
sa3     Available 10-70    IBM 8-Port EIA-232/RS-422A (PCI) Adapter
```

In Example 2-1, you see three native serial port devices (sa0, sa1, and sa2) and a serial port adapter device (sa3) configured for the IBM 8-Port EIA-232/RS-422A (PCI) Adapter, which is inserted in the PCI bus slot 10-70 of the IBM @server pSeries 610 Model 610.

Please note that you have eight serial ports, 0 - 7, for an 8-port asynchronous adapter, because the adapter is connected by an 8-port asynchronous cable with fanout box, as shown in Figure 2-12.

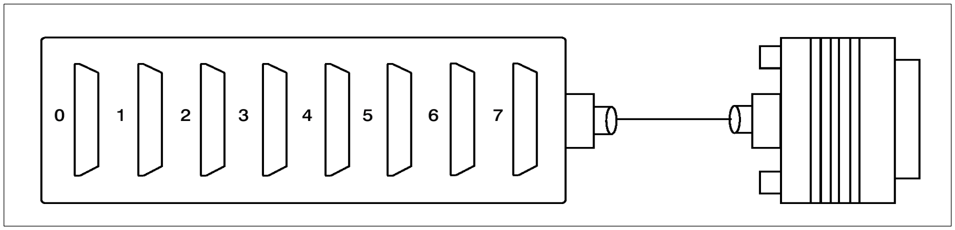


Figure 2-12 8-port asynchronous cable with fanout box

If you use an asynchronous adapter to configure additional serial ports, you have to insert the adapter first, and then install the required device driver.

For further information about asynchronous adapters, please consult with the following publications shipped with the adapter:

- *8-Port Asynchronous PCI Adapter Installation and User's Guide*, SA23-2562
- *128-Port Asynchronous PCI Adapter Installation and User's Guide*, SA23-2563

Serial adapter device drivers

The adapter device driver is a software component in the operating system kernel that handles the physical adapter device. AIX provides several serial adapter device drivers, as shown in Figure 2-11 on page 33, including:

rsdd_rspc Integrated native serial port adapter device driver

cxpadd PCI 128-port and 8-port adapter device driver

These device drivers are provided in separate filesets.

The following example shows how to confirm the device driver name and its fileset for a specific serial adapter device:

```
# lsdev -Cc tty | grep tty3
tty3 Available 10-70-01-00 Asynchronous Terminal
# lsdev -C -l tty3 -F parent
sa3
# lsdev -C -l sa3
sa3 Available 10-70 IBM 8-Port EIA-232/RS-422A (PCI) Adapter
# odmget -q name=sa3 CuDv

CuDv:
        name = "sa3"
        status = 1
        chgstatus = 2
        ddins = "pci/cxpadd"
        location = "10-70"
        parent = "pci0"
        connwhere = "112"
        PdDvLn = "adapter/pci/4f111100"
# odmget -q uniquetype="adapter/pci/4f111100" PdDv

PdDv:
        type = "4f111100"
        class = "adapter"
        subclass = "pci"
        prefix = "sa"
        devid = "0x4f111100"
        base = 1
        has_vpd = 1
        detectable = 1
        chgstatus = 0
        bus_ext = 0
        fru = 1
        led = 1670
        setno = 152
        msgno = 22
        catalog = "devices.cat"
        DvDr = "pci/cxpadd"
        Define = "/usr/lib/methods/define_rspc"
        Configure = "/usr/lib/methods/cfgcxma"
        Change = "/usr/lib/methods/chggen_rspc"
        Unconfigure = "/usr/lib/methods/ucfgcxma"
        Undefine = "/usr/lib/methods/undefine"
        Start = ""
        Stop = ""
        inventory_only = 0
        uniquetype = "adapter/pci/4f111100"
# ls -l /usr/lib/drivers/pci/cxpadd
```

```
-r-xr-xr-x   1 root     system       262822 Mar 13 20:28
/usr/lib/drivers/pci/cxpadd*
# lslpp -w /usr/lib/drivers/pci/cxpadd
  File                                        Fileset               Type
  ----------------------------------------------------------------------------
  /usr/lib/drivers/pci/cxpadd
                                    devices.pci.4f111100.com        File
# lslpp -L devices.pci.4f111100.*
  Fileset                      Level  State  Type  Description (Uninstaller)
  ----------------------------------------------------------------------------
  devices.pci.4f111100.asw   5.1.0.0    C     F    PCI 8-Port Asynchronous
Adapter
                                                   Software
  devices.pci.4f111100.com  5.1.0.25    C     F    Common PCI Asynchronous
Adapter
                                                   Software
  devices.pci.4f111100.diag
                            5.1.0.15    C     F    RISC PC PCI Async 8 Port
Adapter
                                                   Diagnostics
  devices.pci.4f111100.rte  5.1.0.25    C     F    PCI 8-Port Asynchronous
Adapter
                                                   Software
```

In this example, the TTY device tty3 is configured on the serial adapter device sa3. The device driver name of sa3 device is pci/cxpadd, and this driver is included in the fileset devices.pci.4f111100.com. The four filesets, devices.pci.4f111100.*, comprise the required device support for 8-Port Async Adapter, EIA-232/422 (PCI) - FC 2943.

TTY devices

To connect serial devices to AIX systems, you have to configure TTY devices on serial ports. TTY devices are configured as /dev/ttyX, where X is the device instance number. Application programs can send data to or receive data from the serial ports, which are connected to serial devices, by opening TTY devices. A TTY device is represented by a character device file, as shown in Example 2-2.

Example 2-2 A TTY device special file

```
# ls -l /dev/tty0
crw--w--w-   1 root     system    18,  0 Mar 27 10:46 /dev/tty0
```

Each configured TTY device has many attributes. We explain some frequently referenced attributes in Table 2-2 on page 37.

Table 2-2 Common TTY device attributes

SMIT attributes	Purpose	Values
Port number	Specific port number of the serial device used.	0 to N-1 (N is the number of ports on the device).
Enable login	Depends on the type of communication being established (DTE/DCE; DTE/DTE, modem, and so on).	Enable / Disable (default) / Share / Delay.
Baud rate	Defines the baud rate to be used on the host side. Must match the value of the other device.	0 to 230000. Default value is 9600.
Parity	Type of parity to be used.	None (default) / Even / Odd / Mark.
Bits per character	Number of bits per character.	5 / 6 / 7 / 8 (default).
Number of stop bits	Number of stop bits.	1 (default) / 2.
Terminal type	Defines the terminal parameters to be used. Equivalent to AIX variable TERM.	dumb (default) / vt100 / vt220 / ibm3161 / etc.
Flow Control	Type of flow control.	Xon (default) / RTS / Ixany / None.

STREAMS

STREAMS is a kernel mechanism that supports the development of network services and data communication drivers. It defines interface standards for character input and output within the kernel, and between the kernel and user level. The STREAMS mechanism comprises integral functions, utility routines, kernel facilities, and a set of structures.

On TTY devices, the line discipline *ldterm* is loaded as a default STREAM module.

The getty daemon

The getty daemon is responsible for setting and managing serial ports. It is individually invoked by the init process for each TTY device. Therefore, each configured TTY device has an entry in /etc/inittab, as shown in the following example:

```
# grep tty /etc/inittab | grep -v console
tty0:2:off:/usr/sbin/getty /dev/tty0
tty1:2:respawn:/usr/sbin/getty /dev/tty1
```

```
tty2:2:off:/usr/sbin/getty /dev/tty2
tty3:2:respawn:/usr/sbin/getty /dev/tty3
```

The third field (action field) is determined by the Enable LOGIN parameter value of the TTY device, as shown in Table 2-3.

Table 2-3 Relationship between Enable LOGIN and getty daemon

Enable LOGIN parameter	inittab corresponding entry
Disable	ttyX:2:off:/usr/sbin/getty /dev/ttyX
Enable	ttyX:2:respawn:/usr/sbin/getty /dev/ttyX
Share	ttyX:2:respawn:/usr/sbin/getty -u /dev/ttyX
Delay	ttyX:2:respawn:/usr/sbin/getty -r /dev/ttyX

The behavior of getty regarding the four possible configurations of the tty ENABLE Login parameter are described as follows:

- No login prompt will be issued from this host (Enable LOGIN=off)

  This is the default value when creating a tty device, and should be used in situations, such as when you want to configure a device for receiving a login prompt, simulating an asynchronous terminal function. We also strongly recommended you use this configuration for dial-out through the configured tty device.

- A login prompt will be issued from this host (Enable LOGIN=enable)

  When you want the host to spawn a getty on the configured port, you must enable the Enable LOGIN parameter. In this case, the login prompt is displayed as soon as the DCD pin is up (logical 1).

- Connection Over a Modem (Enable LOGIN=share)

  With the tty's Enable LOGIN field set to share, the **/etc/getty** command, which controls the login process, will start up on the port and wait for the modem to turn on a carrier signal. When the modem turns the carrier signal on, the **getty** command attempts to lock the port so that no other processes can use it. If another program (for example, **cu** or **ate**) has already locked the port for dial out, the **getty** command waits until the port is freed by that process before attempting to open it again. If the port is not locked, the **getty** command locks it and sends out a login herald, thus allowing a login to take place on the port.

- Direct Connect (Enable LOGIN=delay)

  With the tty's Enable LOGIN field set to delay, after DCD assertion, /etc/getty waits until the remote system sends a character to the local tty before locking the port and sending a login herald.

This attribute is useful for connections between two hosts that may use BNU commands (like **cu**) for communications in either direction. In this, case, delay allows **cu** to be started on either side of the connection. The **cu** command sends a few carriage returns through, which causes the getty on the other end to send the login screen. When that cu session has ended, then a cu session can be started on the other host without having to change the getty attribute.

2.2.2 Life cycle of a tty session

If you configure a TTY with Enable LOGIN enabled, a getty process is spawned by the init process for that TTY device. In Example 2-3, because the action field of the tty0 entry in /etc/inittab is specified as respawn, init restarts getty after it exits.

Example 2-3 Login enable configuration for tty0

```
# grep tty0 /etc/inittab
tty0:2:respawn:/usr/sbin/getty /dev/tty0
# ps alxww | head -1; ps alxww | grep tty0
F S      UID     PID    PPID   C PRI NI ADDR  SZ  RSS   WCHAN    TTY  TIME
CMD
240001 A        0 14722     1   0  60 20 1f07 760  780   EVENT      -  0:00
/usr/sbin/getty /dev/tty0
```

In this example, notice that the PPID field of the **ps** command output is 1. It means that getty's parent process is init. The TTY field of the output is '-', which means that the TTY device is not ready to use. It also means that DSR or DCD or both are not asserted by peer.

When a new connection comes up, DSR and DCD are asserted, getty becomes a login and sends a login prompt to the TTY device. If a user enters a valid user name and password, the user is authenticated, and then the login process calls exec() to spawn the user's login shell process.

Example 2-4 shows a login session example from a TTY device. You can see the process ID and TTY device of getty in Example 2-3 are same ones in the login shell shown in Example 2-4.

Example 2-4 Remote login section through serial connection

```
AIX Version 5
(C) Copyrights by IBM and by others 1982, 2000.
login: root
root's Password:
*******************************************************************************
*                                                                             *
*                                                                             *
*  Welcome to AIX Version 5.1!                                                *
*                                                                              *
*                                                                              *
*  Please see the README file in /usr/lpp/bos for information pertinent to     *
*  this release of the AIX Operating System.                                   *
*                                                                              *
*                                                                              *
********************************************************************************
Last unsuccessful login: Fri Mar 29 13:44:02 CST 2002 on /dev/pts/0 from
shochst
e1.itsc.austin.ibm.com
Last login: Sat Mar 30 08:18:22 CST 2002 on /dev/tty0

[YOU HAVE NEW MAIL]
root@svr01:/ [284] # tty
/dev/tty0
root@svr01:/ [285] # echo $$[4]
14722
```

You can terminate this session using the following two methods:

- Exit from the login shell

  The login shell close the TTY device, which stops DTR assertion. The connection peers receive the signal down and start the disconnection process.

- Disconnected by peer

  If the peer stops assertion of DCD, or DCD and DSR, the TTY device driver of the serial port raises SIGHUP (hang-up signal), a process that opens the device, typically a login shell. The login shell receives the signal and starts logout process.

Figure 2-13 on page 41 shows the life cycle of a TTY session.

[4] The shell variable $$ holds the current process ID.

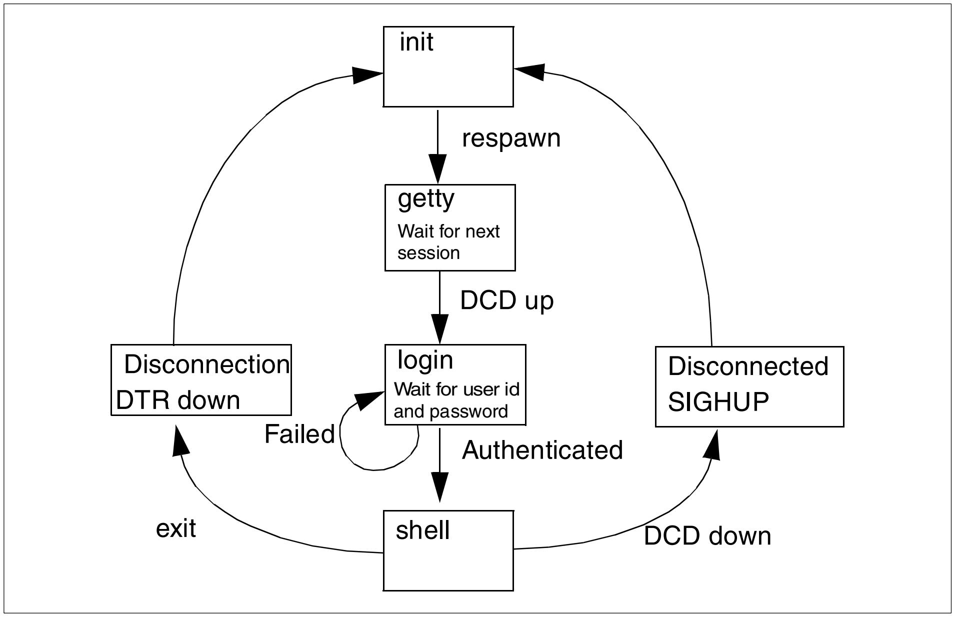


Figure 2-13 Life cycle of a tty session

2.2.3 Serial port configuration

This section explains how to configure and administer serial ports on AIX.

Identifying serial ports

Before configuring serial ports, you have to identify on what serial ports you are going to connect your serial device. To identify the adapter, see “Serial adapter devices” on page 33.

Configuring TTY devices

After verifying the installed serial port adapters, you can configure TTY devices using the following steps. In this example, we assume that you are going to configure a TTY device on port 0 of the 8-port asynchronous adapter configured as sa3 device, as shown in Example 2-1 on page 34.

1. Run **smitty** and select the following menus:

```
# smitty
    Devices
        TTY
            Add a TTY
```

Select the tty rs232 Asynchronous Terminal option, then select the serial port adapter on which you are going to define a serial port device. In this example, select the sa3 device. You will see the SMIT panel shown in Example 2-5.

Example 2-5 Add a TTY

```
                                  Add a TTY

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

[TOP]                                                    [Entry Fields]
  TTY type                                              tty
  TTY interface                                         rs232
  Description                                           Asynchronous Terminal
  Parent adapter                                        sa3
* PORT number                                          [0]                    +
  Enable LOGIN                                          disable               +
  BAUD rate                                            [9600]                 +
  PARITY                                               [none]                 +
  BITS per character                                   [8]                    +
  Number of STOP BITS                                  [1]                    +
  TIME before advancing to next port setting           [0]                    +
  TERMINAL type                                        [dumb]
  FLOW CONTROL to be used                              [xon]                  +
[MORE...35]

F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         Esc+6=Command       Esc+7=Edit          Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do
```

2. Specify the following default parameters on the SMIT panel, then press Enter:
   - PORT number: Specify the appropriate port number; in this case specify 0
   - Enable LOGIN: disable
   - BAUD rate: 9600
   - PARITY: None
   - BITS per character: 8
   - Number of STOP BITS: 1
   - TERMINAL type: dumb
   - FLOW CONTROL to be used: xon

Note: We customize these parameters in a later phase.

3. You can confirm that the new TTY device, tty3, is configured, as shown in the following example:

   ```
   # lsdev -Cc tty
   tty0 Available 01-S1-00-00 Asynchronous Terminal
   tty1 Available 01-S2-00-00 Asynchronous Terminal
   tty2 Available 01-S3-00-00 Asynchronous Terminal
   tty3 Available 10-70-01-00 Asynchronous Terminal
   ```

   You can also confirm the tty3 entry is added to /etc/inittab, as shown in the following example:

   ```
   # lsitab -a | grep tty3
   tty3:2:off:/usr/sbin/getty /dev/tty3
   ```

Connecting serial devices

After configuring a TTY device, you should physically connect your serial device to the serial port on which the TTY device is configured. As we explained in Section 2.1.5, “Serial port cabling” on page 28, you have to select the appropriate cable to connect your serial device.

Tailoring the TTY device attributes

You should customize the TTY device attributes to meet with your specific connection requirements. Because these attributes vary from device to device, we focus on the serial connections scenarios in this section.

Figure 2-14 on page 44 is a reference for the terminology used to explain the various types of connection scenarios and their correspondent Enable LOGIN parameters (or -a login= on command line) on the TTY device configuration.

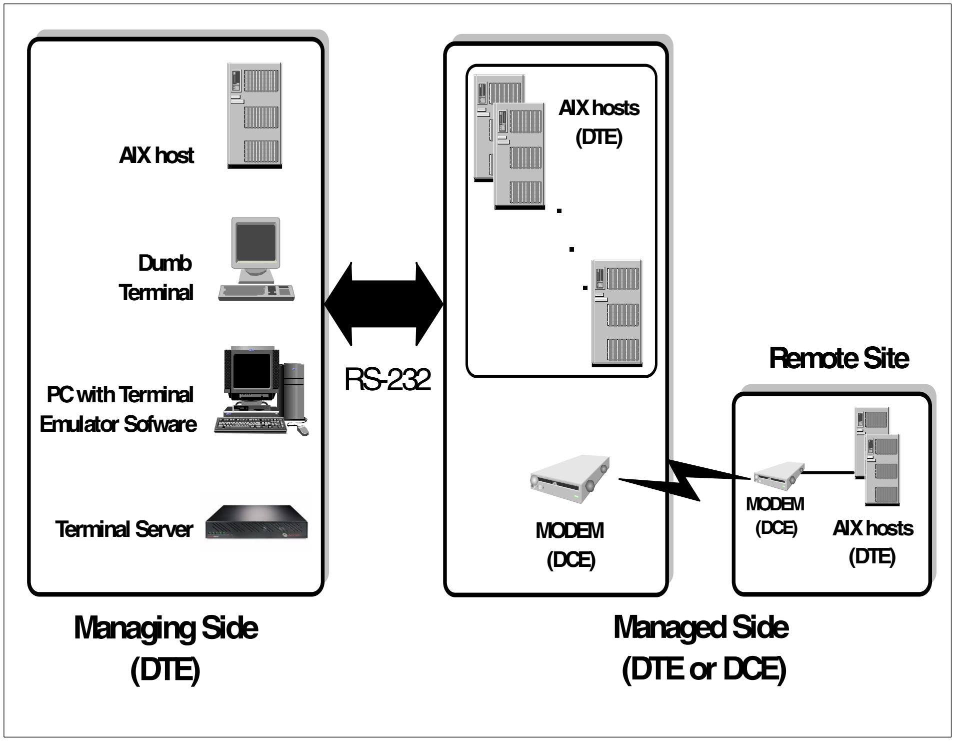


Figure 2-14 Serial device options

As you can see in Figure 2-14, the serial devices, from the perspective of their role in a server farm, are positioned in two distinct categories:

- On the managing side are the devices responsible for gathering information of the actual computer resources of the server farm
- On the managed side are the actual devices to be managed

Note: Although a modem is not a computer resource to be managed, we include it in this category, because you require a set of specific parameters to connect a modem to an AIX system.

As described in Section 2.1.5, “Serial port cabling” on page 28, we believe there are at least five different connection scenarios, as shown in Table 2-4 on page 45. Table 2-4 on page 45 summarizes the required tty ENABLE Login parameter for each scenarios.

Table 2-4 Connection scenarios

Connection Scenarios			**tty ENABLE Login parameter**	
Managing Side	Managed Side	Communication purpose	Managing Side	Managed Side
AIX (DTE)	AIX (DTE)	Receive Login prompt from Managed Server	Disable (Off)	Enable
AIX (DTE)	AIX (DTE)	Full access between hosts in both directions	Delay	Delay
AIX (DTE)	modem (DCE)	Remote host connection - modem may originate and answer calls	Share	N/A
AIX (DTE)	modem (DCE)	Remote host connection - modem only allowed to originate calls	Disable (Off)	N/A
AIX (DTE)	modem (DCE)	Remote host connection - modem only allowed to answer calls	Enable	N/A

2.3 Useful functions and hints

This section explains about some useful functions and hints regarding serial port management on AIX.

2.3.1 Managing asynchronous adapters

If you require many serial connections, you have to use asynchronous adapters. This section provides useful information to administer asynchronous adapters on AIX.

Monitoring asynchronous adapter

This SMIT option allows you to observe the behavior of the control signals on the serial port of an asynchronous adapter. To use this function, do the following:

1. Run **`smitty`** and select the following menus:

   ```
   # smitty
       Devices
           Asynchronous Adapters
               Monitor Async Adapters
   ```

2. Select the adapter you want to monitor and press Enter. You will see the SMIT panel shown in Example 2-6.

Example 2-6 Monitoring asynchronous adapter ports

```
Bus  #: 00          128 Port Asynchronous Subsystem Monitor
Slot #: 70


                           Port 0                         Name: tty3
     +--------------------------------------------------------------+
     |                                                              |
     |                                                              |
     |     TD   RD   RTS  CTS  DSR  CD   DTR  RI  OFC   IFC         |
     |      -    -    +    -    -    -    +   -     -    -          |
     |                                                              |
     +--------------------------------------------------------------+
              KEY: Signal Active = + Inactive = -

  Input Modes :BRKINT:IXANY:IXOFF:IXONA:
 Output Modes :XCASE:ONLCR:TAB2:BS1:
Control Modes :9600 Baud:8 Bits:1 Stop Bits:No Parity:

 Left Arrow Key=Next Port    Right Arrow Key=Previous Port    F12=Loopback Test
 F2=Refresh       F3=Cancel           F8=Image            F10=Exit
```

Example 2-6 shows an example of the control signals for the specific serial port. The current state is dynamically updated so that you can monitor the behavior of these signals. You can change the target serial port to be monitored by using the right and left arrow keys.

Note: On the actual SMIT panel, the + character is displayed as a small filled box indicating that the pin level is high (logical 1).

The stty-cxma command

The **stty-cxma** command, installed in the /usr/lbin/tty directory, is a utility program that sets and displays the configuration information of the serial port configured on PCI 8- and 128-port adapters. It displays device special settings, signal pins' statuses, and all the standard parameters displayed by the normal **stty** command for the TTY device referenced by standard input. Command options can be used to change flow control settings, set transparent print options, force modem control lines, and display other setting parameters.

This command is included in the devices.common.IBM.cx.rte fileset, as shown in the following example:

```
# lslpp -w /usr/lbin/tty/stty-cxma
  File                                        Fileset               Type
  ----------------------------------------------------------------------------
  /usr/lbin/tty/stty-cxma
                                   devices.common.IBM.cx.rte        File
```

Example 2-7 shows an example output of the **stty-cxma** command, which displays all the configuration information of the /dev/tty3 device.

Example 2-7   The stty-cxma command

```
# stty-cxma /dev/tty3
onstr \033[5i offstr \033[4i term
maxcps 100 maxchar 50 bufsize 100
-forcedcd fastcook -altpin -fastbaud (19200)
-rtspace -dtrpace -ctspace -dsrpace -dcdpace
edelay 100
DTR+  RTS+  CTS-  CD-  DSR-  RI-
-aixon astartc = 0x0 astopc = 0x0 -2200flow -2200print
speed 19200 baud; -parity hupcl
eol2 = ^?
brkint -inpck -istrip icrnl -ixany ixoff onlcr tab3
echo echoe echok
```

If you specify -a option on the command line, the **stty-cxma** command displays all the configuration information for this port, as shown in Example 2-8.

Example 2-8   stty-cxma with option -a

```
# stty-cxma -a /dev/tty3
onstr \033[5i offstr \033[4i term
maxcps 100 maxchar 50 bufsize 100
-forcedcd fastcook -altpin -fastbaud (19200)
```

```
-rtspace -dtrpace -ctspace -dsrpace -dcdpace
edelay 100
DTR+  RTS+  CTS-  CD-  DSR-  RI-
-aixon astartc = 0x0 astopc = 0x0 -2200flow -2200print
speed 19200 baud; 0 rows; 0 columns;
eucw 1:1:0:0, scrw 1:1:0:0:
intr = ^C; quit = ^\; erase = ^H; kill = ^U; eof = ^D; eol = ^@
eol2 = ^?; start = ^Q; stop = ^S; susp = ^Z; dsusp = ^Y; reprint = ^R
discard = ^O; werase = ^W; lnext = ^V
-parenb -parodd cs8 -cstopb hupcl cread -clocal -parext
-ignbrk brkint -ignpar -parmrk -inpck -istrip -inlcr -igncr icrnl -iuclc
ixon -ixany ixoff imaxbel
isig icanon -xcase echo echoe echok -echonl -noflsh
-tostop echoctl -echoprt echoke -flusho -pending iexten
opost -olcuc onlcr -ocrnl -onocr -onlret -ofill -ofdel tab3
```

Performance enhancement on asynchronous adapters

IBM has announced a performance enhancement feature addressed by an APAR, IY26815, for the 8-port PCI asynchronous adapters (FC 2943) in AIX 5L Version 5.1. The enhancement allows you to handle interrupt requests on these adapters instead of the default polling approach. After applying[5] the APAR, you can enable this feature using the following steps:

1. Disable the login of TTY devices defined on the adapter using the **`pdisable`** command.
2. Run **`smitty`** and select the following menus:

   ```
   # smitty
       Devices
           Asynchronous Adapters
               IBM 8-Port EIA-232/RS-422A (PCI) Adapter
                   Change/Show Characteristics of a IBM 8-Port EIA-232/RS-422A
   (PCI) Adapter
   ```

3. Select the adapter you want to enable the enhancement. You can select an adapter that has the device description `IBM 8 Port EIA-232/RS 422A (PCI) Adapter`. You will see the SMIT panel shown in Example 2-9.

Example 2-9   Changing characteristics of an asynchronous adapter

```
    Change/Show Characteristics of a IBM 8-Port EIA-232/RS-422A (PCI) Adapter

Type or select values in entry fields.
Press Enter AFTER making all desired changes.
                                                        [Entry Fields]
  Logical Name                                        sa3
  Description                                         IBM 8-Port EIA-232/RS->
```

[5] The system must be rebooted after applying the APAR IY26815.

```
  Status                                                 Available
  Location                                               10-70
  Adapter Uses Interrupts                                enable          +

F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         Esc+6=Command       Esc+7=Edit          Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do
```

4. Change the value disable to enable on the Adapter Uses Interrupts line, as high-lighted in Example 2-9 on page 48 and press Enter.

You can confirm if the performance enhancement function is correctly configured using the **lsattr** command, as shown in the following example:

```
# lsattr -El sa3 |grep intr
intr_level   13         Bus Interrupt Level     False
use_intr     enable     Adapter Uses Interrupts True
```

2.3.2 AIX console logging

AIX sends to a log file, called the console log file (the default console log file is /var/adm/ras/conslog), all the data that is sent to the console device /dev/console. This function is helpful in diagnosing problems that occurred in the system boot phase.

To view the console log file, use the **alog** command as the root user, as shown in the following example:

```
# alog -f /var/adm/ras/conslog -o | more
         0 Sun Mar 24 21:14:42 CST 2002
         0 Sun Mar 24 21:14:42 CST 2002 Saving Base Customize Data to boot disk
         0 Sun Mar 24 21:14:42 CST 2002 Starting the sync daemon
         0 Sun Mar 24 21:14:43 CST 2002 Starting the error daemon
         0 Sun Mar 24 21:14:43 CST 2002 System initialization completed.
         0 Sun Mar 24 21:14:43 CST 2002 Starting Multi-user Initialization
```

Note: We recommend you define an alias for this command, for example:

```
alias dmesg='alog -f /var/adm/ras/conslog -o'
```

Only messages sent to the console can be logged, which means that any other output sent to a device acting as a console, such as command, smitty, and getty outputs are not logged.

For further information about AIX console logging, please refer to the redbook *AIX Version 4.3 Differences Guide*, SG24-2014.

2.3.3 TTY remote reboot

AIX provides a TTY remote reboot function that instructs the AIX kernel to perform a predefined action by typing an remote reboot string[6] on an integrated native serial port. Once the defined remote reboot string is typed on the serial port with the TTY remote reboot enabled, you can control the system that stops responding on the network but is still processing devices interrupts.

Note: Only the integrated serial ports support the TTY remote reboot function.

You can configure the following three actions on the TTY device with TTY remote reboot enabled:

- No Action (default value)
- System reboot
- System dump

To enable this function, do the following:

1. Run **`smitty`** and select the following menus:

   ```
   # smitty
       Devices
           TTY
               Change / Show Characteristics of a TTY
   ```

   Then select the TTY, which you are going to use to enable the TTY remote reboot function. You will see the SMIT panel shown in Example 2-10.

Example 2-10 Change / Show Characteristics of a TTY

```
                   Change / Show Characteristics of a TTY

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

[TOP]                                                   [Entry Fields]
  TTY                                                 tty0
  TTY type                                            tty
  TTY interface                                       rs232
  Description                                         Asynchronous Terminal
  Status                                              Available
  Location                                            01-S1-00-00
  Parent adapter                                      sa0
  PORT number                                        [0]                      +
  Enable LOGIN                                        disable                 +
  BAUD rate                                          [9600]                   +
```

[6] A remote reboot string is a string comprised of an unusual character sequence.

```
  PARITY                                                 [none]            +
  BITS per character                                     [8]               +
  Number of STOP BITS                                    [1]               +
[MORE...35]

F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         Esc+6=Command       Esc+7=Edit          Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do
```

2. Change the value `no` to `reboot` or `dump` on the REMOTE reboot ENABLE line.
3. Enter your remote reboot string on the REMOTE reboot STRING line. Do not use the default value `<#@reb@#>` except for test purposes.
4. Press Enter.

If you set the REMOTE reboot enable parameter to `reboot` or `dump`, once the chosen character sequence is typed on the terminal attached to the TTY device, the character sequence will be erased and a prompt (>) will be presented. You have the following two options at this prompt:

- Press 1[7] on the keyboard

  Instructs the AIX kernel to perform a defined action (reboot or dump).

- Press any other key

  The typed remote reboot string reappears on the screen as if you just typed it now. Therefore, nothing will happen, and the current session on this terminal will continue to be available.

Once a TTY remote reboot is performed, an error log entry is logged, as shown in Example 2-11.

Example 2-11   TTY remote reboot error log entry

```
LABEL:          TTY_RRB
IDENTIFIER:     1960E672

Date/Time:       Wed Mar 27 11:31:01 CST
Sequence Number: 101
Machine Id:      000681734C00
Node Id:         svr01
Class:           O
Type:            INFO
Resource Name:   Remote Reboot

Description
SYSTEM REBOOTED USING TTY REMOTE REBOOT.
```

[7] Use the key on the full keypad, not the numeric keypad.

```
User Causes
SYSTEM REBOOTED USING TTY REMOTE REBOOT.

Detail Data
TTY LOGICAL NAME
tty0
```

It is not possible to enable a password-protected reboot string, since this would require the code checking the password to use the crypt() function. The code checking the string is running at the highest interrupt priority, so any increase in the time taken to service the interrupt may cause other device interrupts to be lost with unpredictable results.

The reboot string is stored as the value attribute in the CuAt ODM class file, as shown in the following example:

```
# odmget -q 'attribute=reboot_string AND name=tty0' CuAt

CuAt:
        name = "tty0"
        attribute = "reboot_string"
        value = "reboot"
        type = "E"
        generic = "DU"
        rep = "s"
        nls_index = 15
```

Because the CuAt ODM class file has the following permission mode, once a user has a chance to log in to the system, they might know the reboot string:

```
# ls -l /etc/objrepos/CuAt
-rw-r--r--   1 root     system        12288 Apr 07 16:18 /etc/objrepos/CuAt
```

Therefore, it is your responsibility to provide physical security on any serial ports with remote reboot enabled.

For further information about TTY remote reboot, please refer to the redbook *AIX Version 4.3 Differences Guide*, SG24-2014.

2.3.4 System hang detection

AIX 5L Version 5.1 offers a new feature called system hang detection. The concept behind the creation of this feature is explained as follows.

All processes[8] run at a priority. This priority is numerically inverted in the range 0-126. Zero is highest priority and 126 is the lowest priority. The default priority for all threads is 60. The priority of a process can be lowered by any user with the **nice** command. Anyone with root authority can also raise a process's priority.

The kernel scheduler always picks the highest priority runnable thread to put on a CPU. It is therefore possible for a sufficient number of high priority threads to completely tie up the machine so that low priority threads can never run. If the running threads are at a priority higher than the default of 60, this can lock out all normal shells and logins to the point where the system appears hung.

The system hang detection feature provides a mechanism to detect this situation and allow the system administrator a means to recover. This feature is implemented as a daemon (shdaemon) that runs at the highest priority. This daemon queries the kernel for the lowest priority thread run over a specified interval. If the priority is above a configured threshold, the daemon can take one of five actions. Each of these actions can be independently enabled, and each can be configured to trigger at any priority and over any time interval. The actions and their default values are shown in Table 2-5.

Table 2-5 The default shdaemon actions

Action	Default enabled	Default priority	Default timeout	Default service
Log an error	no	60	2	
Console message	no	60	2	/dev/console
High priority login shell	yes	60	2	/dev/tty0
Run command at high priority	no	60	2	
Crash and reboot	no	39	5	

For both High priority login shell and Run command at high priority actions, system hang detection will launch a special getty or the specified command at the highest priority. The special getty will print a warning message specifying that it is a recovering getty running at priority 0.

The **shconf** command is used to enable or disable the shdaemon, as well as configuring all the system hang detection environment specified with parameters.

We explain how to configure system hang detection in the following:

1. By default, system hang detection is not enabled (shdaemon has not been started) as shown in the following example:

```
# shconf -d
sh_pp=disable
# ps -ef | grep shdaemon | grep -v grep
... nothing displayed ...
```

[8] Technically, each thread in a process has a priority that is inherited from the main thread created upon the process fork.

2. Run **smitty** and select the following menus:

```
# smitty
    System Environments
        Change/Show Characteristics of Priority Problem Detection
```

   You see the SMIT panel shown in Example 2-12.

Example 2-12 System hang detection SMIT panel

```
           Change/Show Characteristics of Priority Problem Detection

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

[TOP]                                                   [Entry Fields]
  Enable Process Priority Problem                     disable              +
  Log Error in the Error Logging                      disable              +
    Detection Time-out                               [2]
    Process Priority                                 [60]
  Display a warning message on a console              disable              +
    Detection Time-out                               [2]
    Process Priority                                 [60]
    Terminal Device                                  [/dev/console]
  Launch a recovering login on a console              enable               +
    Detection Time-out                               [2]
    Process Priority                                 [56]
    Terminal Device                                  [/dev/tty0]
  Launch a command                                    disable              +
    Detection Time-out                               [2]
    Process Priority                                 [60]
    Script                                           [/]
  Automatically REBOOT system                         disable              +
    Detection Time-out                               [5]
    Process Priority                                 [39]

F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         Esc+6=Command       Esc+7=Edit          Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do
```

3. First, we enable system hang detection with all actions disabled. To do so, you have to change the value `enable` to `disable` on the "Launch a recovering login on a console" field on the SMIT panel.

4. You can verify if the shdaemon is running, as shown in Example 2-13.

Example 2-13 Verifying shdaemon (priority 60)

```
# ps lwx | head -1; ps lwx | grep shdaemon | grep - v grep
     F S      UID     PID    PPID    C PRI NI ADDR  SZ  RSS   WCHAN    TTY  TIME  CMD
```

```
240001 A        0 1017880       1   0  60 20 262b1 480  496   EVENT      -  0:00
/usr/sbin/shdaemon
```

You will notice that shdaemon is running with priority level 60, as shown in the PRI column.

5. You then enable one of the actions using the SMIT panel shown in Example 2-12 on page 54. In this example, we set the `Launch a recovering login on a console` field to enable. shdaemon will change to the highest available priority value 0.
6. To confirm the changed configuration, use the commands shown in the following examples:

```
# shconf -d
sh_pp=enable
# shconf -E -l prio H
sh_pp      enable      Enable Process Priority Problem
pp_errlog  disable     Log Error in the Error Logging
pp_eto     2           Detection Time-out
pp_eprio   60          Process Priority
pp_warning disable     Display a warning message on a console
pp_wto     2           Detection Time-out
pp_wprio   60          Process Priority
pp_wterm   /dev/console Terminal Device
pp_login   enable      Launch a recovering login on a console
pp_lto     2           Detection Time-out
pp_lprio   56          Process Priority
pp_lterm   /dev/tty0   Terminal Device
pp_cmd     disable     Launch a command
pp_cto     2           Detection Time-out
pp_cprio   60          Process Priority
pp_cpath   /           Script
pp_reboot  disable     Automatically REBOOT system
pp_rto     5           Detection Time-out
pp_rprio   39          Process Priority
```

You will also notice that the priority of shdaemon is now changed to 0, as shown in Example 2-14.

Example 2-14 Verifying shdaemon (priority 0)

```
# ps lwx | head -1; ps lwx | grep shdaemon | grep - v grep
     F S      UID     PID    PPID   C PRI NI ADDR  SZ  RSS   WCHAN    TTY  TIME  CMD
240001 A        0 1020394       1   0   0 20 262b1 33296 33312   EVENT -  0:00
/usr/sbin/shdaemon
```

7. You should confirm the following entry is inserted into /etc/inittab so that shdaemon is invoked on every reboot:

```
# lsitab -a |grep shdaemon
```

```
shdaemon:2:respawn:/usr/sbin/shdaemon >/dev/console 2>&1
```

For further information about system hang detection, please refer to the redbook *AIX 5L Differences Guide Version 5.1 Edition*, SG24-5765.

3

Practical use of serial connections

This chapter shows practical usage examples of serial connections in an AIX server farm environment.

This chapter contains the following five sections:

- Section 3.1, “Using terminal server” on page 58
- Section 3.2, “Using a modem” on page 62
- Section 3.3, “Automating administrative operations” on page 75
- Section 3.4, “Using PPP” on page 88
- Section 3.5, “Service Agent” on page 105

3.1 Using terminal server

A very useful management resource on a server farm is the one that allows you to manage multiple serial ports from a single point of control. From this centralized point you can:

- Keep doing the normal administration tasks in case of a network failure.
- Access the TTY remote boot facility (see Section 2.3.3, “TTY remote reboot” on page 50).
- Access the SMS menu to remotely power on a machine (see Section 3.3, “Automating administrative operations” on page 75).

We describe two different ways of implementing this resource. The two approaches are valid, and the choice for each one will depend on your budget and/or specific requirements of your site. If you can implement both, this should be the best environment possible, because the terminal server (appliance) implementation can be thought as a contingency for the administration server, in case of hardware or network failure.

3.1.1 Terminal server

You can purchase a terminal server, which is an appliance with multiple serial ports and one (or more) network interface(s), such as Ethernet. One of the main purposes of such a device is console management of multiple servers, which is accomplished by allowing an administration workstation connected to the terminal server’s network port to gain access to each of the connected serial ports of the managed servers.

From this single administration workstation, you can do all sorts of administrative tasks, without the inconvenience of having to move to the place where the specific serial terminal (attached to the server you want to manage) is located. You can do the same tasks over the networks; however, you may have a problem with the network environment of your server farm, or with the IP stack on your server.

This approach also reduces the physical environment management efforts, and saves space, since only one serial device is attached to all the managed servers.

In Figure 3-1 on page 59, we show an environment with a terminal server attached to the administration workstations through a physically isolated network. The terminal server is also connected via serial connections to the ports of the servers to be monitored.

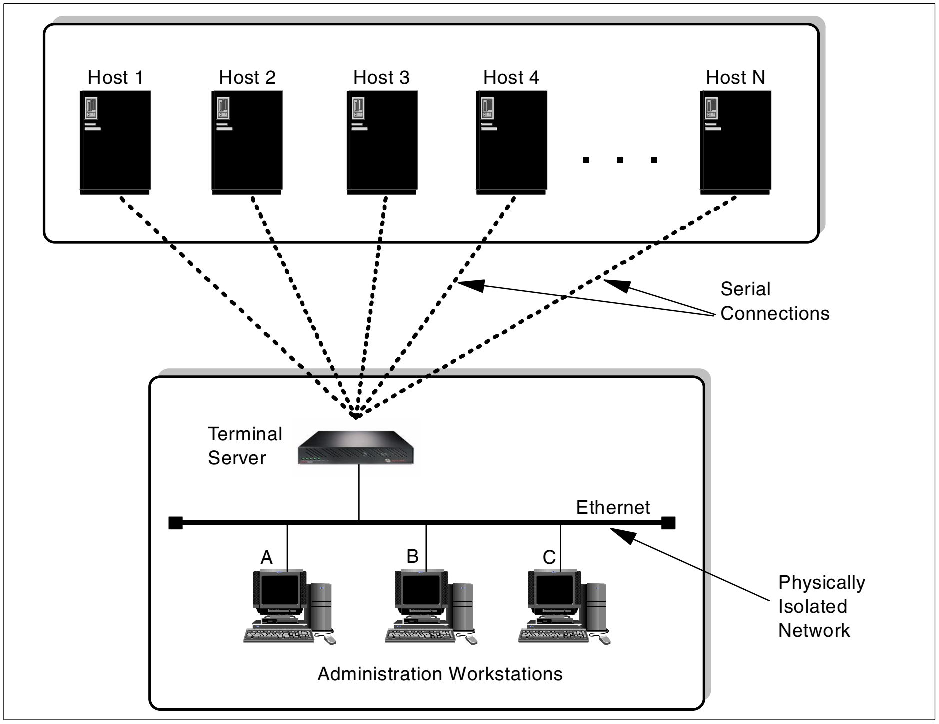


Figure 3-1 Terminal server environment

The following steps explain the required tasks to use terminal servers[1] in our environment to establish an administrative session from the administration workstation A to the serial port of Host 2:

1. Connect each port of the terminal server to its corresponding serial port on the server side using a null-modem cable.
2. You have to configure a TTY device, which is connected to one of serial ports on the terminal server, with the Enable LOGIN parameter set to enable on all the servers. You should specify TERMINAL type (TERM) as vt100. Keep the default 9600 8N1 communication parameters unchanged. For further information how to configure a TTY device, see Section 2.2.3, “Serial port configuration” on page 41.
3. Configure the terminal server box. This can vary significantly depending on the terminal server manufacturer and model. We assume that the terminal server configuration is based on the following:
   – You configure the terminal server to match the communication parameters set on the server’s serial port (typically, 9600-8N1).

[1] In our test environment, we use the Equinox ELS 16-II terminal server.

   - You use the **telnet** command to the terminal server’s IP address to access the desired terminal server serial port,
   - You specify an appropriate TCP port number, which corresponds to the specific serial port you are going to connect to, as an argument of the **telnet** command.

4. In our example, the terminal server IP address is 10.0.0.10 and the base telnet socket number is 30XX, where XX is the connected serial port number on the terminal server.

   You invoke the **telnet** command on administration workstation A, to access Host 2, which is attached to the terminal server’s serial port 2, as shown in the following example:

   ```
   # telnet 10.0.0.10 3002
   Trying...
   Connected to 10.0.0.10.
   Escape character is '^]'.
   ```

   You might have to type Enter to start the login session. Then you will see the login prompt of the target managed server, as shown in the following example:

   ```
   AIX Version 5
   (C) Copyrights by IBM and by others 1982, 2000.
   login:
   ```

Note: Because most terminal server do not support secure network connections, such as OpenSSH (see Section 4.2, “OpenSSH” on page 131), it is strongly recommended that a terminal server should be connected on the physically isolated network, which is dedicated for administrative tasks.

3.1.2 Setting up your own terminal server

You can set up your own terminal server using an AIX system installed with asynchronous adapters instead of purchasing a terminal server. This configuration has some advantages compared to the appliance, because you can secure your terminal server environment using OpenSSH (explained in Section 4.3, “OpenSSH on AIX” on page 135). Therefore, you do not require a physical isolated network for this solution, and you also have the full functionality of AIX, such as executing management programs or scripts. We refer to this AIX system as the *Administration Server* in server farms.

Figure 3-2 on page 61 illustrates an example environment using an Administration Server that is running AIX and using OpenSSH.

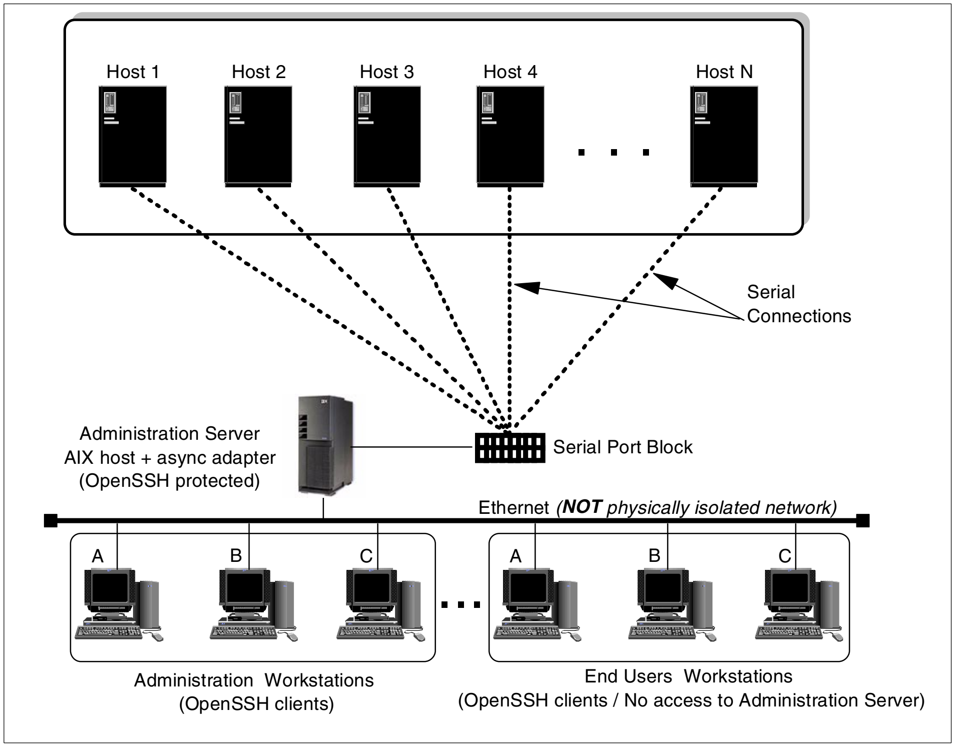


Figure 3-2 Implementing a terminal server using an AIX system

In the following steps, we explain how to set up your own terminal server, as shown in Figure 3-2. We assume that OpenSSH is already installed on both the client and the server sides, as explained in Section 4.3, “OpenSSH on AIX” on page 135.

1. Connect each serial port of the administration server to the serial port of the server side using a null-modem cable.
2. Configure a TTY device, which is connected to one of serial ports on the administration server, with the Enable LOGIN parameter set to enable on all the servers. You should specify TERMINAL type (TERM) as vt100. Keep the default 9600 8N1 communication parameters unchanged. For further information how to configure a TTY device, see Section 2.2.3, “Serial port configuration” on page 41.
3. Configure all the TTY devices that will be connected to the servers ports with the Enable LOGIN parameter set to disable on the administration server.

4. Edit /etc/uucp/Devices on the administration server, as shown in the following example:

```
Direct    tty0    -    9600    direct
Direct    tty1    -    9600    direct
Direct    tty2    -    9600    direct
Direct    tty3    -    9600    direct
Direct    tty4    -    9600    direct
Direct    tty5    -    9600    direct
Direct    tty6    -    9600    direct
Direct    tty7    -    9600    direct
Direct    tty8    -    9600    direct
Direct    tty9    -    9600    direct
Direct    tty10   -    9600    direct
```

5. Login to the administration server from one of administration workstations. You can use OpenSSH, as explained in Section 4.3, “OpenSSH on AIX” on page 135, instead of the **telnet** command.

6. Execute the following command to connect the managed server connected on the /dev/tty1device:

```
# cu -ml tty1
Connected
```

   If you type Enter, you will see the login prompt of the managed server, as shown in the following example:

```
AIX Version 5
(C) Copyrights by IBM and by others 1982, 2000.
login:
```

Note: You can use the ATE program (command name is **ate**) to perform the Administration Server role instead of the **cu** command. See Section 3.2.3, “Using the ATE program” on page 69.

3.2 Using a modem

In a server farm, a modem plays an important role, as it is a widely used (and affordable) way of remote connection to the site where the computer resources are located. This section discusses modem standards, general modem setup, and specific configuration tips for helping you configure your environment.

3.2.1 Modem overview

A modem is a device that allows you to connect one computer to another across ordinary telephone lines. The current telephone system is incapable of carrying the voltage changes required for a direct digital connection. A modem overcomes this limitation by modulating digital information into audio tones for transmission across the phone line, and by demodulating those tones back into digital information on reception. The device name modem is actually an acronym that express these two functions: modulating and demodulating.

Telecommunication standards

As introduced in Section 2.1.3, “Serial communications terminology” on page 23, the term baud is used to refer to modem (or other serial device) speed instead of bps. Baud is actually a measurement of the modulation rate. In older modems, only 1 bit was encoded in each signal change, so modem baud rate was equal to modem speed. Modems that operate at higher speeds, however, still generally operate at 2,400 (or even 1,200) baud, and encode two or more bits per signal change. A modem’s bps rate is calculated by multiplying the number of data bits per signal with the baud (for example, 2,400 baud x 6 bits per signal change = 14,400 bits per second). Most modern modems can communicate at a variety of speeds, for example, 28,800, 14,400, 9,600, 7,800, 4,800, and 2,400 bps.

The older speeds of 300, 1,200, and 2,400 bps were well defined. However, as modem manufacturers began to devise methods for gaining higher speeds, some manufactures started to use a proprietary method incompatible with modems from others. Today, the Telecommunication Standardization Sector of the International Telecommunications Union (ITU-T) (formerly the United Nations Consultative Committee for International Telephony and Telegraphy, abbreviated CCITT) defines standards for most high-speed communications.

These compression algorithms are sensitive to the data being transmitted. If the data has already been compressed (for example, with the **`compress`** command), the data compression methods of high-speed modems offer little or no benefit, and might even reduce data throughput.

When using a modem with data compression technology, the speed of the data terminal equipment/data circuit-terminating equipment (DTE/DCE) connection between the computer and the modem is equal or greater than the nominal data rate of the connection between modems. For example, with a V.32bis modem with V.42bis data compression, the data rate of the modem (the speed at which the modem communicates across telephone lines) is 14,400 bps. When the V.42bis compression is active, actual data throughput can reach 57,600 bps. To accommodate the greater throughput offered by data compression, the speed of the serial line between the computer and the modem must support 57,600 bps or higher.

Note: Some high-speed modems that implement data compression may yield a higher data throughput than the ones whose serial ports can be supported on AIX systems.

ITU-T standards are usually named V.*NN*, where NN stands for a number. Another common standard is the Microcosm Networking Protocol (MNP), which has available versions (called classes) 1-9. MNP is a high-performance, high-speed protocol that was available relatively early, and became something of a *de facto* standard before the advent of the CCITT standards.

ITU-T communications standards

The following list describes some common communications standards defined by the ITU-T:

V.29 9600 bits per second modem standardized for use on point-to-point 4-wire leased telephone-type circuits.

V.32 A family of 2-wire, duplex modems operating at data signalling rates of up to 9600 bit/s for use on the general switched telephone network and on leased telephone-type circuits.

V.32bis A duplex modem operating at data signalling rates of up to 14400 bit/s for use on the general switched telephone network and on leased point-to-point 2-wire telephone-type circuits. A revision of the V32 standard.

V.34 A modem operating at data signalling rates of up to 33600 bit/s for use on the general switched telephone network and on leased point-to-point 2-wire telephone-type circuits.

V.42 Error-correcting procedures for DCEs using asynchronous-to-synchronous conversion.

V.42bis Data compression procedures for data circuit-terminating equipment (DCE) using error correction procedures.

V90 A digital modem and analog modem pair for use on the Public Switched Telephone Network (PSTN) at data signalling rates of up to 56000 bit/s downstream and up to 33600 bit/s upstream.

For further information about the standards, please visit the following URL:

http://www.itu.int/rec/recommendation.asp?type=products&lang=e&parent=T-REC-V

Modem considerations

The following aspects should be taken into consideration when using modems in an AIX environment.

- Supported modems

  Any EIA 232 compliant modem is capable of returning results in response to a command, and capable of communication at one of the following baud rates[2]: 2400, 4800, 9600, 19200, and 38400.

- Data Carrier Detect Handling

  AIX uses the Data Carrier Detect (DCD) signal to monitor the true state of a modem. If the DCD signal on the modem's port is high, the server believes the modem to be in use. It is therefore important to know which circumstances cause this signal to be forced into a high state.

  The DCD signal can be raised high for the following reasons:

  - The use of clocal in the stty attributes for the run-time field on the SMIT TTY Configuration panel.
  - Having the Ignore Carrier Detect field set to enable on the SMIT TTY Configuration panel for TTYs.
  - Connected to a 128-port adapter.
  - The modem forces DCD high with either AT commands or dip switch settings.
  - The TTY port is already in use by an application.

- Data Terminal Equipment or Data Communication Equipment speeds

  With serial communication involving modems, as pictured in Figure 3-3 on page 66, there are three major considerations:

  - DTE interface speed (server to modem). This is the speed the server communicates to the modem.
  - DCE interface speed (modem to server) (sometimes called the serial port interface speed). This is the speed at which the modem communicates to the server.
  - Connection speed (modem to modem). This is the speed at which a modem communicates (or talks) to another modem.

  Most high-speed moderns allow the DCE interface speed to be different than the connection speed. This allows the DTE speed to be locked at a single baud rate while allowing the connection speed to fluctuate up or down as needed for proper communication between modems. Modern high-speed modems hold the data to be transmitted to the server in a buffer and send it

[2] We exclude baud rates under 2400 on purpose, because they are no longer widely used.

when the system can accept it. They can also hold data to be transmitted to the other modem in a buffer and send it as the remote is able to accept it. This kind of data transmission requires the modem and the server to engage in flow control.

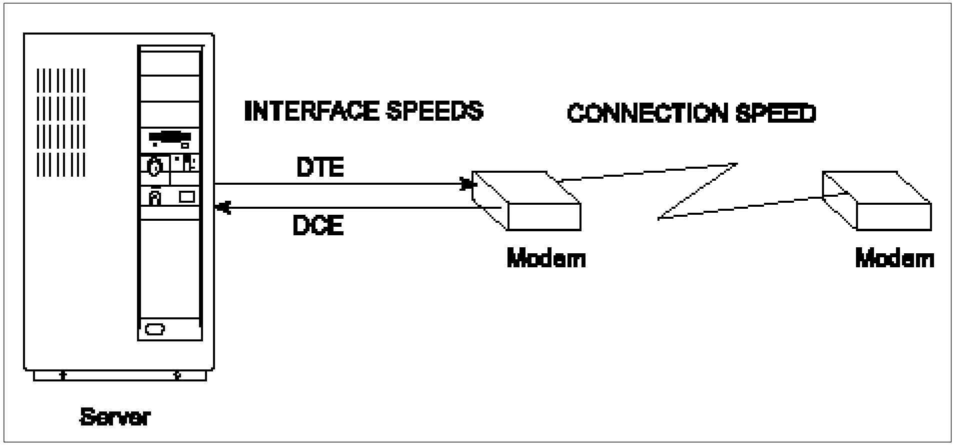


Figure 3-3 Modem speed considerations

- Modem Control Signals

  Modems are often used to initiate and receive calls. It is therefore important to program the modem to negotiate a connection at the highest possible speed and to reset itself to a known state after a connection is stopped. The server will toggle the Data Terminal Ready (DTR) signal from on to off to instruct the modem to terminate the connection. Most modems can be configured to reset themselves when this on-to-off DTR transition occurs.

  For the connection between the server and the modem to be fully functional, the cabling must have the following qualifications:

  – It must meet specifications.

  – It should be properly shielded.

  – At least the following signals should be provided: RxD, TxD, RTS, CTS, SG, DCD, and DTR, as described in Section 2.1.5, “Serial port cabling” on page 28.

AT commands

In the early 1980s, a company called Hayes Microcomputer Products Inc. invented the AT commands set to allow easy interfacing to their auto-dial modems. Since then, it became a *de facto* industry standard, although many modem vendors typically extend the basic AT command set.

The basic functions offered by this command set allow you to:

- Configure the modem phone line interface.
- Configure the modem to the DTE interface.
- Program modem memory (sometimes referred to as *registers*).
- Dial phone numbers.

The configuration process is based on a sequence of inputs (commands) issued on the TTY connected to the modem, which echoes back on the screen the corresponding answer for each command.

The list of Hayes AT commands is quite long, and each manufacturer enhances this list by introducing new commands. So, the better approach when configuring modems is to have the product manual by your side.

If you want more information about the complete Hayes AT command site and other useful information about modems, please visit the following URL:

`http://nemesis.lonestar.org/reference/telecom/modems/index.html`

Useful hints and tips

We summarize a list of guidelines that could help you to configure modems:

- Purchase a modem that supports the standards that satisfies your communications requirement, for example V32/bis, V34, and so on.
- Use an asynchronous port and a cable that support both the DCD input from the modem and the DTR output to the modem.
- Set the hardware dip switches properly, if your modem have switches.
- Start with factory defaults when configuring your modem.
- Configure your modem to use CD as the data carrier detection for the EIA 232 signal. This will give the best indication of the communication link status. You must turn off the clocal flag for the TTY device so that AIX will detect and report status changes in the DCD signal *and* the cable must carry the DCD signal from the modem to the host serial port connector.
- Attempt to configure the modem to disconnect when your terminal or system serial port drops the DTR signal. This gives you a way to force a disconnect and may prevent some unnecessary long distance charges. The hupcl flag must be set on the TTY device to tell AIX to drop DTR on close.
- Always retry things that fail. Sometimes the phone company might make mistakes.

3.2.2 Configuring modems on AIX

In an AIX environment, modems are commonly used with the following applications:

- Basic Network Utilities (BNU) or other implementations of the UNIX-to-UNIX Copy Program (UUCP).
- Asynchronous Terminal Emulation (ATE) program.
- Point-to-Point Protocol (PPP).

Basic modem configuration

Here, we are going to present the necessary basic steps to configure a modem[3] connected to the tty1 device on an AIX system using the **cu** command:

1. Create and configure a TTY device (in this example, tty1) with the Enable LOGIN parameter set to off.
2. Program your modem using the **cu** command:
   a. First, add the following line to the /etc/uucp/Devices file:
      ```
      Direct tty1 - Any direct
      ```
   b. Verify that tty1 has been disabled by issuing the following command:
      ```
      # pdisable tty1
      ```
   c. Type in the following command:
      ```
      # cu -ml tty1
      ```
   d. You will see the message `Connected.` on the screen, which means that you are communicating with the modem. In order to verify if the modem supports AT command sets, type `AT` in uppercase[4] letters and then press Enter. It should respond to you with the message `OK`. If you do not receive this message, then check the following:
      - Cable connections. Please make sure that you are *not* using a null-modem cable between the AIX system and the modem.
      - Verify the results of steps described in a and b.
      - Check specific commands related to your modem model. Although the Hays AT command set is widely used in the industry, some modems might differently implement several AT commands from the standard.
      - Consult with your modem hardware publication to confirm that there are any hardware or software configurations that could override the normal behavior of the signal pins.

[3] We used US Robotics Sportster 14400 FAX/modem, supporting CCITT V32 bis and V42 bis in our example.

[4] Some modems are case sensitive for AT command sets.

e. After receiving the OK reply message, program the modem using the following basic Hayes commands with the presented sequence. After each command is entered, the modem will respond with an OK message on the screen. To terminate the communication with the modem, type ~. in the session.

AT&F Recalls the factory configuration as the active configuration.

ATE1 In command state, echoes characters from the keyboard to the screen. Make sure carrier is not ON on the port or modem.

AT&D2 Monitors the DTR signal. When an on-to-off transition of the DTR signal occurs, the modem hangs up and enters the command state.

AT&C1 Tracks the status of the carrier detect signal.

ATS0=1 Specifies auto answer.

ATS9=12 Carrier-detect response time (typical default value is 6). Possible values are 1 to 255, in tenths of seconds.

AT&W Writes the storable parameters of the current configuration to memory.

~. Terminates the connection

3. After configuring the modem, set the Enable LOGIN parameter to one of the values in Table 2-4 on page 45. The parameter clocal must not be configured, and hupcl must be set on the TTY device.
4. If you are using a remote AIX host connected to the other modem, repeat steps 1, 2, and 3.
5. Now you can configure your communication application, such as BNU or ATE, to meet the specific requirements of your application, such as login prompt, file transfer, terminal connection, and so on.

3.2.3 Using the ATE program

ATE stands for Asynchronous Terminal Emulation, which is an easy-to-use menu-driven application. It can run commands on the connected system and send and receive files using the xmodem protocol. You can also capture and file incoming data from the remote system.

ATE accepts both DTE/DTE (using null-modem cable) and DTE/DCE (using straight cable) connections for communication between the local and the remote systems. When you start ATE with the **ate** command, it displays the Unconnected Main Menu, which enables the user to:

- Temporarily change the characteristics of ATE (modify and alter)

- Connect to another system (directory and connect)
- Get help (help)
- Execute workstation operating system commands on the system (perform)
- Leave ATE (quit)

Note: To use ATE, the user has to be a member of the uucp group.

In the following steps, we explain how to use ATE to dial from a local AIX system to a remote AIX system, as shown in Figure 3-4.

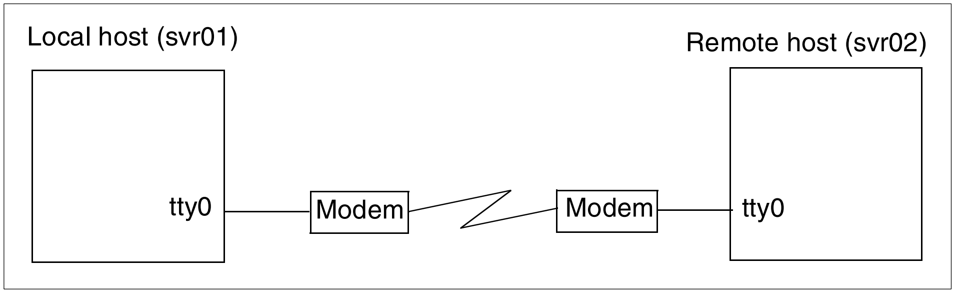


Figure 3-4 ATE connection configuration example

Note: You can also use ATE to connect to the remote system without modems (DTE/DTE connection).

1. Verify that you have already installed the ATE program (/usr/bin/ate), as shown in the following example:

   ```
   # whence ate
   /usr/bin/ate
   # ls -l /usr/bin/ate
   -r-xr-xr-x   1 bin      bin          114392 Apr 08 2001  /usr/bin/ate
   ```

   Otherwise, install the bos.net.ate fileset, which includes the ATE program, from the AIX 5L Version 5.1 BOS installation media.

2. Configure the TTY devices and the modems on both sides, as explained in “Basic modem configuration” on page 68. You have to set the tty Enable LOGIN to `disable` on your local host (dialing side), and to `enable` on the remote host (receiving side). You must *not* set clocal and set the hupcl parameters of the TTY devices attributes on both sides.

3. Issue the **`ate`** command on the local host (svr01). You will see the main panel shown in Example 3-1 on page 71.

Example 3-1 ATE Unconnected Main Menu

```
Node: svr01          UNCONNECTED MAIN MENU
-------------------------------------------------------------------------------
     COMMAND     DESCRIPTION
     -------     -----------------------------------------------
     Connect     Make a connection
     Directory   Display a dialing directory.

     Help        Get help and instructions.
     Modify      Modify local settings.
     Alter       Alter connection settings.
     Perform     Perform an Operating System command.
     Quit        Quit the program.
-------------------------------------------------------------------------------
     The following keys can be used during a connection:
         ctrl b Start or stop recording display output.
         ctrl v Display main menu to issue a command.
         ctrl r Return to a previous screen at any time.
-------------------------------------------------------------------------------
Type the first letter of the command and press Enter.
>
```

4. In this example, we assume that the default AIX communication parameters (9600 8N1 Xon/Xoff) are set for the TTYs and the modems on both sides. Therefore, you have to change the ATE default parameters to adapt them to your configuration. To do so, select the Alter connection setting menu by typing the character a. You will see the setting panel shown in Example 3-2.

Example 3-2 Alter connection settings menu

```
Node: svr01          ALTER CONNECTION SETTINGS
-------------------------------------------------------------------------------
 COMMAND   DESCRIPTION             CURRENT          POSSIBLE CHOICES
---------  ----------------------  --------  ------------------------------
Length     Bits per character      8         7,8
Stop       Number of stop bits     1         1,2
Parity     Parity setting          0         0=none, 1=odd, 2=even
Rate       Number of bits/second   1200      50,75,110,134,150,300,600,
                                             1200,1800,2400,4800,9600,19200

Device     /dev name of port       tty0      tty0-tty16
Initial    Modem dialing prefix    ATDT      ATDT, ATDP, etc.
Final      Modem dialing suffix              0 for none, valid modem suffix
Wait       Wait between redialing  0         seconds between tries
Attempts   Maximum redial tries    0         0 for none, a positive integer

Transfer   File transfer method    p         p=pacing, x=xmodem
Character  Pacing char or number   0         0 for none, a single char/integer
-------------------------------------------------------------------------------
```

```
To change a current choice, type the first letter of the command followed by
your new choice (example:  r 300) and press Enter.
>
```

To change these parameters, type the following at the > prompt:

```
> r 9600
> d ttyX (X is a serial port number that you are going to connect.)
```

5. Now, you are ready to dial to the remote host. Return to the main menu (just press Ctrl-r) and type `c <phone number>`. You should be able to connect to the remote host and see the login prompt of the remote host.

If you want to return to the local host, you can always suspend your remote session by pressing Ctrl-r.

You can use the following two configuration tips to better use the **ate** command:

1. Instead of changing the configuration parameters at the Alter Connection Settings menu upon every connection attempt, you can edit the ate.def file to store these settings. The file is automatically created on the current directory upon the first invocation of the **ate** command. The **ate** command reads this configuration file from the current directory only.
2. If you have to dial to different phone numbers, another useful utility is the directory file (/usr/lib/dir by default), on which you can associate names to your dialing settings (phone number, baud rate, and so on). You can access these lists from the Unconnected Main Menu using the option d (Directory) to choose the host to dial. You can specify your directory file on a different path by editing the DIRECTORY parameter in the ate.def file.

Transferring files using xmodem

You can transfer files between systems over the **ate** command session. The **xmodem** command is used with the ATE program to transfer files, using the xmodem protocol, which is an 8-bit transfer protocol that detects data transmission errors.

In the following example, we describe the required steps for sending a file from a local host (svr01) to a remote host (svr02):

1. Verify that you have already installed the **xmodem** command on both sides. The bos.net.ate fileset also includes this command, as shown in the following example:

```
# whence xmodem
/usr/bin/xmodem
# lslpp -w /usr/bin/xmodem
  File                                        Fileset             Type
----------------------------------------------------------------------------
```

```
/usr/bin/xmodem                             bos.net.ate          File
```

2. You must have an already established ATE connection between the local and remote hosts, as explained in Section 3.2.3, “Using the ATE program” on page 69.
3. Set the TRANSFER parameter to xmodem by typing `x` on the Alter Connection Settings menu.
4. After logging in to the remote host (in this example, on svr02), you should type the following command on the remote host:

   ```
   root@svr02:/ # xmodem -r /tmp/target
   ate: 0828-005 The system is ready to receive file /tmp/target.
                 Use ctrl-X to stop xmodem.
   ```

   The option -r instructs the **xmodem** command to receive the specified file from the local host and wait until the target file has sent it from the local host.

   > **Note:** The remote file name does not have to be same as the local file name.

5. Press Control-v to access the connected main menu, as shown in Example 3-3.

Example 3-3 Connected main menu

```
Node: svr01          CONNECTED MAIN MENU
-------------------------------------------------------------------------------
     COMMAND      DESCRIPTION
     -------      -----------------------------------------------
     Send         Send a file over the current connection.
     Receive      Receive a file over the current connection.
     Break        Send a break signal over the current connection.
     Terminate    Terminate the connection.

     Help         Get help and instructions.
     Modify       Modify local settings.
     Alter        Alter connection settings.
     Perform      Perform an Operating System command.
     Quit         Quit the program.
-------------------------------------------------------------------------------
     The following keys can be used during a connection:
         ctrl b Start or stop recording display output.
         ctrl v Display main menu to issue a command.
         ctrl r Return to a previous screen at any time.
-------------------------------------------------------------------------------
Type the first letter of the command and press Enter.
>
```

6. Press s to send a file. You will be prompted to enter the source file that you are going to send:

   ```
   Type the name of the file you wish to send and press Enter.  To use
   the last file name (), just press Enter.
   >
   ```

   In this example, we typed /tmp/source as the source file name, then pressed Enter.

7. You will see the sending message while your file is transferring, as shown in Example 3-4.

Example 3-4   Transferring file using ATE

```
ate: 0828-024 The program is ready to send file /tmp/source.
           You will receive another message when the file transfer is complete.
ate: 0828-025 The system is sending block 1.
ate: 0828-025 The system is sending block 2.
ate: 0828-025 The system is sending block 3.
ate: 0828-025 The system is sending block 4.
ate: 0828-025 The system is sending block 5.
ate: 0828-025 The system is sending block 6.
ate: 0828-025 The system is sending block 7.
ate: 0828-025 The system is sending block 8.
ate: 0828-025 The system is sending block 9.
ate: 0828-025 The system is sending block 10.
ate: 0828-025 The system is sending block 11.
ate: 0828-025 The system is sending block 12.
ate: 0828-025 The system is sending block 13.
ate: 0828-025 The system is sending block 14.
ate: 0828-015 The file transfer is complete.
ate: 0828-040 Press Enter.
```

8. After the file is successfully transferred, press Enter to return to the shell prompt on the remote host.

In most cases, you will notice that the size of remote file is larger than the size of the local file, as shown in the following example:

```
root@svr02:/ # ls -l /tmp/target
-rwxr--r--   1 root     system         1792 Apr 05 10:11 /tmp/target

root@svr01:/ # ls -l /tmp/source
-rw-r--r--   1 root     system         1748 Apr 05 10:05 /tmp/source
```

The reason is that the **xmodem** command sends data with a 128-byte block basis (128 bytes times 14 is 1792 bytes). If the size of the source file is not a multiple of 128 bytes, then the last block is padded with Control-z characters. To avoid this situation, do the following on the local host before sending files:

```
$ tar -cvf - file_name1 file_name2 ... | compress > archived_file.tar.Z
$ dd bs=128 conv=sync if=archived_file.tar.Z of=file_to_transfer.tar.Z
```

The specified options `bs=128 conv=sync` instructs the **dd** command to create the output file with the size of multiplies of 128 bytes and to pad the last block with null-character (0x00) if the original last block size is less than 128 bytes.

After transferring this file using the **xmodem** command, do the following on the remote host to extract the original files:

```
$ uncompress transfered_file.tar.Z
$ tar -xvf transfered_file.tar
```

Because the **uncompress** command ignores the trailing null characters at the file end, you can extract exactly the same files on the remote host.

Receiving files from the remote system

In order to receive files from svr02 to svr01, do the following:

1. Repeat steps 1, 2, and 3 from “Transferring files using xmodem” on page 72.
2. After connecting to svr02, issue the **xmodem -s <source file>** command on the shell prompt on svr02.
3. Repeat step 5 from “Transferring files using xmodem” on page 72.
4. On the Connected Main Menu, press r to receive the file from svr02, and type the local file name to be received on svr01, then press Enter.

3.3 Automating administrative operations

This section explains an advanced topic that enables automated operations over a serial connection. When you connect two AIX systems using a serial port connection, you can use the **cu** or the **ate** command to login to a remote system. Once you logged in to the remote system, you can issue any commands as if you were in the telnet session. However, you have to log in to the remote system first anyway. Because the login and the password prompt set a non-canonical mode to a TTY, you cannot use the shell’s redirect function to automate the login process to the remote system.

There are several software tools that simulate keyboard inputs by human. The most famous tool is *Expect*, which is a *Tcl* (tool command language) based scripting language.

The AIX toolbox for Linux applications provides the installable Expect package for AIX, found at:

http://www-1.ibm.com/servers/aix/products/aixos/linux/download.html

Another useful tool is the *Expect module* for Perl, which extends the Perl language to implement similar functions provided by the Expect language.

By using these language tools, you can write scripts that can send user input on the prompts from many application programs instead of typing it yourself.

In this section, we select Perl with the Expect module (referred to hereafter as Expect/Perl) to write scripts, simply because we are familiar with Perl rather than the Expect language itself. Example scripts presented in this section will break through the password prompt on the remote system to automatically issue administrative operations.

3.3.1 Install Perl/Expect

Installing Perl/Expect is an easy task. Because Perl[5] is installed by default on AIX 5L Version 5.1, you only have to download and install the several modules required for Perl/Expect.

To install these modules, visit the CPAN (Comprehensive Perl Archive Network) site that archives all the modules for Perl. You can access CPAN at the following URL:

http://www.cpan.org

The following steps explain how to install the Expect module on AIX 5L Version 5.1:

1. Download the following three modules[6] from CPAN:
   - Expect-1.15.tar.gz
   - IO-Stty-IO-Stty-.02.tar.gz
   - IO-Tty-1.02.tar.gz
2. Install IBM C compiler V5 and the GNU **gzip**[7] command.
3. Make sure you have a minimum of 1 MB free space in /usr.
4. Install these modules, as shown in Example 3-5.

Example 3-5 Install modules for Perl

```
$ cd /work/src/Expect_modules
```

[5] Starting from AIX Version 4.3.3, Perl is installed by default as the perl.rte fileset.

[6] You might see newer versions of these modules on CPAN.

[7] The GNU **gzip** command is provided by the AIX toolbox for Linux applications.

```
$ gzip -d -c ../source_package/IO-Tty-1.02.tar.gz | tar -xf -
$ cd IO-Tty-1.02
$ perl Makefile.PL
$ make
$ su -
# cd /work/src/Expect_modules/IO-Tty-1.02
# make install
# exit
$
$ cd /work/src/Expect_modules
$ gzip -d -c ../source_package/IO-Stty-.02.tar.gz | tar -xf -
$ cd IO-Stty-.02
$ perl Makefile.PL
$ make
$ su -
# cd /work/src/Expect_modules/IO-Stty-.02
# make install
# exit
$
$ cd /work/src/Expect_modules
$ gzip -d -c ../source_package/Expect-1.15.tar.gz|tar -xf -
$ cd Expect-1.15
$ perl Makefile.PL
$ make
$ su -
# cd /work/src/Expect_modules/Expect-1.15
# make install
# exit
```

5. You will see the following files installed under the /usr/opt/perl5/lib/site_perl/5.6.0[8] directory:

```
# ls -lR | grep -v CT
total 208
-r--r--r--   1 root     system       52845 Mar 19 05:59 Expect.pm
-r--r--r--   1 root     system       41666 Mar 19 05:59 Expect.pod
drwxr-xr-x   2 root     system         512 May 10 14:46 IO
drwxr-xr-x   4 root     system         512 May 10 14:43 aix
./IO:
total 40
-r--r--r--   1 root     system       14416 Jan 05 1998  Stty.pm
-r-xr-xr-x   1 root     system         198 Dec 05 1997  stty.pl

./aix:
total 16
drwxr-xr-x   3 root     system         512 May 10 14:43 IO
drwxr-xr-x   4 bin      bin            512 May 10 14:47 auto
```

[8] The Perl version provided by AIX 5L Version 5.1 is 5.6.0.

```
./aix/IO:
total 48
-r--r--r--   1 root     system        8517 Apr 02 06:29 Pty.pm
drwxr-xr-x   2 root     system         512 May 10 14:43 Tty
-r--r--r--   1 root     system        7277 Apr 02 06:29 Tty.pm

./aix/IO/Tty:
total 16
-r--r--r--   1 root     system        7184 May 10 14:37 Constant.pm

./aix/auto:
total 24
drwxr-xr-x   2 root     system         512 May 10 14:47 Expect
drwxr-xr-x   4 root     system         512 May 10 14:46 IO

./aix/auto/Expect:
total 8
-rw-r--r--   1 root     system         120 May 10 14:47 .packlist

./aix/auto/IO:
total 16
drwxr-xr-x   2 root     system         512 May 10 14:46 Stty
drwxr-xr-x   2 root     system         512 May 10 14:43 Tty

./aix/auto/IO/Stty:
total 8
-rw-r--r--   1 root     system          92 May 10 14:46 .packlist

./aix/auto/IO/Tty:
total 80
-rw-r--r--   1 root     system         372 May 10 14:43 .packlist
-r--r--r--   1 root     system           0 May 10 14:38 Tty.bs
-r-xr-xr-x   1 root     system       34222 May 10 14:38 Tty.so
```

3.3.2 A simple exercise using Perl/Expect

Figure 3-5 on page 79 illustrates the concept of how a script written in Perl/Expect interacts with an external program. Upon invoking the external program, the script sets up pseudo-device sets, as shown in Figure 3-5 on page 79. When the program writes some data to standard out stream, the script reads from the pseudo-control device. If the data read from the device matches with the *expected* one expressed by the expect() function in the script, then the script writes back a response to the device using the send() function. The program reads the data sent by the script from the standard in stream.

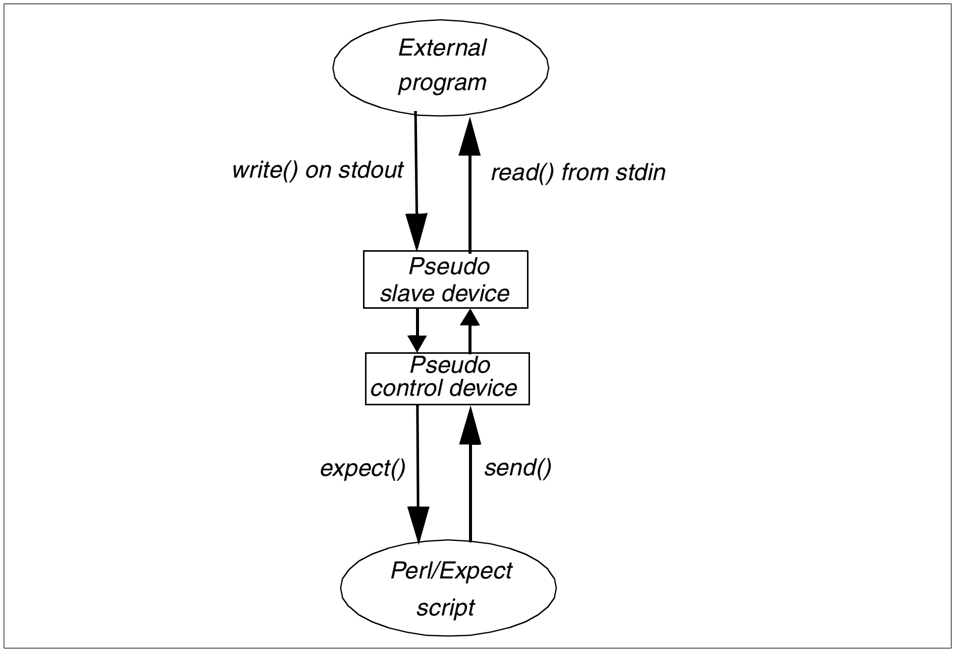


Figure 3-5 Interaction between Perl/Expect and external program

Look at the simple Perl/Expect script shown in Example 3-6.

Example 3-6 Simple example for Perl/Expect

```
 1: #!/usr/bin/perl
 2: use Expect;
 3:
 4: $e = Expect->spawn("/bin/sh");
 5: $e->expect(undef, "# ");
 6: $e->send("tty\r");
 7: $e->expect(undef, "# ");
 8: $e->send("echo $?\r");
 9: $e->expect(undef, "# ");
10: $e->send("ps -afe | grep " . $$ . "\r");
11: $e->expect(undef, "# ");
12: $e->hard_close();
13: printf("done.\n");
14: exit 0;
```

The statement `use Expect` in the second line in Example 3-6 declares that you are going to use the Expect module in your Perl script.

The fourth line spawns a child process to execute an external program (in this example, /bin/sh):

```
$e=Expect->spawn("/bin/sh");
```

The spawn() function sets up the pseudo terminal device pair and executes a specified external program so that stdin and stdout of the external process are mapped to the pseudo terminal. The variable $e stores an instance of the Expect class that is returned by the spawn() function. You have to access several methods of this instance variable to interact with the spawned external program.

The fifth line instructs the script to wait until it receives the root user’s prompt “# “from the spawned shell process:

```
$e->expect(undef, "# ");
```

The first argument of the expect() function is a time-out value and the second argument is a pattern string that you are expecting to be sent from the external program. If the undef value is specified as the time-out value, the script waits forever until the expected pattern is sent by the external program.

Note: This line assumes that your root user’s shell prompt string contains the pound sign (#). Otherwise, you have to modify this line to match with your customized shell prompt string.

Once it receives the root user’s command prompt, it sends the **tty** command to the external program, as shown in the sixth line:

```
$e->send("tty\r");
```

When the script sends a string to the external program using the send() function, you normally have to add the ‘\r’ character at the end of the sending string to simulate the Enter key on the standard input. Otherwise, the standard input stream is not flushed, the external program cannot receive the data sent from the script.

On the other hand, when the external program prints the new line character to flush the standard output, the script receives two bytes of data, “\r\n”. These conversions are performed by the pseudo terminal device pair.

If you invoke this script, extest, it generates the output shown in the following example:

```
# ./extest
# tty
/dev/pts/2
# echo 0
0
```

```
# ps -afe | grep 10908
    root 10908  9002  11 21:43:51  pts/1  0:00 perl ./extest
    root 19676 10908   1 21:43:51  pts/2  0:00 /bin/sh
    root 30624 19676   1 21:43:51  pts/2  0:00 grep 10908
# done.
```

You will notice that the TTY device name (/dev/pts/2) of the spawned child process /bin/sh is different from the one (pts/1) of the script itself in this example.

3.3.3 Automated login using the serial connection

Example 3-7 shows an example script that does the following:

1. Spawns the **cu** command to log in through the /dev/tty0 device (line 8).
2. Waits for the Connected string from the **cu** command (line 12).
3. Types Enter several times to flush out the garbled messages on the standard input until the login prompt is received (lines 14 - 18).
4. Sends the login user name stored in the $userid variable (line 23).
5. Sends the root user's password stored in the password variable (line 27).
6. Waits for the root user's prompt string on the remote system (line 29).
7. Starts to write to the log file on the local system using the log_file function (line 31).
8. Sends the following command to be executed on the remote system (line 33):

   ```
   LANG=C errpt -d H
   ```

9. Waits again for the root user's prompt string on the remote system (line 35).
10. Stops to write to the log file on the local system (line 37).
11. Prints the message done. (line 41).
12. Exits with exit code 0. (line 42).

Example 3-7 Automated login script using Perl/Expect

```
 1: #!/usr/bin/perl
 2: use Expect;
 3: $userid = "root";
 4: $password = "XXXXXX";
 5: $logfile = "./hwerr.log";
 6:
 7: # Spawn the cu command to access the serial port.
 8: $e = Expect->spawn("cu -l tty0");
 9:
10: # You should recieve "Connected" message string from cu,
11: # upon successful openning of the serial port.
12: &timeout unless($e->expect(10, "Connected"));
```

```

# Type ENTER until get login prompt.
for ($c=0; $c<5; $c++) {
    $e->send("\r");
    last loginp if ($e->expect(10, 'login: '));
}
&timeout;

loginp:
# Type userid<ENTER>
$e->send("$userid\r");
# Wait for 'Password:' prompt.
&timeout unless ($e->expect(10, 'Password:'));
# Type password<ENTER>
$e->send("$password\r");
# Wait for prompt
&timeout unless ($e->expect(10, "# "));
# Turn logging on
$e->log_file($logfile, "w");
# Type errpt command
$e->send("LANG=C errpt -d H\r");
# Wait for prompt
&timeout unless ($e->expect(10, "# "));
# Turn logging off
$e->log_file(undef);

sleep(1);
$e->hard_close();
printf("done.\n");
exit 0;

sub timeout {
    sleep(1);
    $e->hard_close();
    printf("timeout!\n");
    exit 1;
}
```

Note: The script contains the root user’s password in clear text format, as shown in the high-lighted line. Therefore, you should securely store the script. The script must have permission mode 0700, as a minimum security.

Example 3-8 on page 83 shows the output from this script in our environment.

Example 3-8 Output from the automated login script

```
Connected

... removed output message lines ...

AIX Version 5
(C) Copyrights by IBM and by others 1982, 1996.
login:

... removed output message lines ...

AIX Version 5
(C) Copyrights by IBM and by others 1982, 1996.
login: root
root's Password:
********************************************************************************
*                                                                              *
*                                                                              *
*  Welcome to AIX Version 5.1!                                                 *
*                                                                              *
*                                                                              *
*  Please see the README file in /usr/lpp/bos for information pertinent to     *
*  this release of the AIX Operating System.                                   *
*                                                                              *
*                                                                              *
********************************************************************************
Last unsuccessful login: Mon Apr  1 17:46:46 2002 on /dev/pts/0 from
tcenter1.hakozaki.ibm.com
Last login: Fri Apr  5 22:03:38 2002 on /dev/tty0

# LANG=C errpt -d H
IDENTIFIER TIMESTAMP  T C RESOURCE_NAME  DESCRIPTION
75CE5DC5   0405112402 I H tok0           ADAPTER ERROR
DADF69E4   0405112302 U H SYSINTR        UNDETERMINED ERROR
75CE5DC5   0405112302 I H tok0           ADAPTER ERROR
071F4755   0404094802 P H sysplanar0     ENVIRONMENTAL PROBLEM
75CE5DC5   0325092502 I H tok0           ADAPTER ERROR
0BA49C99   0323030102 T H scsi0          SCSI BUS ERROR
# done.
```

On the local system, the hwerr.log file also holds the same content with the **errpt** command output in Example 3-8 as shown in the following example:

```
LANG=C errpt -d H
IDENTIFIER TIMESTAMP  T C RESOURCE_NAME  DESCRIPTION
75CE5DC5   0405112402 I H tok0           ADAPTER ERROR
DADF69E4   0405112302 U H SYSINTR        UNDETERMINED ERROR
75CE5DC5   0405112302 I H tok0           ADAPTER ERROR
```

```
071F4755   0404094802 P H sysplanar0     ENVIRONMENTAL PROBLEM
75CE5DC5   0325092502 I H tok0           ADAPTER ERROR
0BA49C99   0323030102 T H scsi0          SCSI BUS ERROR
#
```

3.3.4 Power on server using Perl/Expect

You can power on a remote AIX system using Perl/Expect by accessing the service processor firmware menu on the integrated serial port of the target system. It is useful when you have to power on the system without accessing the physical reset button. To use this function, the target system must be:

- Equipped with the service processor

  Some older RS/6000 models are not equipped with the service processor.

- Plugged into the power outlet

  While the system is plugged into the power outlet, the service processor is awake and displays the configuration menu on the integrated serial ports.

- AIX operating system shut down

  The service processor does not interact with the integrated serial ports while the operating system is running.

Before writing your script, you have to be familiar with the service processor menu of the target system, because the service processor menu depends on the individual pSeries and RS/6000 models.

Example 3-9 is an example script that remotely powers on the RS/6000 model 44P-270 server.

Example 3-9 Power on script using Perl/Expect

```
#!/usr/bin/perl
use Expect;

# Execute cu to get access the firmware menu using the serial connection.
$e = Expect->spawn("cu -l tty0");

# You should recieve "Connected" message from cu, when you successfully
# open the serial port.
&timeout unless($e->expect(10, "Connected"));

# Type Enter anyway.
$e->send("\r");

# Let's see on which menu we are on.
while (1) {
    $match=$e->expect(10, "MAIN MENU"
        , "Press Return to Continue", "Return to Previous Menu");
    if ($match == 1) {
        # We are in the main menu. Wait for '>' prompt and go to the next.
        &timeout unless ($e->expect(10, "-re", "[0-9]>"));
last;
    } elsif ($match == 2) {
        # We see the prompt to confirm the error message, type Enter.
        $e->send("\r");
    } elsif ($match == 3) {
        # We are in the sub-menu, wait for '>' prompt.
        &timeout unless ($e->expect(10, "-re", "[0-9]>"));
        # Type 99<Enter>.
        $e->send("99\r");
        # Wait for a second.
        sleep(1);
        # Type Enter to go back to the main menu.
        $e->send("\r");
    } else {
        &timeout;
    }
}

printf("\n==== I'm in the top menu now! ====\n");
# Type 2<Enter> to go to POWER CONTROL MENU.
$e->send("2\r");
printf("\n==== try to go to power control menu ====\n");
# Wait for '>' prompt.
&timeout unless ($e->expect(10, "-re", "[0-9]>"));
printf("\n==== I'm in the power menu now! ====\n");
# Type 4<Enter> to choose POWER ON.
$e->send("4\r");
printf("\n==== try to power on. ====\n");
# Wait for the confirmation prompt.
&timeout unless ($e->expect(10, "to continue, any other key to abort."));
# Type y to continue.
$e->send("y\r");
# Wait until cu is disconnected by the remote system.
&timeout unless ($e->expect(600, "Lost carrier"));

sleep(1);
$e->hard_close();
printf("done.\n");
exit 0;

sub timeout{
    sleep(1);
    $e->hard_close();
    printf("timeout!\n");
    exit 1;
```

```
66: }
```

Because the service processor menu is composed of several sub-panels, your script has to remember where it is currently located. Especially for the first connection time, the script cannot assume that it is in a sub-panel. Therefore, your script has to type Enter or the **exit** command to go back to the main menu (line 15 - 37).

In line 16 and 17, you see three strings that are specified as the pattern argument of the expect() function:

```
16:     $match=$e->expect(10, "MAIN MENU"
17:         , "Press Return to Continue", "Return to Previous Menu");
```

You can specify multiple patterns in the expect() function. If the first pattern matches with the input, the function returns the value 1; if the second pattern matches, the function returns the value 2, and so on.

The firmware menu prompt string depends on which serial port is used for access on the target system. If you see the prompt 1>, then you are logging in from the first integrated serial port on the system. If you see 2>, then the second integrated serial port is used. To simply manage this varying prompt string, you can use the “-re” option of the expect() function, as shown in the following example (line 27 and 44):

```
$e->expect(10, "-re", "[0-9]>")
```

If you specify the “-re” option as the second argument of the expect() function, it instructs the script to treat the third argument (pattern) as a regular expression. Therefore, this line returns with any number of the prompt string.

Example 3-10 shows the output of this script on the RS/6000 model 44P-270 server.

Example 3-10   Example output of power on script by Perl/Expect

```
Connected

Illegal Value Entered
                   (Press Return to Continue)

... removed lines on purpose ...

Service Processor Firmware
                           Firmware level: sh000808
                       Copyright 2000, IBM Corporation


                              MAIN MENU
```

```
           1. Service Processor Setup Menu
           2. System Power Control Menu
           3. System Information Menu
           4. Language Selection Menu
           5. Call-In/Call-Out Setup Menu
           6. Set System Name
          99. Exit from Menus

 1>
==== I'm in the top menu now! ====
 2
==== try to go to the power control menu ====

... removed lines on purpose ...

SYSTEM POWER CONTROL MENU

         1. Enable/Disable Unattended Start Mode:
              Currently Disabled

         2. Ring Indicate Power-On Menu
         3. Reboot/Restart Policy Setup Menu
         4. Power-On System
         5. Power-Off System
         6. Enable/Disable Fast System Boot:
              Currently Enabled

         7. Boot Mode Menu
        98. Return to Previous Menu
        99. Exit from Menus

 1>
==== I'm in the power menu now! ====
4

==== try to power on. ====
      WARNING:  POWERING SYSTEM WILL EXIT MENUS!
  Enter "Y" to continue, any other key to abort.y
System Powering On.

E212
E211
E306
E307
E308
E309
E315
E316
```

```
E317
E318
E3E2
E302
E303
E304
E311
E312
E313
E220
E216
Lost carrier.
done.
```

3.4 Using PPP

PPP stands for Point-to-Point Protocol, which provides a standard method for transporting a multiprotocol datagram over point-to-point media. Typically, PPP is used to establish TCP/IP network connection over the serial connections. PPP is comprised of three main layers:

- A method for encapsulating multiprotocol datagrams. PPP supports the TCP/IP network layer protocols.
- A Link Control Protocol (LCP) for establishing, configuring, and testing the data-link connection. PPP implements this through streams kernel extensions.
- A family of Network Control Protocols (NCPs) for establishing and configuring different network layer protocols. PPP supports Internet Protocol Control Protocol (IPCP) for negotiating a TCP/IP connection.

For further information about serial communications, please refer to the AIX 5L Version 5.1 *System Management Guide: Communications and Networks*.

Note: Before deploying PPP in your server farms, you have to understand the network security policy required at your site. PPP is quite useful and convenient for administrators, but it might be a security weakness. We strongly recommend you implement the dial-out PPP configuration only in your server farms, for example, to be used for Service Agent (explained in Section 3.5.1, “Understanding Service Agent” on page 105).

3.4.1 Configuring PPP on AIX

AIX PPP differentiates between client and server. An AIX system can act as both a client and a server. The distinction is made to simplify configuration. PPP servers tend to allocate a pool of IP addresses among the connections that are being made. There is some correlation between the media devices, and AIX PPP breaks this correlation. All PPP server connections are allocated on a first-available basis. This facilitates the separation of PPP from the media. The attachment process must make a request to be linked to the proper type of link.

You can use the Web-based System Manager or SMIT to configure the Asynchronous Point-to-Point Protocol. At a minimum, when you initially configure your system, you must choose the following tasks. You must have root privileges to perform these tasks.

- Add a Link Configuration
- Add a Server Interface (if you are setting up the machine as a PPP server)
- Add a Demand Interface (if you want the machine to support demand connections)
- Manipulate PAP or CHAP Users/Passwords (if you want the machine to support PPP authentication)
- Start PPP to effect your changes (or Stop then Start PPP, if PPP is currently running)

When setting up a PPP server, the answering pppattachd daemon is started from the profile script of a special PPP user. The security of this user is dependent on the type of Internet server you are using and who will know the password of the server. In many cases, you will want this special user ID only available while starting a PPP session. This is especially true if you are setting up a server for outside users to dial into. Special tasks specific to the PPP user include:

- Make this user a member of the uucp group.
- Do not allow other users to `su` to this user.
- Do not allow a remote login to the user (telnet or rlogin).
- Disable FTP to the user.

Configuration example

We explain how to configure PPP on AIX in the following steps. In this example, we configure svr01 as a PPP client (IP address: 10.0.2.2 / 255.255.255.0) and to svr02 as a PPP server (IP address 10.0.2.1 / 255.255.255.0), as shown in Figure 3-6.

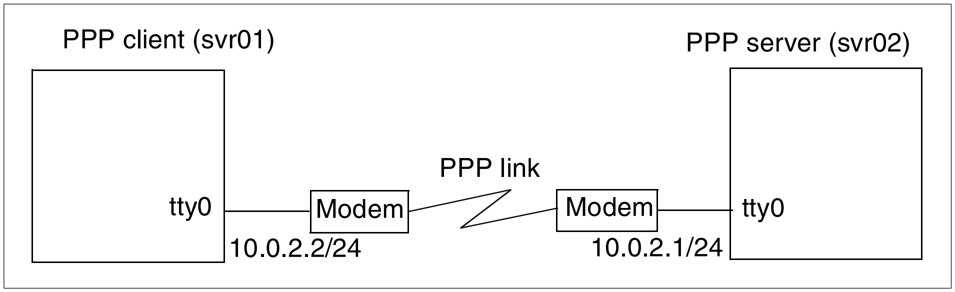


Figure 3-6 PPP sample configuration

Verifying the TTY and modem configurations

1. Verify that the required fileset is installed on both server and client, as shown in the following example:

```
# lslpp -l | grep bos.net.ppp
  bos.net.ppp            5.1.0.10  COMMITTED  Async Point to Point Protocol
```

2. Verify the modem and TTY configurations to be used for the PPP connection on both sides. Before proceeding the following steps, we strongly recommend that you test the connectivity between systems using the **ate** command explained in Section 3.2.3, “Using the ATE program” on page 69.

   The required AT command sequences to be used for the PPP connection depend on your modem. In this example, we used the following AT command sequences to configure modems on both side:

```
at&f
OK
ate1
OK
at&d2
OK
at&c1
OK
ats0=1
OK
ats9=12
OK
atx0
OK
```

```
at&a0
OK
at&w
~.
```

Sometimes you have to modify the AT command sequences so that you can avoid chat script problem in later phase. In our example, we have to add the following two AT commands to be used for the PPP connection, in addition to the command sequences required in the **ate** command example:

ATX0 Disable to echo back the 9600 string with a result code from the modem.

AT&A0 Disable to echo back the /ARQ string with a result code from the modem.

The Enable LOGIN parameter of the TTY device must be set Enable on the PPP server (svr02) and Disable on the PPP client (svr01).

Configuring the PPP server

1. First, you have to create the PPP link control configuration by selecting the following SMIT panels:

```
# smitty
    Communications Applications and Services
        PPP
            Link Control Configuration
                Add a Link Configuration
```

   Specify the following values for the parameters on the SMIT panel, as shown in Example 3-11, then press Enter:

   - PPP subsystem name: ppp_srv
   - Max server connections: 1
   - Max client connections: 0
   - Max demand interface: 0
   - Max IP interface: 1
   - Max async HDLC attachments:1

Example 3-11 LINK Configuration on PPP server

```
                          LINK Configuration

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

[TOP]                                                    [Entry Fields]
  PPP subsystem name                                  [ppp_srv]
  max server connections                              [1]                    #
  max client connections                             [0]                  #
  max demand connections                             [0]                  #
  max ip interfaces                                  [1]                  #
  max async hdlc attachments                         [1]                  #
  mru                                                []                   #
  async character map                                []                   #
  transmit async character map                       []                  #
  negotiate MRU                                       yes                 +
  negotiate magic number                              yes                 +
  negotiate async map                                 yes                 +
  negotiate protocol compression                      yes                 +
[MORE...8]

F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         Esc+6=Command       Esc+7=Edit          Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do
```

This panel will generate the configuration file /etc/ppp/lcp_config, as shown in the following example:

```
root@svr02:/ [102] # cat /etc/ppp/lcp_config
server_name ppp_srv
lcp_server 1
lcp_client 0
lcp_demand 0
num_if 1
num_hdlc 1
```

2. You have to assign an IP address for the server by selecting the following SMIT panels:

```
# smitty
    Communications Applications and Services
        PPP
            PPP IP Interfaces
                Add a Server Interface
```

Specify the following values for the parameters on the SMIT panel, as shown in Example 3-12, then press Enter:

– Local IP address: 10.0.2.1

– Starting Remote IP address: 10.0.2.2

– Number of addresses: 1

– Netmask: 255.255.255.0

Example 3-12 PPP IP Configuration

```
                          PPP IP Configuration

Type or select values in entry fields.
```

```
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
  Local IP address                                 [10.0.2.1]
  Starting Remote IP address                       [10.0.2.2]
  Number of addresses                             [1]
#
  Netmask                                          [255.255.255.0]

F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         Esc+6=Command       Esc+7=Edit          Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do
```

This panel will generate the configuration file /etc/ppp/if_conf, as shown in the following example:

```
root@svr02:/ [124] # cat /etc/ppp/if_conf
interface
server
local_ip 10.0.2.1
remote_ip 10.0.2.2
netmask 255.255.255.0
```

3. Create a PPP login user (pppuser) by selecting the following SMIT panels:

```
# smitty
    Security & Users
        Users
            Add a User
```

Specify the following values for the parameters on the SMIT panel, as shown in Example 3-13, then press Enter:

- User NAME: pppuser
- Primary Group: uucp
- Group Set: uucp
- Administrative Groups: uucp
- Another user can SU to user?: false
- SU Groups: nobody
- User can login?: true
- User can login remotely?: false
- Valid TTYs: /dev/tty0

Example 3-13 Add a User

```
                                  Add a User

Type or select values in entry fields.
```

```
Press Enter AFTER making all desired changes.

[TOP]                                                      [Entry Fields]
* User NAME                                             [pppuser]
  User ID                                               []                      #
  ADMINISTRATIVE USER?                                   false                  +
  Primary GROUP                                         [uucp]                  +
  Group SET                                             [uucp]                  +
  ADMINISTRATIVE GROUPS                                 [uucp]                  +
  ROLES                                                 []                      +
  Another user can SU TO USER?                           false                  +
  SU GROUPS                                             [nobody]                +
  HOME directory                                        []
  Initial PROGRAM                                       []
  User INFORMATION                                      []
  EXPIRATION date (MMDDhhmmyy)                          [0]
  User can LOGIN?                                        true                   +
  User can LOGIN REMOTELY(rsh,tn,rlogin)?                false                  +
  Allowed LOGIN TIMES                                   []
  Number of FAILED LOGINS before                        [0]                     #
       user account is locked
  Login AUTHENTICATION GRAMMAR                          [compat]
  Valid TTYs                                            [/dev/tty0]
[MORE...29]

F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         Esc+6=Command       Esc+7=Edit          Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do
```

After you press Enter, you will be prompted twice to assign a password for the pppuser. In this example, we entered ppp for the password.

4. Modify the pppuser’s .profile to start pppattachd. To do so, add the following line to ~ppuser/.profile:

   ```
   exec /usr/sbin/pppattachd server 2>/dev/null
   ```

5. Issue the following command to prohibit the password change prompt upon the first login attempt, because the chat script cannot handle it:

   ```
   # pwdadm -f NOCHECK pppuser
   ```

6. To capture diagnostic output during the connection attempt phase, do the following:

   a. Add the following line to the /etc/syslog.conf file:

      ```
      *.debug            /tmp/ppp
      ```

   b. Create the /tmp/ppp file:

      ```
      # touch /tmp/ppp; chmod u+w /tmp/ppp
      ```

   c. Refresh the syslogd daemon to reread /etc/syslog.conf:

```
# refresh -s syslogd
```

7. To start the pppcontrold daemon, select the following SMIT panels:

```
# smitty
    Communications Applications and Services
        PPP
            Start PPP
```

   Verify the value is `both` in the only field on the SMIT panel, as shown in Example 3-14, then press Enter:

Example 3-14 Start PPP

```
                               Start PPP

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

                                                        [Entry Fields]
* START PPP now, on system restart or both            both                    +

F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         Esc+6=Command       Esc+7=Edit          Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do
```

8. Confirm the pppcontrold process is running, as shown in the following example:

```
root@svr02:/ [154] # lssrc -s pppcontrold
Subsystem         Group            PID     Status
 pppcontrold      uucp             16320   active
root@svr02:/ [155] # ps -ef | grep pppcontrold | grep -v grep
root 16320  3920   0 14:04:53      -  0:00 /usr/sbin/pppcontrold
```

Configuring the PPP client

1. First, you have to create the PPP link control configuration by selecting the following SMIT panels:

```
# smitty
    Communications Applications and Services
        PPP
            Link Control Configuration
                Add a Link Configuration
```

   Specify the following values for the parameters on the SMIT panel, as shown in Example 3-11 on page 91, then press Enter:

   – PPP subsystem name: ppp_cli

   – Max server connections: 0

  - Max client connections: 1
  - Max demand interface: 0
  - Max IP interface: 1
  - Max async HDLC attachments:1

Example 3-15 LINK Configuration on PPP client

```
                         LINK Configuration

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

[TOP]                                                    [Entry Fields]
  PPP subsystem name                                  [ppp_cli]
  max server connections                              [0]                     #
  max client connections                              [1]                     #
  max demand connections                              [0]                     #
  max ip interfaces                                   [1]                     #
  max async hdlc attachments                          [1]                     #
  mru                                                 []                      #
  async character map                                 []                      #
  transmit async character map                        []                     #
  negotiate MRU                                        yes                    +
  negotiate magic number                               yes                    +
  negotiate async map                                  yes                    +
  negotiate protocol compression                       yes                    +
[MORE...8]

F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         Esc+6=Command       Esc+7=Edit          Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do
```

2. Create the /etc/ppp/if_conf file, as shown in the following example:

```
root@svr01:/ [165] # cat /etc/ppp/if_conf
interface
client
local_ip 10.0.2.2
remote_ip 10.0.2.1
netmask 255.255.255.0
```

   You will notice the second line is changed to *client*, from *server* in the /etc/ppp/if_conf file on the PPP server.

3. Start the pppcontrold daemon using the same step used in the server configuration. Then you will see the pp0 interface, as shown in Example 3-16 on page 97:

Example 3-16 Configured pp0 network interface

```
root@svr01:/ [167] # ifconfig pp0
pp0: flags=e000030<POINTOPOINT,NOTRAILERS,GROUPRT,64BIT>
        inet 10.0.2.2 --> 10.0.2.1 netmask 0xffffff00
```

You never succeed in pinging the local interface, as shown in the following example:

```
root@svr01:/ [172] # ping 10.0.2.2
PING 10.0.2.2: (10.0.2.2): 56 data bytes
... wait several minutes ...
^C
----10.0.2.2 PING Statistics----
6 packets transmitted, 0 packets received, 100% packet loss
```

The reason is that the pp0 interface is brought up with the POINTTOPOINT flag, as shown in Example 3-16. A network interface, which is set with the POINTTOPOINT flag, does not response to the ICMP echo messages.

You also do not succeed in pinging the remote interface, as shown in the following example, simply because there is no established PPP connection so far:

```
root@svr01:/ [168] # ping 10.0.2.1
PING 10.0.2.1: (10.0.2.1): 56 data bytes
0821-069 ping: sendto: The network is not currently available.
ping: wrote 10.0.2.1 64 chars, ret=-1
```

4. To capture diagnostic output during the connection attempt phase, do the same step used in the server configuration.
5. Now you have to create a chat script that initiates dial-out calls from the PPP client to the PPP server. AIX provides a template script, /etc/ppp/dial_out.example. To create your own chat script, do the following:

   a. Create a copy of this template to another file, as shown in the following example:

   ```
   # cp -p /etc/ppp/dial_out.example /etc/ppp/my_dial
   ```

   b. Modify some values in the script to adapt the script to your PPP configuration. In this example, we modified the following lines in /etc/ppp/my_dial:

   ```
   USER=pppuser
   PASSWORD=ppp
   NUMBER=XXX-YYYY          # phone number of svr02
   TTY=tty0                 # tty device connected to the modem on svr01.
   CHATFILE=./dial_svr02
   ```

c. Edit the line of the /usr/sbin/pppattachd, inserting the word `exec` at the beginning of the line so that it can start the dial-up connection, as shown in the following example:

```
exec /usr/sbin/pppattachd client $TTY connect "/usr/sbin/pppdial -v -f
$CHATFILE"
```

This command reads an actual chat script to initiate dial-up calls. The actual chat script file is specified by the variable $CHATFILE (in this example, /etc/ppp/dial_svr02). The file is automatically generated by the /etc/ppp/my_dial script upon every connection attempt. Therefore, direct editing of this file does not make sense.

d. Customize the etc/ppp/my_dial file so that it will generate an appropriate chat script. In this example, we edit the file as in the following example:

```
\"\”                 Expect nothing.
at                   Send a basic AT command to modem.
OK                   Expect that modem responds OK.
atdt$NUMBER          Dial out svr02 phone number.
CONNECT              Expect that modem responds CONNECT.
\"\"                 Send a null string followed by Carriage Return or ^M.
ogin:                Eexpect the string “ogin:”.
$USER                Send user name, in this example pppuser.
\"\"                 Eexpect nothing from modem.
ssword:              Expect the string “ssword:”.
$PASSWORD            Send password, in our example ppp.
```

Note: The comments in the example are presented for illustrative purposes only. You cannot write your comments in the actual script.

Establishing the PPP connection

Once you run the /etc/ppp/my_dial script on the PPP client, it automatically generates the chat script /etc/ppp/dial_svr02 and invokes /usr/sbin/pppattachd, then the pppattached process reads the chat script to establish the PPP connection, as shown in Figure 3-6 on page 90.

You can verify that the PPP connection is working correctly by issuing the **`ping`** command to send ICMP echo messages to the PPP server from the client, as shown in the following example:

```
root@svr01:/ [353] # ping 10.0.2.1
PING 10.0.2.1: (10.0.2.1): 56 data bytes
64 bytes from 10.0.2.1: icmp_seq=0 ttl=255 time=316 ms
64 bytes from 10.0.2.1: icmp_seq=1 ttl=255 time=316 ms
64 bytes from 10.0.2.1: icmp_seq=2 ttl=255 time=315 ms
64 bytes from 10.0.2.1: icmp_seq=3 ttl=255 time=316 ms
^C
----10.0.2.1 PING Statistics----
```

```
4 packets transmitted, 4 packets received, 0% packet loss
```

You never succeed in pinging the local IP address of the pp0 interface, even after the PPP connection is established, as shown in the following example:

```
root@svr01:/ [354] # ping 10.0.2.2
PING 10.0.2.2: (10.0.2.2): 56 data bytes
   ... wait several minutes ...
^C
----10.0.2.2 PING Statistics----
5 packets transmitted, 0 packets received, 100% packet loss
```

Problem determination

Once you establish the PPP connection, you see the detailed messages logged by syslog in the /tmp/ppp file, as shown in Example 3-17.

Example 3-17 Successful PPP connection syslog messages

```
Apr  9 10:52:23 svr01 pppattachd[4074]: starting attachment daemon
Apr  9 10:52:23 svr01 pppattachd[4720]: open /dev/tty0
Apr  9 10:52:23 svr01 pppdial[4076]: send (at^M)
Apr  9 10:52:23 svr01 pppdial[4076]: expect (OK)
Apr  9 10:52:23 svr01 pppdial[4076]: at^M^M
Apr  9 10:52:23 svr01 pppdial[4076]: OK -- got it
Apr  9 10:52:23 svr01 pppdial[4076]: send (atdtXXX-YYYY^M)
Apr  9 10:52:23 svr01 pppdial[4076]: expect (CONNECT)
Apr  9 10:52:23 svr01 pppdial[4076]: ^M
Apr  9 10:52:39 svr01 pppdial[4076]: atdtXXX-YYYY^M^M
Apr  9 10:52:39 svr01 pppdial[4076]: CONNECT -- got it
Apr  9 10:52:39 svr01 pppdial[4076]: send (^M)
Apr  9 10:52:39 svr01 pppdial[4076]: expect (ogin:)
Apr  9 10:52:39 svr01 pppdial[4076]: ^M
Apr  9 10:52:39 svr01 pppdial[4076]: ^M
Apr  9 10:52:40 svr01 pppdial[4076]:
Apr  9 10:52:40 svr01 last message repeated 24 times
Apr  9 10:52:40 svr01 pppdial[4076]: ^MAIX Version 5
Apr  9 10:52:40 svr01 pppdial[4076]: ^M(C) Copyrights by IBM and by others
1982, 2000.
Apr  9 10:52:40 svr01 pppdial[4076]: ^Mlogin: -- got it
Apr  9 10:52:40 svr01 pppdial[4076]: send (pppuser^M)
Apr  9 10:52:40 svr01 pppdial[4076]: send (ssword:^M)
Apr  9 10:52:40 svr01 pppdial[4076]: expect (ppp)
Apr  9 10:52:40 svr01 pppdial[4076]:
Apr  9 10:52:40 svr01 pppdial[4076]: ^M
Apr  9 10:52:40 svr01 pppdial[4076]:
Apr  9 10:52:40 svr01 last message repeated 23 times
Apr  9 10:52:40 svr01 pppdial[4076]: ^MAIX Version 5
Apr  9 10:52:40 svr01 pppdial[4076]: ^M(C) Copyrights by IBM and by others
1982, 2000.
```

```
Apr  9 10:52:40 svr01 pppdial[4076]: ^Mlogin: ppp -- got it
Apr  9 10:52:41 svr01 pppattachd[4720]: attachd name
Apr  9 10:52:41 svr01 pppattachd[4720]: attachment connection established
Apr  9 10:52:42 svr01 /usr/sbin/pppcontrold[16818]: msgid badebe01
Apr  9 10:52:42 svr01 /usr/sbin/pppcontrold[16818]: LOWERUP 5dc
Apr  9 10:52:42 svr01 /usr/sbin/pppcontrold[16818]: msgid badebe03
Apr  9 10:52:42 svr01 /usr/sbin/pppcontrold[16818]: msgid badebe03
Apr  9 10:52:42 svr01 pppauthd[15270]: UPAP init
Apr  9 10:52:42 svr01 pppauthd[15270]: ChapInit
Apr  9 10:52:42 svr01 pppauthd[15270]: upap_lowerup
Apr  9 10:52:45 svr01 /usr/sbin/pppcontrold[16818]: msgid badebc03
Apr  9 10:52:45 svr01 /usr/sbin/pppcontrold[16818]: /etc/ifconfig pp0 10.0.2.2 
10.0.2.1 netmask 255.255.255.0 >/dev/null 2>&1
```

This log file is very useful diagnosing problems caused by your chat script. For example, the default /etc/ppp/dial_out.example has the lines to generate a chat script, as shown in Example 3-18.

Example 3-18   Generating a chat script from /etc/ppp/dial_out.example

```
#
#  The following script creates a generic chat script for general modem
#  connection.  Depending on your modem settings, you may have to add more
#  text to the chat file to set your modem.
#
cat << EOF > $CHATFILE
\"\"
at
OK
atdt$NUMBER
CONNECT
\"\"
ogin
$USER
ssword
$PASSWORD
EOF
```

In our environment, if we use this script without modification, we could not establish the PPP connection, and syslog logged many garbled messages in /etc/ppp, as shown in Example 3-19:

Example 3-19   Unsuccessful PPP connection syslog messages

```
Apr  9 11:02:44 svr01 pppdial[4726]: ^MAIX Version 5
Apr  9 11:02:44 svr01 pppdial[4726]: ^M(C) Copyrights by IBM and by others 
1982, 2000.
Apr  9 11:02:44 svr01 pppdial[4726]: ^Mlogin: pppuser^M
Apr  9 11:02:44 svr01 pppdial[4726]:
*******************************************************************************
^M
Apr  9 11:02:44 svr01 pppdial[4726]: * *^M
Apr  9 11:02:44 svr01 pppdial[4726]: * *^M
Apr  9 11:02:44 svr01 pppdial[4726]: *  Welcome to AIX Version 5.1! *^M
Apr  9 11:02:44 svr01 pppdial[4726]: * *^M
Apr  9 11:02:44 svr01 pppdial[4726]: * *^M
Apr  9 11:02:44 svr01 pppdial[4726]: *  Please see the README file in
/usr/lpp/bos for information pertinent to    *^M
Apr  9 11:02:44 svr01 pppdial[4726]: *  this release of the AIX Operating
System.                                  *^M
Apr  9 11:02:44 svr01 pppdial[4726]: * *^M
Apr  9 11:02:45 svr01 pppdial[4726]: * *^M
Apr  9 11:02:45 svr01 pppdial[4726]:
******************************************************************************^
M
Apr  9 11:02:45 svr01 pppdial[4726]: Last unsuccessful login: Sat Apr  6
06:39:4 8 CST 2002 on /dev/pts/7 from loopback^M
Apr  9 11:02:45 svr01 pppdial[4726]: Last login: Tue Apr  9 10:56:15 CDT 2002
on /dev/tty1^M
Apr  9 11:02:45 svr01 pppdial[4726]: ^M
Apr  9 11:03:20 svr01 pppdial[4726]: ~^?}#@!}!}!} }8}!}$}\}"}&} } } }
}}&l7R}/}'
}"}(}"yR~~^?}#@!}!}!} }8}!}$}\}"}&} } } } }}&l7R}/}'}"}(}"yR~~^?}#@!}!}!}
}8}!}$
}\}"}&} } } } }}&l7R}/}'}"}(}"yR~~^?}#@!}!}!} }8}!}$}\}"}&} } } }
}}&l7R}/}'}"}(
}"yR~~^?}#@!}!}!} }8}!}$}\}"}&} } } } }}&l7
Apr  9 11:03:20 svr01 pppdial[4726]: NO CARRIER^M
Apr  9 11:03:29 svr01 pppdial[4726]: 0838-102 alarm occurred
Apr  9 11:03:29 svr01 pppdial[4726]: 0838-086 Expect failed cause unknown
Apr  9 11:03:29 svr01 pppattachd[3856]: 0838-021 Connection  program failed
```

To establish the PPP connection, we had to modify /etc/ppp/my_dial by adding the following line right after the $USER line:

```
\"\"
```

Note: We strongly recommend you modify /etc/syslog.conf to not generate detailed messages after you have established the PPP connection, because the syslog file contains the dial-out number (see the high-lighted lines in Example 3-17).

3.4.2 Using PPP authentication protocols

Although it is not mandatory, we recommend you use an authentication protocol named CHAP to configure the PPP connection. AIX implements the following two PPP authentication protocols: PAP and CHAP.

PAP Password Authentication Protocol is the simplest of the PPP authentication protocols. User credentials are transmitted in plain text at the beginning of a PPP session. At the request of an authenticator, the client responds with both a PAP name and a password in a single transaction. The authenticator validates this information and replies with a positive or negative acknowledgment.

CHAP Challenge Handshaking Authentication Protocol is a more secure PPP authentication protocol. It performs an authentication process that requires a challenge and a response. A CHAP authenticator challenges its peer with its CHAP name and a random string. The client must transform this random string with a computation algorithm and a CHAP secret key. It then returns the result with its own name. The challenger evaluates the reply with its own copy of the secret key. Then it forwards a success or failure acknowledgment. The challenge response and response computation are all built inside PPP software. Users need to supply only a CHAP name and secret key known by both sides of the connection. The important security characteristic of CHAP is that PPP endpoints never send keys in plain text upon the PPP connection attempt.

By using CHAP for the PPP connections, not only can you secure the PPP authentication, but you can also use the IPCP negotiation. Therefore, you do not have to create the /etc/ppp/if_conf file to assign an IP address on the PPP client.

Using CHAP

To use CHAP for the PPP connection, do the following, in addition to the steps explained in Section 3.4.1, “Configuring PPP on AIX” on page 89:

1. Create a CHAP user on both sides by selecting the following SMIT panels:

   ```
   # smitty
       Communications Applications and Services
           PPP
               CHAP Authentication
                   Add a User
   ```

   Specify the following values for the parameters on the SMIT panel, as shown in Example 3-20, then press Enter:

   Configure the shown fields as follows:

   – Peer name: ppp_cli

     Specify the same name used in the PPP subsystem name on the PPP client.

   – Authenticator name: ppp_srv

     Specify the same name used in the PPP subsystem name on the PPP server.

   – Password: chap

     Specify an arbitrary string. The password must be same on both sides.

Example 3-20 Add a CHAP User

```
                                Add a CHAP User

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

                                                        [Entry Fields]
  Peer name                                          [ppp_cli]
  Authenticator name                                 [ppp_srv]
  Password                                           [chap]

F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         Esc+6=Command       Esc+7=Edit          Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do
Type or select values in entry fields.
Press Enter AFTER making all desired changes.
```

2. Modify the ~pppuser/.profile on the PPP server to instruct pppattached to use CHAP, as shown in the following example:

   ```
   exec /usr/sbin/pppattachd server authenticate chap 2>/dev/null
   ```

3. Modify the /etc/ppp/my_dial file on the PPP client, as shown in the following example:

   ```
   exec /usr/sbin/pppattachd client $TTY peer chap connect "/usr/sbin/pppdial
   -v -f $CHATFILE"
   ```

4. Remove the /etc/ppp/if_conf on the PPP client. If you use CHAP for the PPP connection, the IP address on the PPP client is assigned from the PPP server through the IPCP negotiation. Therefore, you do not have to have this file on the PPP client.
5. Restart the pppcontrold subsystem on the both sides using the following commands:

   ```
   # stopsrc -s pppcontrold
   # startsrc -s pppcontrold
   ```

Now you are ready to use CHAP for the PPP connection. If you invoke /etc/ppp/my_dail, then you will see the messages in /tmp/ppp logged by syslog, as shown in Example 3-21.

Example 3-21 Successful CHAP connection

```
Apr 10 22:07:56 svr01 pppdial[16798]: ^MAIX Version 5
Apr 10 22:07:56 svr01 pppdial[16798]: ^M(C) Copyrights by IBM and by others
1982, 2000.
Apr 10 22:07:56 svr01 pppdial[16798]: ^Mlogin: ppp -- got it
Apr 10 22:07:57 svr01 pppattachd[15318]: attachd name
Apr 10 22:07:57 svr01 pppattachd[15318]:  attachment connection established
Apr 10 22:07:58 svr01 pppauthd[18844]: UPAP init
Apr 10 22:07:58 svr01 pppauthd[18844]: ChapInit
Apr 10 22:07:58 svr01 pppauthd[18844]: upap_lowerup
Apr 10 22:07:58 svr01 pppauthd[18844]: Remote message: 0838-261 Welcome to PPP.
Apr 10 22:07:58 svr01 /usr/sbin/pppcontrold[9316]: msgid badebe01
Apr 10 22:07:58 svr01 /usr/sbin/pppcontrold[9316]: LOWERUP 5dc
Apr 10 22:07:58 svr01 /usr/sbin/pppcontrold[9316]: msgid badebe03
Apr 10 22:07:58 svr01 /usr/sbin/pppcontrold[9316]: msgid badebe03
Apr 10 22:07:58 svr01 pppauthd[18844]: Remote message: 0838-261 Welcome to PPP.
Apr 10 22:07:58 svr01 /usr/sbin/pppcontrold[9316]: msgid badebc03
Apr 10 22:07:58 svr01 /usr/sbin/pppcontrold[9316]: /etc/ifconfig pp0 10.0.2.2
10.0.2.1 >/dev/null 2>&1
Apr 10 22:08:03 svr01 pppauthd[18844]: Remote message: 0838-261 Welcome to PPP.
Apr 10 22:08:08 svr01 pppauthd[18844]: Remote message: 0838-261 Welcome to PPP.
Apr 10 22:08:13 svr01 pppauthd[18844]: Remote message: 0838-261 Welcome to PPP.
```

The high-lighted two lines in Example 3-21 tell you that CHAP is correctly configured on the PPP connection.

Please note that the /etc/ppp/chap-secrets file contains the secret key to be used for the CHAP authentication. Therefore, the file must be set with the permission bit 0600. If it is readable from group members or other users, then syslog logs the following warning message:

```
Apr 10 22:08:13 svr01 pppauthd[18844]: 0838-251 Warning - secret file
/etc/ppp/chap-secrets has world and/or group access
```

3.5 Service Agent

The Electronic Service Agent is a software that resides on your system to monitor events and transmit data to IBM. This is a no-charge function from IBM. Service Agent monitors system hardware error logs and automatically reports problems, when identified, to IBM if the machine is under a service agreement or within the warranty period. The Service Agent collects system inventory and performance information to enable service improvements and the ability to provide electronic services offerings. Knowing about a malfunction early allows for proactive service delivery, which assists you in maintaining higher availability and performance. It allows IBM to respond faster and with the correct repair parts. In some cases, IBM may notify a customer of a failing situation and provide a resolution prior to an outage. Information collected through Service Agent will be made available to IBM service support representatives when they are answering questions or diagnosing problems.

For information about Service Agent, visit the following URL:

http://www.ibmlink.ibm.com/usalets&parms=H_601-011

3.5.1 Understanding Service Agent

This section provides general information about Electronic Service Agent for IBM RS/6000 and IBM @server pSeries servers.

What is Service Agent

The IBM Electronic Service Agent application is designed for monitoring your RS/6000 and pSeries servers to diagnose and report hardware problems. In addition to promoting greater system availability, Service Agent offers you the following benefits:

- Identifies and isolates problems quickly.
- Automatically analyzes error data.
- Gives you a structured view of identified hardware events.
- Consistently and automatically implements hardware service.

After installing Service Agent, you register the systems to be monitored to IBM, and then Service Agent will monitor your system's functional hardware. Only systems under IBM Warranty or Maintenance Agreement (MA) can use Service Agent to report errors, because Service Agent checks the licenses whenever a call is made to IBM.

Service Agent is not intended to be a replacement for the maintenance package for the pSeries and the RS/6000 systems. All service calls should start with the use of the standard maintenance package. You should consider Service Agent as an additional service tool for the system.

What does Service Agent do

Here are some of the actions you can accomplish using Service Agent for pSeries and RS/6000:

- Automatic problem analysis
- Definable threshold levels for error reporting
- Automatic problem reporting; service calls placed to IBM without manual intervention
- Automatic customer notification
- Commonly viewed hardware errors; you can view hardware event logs for any monitored machine on the network from the Service Agent user interface
- Network environment support with minimum telephone lines for modems
- VPD data can be sent to IBM

Supported models

Service Agent supports all the pSeries and RS/6000 models, including RS/6000 SP nodes, that have concurrent diagnostics installed.

How Service Agent works

Machines are defined and Service Agent installed by using the Service Agent user interface. After machines are defined, they are registered with the IBM Service Agent Server (SAS). During the registration process, an electronic key is created that becomes part of your resident Service Agent program. This key is used each time Service Agent places a call for service. Service Agent Server checks the current customer service status from the IBM entitlement database; if this reveals that you are not on Warranty or MA, then the call home function fails to make a PMR (problem management record) in IBM and the status of the PMR gets turned to *FAILED* on the GUI.

If the current call shows that the MA Expiry (Maintenance Agreement Expiration) date is greater than the local key indicates, then the key is extended to the new expiration date and you are sent a message indicating extension of service coverage. Service Agent provides early warning notification of upcoming Warranty or Maintenance Agreement expiration by sending renewal reminders at 90, 60, and 30 days prior to expiration. This feature is activated after you register with IBM.

Service Agent is not designed to arbitrarily pick up just any general information without it having been programed to do so. There is some data that Service Agent does bring to IBM to help with problem resolution. In some cases, this information may very likely be used by IBM for other purposes. This information consists of the problem or error information itself and Vital Product Data (VPD) or Inventory data.

In the event you are concerned about whether this information is sensitive you can review the actual data that is being sent to IBM using either the Service Agent User Interface or, from the command line, using file display programs. If, after reviewing the data and determining you do not want Service Agent to send data, you can take several different steps to prevent data from being sent to IBM. For example:

1. Within Service Agent, you can turn off the VPD gathering feature; thus, VPD is not sent to IBM.
2. After registering, you could turn off the modem itself and configure the Service Agent Notification process to notify a help desk using e-mail or have the help desk monitor Service Agent (in real time) using the Service Agent Alerts function. Then, when Service Agent detects an error, you can call to IBM manually (instead of Service Agent calling).

Today, the only data, besides error information, being sent to IBM is Vital Product Data (VPD), which is generated by either the **lscfg** command or the new Inventory Scout (invscout) program. You can run either of these commands on a system and determine if this information is of a sensitive nature or not.

When you install the Service Agent program on an AIX system, that host becomes the Gateway server by default. All configuration and setup data for monitored systems is maintained on the central database on the Gateway server. Figure 3-7 on page 108 illustrates a typical network environment monitored by Service Agent.

.

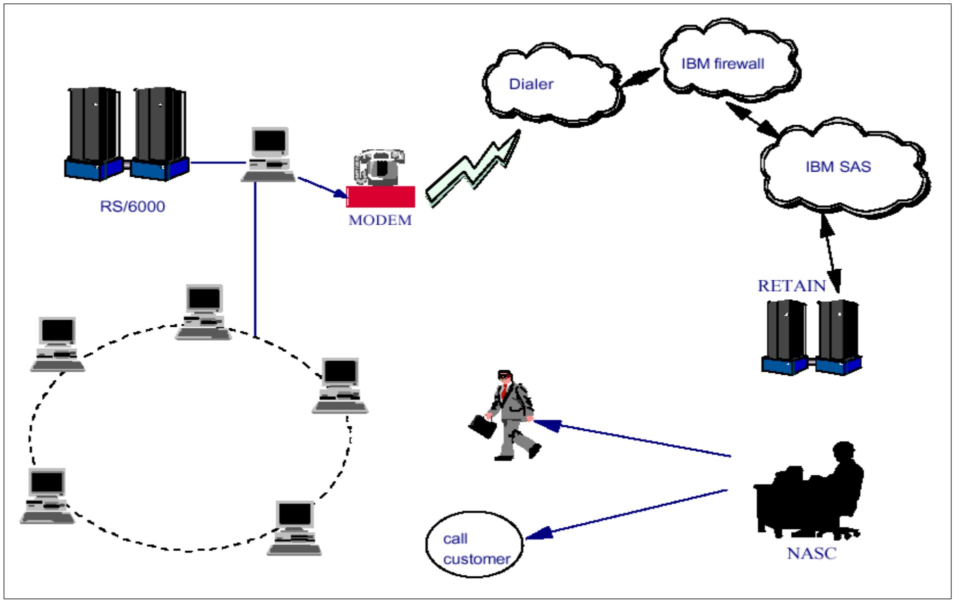


Figure 3-7 Electronic Service Agent process

Modems and TCP/IP configuration

Service Agent uses TCP/IP to communicate with its managed systems. Service Agent receives data from its managed systems over the TCP/IP network and then sends this data to the gateway server, then the gateway server sends the data to IBM. To send data to IBM, you must have at least one pSeries server that is configured with a modem.

Understanding the monitoring system

There are three major components or processes that make up the Service Agent monitoring system:

1. The Electronic Server System (ESS)

   The ESS process runs only on the Gateway Server and handles all requests for data input and retrieval from the centralized database.

2. The On-Demand Server (ODS) process

   The ODS runs on all hosts defined and handles all Service Agent monitoring and communication activities for that host. The ODS retrieves and sends data to the ESS process as necessary, or makes a call to the IBM Service Agent Server (SAS).

Within the ODS, Service Agent automatically monitors and reports back to IBM about the following three major events:

- General health check
- Changes in the Vital Product Data
- Supported error events determined by Service Agent to be valid

These events are reported to IBM directly using a modem that is attached to the Gateway server. The actual process running and dial out times are fully configurable within the Advanced Service Agent Configuration interface. If no errors are reported to IBM during the health check interval, Service Agent reports to IBM that its because the machine is healthy, and not because there is a system or communication problem.

When a supported error event is detected, Service Agent starts actions to prepare and send a request for service. It logs the event and reports the problem to the IBM problem management system for remote analysis and action. If communications to the IBM Service Agent Server (SAS) have been successful, a Problem Management Record (PMR) number is returned and logged in the database with an Open status. You must define and register the machines to be monitored by Service Agent to enable error detection. If you do not, Service Agent will not capture error information.

3. The User Interfaces (Basic and Advanced)

   a. Basic and Advanced graphical user interfaces

      The Graphical User Interfaces (GUI), Basic and Advanced, allow the user to setup and define hosts or machines that Service Agent monitors.

      The Basic User Interface is designed to allow a first time user to configure the Service Agent system with as little user input as possible, utilizing predefined defaults for a single level network environment.

      The Advanced User Interface is used for advanced functions and customization of the system as well as configuration for complex systems and multilevel networks. Both interfaces utilize a logon password that is defaulted to `password`. We recommend that this password be changed after the initial install and stored in a safe place for security purposes.

      Additionally, Service Agent can send e-mail messages or pager notifications to contacts relating all or limited machine problem information. The e-mail and pager notification functions must be configured before they become active.

   b. Basic and Advanced ASCII user interfaces

      These two ASCII interfaces provide similar or the same functions as the GUI Basic and Advanced interfaces described above.

The main difference is that the ASCII interfaces are not dependent on graphical-capable terminal, but may be run from any type, including the most basic ASCII terminal.

Navigation is done completely by keyboard data entry instead of the clicking and scrolling prevalent in the GUI interfaces.

If you intend to use one of the ASCII interfaces, we recommend that you become familiar with how Service Agent works, Service Agent prerequisites, and obtaining and installing Service Agent.

Reviewing prerequisites

The prerequisites identify certain levels of code, application programs, supported levels of machine types, disk space, or similar requirements that must be present before you install, configure, and use Service Agent. You can find this prerequisites on the *Electronic Service Agent for pSeries and RS/6000 & pSeries User's Guide*, LCD4-1060.

3.5.2 Installing Service Agent

We are going to now describe the basic steps for installing the Service Agent program.

Downloading the Service Agent installation package

You can download the installation package of Service Agent at the following URL:

ftp://ftp.software.ibm.com/aix/service_agent_code/aix/service_agent_code

To download the installation package, you can use anonymous ftp or Web browser on your system. After downloading the package, we assume that you have the installation package in the /tmp directory, as shown in the following example:

```
# ls -l /tmp/svcagent.installp
-rw-r--r--   1 root     system     13004800 Apr 11 14:22 /tmp/svcagent.installp
# installp -ld /tmp
  Fileset Name                Level                     I/U Q Content
  ====================================================================
  svcagent                    2.3.0.0                    I  N usr,root
#   IBM Electronic Service Agent For RS6000
```

Installing the Service Agent fileset

To install the Service Agent fileset, svcagent, do the following:

1. Log on to the system as the root user.
2. Type `inutoc /tmp` on the command line.

3. Type `installp -acgXNd /tmp` all on the command line.
4. Verify the fileset is correctly installed, as shown in the following example:

```
# lslpp -l | grep svcagent
  svcagent                 2.3.0.0  COMMITTED  IBM Electronic Service Agent
```

Basic Service Agent configuration

There are two interfaces available for configuring Electronic Service Agent for RS/6000:

- Basic Service Agent Configuration

  Used for initial, simple single networks

  > **Note:** The first time you configure Electronic Service Agent, you must go through the Basic Service Agent Configuration and complete the Network and Gateway server required fields.

- Advanced Service Agent Configuration

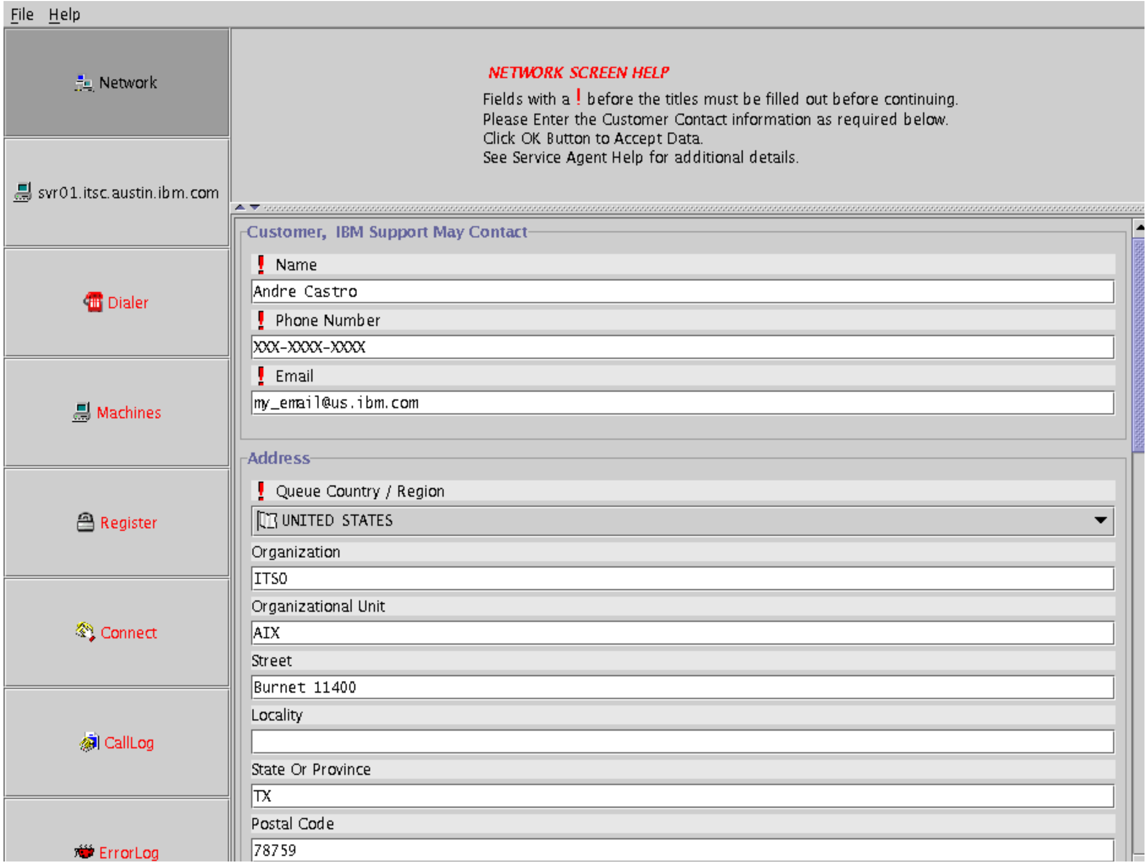
  Used for complex configurations and customization of parameters. This is the primary user interface for working with the Service Agent program. For example, the Advanced Service Agent configuration is used to perform a test call or test e-mail to IBM.

The following steps show the required configuration steps for the Basic Service Agent (we assume that you are working on the CDE environment):

1. Type **`smitty`** on the command line, then select the following panels:

   ```
   Problem Determination
       Service Agent Gateway
           Basic Agent Configuration
               Graphics Version.
   ```

2. You will see the launching GUI application. You should see the password prompt after several minutes. Otherwise, check that the server process is running by selecting Service Agent Status from the SMIT menu.
3. Once the password prompt, type the default password `password`.
4. The Service Agent Basic Configuration Menu appears, as shown in Figure 3-8 on page 112, and you are able to fill all the required fields to configure Service Agent.



NETWORK SCREEN HELP

Fields with a ! before the titles must be filled out before continuing.
Please Enter the Customer Contact information as required below.
Click OK Button to Accept Data.
See Service Agent Help for additional details.

Customer, IBM Support May Contact

! Name: Andre Castro

! Phone Number: XXX-XXXX-XXXX

! Email: my_email@us.ibm.com

Address

! Queue Country / Region: UNITED STATES

Organization: ITSO

Organizational Unit: AIX

Street: Burnet 11400

Locality:

State Or Province: TX

Postal Code: 78759

Figure 3-8 Service Agent Basic Configuration Menu

Basic Service Agent configuration panel

The Basic Service Agent Configuration panel (see Figure 3-8) is divided into three major areas:

- The left-most area is the Properties area. In the Properties area, you see several buttons that allow you to select the following properties:
  - Network
  - Gateway
  - Dialer
  - Machines
  - Register
  - Connect

  - Call Log
  - Error Log

  An auto-prompt feature is incorporated that takes you to the next logical properties folder when an update has taken place.

- The upper pane of the screen displays brief help information that pertains to the currently selected property and parameters.
- The lower pane contains the parameter update, selection and display fields. Depending on the actual property or folder button selected, you can modify parameters by clicking on the appropriate fields and typing in new values. After all the data has been typed, click OK (OK is located at the bottom of the screen) to save the data. If an exclamation mark (!) precedes the parameter name, then that parameter is mandatory. If a icon in the shape of a pad lock precedes a parameter, then that parameter cannot be modified.

In order to configure the Basic Service Agent, you will need to collect information described in the *Electronic Service Agent for pSeries and RS/6000 & pSeries User's Guide*, LCD4-1060, such as the following items:

- Hostname
- Machine Type
- Model
- Serial Number
- Processor-Id

Note: The Basic Service Agent configuration will create the PPP connection for you. You have to use the default PPP user to avoid connection problems.

4

Secure network connection on AIX

Network security, which is briefly explained in Section 1.5.3, “Network security” on page 17, is a broad topic that we cannot cover in detail in this chapter. Therefore, we use an approach that explains some major points in network security in this chapter. This chapter contains the following four sections:

- Section 4.1, “Securing network services on AIX” on page 116
- Section 4.2, “OpenSSH” on page 131
- Section 4.3, “OpenSSH on AIX” on page 135
- Section 4.4, “PuTTY: An SSH2 protocol client on Windows” on page 159

The first section explains security issues concerning several services that are available by default on AIX. In the second section, we provide basic introduction of OpenSSH, which is widely accepted as a replacement for several services explained in the first section. The third section explains how to use OpenSSH on AIX using real examples, and the fourth section also provides real usage examples of an SSH2 protocol client on Microsoft Windows.

AIX 5L Version 5.1 supports the IP security (IPsec) function extensively. For further information about IPsec, please refer to *A Comprehensive Guide to Virtual Private Networks, Volume III: Cross-Platform Key and Policy Management*, SG24-5309-00.

4.1 Securing network services on AIX

AIX offers many network services configured by default. Although some services are convenient and easy to use, they are unfortunately considered weak or vulnerable on all the UNIX operating systems, not only on AIX.

In this section, we first provide basic understanding about several standard network services on AIX, including:

- "R-commands" on page 116
- "The telnet service" on page 118
- "The ftp service" on page 119

Although, these services can be configured with the Kerberos authentication method using separate purchase-able software products[1] on AIX, we exclude these advanced configuration from this redbook on purpose to simply explain the basic concept of these services.

If you have to use other network services that are also considered weak, you should consider implementing the alternative methods explained in the following sections:

- Section 4.1.1, "Verify your network services on AIX" on page 119
- Section 4.1.2, "Alternative FTP daemon" on page 122
- Section 4.1.3, "TCP wrapper" on page 127

> **Note:** IBM provides managed security services to check the security of your systems, found at:
>
> `http://www.ers.ibm.com/`

R-commands

The r-commands, where "r" stands for *remote*, is a so-called generic name for the **`rcp`**, **`rlogin`**, and **`rsh`** (**`remsh`**)[2] commands. These commands are used for the following purpose:

`rcp` Transfers files between a local and a remote host or between two remote hosts.

`rlogin` Connects a local host with a remote host.

[1] To configure the Kerberos Version 5 authentication method for r-commands, ftp, and telnet on AIX 5L Version 5.1, you need DCE (Distributed Computing Environment) Version 3.1 for AIX.

[2] You should not confuse the **`rsh`** command with the Rsh (/usr/bin/rsh), which stands for restricted shell.

rsh (remsh) Executes the specified command at the remote host or logs into the remote host.

These commands are installed in the /usr/bin directory and included in the bos.net.tcp.clients fileset, as shown in Example 4-1.

Example 4-1 r-commands included in bos.net.tcp.client[3]

```
# ls -l /usr/bin/rcp /usr/bin/remsh /usr/bin/rlogin /usr/bin/rsh
-r-sr-xr-x   1 root     system       319972 Apr 08 2001  /usr/bin/rcp
-r-sr-xr-x   2 root     system       303506 Feb 10 14:11 /usr/bin/remsh
-r-sr-xr-x   1 root     bin          306328 Apr 08 2001  /usr/bin/rlogin
-r-sr-xr-x   2 root     system       303506 Feb 10 14:11 /usr/bin/rsh
# lslpp -w /usr/bin/rcp /usr/bin/remsh /usr/bin/rlogin /usr/bin/rsh
  File                                        Fileset               Type
  ----------------------------------------------------------------------------
  /usr/bin/rcp                                bos.net.tcp.client    File
  /usr/bin/remsh                              bos.net.tcp.client    Hardlink
  /usr/bin/rlogin                             bos.net.tcp.client    File
  /usr/bin/rsh                                bos.net.tcp.client    File
```

As an example of the usage of these commands, Figure 4-1 illustrates the basic execution process flow of the **rsh** command.

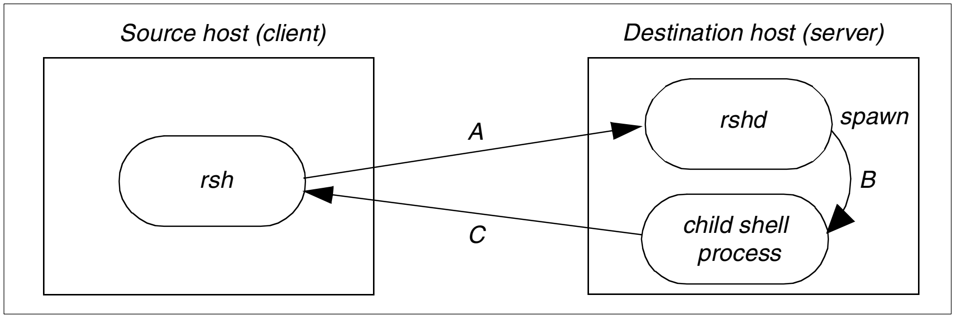


Figure 4-1 Execution process flow of rsh

We explain these flow in the following list:

1. On the source host (client), the **rsh** command is invoked to connect to the destination host (server), as shown in A.
2. The rshd daemon attempts to validate the specified user using the following steps:
   a. The rshd daemon looks up the configured name service to be used for the user name and password, for example, the /etc/passwd file or NIS passwd map.

[3] **rsh** and **remsh** are hard linked to the same file.

   b. If the user ID is not 0, rshd searches the /etc/hosts.equiv file to verify that the client name is listed; then the rshd daemon validates the user.
   c. If the $HOME/.rhosts file exists, rshd tries to authenticate the user by checking the $HOME/.rhosts file.
   d. If either of the previous attempts failed, rshd shows you a password prompt of the user for the authentication.

3. Once rshd validates the user, it spawns a child shell of the user, as shown in B. The shell inherits the network connections established by the rshd daemon and it passes the command line specified on the rsh command line. The shell sends the output to the client using the inherited network connection, as shown in C. When the remote command terminates, the local rsh process exits.

The /etc/host.equiv file is a system-wide configuration file for r-commands. The $HOME/.rhosts file is a user-basis configuration file for each user. To facilitate r-commands service, both files must reside on the server and have the same format as shown in the following example:

```
hostname [username]
```

The first field of this format expresses the host name, which is allowed to access to the server. If a special character '+' is specified, any host is allowed to access the server. The optional second field expresses the user name to which access to the server is granted. If a special character '+' is specified, any user is granted the access; in other words, no authentication is attempted.

Although the usage of r-commands is quite simple and convenient, it may provide a security hole on your system. We strongly recommend you to disable these services on your system.

The white paper provided by CERT, *rlogin(1): The Untold Story*, provides detailed information about security vulnerability for r-commands. It can be found at:

http://www.cert.org/archive/pdf/rlogin1_98tr017.pdf

> **Note:** The r-commands service transmit all the data between client and server in clear text.

The telnet service

The telnet service, the most popular application in TCP/IP, is based on a client and server architecture, as described in the following:

telnet The **telnet** command, installed as /usr/bin/telnet, is an application client that supports the telnet service.

telnetd The **telnetd** daemon, installed as /usr/sbin/telnetd, is a server daemon process that supports the telnet service. The **telnetd** daemon process listens at port 23 by default, as specified in the /etc/services file. The telnetd daemon is invoked from inetd (an internet super daemon process) upon receiving the connection request to the telnet service.

Note: The telnet service transmits all the data between client and server in a clear text.

The ftp service

The ftp service, also a popular application in TCP/IP, is based on client and server architecture, as described in the following:

ftp The **ftp** command, installed as /usr/bin/ftp, is an application client that supports the ftp service to be used for transferring files between a local and a remote host.

ftpd The **ftpd** daemon, installed as /usr/sbin/ftpd, is a server daemon process that supports the ftp service. The ftpd daemon is invoked from inetd upon receiving the connection request to the ftp service.

You can use the $HOME/.netrc file to automatically log into remote hosts for ftp service. This file is created on the client side of ftp service and has the format shown in the following example:

```
machine svr01.austin.ibm.com login bob password
```

Because the $HOME/.netrc file has user names and passwords written in a clear text, a malicious user can steal the password if the permission mode[4] of this file is improperly set.

Note: The ftp service transmits all the data between client and server using a non-encrypted format.

4.1.1 Verify your network services on AIX

The basic concept of how to secure AIX systems is simple, though implementation tasks are sometimes complex—unless absolutely required, disable all the pre-configured network services. For example, you should stop all the network services on the Web server, except the httpd process.

[4] We strongly recommend you do not use this file. If you have to use this file, set the permission mode as 0600.

On AIX, there are three places where programs are commonly started:

- /etc/inittab
- /etc/rc*
- /etc/inetd.conf

You should verify programs that use network services in these files and directories, including the following services.

Disabling NFS and NIS services

The Network File System (NFS) services and Network Information Service (NIS) are started from /etc/rc.nfs. Although these services are quite popular and convenient for sharing information among systems, NFS and NIS are widely considered as insecure on all UNIX operating systems, including AIX. Therefore, unless you absolutely require these services on your systems, we recommend you disable them.

To disable the NFS and NIS services, issue the following commands:

- NFS

  ```
  # stopsrc -g nfs
  ```

- NIS

  ```
  # stopsrc -g nis
  ```

You should also comment out the following entry from /etc/inittab:

```
:rcnfs:23456789:wait:/etc/rc.nfs > /dev/console 2>&1 # Start NFS Daemons
```

> **Note:** Network installation manager (NIM) uses NFS to install filesets over the network. Therefore, you cannot stop these services in an environment that uses NIM.

Customizing /etc/inetd.conf

The internet super daemon, inetd, is invoked and reads the configuration file, /etc/inetd.conf, upon system startup. It spawns a defined daemon program upon a connection request to a specific service in /etc/inetd.conf.

We recommend that you comment out many lines in /etc/inetd.conf, as shown in Example 4-2 on page 121. Please note that the telnet[5] service is secured using the TCP wrapper explained in Section 4.1.3, “TCP wrapper” on page 127.

[5] We recommend you to use OpenSSH instead of telnet, as explained in Section 4.3, “OpenSSH on AIX” on page 135.

Example 4-2 Customize /etc/inetd.conf

```
telnet   stream tcp  nowait  root   /usr/sbin/tcpd /etc/leland/telnetd -a user
#ftp     stream tcp6 nowait  root   /sbin/ftpd ftpd
#shell   stream tcp6 nowait  root   /sbin/rshd rshd
#login   stream tcp6 nowait  root   /sbin/rlogind rlogind
#exec    stream tcp6 nowait  root   /usr/sbin/rexecd rexecd
#comsat  dgram  udp  wait    root   /usr/sbin/comsat comsat
#uucp    stream tcp  nowait  root   /usr/sbin/uucpd uucpd
#bootps  dgram  udp  wait    root   /usr/sbin/bootpd bootpd /etc/bootptab
#finger  stream tcp  nowait  nobody /usr/sbin/fingerd fingerd
#systat  stream tcp  nowait  nobody /usr/bin/ps ps -ef
#netstat stream tcp  nowait  nobody /usr/bin/netstat netstat -f inet
#tftp    dgram  udp6 SRC     nobody /usr/sbin/tftpd tftpd -n
#talk    dgram  udp  wait    root   /usr/sbin/talkd talkd
#ntalk   dgram  udp  wait    root   /usr/sbin/talkd talkd
#echo    stream tcp  nowait  root   internal
#discard stream tcp  nowait  root   internal
#chargen stream tcp  nowait  root   internal
#time    stream tcp  nowait  root   internal
#echo    dgram  udp  wait    root   internal
#discard dgram  udp  wait    root   internal
#chargen dgram  udp  wait    root   internal
#time    dgram  udp  wait    root   internal
#imap2   stream tcp  nowait  root   /usr/sbin/imapd imapd
#pop3    stream tcp  nowait  root   /usr/sbin/pop3d pop3d
#pcnfsd  sunrpc_udp  udp     wait   root    /usr/sbin/rpc.pcnfsd pcnfsd
```

> **Note:** You have to refresh inetd using the **refresh -s inetd** command after modifying /etc/inetd.conf.

If you selected Trusted Computing Base (TCB) upon AIX installation, you can run the **securetcpip** command to easily disable non secure services. The command disables non-trusted services, such as rsh, rlogin, and so on. It comments out the entries of non-trustable services in /etc/inetd.conf and disallows permissions for the individual daemons. To invoke the **securetcpip** command, do the following:

1. **stopsrc -g tcpip**
2. **securetcpip**
3. **startsrc -g tcpip**

To use the Trusted Computing Base, you have to set Install Trusted Computing Base option to YES upon AIX installation.

Basic administration tasks

The following issues describe how to verify network security issues on AIX:

- List which processes and services are running currently using the following command:

  ```
  ps –ax | more
  ```

- Display processes that are listening on network sockets using the following command:

  ```
  netstat –a | more
  ```

- Another good task is checking open ports (see “Port scanning” on page 280).

For further information about network security on AIX, please refer to the following IBM Redbooks:

- *AIX 4.3 Elements of Security*: *Effective and Efficient Implementation*, SG24-5962
- *Additional AIX Security tools on IBM @server pSeries, IBM RS/6000, and SP/Cluster*, SG24-5971

4.1.2 Alternative FTP daemon

AIX ftpd is suitable in a simple and secure network environment where there is only trusted users. In a server farm, you may have many different types of users: web administrators who need to upload new images or content, support staff who need to upload software or patches, anonymous users who need to download files, personnel who need to download log files, and even customers who need to update their own sites. In this environment, you need something where users and groups can be defined and activity logs can be kept. The solution to this is to use Washington University FTP daemon, or *WU-FTPD*.

For detailed information about WU-FTPD, please visit the following URL:

http://www.wu-ftpd.org/

Installing WU-FTPD

You can install WU-FTPD from the AIX toolbox for Linux applications for POWER Systems CD-ROM media. To install WU-FTPD, do the following:

1. Insert the AIX toolbox for Linux applications for POWER Systems CD-ROM media into the CD-ROM drive.
2. Enter the following command as root:

   ```
   # mount -v cdrfs -o ro /dev/cd0 /mnt
   # cd /mnt/RPMS/ppc
   # ls -1 fileutils* wu-ftpd*
   ```

```
fileutils-4.1-2.aix4.3.ppc.rpm
wu-ftpd-2.6.2-1.aix4.3.ppc.rpm
# rpm -Uvh fileutils-4.1-2.aix4.3.ppc.rpm
fileutils
##################################################
# rpm -Uvh wu-ftpd-2.6.2-1.aix4.3.ppc.rpm
wu-ftpd
##################################################
```

3. After you install WU-FTPD, you have to modify /etc/inetd.conf to use the new daemon provided by WU-FTPD, as shown in the following example:

```
ftp     stream  tcp nowait root /opt/freeware/sbin/in.ftpd  in.ftpd -a
```

4. Then you have to refresh the inetd subsystem using **`refresh -s inetd`**.
5. You should confirm WU-FTPD is configured correctly, as shown in Example 4-3.

Example 4-3 Testing WU-FTPD

```
# ftp localhost
Connected to loopback.
220 svr06.itsc.austin.ibm.com FTP server (Version wu-2.6.2(1) Fri Feb 8
15:26:34 CST 2002) ready.
Name (localhost:root):
331 Password required for root.
Password:
230 User root logged in.
ftp> quit
221-You have transferred 0 bytes in 0 files.
221-Total traffic for this session was 225 bytes in 0 transfers.
221-Thank you for using the FTP service on svr06.itsc.austin.ibm.com.
221 Goodbye.
```

You will notice the ftp login banner is slightly different from the highlighted version number in the example.

Configuring WU-FTPD

Configuration of WU-FTPD is performed using several files:

- /etc/ftpaccess

  Main configuration file

- /etc/ftpusers

  List of users that *cannot* use the ftp service

- /etc/ftphosts

  Specific list of hosts allowed or denied access

- /etc/ftpserver

  Used to configure virtual FTP servers

- /etc/ftpgroup

  Used for enhanced security access

Upon installation, a simple example of the /etc/ftpaccess file is installed. Example 4-4 shows a customization sample of /etc/ftpaccess.

Example 4-4 The /etc/ftpaccess file

```
class   all   real,guest,anonymous  *
guestgroup webmaster
guestgroup customer
guestuser weblog

deny-uid %-199 %65535

limit   real      10   Any      /etc/msgs/msg.dead
limit   guest      5   Any      /etc/msgs/msg.dead
limit   anonymous 20   Any      /etc/msgs/msg.dead

greeting brief

defumask 0002 all
loginfails 3

readme  README*    login
readme  README*    cwd=*

message /welcome.msg            login
message .message                cwd=*

compress       yes              all
tar            yes              all

passwd-check rfc822 warn

noretrieve /etc/passwd /etc/group

delete no anonymous
overwrite no anonymous
rename no anonymous
chmod no anonymous
umask no anonymous

# Anonymous user - no upload, download only
upload /www/ftp * no
```

```
# Web Masters - access to web site only
upload /www/a/dev * yes root webm 0664 dirs
upload /www/a/live * yes root webm 0664 dirs

# Stats people - download only
upload /logs/ihs * no weblog weblog 0664 nodirs

# Customer - access to customer area
upload /www/a/dev/infof/documents/www * yes ibyt customer 0664 nodirs
upload /www/a/dev/edwardsm/documents/www * yes jonesj customer 0664 nodirs

path-filter anonymous /etc/pathmsg ^[-A-Za-z0-9_\]*$^\.^-

log commands real
log transfers anonymous,real inbound,outbound

shutdown /etc/shutmsg

email parkera@svr06.itso.austin.ibm.com
```

In this example, we define three types of users:

- Real users

  Trusted users on the system.

- Guest users

  Users defined on the box, but with restricted access.

- Anonymous users

  Unknown users with very restricted access. This is not recommended under normal circumstances. If you must have anonymous access do not allow file uploads.

Each user must be defined on the system as normal. A new user, ftp, must be set up for the anonymous ftp access user account. This user's home directory is defined as /www/ftp in this example. When the anonymous user logs in, WU-FTP performs the chroot() system call. This sets the effective root directory to be the user's home directory. This means the ftp user cannot, for example, get into the real /etc directory. This does present us a small problem; with no access to /etc/passwd, the **ls** command cannot list who owns a certain file. In fact, we cannot even have the **ls** command! Therefore, we have to provide a few small files to be used for the anonymous ftp environment. The easiest work around to this issue is to set up the ftp user with their home directory, then use the /usr/samples/tcpip/anon.ftp script to set up the environment.

Once you have run the anon.ftp script, you will see a directory structure as shown in Example 4-5. You will notice there is a file named /www/ftp/ls that is a copy of the **/usr/bin/ls** command. Because this environment does not have /etc/passwd or /etc/group files to be used for processing credential listing files' owner, you have to manually copy these files in the /www/ftp/etc directory. Make sure that you edit these files once copied in, and that only two users, root and ftp, should be listed in these files.

Example 4-5 Anonymous ftp directory structure

```
root@svr06:/www/ftp [518] # find . -print
.
./.profile
./bin
./bin/ls
./etc
./pub
./lib
./lib/libc.a
./lib/libcurses.a
./lib/libcrypt.a
./dev
./dev/null
./usr
./usr/lpp
./usr/lpp/msg
./usr/lpp/msg/en_US
root@svr06:/www/ftp [519] # rm .profile
root@svr06:/www/ftp [520] # cp /etc/passwd etc
root@svr06:/www/ftp [521] # cp /etc/group etc
root@svr06:/www/ftp [522] # vi etc/passwd etc/group
root@svr06:/www/ftp [523] # cd etc
root@svr06:/www/ftp/etc [524] # cat passwd
root:!:0:0::/:/usr/bin/ksh
ftp:*:210:1:Anonymous ftp:/www/ftp:/usr/bin/ksh
root@svr06:/www/ftp/etc [525] # cat group
system:!:0:root
```

The anonymous user is only allowed to download files under the /www/ftp directory. It is possible to allow the anonymous user to upload files; however, this is against our security rules.

The guest users have similar restrictions placed upon them, however they are all allowed to upload, except the weblog users. You will need to create a similar directory structure as shown in Example 4-5 for these users. If you do not, they will not be able to list files in their directory.

The webmasters can upload files to the system. However, these file will be owned by root, though everyone will be granted read access. The person who reads these files will probably be the system administrator of the web server, which runs with very low privileges. Web masters are permitted to create sub directories for the content.

Other users have normal access to the system. They are not restricted by the chroot() system call. The root user and system accounts are not allowed to log in using ftp.

4.1.3 TCP wrapper

TCP wrapper protects the inetd daemon by defining access controls to services provided by the daemon. When a connection comes in for a service and port combination, inetd first spawns the TCP wrapper program (tcpd). The tcpd program then verify the incoming connection request with the defined access control rules to ensure that it is an allowed request for the requested service and port combination.

From the standpoint of inetd, the entire tcpd process is transparent. The only required modification to inted is you have to modify the /etc/inetd.conf file. The file has to be modified to start tcpd instead of the requested server program. TCP wrapper takes care of starting the requested server program.

Note: TCP wrapper can handle TCP-based services invoked from inetd.

TCP wrapper is a common tool in network environments where high security is required.

For further information about TCP wrapper, please visit the following URL:

`http://www.cert.org/security-improvement/implementations/i041.07.html`

Installing TCP wrapper on AIX

To download TCP wrapper, visit the following URL:

`http://www.bullfreeware.com/`

Then download the package named tcp_wrappers-7.6.1.0.exe.

Once you download the package file, you can install TCP wrapper using the steps shown in Example 4-6 on page 128.

Example 4-6 Installing TCP wrapper

```
# pwd
/tmp
# chmod +x tcp_wrappers-7.6.1.0.exe
# ./tcp_wrappers-7.6.1.0.exe
UnZipSFX 5.32 of 3 November 1997, by Info-ZIP (Zip-Bugs@lists.wku.edu).
  inflating: tcp_wrappers-7.6.1.0.bff
  inflating: tcp_wrappers-7.6.1.0.bff.asc
# installp -a -d./tcp_wrappers-7.6.1.0.bff all
# lslpp -l | grep tcp_wrap
  freeware.tcp_wrappers.rte  7.6.1.0  COMMITTED  TCP/IP daemon security wrapper
```

Configuring TCP wrapper

To configure TCP wrapper, you have to modify the following files:

- /etc/inetd.conf

  The configuration file for inetd has to be modified to specify tcpd instead of the specified program.

- /etc/hosts.allow

  This file defines hosts and services that are allowed access.

- /etc/hosts.deny

  This file defines hosts and services that are denied access.

The two access files, /etc/hosts.allow and /etc/hosts.deny, are parsed by tcpd in the following order (if matched, the parse will stop):

1. Access will be allowed when a host and service matches an entry in the /etc/hosts.allow file.
2. Access will be denied when a service and client matches an entry in the /etc/hosts.deny file.
3. All access will be granted.

In a server farm, there should only be limited hosts that are allowed access. Therefore, the security policy in a server farm is summarized as: denying access to every service from everywhere, unless specifically allowed.

Example 4-7 and Example 4-8 on page 129 show the TCP wrapper configuration files that implement this security policy.

Example 4-7 /etc/hosts.allow

```
# cat /etc/hosts.allow
# Local host
ALL: 127.
```

```
# Administration server
ALL: 10.10.10.10
# Internal network
ALL: 10.20
# IBM internal
ftpd: .ibm.com
```

Example 4-8 /etc/hosts.deny

```
# cat /etc/hosts.deny
ALL: ALL
```

These configuration files defines the rules that allow:

- The localhost to access any of its local services.
- The server with IP address 10.10.10.10 to access any service.
- Any hosts on the network address 10.20.0.0 to access any service.
- Any host in the DNS domain .ibm.com to access the ftp service only.

Once these configuration files are set up, you have to modify /etc/inetd.conf to instruct inetd to use TCP wrapper. Example 4-9 shows an example that the ftp service is wrapped.

Example 4-9 Sample wrapper /etc/inetd.conf file

```
# grep ftpd /etc/inetd.conf
#ftp     stream  tcp6    nowait  root    /usr/sbin/ftpd         ftpd
ftp     stream  tcp     nowait  root    /usr/local/bin/tcpd     ftpd
#tftp    dgram  udp6    SRC     nobody  /usr/sbin/tftpd         tftpd -n
```

Note: We recommend you to comment out the original line instead of removing it.

Any connection problems are logged using syslogd. Example 4-10 shows an example of the connection message logged for tcpd.

Example 4-10 Connection messages logged for tcpd

```
root@svr01:/ [190] # telnet svr06
Trying...
Connected to svr06.itsc.austin.ibm.com.
Escape character is '^]'.
Connection closed.


root@svr06:/ [501] # tail /var/adm/messages
got a message (1, 0x10)
```

```
logmsg: pri 24, flags 0, from svr06, msg Apr  4 14:42:34 telnetd[32114]:
refused connect from 10.30.0.1
Logging to UNUSED
readfds = 0x30
```

In this example, the telnet connection is refused because the IP address 10.30.0.1 of svr06 is not granted access to the telnet service.

TCP wrapper provides a useful command (**tcpdchk**) to verify the configuration. It examines your configuration and reports all potential and real problems it can find. The program examines the access control files, and compares the entries in these files against entries in inetd.conf.

Example 4-11 shows an example output from the **tcpdchk** command using configuration files listed in Example 4-7 on page 128 and Example 4-8 on page 129.

Example 4-11 Verifying rules of TCP wrapper

```
# /usr/local/bin/tcpdchk -v
Using network configuration file: /etc/inetd.conf

>>> Rule /etc/hosts.allow line 2:
daemons:  ALL
clients:  127.
access:   granted

>>> Rule /etc/hosts.allow line 4:
daemons:  ALL
clients:  10.10.10.10
access:   granted

>>> Rule /etc/hosts.allow line 6:
daemons:  ALL
clients:  10.20
access:   granted

>>> Rule /etc/hosts.allow line 8:
daemons:  ftpd
clients:  .ibm.com
access:   granted

>>> Rule /etc/hosts.deny line 1:
daemons:  ALL
clients:  ALL
access:   denied
```

Note: TCP wrapper does not provide all the functions provided that Firewall provides. For example, if a malicious user change the IP address field in the IP protocol header, TCP wrappers will allow the access.

4.2 OpenSSH

This section provides you basic understanding about OpenSSH. OpenSSH is reliable and secure, and is widely accepted in the IT industry to replace the r-commands, telnet, and ftp services, providing secure encrypted sessions between two hosts over the networks.

4.2.1 Introduction to OpenSSH

The OpenSSH is a free software tool that supports SSH1 and SSH2 protocols. It is composed of several commands and configuration files to provide the following functions:

- Authentication
- Encryption
- Integrity
- Port forwarding

OpenSSH is based on client and server architecture. An OpenSSH daemon process (sshd) is typically running on the UNIX host and waiting for the connection from clients. Several client programs are available for UNIX and Microsoft Windows environments.

OpenSSH supports public- and private-key pairs for authentication and encryption of channels to ensure secure network connections.

For further information about OpenSSH, please visit the following URL:

http://www.openssh.org

History of OpenSSH

Secure shell was developed by Tatu Ylonen, when he worked as a researcher at Helsinki University in 1995. He was a victim of a password-sniffing attack and decided to do something about it himself. The SSH1 protocol implementation was initially released as open source to the world.

Because several security defects were discovered in the original SSH1 protocol, a new version SSH2 was developed in 1997. Although the SSH2 protocol required a strict license agreement in its early days, the agreement was relaxed later and several free implementations of SSH were developed. The most common free implementation is OpenSSH, which was initially developed by Markus Friedl for OpenBSD. It was very quickly ported to other UNIX based operating systems, including AIX.

SSH1 and SSH2 protocols

Although OpenSSH supports the SSH1 and SSH2 protocols, we use the SSH2 protocol only throughout this redbook because of the following facts:

- The SSH2 protocol has better security, performance, and portability than SSH1. For example, SSH2 uses several session keys, one for each direction, unlike SSH1, which uses only one session key.
- It supports DSA and other public key algorithms for encryption and authentication (SSH1 only supports RSA).
- It supports both SOCKS and sftp (secure file transfer protocol).

Note: The SSH1 and SSH2 protocols are not compatible.

Encryption algorithms

OpenSSH supports the following symmetric encryption algorithms to be used with the SSH2 protocol:

- 3DES
- Cast128-cbc
- Blowfish
- Twofish
- Arcfour

OpenSSH supports the following asymmetric authentication algorithms for digital signature encryption to be used with the SSH2 protocol:

- RSA (Rivest-Shamir-Adleman algorithm)
- DSA (digital signature algorithm)

OpenSSH supports several authentication methods to be used with the SSH2 protocol, including:

- Password authentication
- Public key authentication

- Host-based authentication
- Kerberos authentication[6]

Note: Throughout this redbook, we only use two common methods: password and public key authentication.

For information about cryptographic concepts, private key, public key or DSA and RSA digital signatures, please refer to the redbook *TCP/IP Tutorial and Technical Overview*, GG24-3376.

4.2.2 OpenSSH architecture

An OpenSSH environment is composed of the following components:

Server A program running as a daemon that listens for secure shell connections.

Client A program that connects to the secure shell server.

User key An asymmetric key used by the client to provide a user identity.

Host key An asymmetric key used by the server to provide the server identity.

Session key A symmetric key for encrypting the communication. Symmetric encryption is faster than asymmetric encryption. User data is transmitted using a symmetric algorithm.

The user key and the host key are composed of a public- and private-key pair. Before making a secure connection using OpenSSH, the client should copy its public key to the remote host to which you want to connect using public key authentication. The private key remains on the host that the key is generated on and must be securely protected.

Upon generating the user key files, the private key can be encrypted with the passphrase[7] you entered. The public key is not encrypted by the passphrase.

Note: To prevent a non-authorized user from using the private key, we recommend that you should always encrypt your user private key.

[6] To use Kerberos authentication, you have to incorporate required patches to OpenSSH, found at http://www.sxw.org.uk/computing/patches/openssh.html.

[7] Passphrase is a string of characters longer than the usual password used in creating a digital signature or in the encryption or decryption of a message.

To provide a stronger authentication mechanism, the private key is used to decrypt the connection data, which is sent by the remote peer, and encrypted with the local peer's public key.

To establish a secure connection between a client and a server, OpenSSH works as follows:

1. An OpenSSH client requests a connection to an OpenSSH Server.
2. The OpenSSH client and the server disclose the supported secure shell protocol versions to each other.
3. The server presents its public host key and a challenge (sequence of random bytes) to the client.
4. The client generates a new session key and a challenge, then encrypts them with the host public key sent from the server. Then, it sends the new session key and the challenge to the server.
5. The server decrypts the session key and the challenge using its own host private key.
6. Once the decrypting is performed successfully, the secure connection is established. Now, both sides can use a secure network connection by encrypting and decrypting all the data using the peer's session key.

Operating system authentication is required to determine whether an user is allowed to access to a certain user account.

The standard for the SSH2 protocol is defined by the IETF Security Working Group, found at:

http://www.ietf.org/html.charters/secsh-charter.html

4.2.3 OpenSSH configuration files

The OpenSSH client and the server require different sets of configuration files, which are explained in this section.

Server configuration files

OpenSSH requires several configuration files stored in the /etc/ssh directory as shown in the following example:

```
# pwd
/etc/ssh
# ls -l sshd_config ssh_host_dsa*
-rw-------   1 root     system          668 Apr 02 17:17 ssh_host_dsa_key
-rw-r--r--   1 root     system          620 Apr 02 17:17 ssh_host_dsa_key.pub
-rw-------   1 root     system         2053 Apr 02 17:19 sshd_config
```

These files are used for the following purpose:

sshd_config	OpenSSH server configuration file
ssh_host_dsa_key	Host DSA private key file
ssh_host_dsa_key.tab	Host DSA public key file

Client configuration files

OpenSSH client commands require several configuration files stored in the user's $HOME/.ssh directory, as shown in the following example:

```
emilia@svr02 $ pwd
/home/emilia
emilia@svr06 $ ls -l .ssh
total 6
-rw-------   1 emilia   staff          736 May 05 09:16 id_dsa
-rw-r--r--   1 emilia   staff          619 May 05 09:16 id_dsa.pub
-rw-r--r--   1 emilia   staff          624 May 05 09:22 known_hosts
```

These files are used for the following purposes:

id_dsa	User DSA private key file.
id_dsa.pub	User DSA public key file.
known_hosts	This file is automatically created to store the remote system's host public key after the user logs into the remote system using OpenSSH.

There is a system-wide client configuration file for each client, named /etc/ssh_config. User configuration files in the $HOME/.ssh directory always override the definition in this file.

4.3 OpenSSH on AIX

This section explains how to install and configure OpenSSH on AIX. Starting in April 2002, AIX 5L Version 5.1 offers OpenSSH[8] in the updated Bonus Pack CD-ROM media as several AIX standard install packages (installp format). OpenSSH is also available in several RPM format packages, provided by the AIX toolbox for Linux applications.

The OpenSSH program contained in the Bonus Pack CD-ROM media is offered *AS-IS* and is licensed under the terms and conditions of the IBM International Program License Agreement (IPLA) for Non-Warranted Programs.

[8] Although OpenSSH Version 2.9.9 is currently provided, it is subject to change in accordance with OpenSSH development in the future.

4.3.1 Installing and configuring OpenSSH on AIX

If you have the AIX 5L for POWER Version 5.1 Bonus Pack 5765-E61 CD-ROM media, you can install OpenSSH using the following steps. Otherwise, you should follow the steps explained in:

- "Downloading RPM format packages" on page 138
- "Installing OpenSSH RPM packages" on page 140

Note: The OpenSSH packages are also available at the following site:

http://oss.software.ibm.com/developerworks/projects/opensshi

Once security holes of OpenSSH are detected, IBM will rebuild the packages and announce the latest version on this site.

Using installp format packages

Before installing the OpenSSH provided in the installp format packages, you have to install Open Secure Sockets Layer (OpenSSL). Currently, OpenSSL is provided in an RPM package in the AIX toolbox for Linux applications. If you have the AIX toolbox for Linux applications for POWER Systems CD-ROM media, then you can simply install the package using the following method:

1. Insert the AIX toolbox for Linux applications for POWER Systems CD-ROM media into the CD-ROM drive.
2. Enter the following command as the root user:

```
# mount -v cdrfs -o ro /dev/cd0 /mnt
# cd /mnt/RPMS/ppc
# ls -1 openssl*
openssl-0.9.6b-2.aix4.3.ppc.rpm
openssl-devel-0.9.6b-2.aix4.3.ppc.rpm
openssl-doc-0.9.6b-2.aix4.3.ppc.rpm
# rpm -Uvh openssl-0.9.6b-1.aix4.3.ppc.rpm
openssl
##################################################
```

   The two packages openssl-devel and openssl-doc, are not mandatory packages for using OpenSSH on AIX. These are development tools and documentation for OpenSSH.

3. If the package is correctly installed, you can verify the installation status using either of the following commands:

```
# lslpp -L | grep openssl
  openssl                   0.9.6b    C    R    Secure Sockets Layer and
# rpm -q openssl
openssl-0.9.6b-2
```

4. Unmount the /mnt filesystem and eject the CD-ROM media:

   ```
   # cd /
   # umount /mnt
   ```

If you do not have the AIX toolbox for Linux applications for POWER Systems CD-ROM media, then download the package using the steps explained in "Downloading RPM format packages" on page 138.

The OpenSSH support included in the updated Bonus Pack CD-ROM media is composed of the filesets shown in Example 4-12.

Example 4-12 OpenSSH filesets

```
Fileset Name                Level                     I/U Q Content
  ====================================================================
  openssh.base.client         2.9.9.0                    I  N usr,root
#   Open Secure Shell Commands

  openssh.base.server         2.9.9.0                    I  N usr,root
#   Open Secure Shell Server

  openssh.license             2.9.9.0                    I  N usr
#   Open Secure Shell License

  openssh.man.en_US           2.9.9.0                    I  N usr
#   Open Secure Shell Documentation - U.S. English

  openssh.msg.en_US           2.9.9.0                    I  N usr
#   Open Secure Shell Messages - U.S. English
```

To install these filesets, do the following:

1. Insert the Bonus Pack CD-ROM media into the CD-ROM drive.
2. Issue the following command as the root user:

   ```
   # smitty install_latest
   ```

3. Type `/dev/cd0` on the "INPUT device / directory for software" field.
4. Press F4 key on the "SOFTWARE to install" field on the panel, as shown in Example 4-13.

Example 4-13 Install Software

```
                                Install Software

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

                                                        [Entry Fields]
* INPUT device / directory for software               /dev/cd0
```

```
* SOFTWARE to install                               [_all_latest]            +
  PREVIEW only? (install operation will NOT occur)   no                      +
  COMMIT software updates?                           yes                     +
  SAVE replaced files?                               no                      +
  AUTOMATICALLY install requisite software?          yes                     +
  EXTEND file systems if space needed?               yes                     +
  OVERWRITE same or newer versions?                  no                      +
  VERIFY install and check file sizes?               no                      +
  Include corresponding LANGUAGE filesets?           yes                     +
  DETAILED output?                                   no                      +
  Process multiple volumes?                          yes                     +
  ACCEPT new license agreements?                     yes                     +
  Preview new LICENSE agreements?                    no                      +

F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         Esc+6=Command       Esc+7=Edit          Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do
```

5. Select the filesets listed in Example 4-12 on page 137 by pressing the F7 key, then press Enter.
6. Press the Tab key on the “ACCEPT new license agreements?” field. The field should be changed to be `yes`.
7. Press Enter.

These filesets are installed in the /usr filesystem as shown in the following example:

```
# lslpp -w /usr/bin/ssh
  File                                        Fileset               Type
  ----------------------------------------------------------------------------
  /usr/bin/ssh                                openssh.base.client   File
# lslpp -w /usr/sbin/sshd
  File                                        Fileset               Type
  ----------------------------------------------------------------------------
  /usr/sbin/sshd                              openssh.base.server   File
```

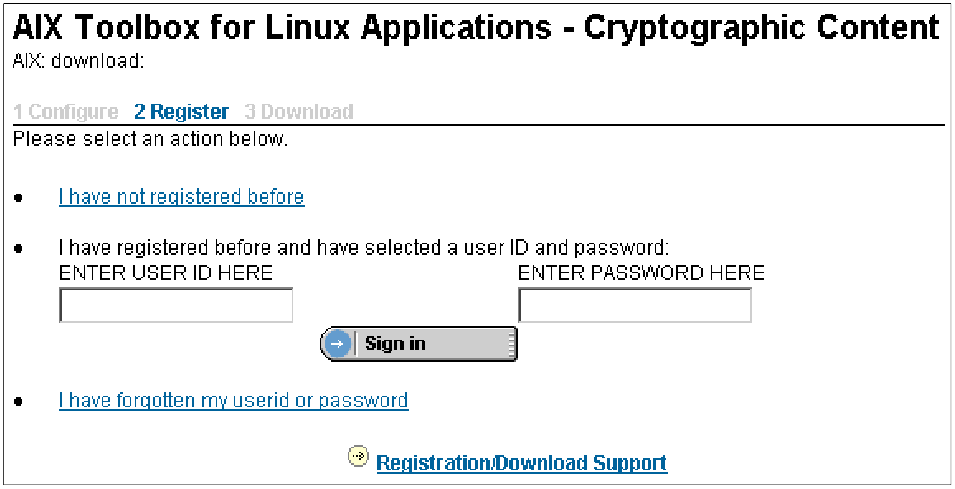
Downloading RPM format packages

To download the RPM format packages for OpenSSH, do the following:

1. Access the following URL:

   http://www.ibm.com/servers/aix/products/aixos/linux/download.html

2. Click AIX Toolbox Cryptographic Content on the sorted content download in the right upper area; you will see the panel shown in Figure 4-2 on page 139.



AIX Toolbox for Linux Applications - Cryptographic Content

AIX: download:

Please select an action below.

- I have not registered before
- I have registered before and have selected a user ID and password:

  ENTER USER ID HERE

  ENTER PASSWORD HERE

- I have forgotten my userid or password

Figure 4-2 User registration panel

3. If you have already registered your user ID on this site, enter your ID and password, then click the Sign-in button. Otherwise, click Registration/Download Support to register yourself.
4. Click the Accept License button at the bottom of the panel that appears.
5. You will see the RPM packages in Cryptographic Content for AIX.
6. Select the packages listed in Table 4-1, then click the Download now button for each.

Table 4-1 shows all the RPM packages needed to use OpenSSH on AIX Version 4.3.3.

Table 4-1 RPM packages for OpenSSH on AIX Version 4.3.3

Package file name	Description
openssh-2.9.9p2-6.aix4.3.ppc.rpm[a]	Open Source Secure Shell
openssh-clients-2.9.9p2-6.aix4.3.ppc.rpm	OpenSSH Secure Shell protocol clients
openssh-server-2.9.9p2-6.aix4.3.ppc.rpm	OpenSSH Secure Shell protocol server (sshd)
openssl-0.9.6b-2.aix4.3.ppc.rpm	Secure Sockets Layer and cryptography libraries and tools
openssl-doc-0.9.6b-2.aix4.3.ppc.rpm	OpenSSL miscellaneous files (optional package)

Package file name	Description
prngd-0.9.23-1.aix4.3.ppc.rpm	Pseudo random number generator daemon

a. This RPM package must be installed before installing any OpenSSH packages.

These packages are available for installation using the RPM package manager. On AIX 5L Version 5.1, the RPM package manager is installed by default (see Section 6.1.6, “RPM (Red Hat Package Manager)” on page 218).

Installing OpenSSH RPM packages

To install these packages, execute the commands shown in Example 4-14.

Example 4-14 Install RPM packages for OpenSSH

```
# rpm -i openssl-0.9.6b-2.aix4.3.ppc.rpm
# rpm -i prngd-0.9.23-1.aix4.3.ppc.rpm
0513-059 The prngd Subsystem has been started. Subsystem PID is 16160
# rpm -i openssh-2.9.9p2-6.aix4.3.ppc.rpm
# rpm -i openssh-server-2.9.9p2-6.aix4.3.ppc.rpm
0513-071 The sshd Subsystem has been added.
0513-059 The sshd Subsystem has been started. Subsystem PID is 14720.
# rpm -i openssh-clients-2.9.9p2-6.aix4.3.ppc.rpm
```

In this example, we assume that you downloaded the package files on the current directory.

To verify these packages has been correctly installed, do the following:

```
# rpm -qa | egrep '(openssl|openssh|prng)'
openssl-0.9.6b-2
openssh-2.9.9p2-6
openssh-clients-2.9.9p2-6
openssh-server-2.9.9p2-6
prngd-0.9.23-1
```

These packages are installed under the /opt/freeware directory, and several symbolic links are created in /usr/bin or /usr/sbin, as shown in the following example:

```
# ls -l /usr/bin/ssh
lrwxrwxrwx   1 root     system          26 Mar 19 18:04 /usr/bin/ssh@ ->
../../opt/freeware/bin/ssh*
# ls -l /usr/sbin/sshd
lrwxrwxrwx   1 root     system          28 Mar 19 18:04 /usr/sbin/sshd@ ->
../../opt/freeware/sbin/sshd*
```

Therefore, you can invoke these commands installed from RPM packages by using the same path names with the installation explained in “Using installp format packages” on page 136.

4.3.2 Managing the OpenSSH server

This section explains how to manage OpenSSH server on AIX.

Server process startup

The following entry in /etc/inittab invokes all the scripts starting from *S* under the /etc/rc.d/rc2.d directory upon system startup:

```
l2:2:wait:/etc/rc.d/rc 2
```

In the /etc/rc.d/rc2.d directory, the following example shows the required symbolic-link to start sshd:

```
#ls -l /etc/rc.d/rc2.d | grep ssh
lrwxrwxrwx   1 root system 14 Mar 19 18:04 K55sshd -> ../init.d/sshd
lrwxrwxrwx   1 root system 14 Mar 19 18:04 S55sshd -> ../init.d/sshd
```

The prngd is started from the following entry in /etc/inittab:

```
prng:2:wait:/usr/bin/startsrc -s prngd
```

Tailoring the default /etc/ssh/sshd_config file

In order to specify the SSH2 protocol to be used for OpenSSH, add the following line to the /etc/ssh/sshd_config file:

```
Protocol 2
```

To verify if the SSH2 protocol has been correctly configured, see “Verifying the SSH protocol version” on page 143.

The default /etc/ssh/sshd_config file contains entries for both RSA and DSA keys as shown in the following example:

```
# HostKeys for protocol version 2
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
```

We strongly recommend that you comment out the second line so that sshd should select DSA as the public key encryption mechanism.

If you modify /etc/ssh/sshd_config, you have to stop and restart ssh.

Start and stop servers

The prngd and sshd servers are registered as subsystems on AIX. You can control these subsystems using the commands as shown in Table 4-2.

Table 4-2 Start and stop subsystems

Subsystem name	Start	Stop
prngd	**`startsrc -s prngd`**	**`stopsrc -s prngd`**
sshd	**`startsrc -s sshd`**	**`stopsrc -s sshd`**

Generating key files for servers

Upon installation of OpenSSH on AIX, the following key files are automatically created:

- DSA key files:

Private key file	/etc/ssh/ssh_host_dsa_key
Public key file	/etc/ssh/ssh_host_dsa_key.pub

- RSA key files:

Private key file	/etc/ssh/ssh_host_rsa_key
Public key file	/etc/ssh/ssh_host_rsa_key.pub

If you want to generate new host key files, then use one of the following methods:

- For DSA encryption:

```
# ssh-keygen -t dsa -f /etc/ssh/ssh_host_dsa_key
```

- For RSA encryption:

```
# ssh-keygen -t rsa -f /etc/ssh/ssh_host_dsa_key
```

Example 4-15 describes how to generate the host DSA public and private key files. If the key files already exist, enter `y` to overwrite. When prompted to enter the passphrase, just press Enter.

> **Note:** If you enter a passphrase, then sshd never starts, because it cannot decrypt the private key upon startup.

Example 4-15 Regenerating host DSA keys

```
# ssh-keygen -t dsa -f /etc/ssh/ssh_host_dsa_key
Generating public/private dsa key pair.
/etc/ssh/ssh_host_dsa_key already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):<Enter>
Enter same passphrase again:<Enter>
Your identification has been saved in /etc/ssh/ssh_host_dsa_key.
Your public key has been saved in /etc/ssh/ssh_host_dsa_key.pub.
The key fingerprint is:
56:06:7e:53:38:e6:70:43:c6:35:99:41:0e:35:6e:99 root@svr01.itsc.austin.ibm.com
```

The host private key file must be kept secure. The file permission mode of this file must be set 0600 so that only root user can access to this file, as shown in the following example:

```
# ls -l /etc/ssh/*_key
-rw-------   1 root     system          668 May 03 11:49
/etc/ssh/ssh_host_dsa_key
-rw-------   1 root     system          515 May 03 11:49 /etc/ssh/ssh_host_key
-rw-------   1 root     system          887 May 03 11:49
/etc/ssh/ssh_host_rsa_key
```

If you generate new host key files, you have to stop and restart the OpenSSH server (sshd).

Note: We recommend that you generate new DSA key files on your systems periodically, for example, once a month, to provide higher security.

Verifying the SSH protocol version

To verify the SSH protocol version, you can use the **telnet** command and explicitly specify the port number 22. If you see SSH-2.X at the beginning in the last line, as shown in the following example, then the SSH2 protocol is correctly configured on the local host:

```
# telnet localhost 22
Trying...
Connected to loopback.
Escape character is '^]'.
SSH-2.0-OpenSSH_2.9.9p2
```

If you see SSH-1.9X at the beginning in the last line, then the SSH1 protocol is used. If you see the following output, then the sshd daemon is not running. To start the sshd daemon, issue **startsrc -s sshd**.

```
# telnet localhost 22
Trying...
telnet: connect: A remote host refused an attempted connect operation.
```

To terminate the hanged telnet session, press Ctrl-[and type q on the Telnet> prompt.

4.3.3 Generating user key files for the OpenSSH client commands

Although it is not mandatory, we strongly recommend that you generate user public/private key pair files for the OpenSSH client commands. This is because you can use a more secure authentication mechanism provided by the public key pair, explained in Section 4.2.2, “OpenSSH architecture” on page 133, if you have user key files on the OpenSSH client side. If you configure OpenSSH client commands to use user key files, you will be prompted to enter the passphrase upon any remote connection attempt. Otherwise, you will be prompted to enter the AIX login password to the remote server.

To generate user key files, you also use the **ssh-keygen** command (see “Generating key files for servers” on page 142).

Example 4-16 shows you how to create DSA key files for user emilia. Unless you want to specify a different name for your key files, just press Enter. Then you will be prompted to enter your passphrase twice.

Note: We recommend that you enter a passphrase on the client’s key.

The generated user private key file, $HOME/.ssh/id_dsa, is encrypted with the passphrase that you typed in. The user public key file is generated as $HOME/.ssh/id_dsa.pub.

Example 4-16 Generating user key files

```
emilia@svr02 $ ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/home/emilia/.ssh/id_dsa):<Enter>
Enter passphrase (empty for no passphrase):<Passphrase>
Enter same passphrase again:<Passphrase>
Your identification has been saved in /home/emilia/.ssh/id_dsa.
Your public key has been saved in /home/emilia/.ssh/id_dsa.pub.
The key fingerprint is:
2a:60:5f:aa:fa:47:b5:64:3f:c8:9b:f5:da:56:f6:45
emilia@svr02 $ ls $HOME/.ssh/
id_dsa      id_dsa.pub
```

Note: Do not forget the passphrase; otherwise, you have to generate your user key files again.

First connection attempt to the OpenSSH server

Before the first connection attempt using OpenSSH, the client has no idea about the remote server's host keys. If another host managed by a malicious user is masquerading as the target server, the client might not realize it. As an analogy, imagine you are calling a person for the first time. How can you trust the voice belongs to the correct person if you have never heard his or her voice?

To avoid this security risk, you should receive a host key fingerprint from the administrator who manages the server you are going to connect before the first connection attempt. Upon the first connection attempt using OpenSSH, you will see the fingerprint of the remote server. You should verify whether this fingerprint matches with the one received from the administrator.

Example 4-17 shows the first connection attempt using OpenSSH. The DSA host key fingerprint is high-lighted. After verifying it matches with the one received from the administrator of the server that you are connecting to, enter `yes`. Then you will be prompted the login password to the remote server.

In this example, the DSH host key fingerprint shown in Example 4-17 is the same with the one shown in Example 4-15 on page 142. Therefore, you are sure that you are connecting to the correct server, svr01.

Example 4-17 First connection attempt using OpenSSH

```
emilia@svr02 $ ssh -l valen svr01.itsc.austin.ibm.com
The authenticity of host 'svr01 (XXX.XXX.XXX.XXX)' can't be established.
DSA key fingerprint is 56:06:7e:53:38:e6:70:43:c6:35:99:41:0e:35:6e:99.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'svr01,XXX.XXX.XXX.XXX' (DSA) to the list of known
hosts.
valen@svr01's password:<password>
Last login: Mon Mar 25 15:26:57 CST 2002 on ssh from snecac
*******************************************************************************
*                                                                             *
*                                                                             *
*  Welcome to AIX Version 5.1!                                                *
*                                                                             *
*                                                                             *
*  Please see the README file in /usr/lpp/bos for information pertinent to    *
*  this release of the AIX Operating System.                                  *
*                                                                             *
*                                                                             *
*******************************************************************************
valen@svr01 $
```

After the first connection, the $HOME/.ssh/known_hosts file is created on the client (if not created already). The file contains the connected server's public host key. Therefore, you will not be prompted to confirm server's fingerprint from the second connection attempt.

4.3.4 Using a passphrase

In order to exploit the security functionality provided by OpenSSH (explained in Section 4.2.2, "OpenSSH architecture" on page 133), you should transfer your user public key file from the client to the server.

To transfer the public key file, do the following:

1. Generate your user key files on the client, as explained in Example 4-16 on page 144.
2. Transfer the public user key file, $HOME/.ssh/id_dsa.pub, to the remote server. The following example shows how to transfer the file using the **scp** command explained in Section 4.3.6, "Using the scp command" on page 149:

   ```
   emilia@svr02 $ scp .ssh/id_dsa.pub valen@svr01:/home/valen/copied_pub_key
   ```

   You will be prompted to enter the remote user password to the server.
3. Log in to the remote server using the **ssh** command, and then concatenate the transferred user public key file to the $HOME/.ssh/authorized_keys file, as shown in the following example:

   ```
   valen@svr01 $ mkdir .ssh
   valen@svr01 $ touch .ssh/authorized_keys
   valen@svr01 $ cat copied_pub_key >> .ssh/authorized_keys
   valen@svr01 $ rm copied_pub_key
   ```

4. Verify the file permissions, as shown in the following example:

   ```
   valen@svr01 $ chmod 0755 $HOME/.ssh
   valen@svr01 $ chmod 0644 $HOME/.ssh/authorized_keys
   ```

Once your user public key is transferred, you will be prompted to enter the passphrase you entered when creating the user public key, upon remote connection attempt using OpenSSH.

Example 4-18 shows a connection attempt using a passphrase. In this example, as high-lighted, you should enter the passphrase you entered upon generation of the user public key created as /home/emilia/.ssh/id_dsa on svr02.

Example 4-18 Login using a passphrase

```
emilia@svr02 $ ssh svr01.itsc.austin.ibm.com -l valen
Enter passphrase for key '/home/emilia/.ssh/id_dsa':<Passphrase>
Last login: Mon Mar 25 15:31:57 CST 2002 on ssh from svr02
********************************************************************************
```

```
*                                                                             *
*                                                                             *
*  Welcome to AIX Version 5.1!                                                *
*                                                                             *
*                                                                             *
*  Please see the README file in /usr/lpp/bos for information pertinent to    *
*  this release of the AIX Operating System.                                  *
*                                                                             *
*                                                                             *
*******************************************************************************
valen@svr01 $
```

Note: We strongly recommend that you configure the OpenSSH client commands to use a passphrase.

4.3.5 Using the ssh command

Although the **ssh** command provides many command line options as shown in Example 4-19 on page 148, you can use it as if you were using the **rsh** command.

If you do not specify the command to be remotely executed as a command line argument on the command line, as shown in Example 4-18 on page 146, then you log in to the remote system through the secure connection established by OpenSSH.

If you specify a command argument, the **ssh** command remotely executes the specified command on the remote server and returns the command output on the local controlling terminal.

The following example shows the **ssh** command execution that remotely invokes two commands, **hostname** and **whoami**, on svr01. The last two lines are the output of these commands.

```
emilia@svr02 $ ssh svr01.itsc.austin.ibm.com -l valen ' hostname; whoami'
Enter passphrase for key '/home/emilia/.ssh/id_dsa':<Passphrase>
svr01.itsc.austin.ibm.com
valen
```

In this example, we enter the passphrase instead of the login password, because we followed the steps explained in Section 4.3.4, “Using a passphrase” on page 146.

Note: If your remote user name is the same as your local one, you do not have to specify it using the -l optional flag.

Example 4-19 ssh command line options

```
$ ssh --help
Usage: ssh [options] host[command]
Options:
  -l user     Log in using this user name.
  -n          Redirect input from /dev/null.
  -F config   Config file (default: ~/.ssh/config).
  -A          Enable authentication agent forwarding.
  -a          Disable authentication agent forwarding.
  -X          Enable X11 connection forwarding.
  -x          Disable X11 connection forwarding.
  -i file     Identity for public key authentication (default:
~/.ssh/identity).
  -t          Tty; allocate a tty even if command is given.
  -T          Do not allocate a tty.
  -v          Verbose; display verbose debugging messages.
              Multiple -v increases verbosity.
  -V          Display version number only.
  -P          Don't allocate a privileged port.
  -q          Quiet; don't display any warning messages.
  -f          Fork into background after authentication.
  -e char     Set escape character; ``none'' = disable (default: ~).
  -c cipher   Select encryption algorithm: ``3des'', ``blowfish''
  -m macs     Specify MAC algorithms for protocol version 2.
  -p port     Connect to this port. Server must be on the same port.
  -L listen-port:host:port   Forward local port to remote address
  -R listen-port:host:port   Forward remote port to local address
              These cause to listen for connections on a port, and
              forward them to the other side by connecting to host:port.
  -D port     Enable dynamic application-level port forwarding.
  -C          Enable compression.
  -N          Do not execute a shell or command.
  -g          Allow remote hosts to connect to forwarded ports.
  -1          Force protocol version 1.
  -2          Force protocol version 2.
  -4          Use IPv4 only.
  -6          Use IPv6 only.
  -o 'option' Process the option as if it was read from a configuration file.
  -s          Invoke command (mandatory) as SSH2 subsystem.
  -b addr     Local IP address.
```

For further information about the **ssh** command, please refer to the online command reference by issuing the following command:

```
$ MANPATH=/usr/local/man:$MANPATH; export MANPATH
$ man ssh
```

4.3.6 Using the scp command

The secure copy gives you the ability to copy files between two hosts using a secure connection. Although the **scp** command provides several command line options, as shown in the following example, you can use it as easily as you use the **rcp** command:

```
$ scp --help
usage: scp [-pqrvBC46] [-F config] [-S ssh] [-P port] [-c cipher] [-i identity]
           [-o option] f1 f2
   or: scp [options] f1 ... fn directory
```

In Example 4-20, the file carol_file on svr02 is copied as /home/valen/marita_file on svr01.

Example 4-20 Copying a file using scp

```
emilia@svr02 $ ls -l carol_file
-rw-r--r--   1 emilia staff            64 Apr 03 21:18 carol_file
emilia@svr02 $ scp carol_file valen@svr01:/home/valen/marita_file
Enter passphrase for key '/home/emilia/.ssh/id_dsa':<Passphrase>
carol_file 100% |*****************************|    64       00:00
emilia@svr02 $ ssh -l valen svr01 'ls -l marita_file'
-rw-r--r--   1 valen staff            64 Apr 03 21:20 marita_file
```

For further information about the **scp** command, please refer to the online command reference by issuing the following command:

```
$ MANPATH=/usr/local/man:$MANPATH; export MANPATH
$ man scp
```

4.3.7 Using the sftp command

If you are familiar with the user interface of the **ftp** command, you can use the **sftp**[9] command to securely transfer files. If you invoke the **sftp** command, as shown in the following example, you will be prompted by the sftp> subcommand prompt to enter subcommands after entering the passphrase:

```
emilia@svr02 $ sftp valen@svr01.itsc.austin.ibm.com
Connecting to svr01.itsc.austin.ibm.com...
Enter passphrase for key '/home/emilia/.ssh/id_dsa': <Passphrase>
sftp>
```

Example 4-21 shows the available sub-commands of the **sftp** command.

Example 4-21 Subcommands for the sftp command

```
sftp> help
Available commands:
```

[9] The **sftp** command connects to sftp-server specified in /etc/ssh/sshd_config instead of ftpd.

```
cd path                       Change remote directory to 'path'
lcd path                      Change local directory to 'path'
chgrp grp path                Change group of file 'path' to 'grp'
chmod mode path               Change permissions of file 'path' to 'mode'
chown own path                Change owner of file 'path' to 'own'
help                          Display this help text
get remote-path [local-path]  Download file
lls [ls-options [path]]       Display local directory listing
ln oldpath newpath            Symlink remote file
lmkdir path                   Create local directory
lpwd                          Print local working directory
ls [path]                     Display remote directory listing
lumask umask                  Set local umask to 'umask'
mkdir path                    Create remote directory
put local-path [remote-path]  Upload file
pwd                           Display remote working directory
exit                          Quit sftp
quit                          Quit sftp
rename oldpath newpath        Rename remote file
rmdir path                    Remove remote directory
rm path                       Delete remote file
symlink oldpath newpath       Symlink remote file
version                       Show SFTP version
!command                      Execute 'command' in local shell
!                             Escape to local shell
?                             Synonym for help
```

For further information about the **sftp** command, please refer to the online command reference by issuing the following command:

```
$ MANPATH=/usr/local/man:$MANPATH; export MANPATH
$ man sftp
```

4.3.8 Authentication agents

You have to enter your passphrase for every remote login attempt using OpenSSH. The authentication agent ssh-agent allows you to use the OpenSSH client commands without a passphrase. The agent stores the authentication private keys in memory.

The authentication agent will spawn the specified command on the command line. Typically, you specify the variable `$SHELL` to invoke the child shell process, as shown in the following example:

```
$ ssh-agent $SHELL
```

To verify that the ssh-agent process has been started, issue the following command:

```
$ ps -ef | grep ssh-agent | grep -v grep
    emilia 25096 25746   0 18:09:23      -  0:00 ssh-agent /usr/bin/ksh
```

After invoking the authentication agent, you have to load your private key into the agent using the **ssh-add** command. To load your private key into the agent, do the following:

```
$ ssh-add $HOME/.ssh/id_dsa
Enter passphrase for id_dsa: <Enter your passphrase>
Identity added: id_dsa (id_dsa)
```

You will be prompted to enter your passphrase.

Note: Although it is convenient to use, you should be aware of the potential security risk while you are using the authentication agent. If a malicious user takes over your terminal, he/she can remotely log in to any system without a prompted passphrase. You should terminate the authentication agent before you leave your terminal.

To terminate the authentication agent, issue the following command:

```
$ ssh-agent -k
```

The agent process exits after relinquishing the stored private key from the memory.

Automate OpenSSH sessions

In some cases, you have to balance the security and business requirements. When remote batch processing is required, you should have an alternative to automate OpenSSH sessions. However, if you keep running the authentication agent, you might nullify all the effort for network security in server farms.

To solve this dilemma, we provide you a script written in Perl/Expect, which is explained in Section 3.3, “Automating administrative operations” on page 75.

The launcher script, shown in Example 4-22 on page 152, invokes the authentication agent on local host with a passphrase, connects to the remote server using the **ssh** command, and shows permanent hardware error entries using the **errpt** command. Then, the script exits from the remote server and terminates the authentication agent.

Example 4-22 The launcher script

```
#!/usr/bin/perl
#
# launcher -- Launch the ssh-agent process without entering passphrase
#             then connect to the remote host using the ssh command
#             to automate viewing the AIX error log.

use Expect;
$logfile = "./launcher.log";
$pass = "passphrase";
$ENV{"SHELL"} = "/bin/sh";
# Start the ssh-agent process and get the SSH_* environment values.
open(TT, "/usr/bin/ssh-agent -s 2>&1 | head -2 |");
while (<TT>) {
    if (/^(SSH[^=]*) = ([^;]*);.*$/) {
        $ENV{$1} = $2;
        }
}
close(TT);

# Start a new shell.
$e = Expect->spawn("/bin/sh");
# Wait for the command line prompt.
&timeout unless ($e->expect(10, "# "));
# Execute ssh-add to add an identity to the ssh-agent process.
$e->send("/usr/bin/ssh-add\r");
&timeout unless ($e->expect(10, "-re", "Enter passphrase for.*:"));
$e->send($pass . "\r");

# Wait for the command line prompt.
&timeout unless ($e->expect(10, "# "));
# Invoke ssh to connect to svr01.
$e->send("ssh svr01\r");

# Wait for the command line prompt.
&timeout unless ($e->expect(10, "# "));
# Set the environment variable LANG=C.
$e->send("export LANG=C\r");
# Wait for the command line prompt.
&timeout unless ($e->expect(10, "# "));
# Turn logging on.
$e->log_file($logfile,"w");
# Execute the errpt command.
$e->send("errpt -d H\r");
# Wait for the command line prompt.
&timeout unless ($e->expect(10, "# "));
# Turn logging off.
$e->log_file(undef);
```

```
# Exit the shell connected via ssh.
$e->send("exit\r");
# Wait for the command line prompt.
&timeout unless ($e->expect(10, "# "));
# Terminate the ssh-agent process.
$e->send("/usr/bin/ssh-agent -k\r");
# Wait for the command prompt of the original shell.
&timeout unless ($e->expect(10, "# "));

sleep(1);
$e->hard_close();
printf("done.\n");
exit 0;

sub timeout{
    sleep(1);
    $e->hard_close();
    printf("timeout!\n");
    exit 1;
}
```

> **Note:** The script contains the passphrase in clear text format, as shown in the high-lighted line. Therefore, you should securely store the script. The script must have permission mode of 0700 as a minimum security.

Example 4-23 shows the execution example of this script. Please note that we have invoked only the launcher script, as shown in the first line in Example 4-23.

Example 4-23 Execution example of the launcher script

```
# launcher
# /usr/bin/ssh-add
Enter passphrase for //.ssh/identity:
Identity added: //.ssh/identity (//.ssh/identity)
root@svr02:/ [304] # ssh svr01
Last unsuccessful login: Thu Apr  4 16:54:32 CST 2002 on /dev/pts/5 from snecac
Last login: Tue Apr  9 14:54:55 CDT 2002 on ssh from svr02.itsc.austin.ibm.com

*******************************************************************************
*                                                                             *
*                                                                             *
*  Welcome to AIX Version 5.1!                                                *
*                                                                             *
*                                                                             *
*  Please see the README file in /usr/lpp/bos for information pertinent to    *
*  this release of the AIX Operating System.                                  *
*                                                                             *
*                                                                              *
********************************************************************************
[YOU HAVE NEW MAIL]
root@svr01 # export LANG=C
root@svr01 # errpt -d H
IDENTIFIER TIMESTAMP  T C RESOURCE_NAME  DESCRIPTION
499B30CC   0327153102 T H ent1           ETHERNET DOWN
499B30CC   0326192802 T H ent1           ETHERNET DOWN
499B30CC   0319083002 T H ent1           ETHERNET DOWN
root@svr01 # exit
Connection to svr01 closed.
# /usr/bin/ssh-agent -k
unset SSH_AUTH_SOCK;
unset SSH_AGENT_PID;
echo Agent pid 8512 killed;
# done.
```

4.3.9 TCP/IP forwarding

TCP/IP forwarding is a powerful function that allows you to securely establish TCP connections between local and remote hosts by forwarding connection ports. TCP based protocols[10], such as SMTP and POP3 (both are used by e-mail services), can exploit TCP/IP forwarding using the secure connection. You can also use multiple tunnels among multiple hosts by concatenating tunnels. This allows you to extend the secure connection to any hosts, for example, beyond the firewall, if it allows the connection using OpenSSH.

TCP/IP forwarding is classified into two modes according to where the port is forwarded:

Local forwarding A connection is forwarded on the client host.

Remote forwarding[11] A connection is forwarded on the remote host.

Figure 4-3 on page 155 illustrates the concept of local TCP/IP forwarding. You want to connect an application client process (shown as A) on the client to an application server process (shown as B) on the application server. The application client uses the port specified as *local port* and the application server is listening on the port specified as *remote port*.

[10] As for the FTP protocol, we recommend you use the **sftp** command rather than TCP/IP forwarding.

[11] Throughout this section, we explain local forwarding only, because remote forwarding is not commonly used.

Before invoking the application client process, you have to establish a secure connection (shown as C) using the **ssh** command between the client and the SSH server. The ssh client process is instructed to enable a local port forwarding upon invocation, allowing you to connect the client application process to itself. If you invoke the application client process, then it can connect to the application server process using the secure tunnel provided by OpenSSH. Please note that we assume that OpenSSH uses the default port number 22.

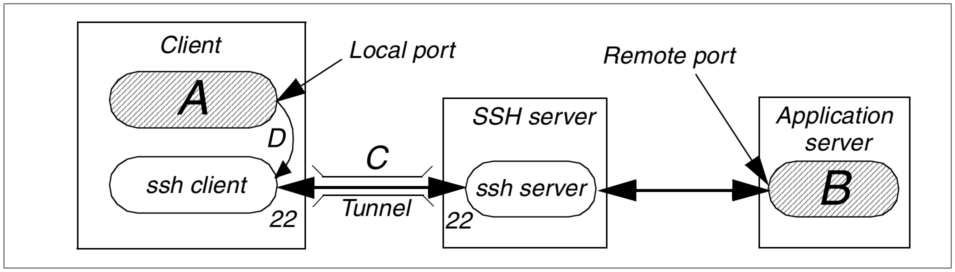


Figure 4-3 TCP/IP forwarding (local port)

To use TCP/IP forwarding, you have to invoke the **ssh** command before the application connection using a complex command syntax, as shown in the following example:

```
$ ssh -L <local_port>:<remote_host>:<remote_port> <user_name>@<ssh_server>
```

Where:

local_port Port number to be used for the local application client process. You must have root privilege to use a local port number less than or equal to 1024.

remote_host Host name where the application server process is running.

remote_port Port number where the application server process is listening.

user_name User name on the OpenSSH server.

ssh_server Host name of the OpenSSH server.

Note: The two hosts, remote_host and ssh_server, can be the same.

Figure 4-3 illustrates an example where a Web browser (application client process) on the client (svr02) connects to the httpd process (application server process) on the server (svr03) through the OpenSSH server (svr01). We assume that the httpd process is listening on port number 80.

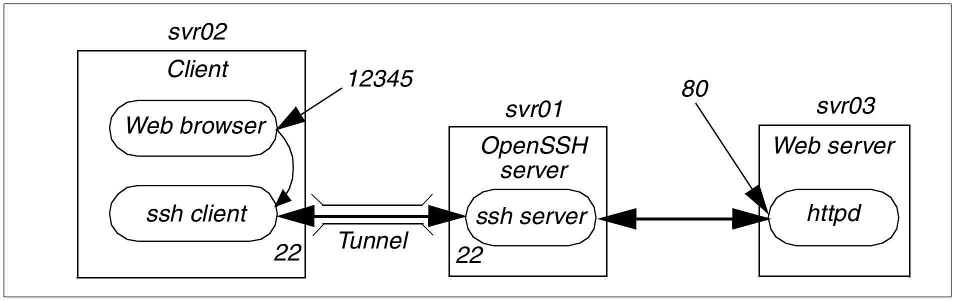


Figure 4-4 TCP/IP forwarding (local port)

To use TCP/IP forwarding as shown in Figure 4-3 on page 155, do the following:

1. On svr02, execute the following commands:

   ```
   emilia@svr02 $ ssh -L 12345:svr03:80 valen@svr01
   ```

   In this example, the specified command line options are:

   -L Allows local TCP/IP forwarding

   12345 Port number to be used for forwarded connection for the Web browser on svr02

   svr03 Host name where the httpd process is running

   80 Port number where the httpd process is listening

   svr01 Host name of the OpenSSH server

   It returns the command prompt on svr01, as shown in the following example:

   ```
   valen@svr02 $
   ```

   Note: Do not close this ssh session while you are using TCP/IP forwarding.

2. On svr02, you will notice that the port number 12345 is in `LISTEN` status, as shown in the following example:

   ```
   root@svr02:/ [130] # netstat -an | grep 12345
   tcp4       0      0  127.0.0.1.12345       *.*                  LISTEN
   ```

3. On svr02, invoke a Web browser, such as Netscape Navigator. Disable any proxy and socks configuration for the web browser and visit the URL `http://localhost:12345`. You can connect to the httpd process on svr03 using TCP/IP forwarding established by OpenSSH.

4.3.10 X11 forwarding

X11 forwarding is an extension to TCP/IP forwarding. Because there are so many graphical user interface (GUI) applications running on X[12], secure connections for remote X clients and servers are strongly demanded.

Figure 4-5 on page 158 illustrates a connection example between X clients and a X server. On svr02, the X client application process (Xclient2) connects to a local X server process. On svr01, another X client application process (Xclient1) connects to the remote X server process over the network.

To facilitate these connection topologies, the X protocol defines the DISPLAY environment value specified as *hostname:N.M*:

hostname The host name where the X server process is running.

N An integer value that defines a display number.

M An integer value that defines a virtual display number.

You use the integer value 0 for both N and M, unless you have multiple graphics displays.

In Figure 4-5 on page 158, the local X client process (Xclient2) uses `localhost:0.0` as the DISPLAY environment value and the remote X client process (Xclient1) uses `svr02:0.0`.

[12] The authentication method to be used for the remote connections between X clients and servers is widely considered as vulnerable.

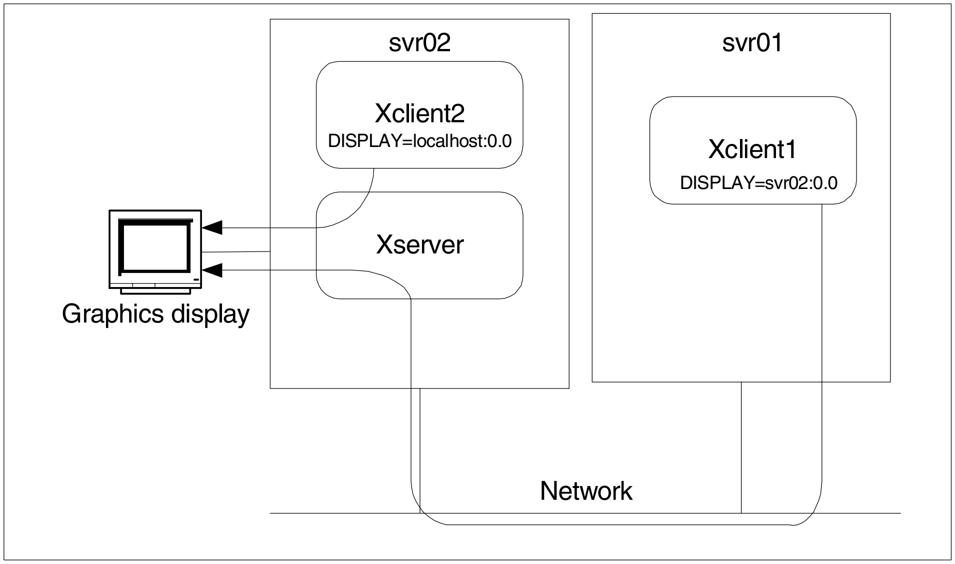


Figure 4-5 X11 remote display session

You can create a secure tunnel that protects remote connections between the X client and server processes over the network by using X11 forwarding.

To use X11 forwarding, do the following:

1. Verify whether the remote OpenSSH server you are going to connect to supports X11 forwarding. If the /etc/ssh/sshd_config file on the remote server has the line shown in the following example, you have to change the `no` string to `yes` and restart the sshd process:

   ```
   X11forwarding no
   ```

2. On svr02, verify whether the local X server process grants remote connections using the **xhost** command. If the command returns the following output:

   ```
   emilia@svr02 $ xhost
   access control enabled, only authorized clients can connect
   ```

   then you have to issue the following command to instruct the X server process to grant remote connections from svr01:

   ```
   emilia@svr02 $ xhost +svr01.itsc.austin.ibm.com
   svr01.itsc.austin.ibm.com being added to access control list
   ```

3. On svr02, execute the following commands:

   ```
   emilia@svr02 $ ssh -l valen -X svr01
   ```

   The -X option instructs the **ssh** command to enable X11 forwarding.

The command returns the command prompt on svr01, as shown in the following example:

```
valen@svr02 $
```

Note: Do not close this ssh session while you are using X11 forwarding.

4. Verify the DISPLAY environment value on the remote server:

```
valen@svr01 $ echo $DISPLAY
svr01.itsc.austin.ibm.com:10.0
```

Now, if you invoke X client applications, such as **xclock** and Netscape Navigator, in this session, those clients connect to the local X server process on svr02.

4.4 PuTTY: An SSH2 protocol client on Windows

PuTTY is a free software tool that supports several protocols, such as telnet, SSH1, and SSH2, on the Microsoft Win32[13] platform for the IA32 architecture.

We explain how to use PuTTY on Microsoft Windows clients to be used by system administrators to secure the network connection between these clients and AIX systems running the OpenSSH server.

For detailed information about PuTTY, please visit the following URL:

http://www.chiark.greenend.org.uk/~sgtatham/putty

4.4.1 Installing PuTTY

To install PuTTY, visit the following URL:

http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

This URL contains the following executables that are used in this section:

putty.exe	PuTTY secure shell client.
puttygen.exe	Key generator utility for RSA and DSA keys.
pageant.exe	Authentication agent to be used with PuTTY.

Because these executable programs do not provide any specific installation procedures, you can simply download them into an appropriate directory, such as C:\Program Files\PuTTY.

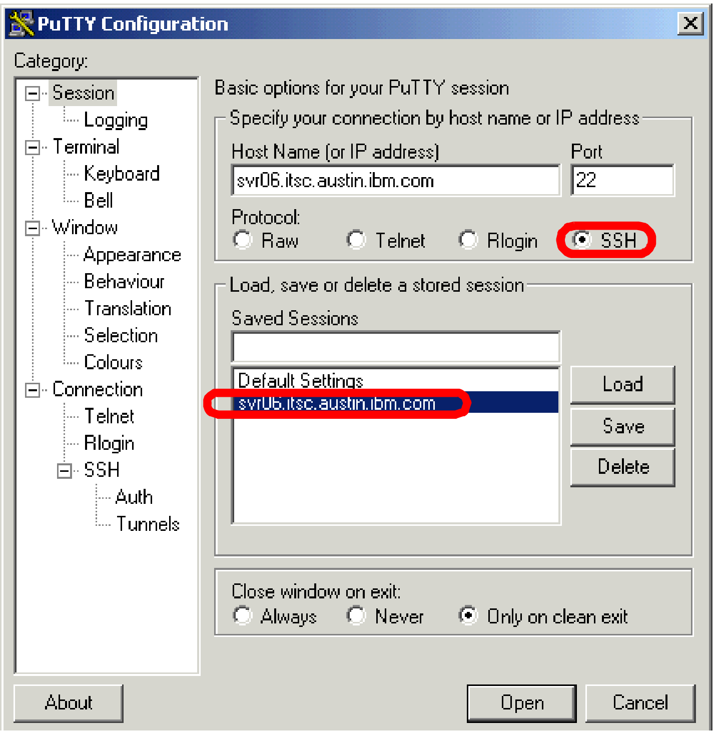
[13] The Microsoft Win32 platform is comprised of the Microsoft Windows 95, 98, Me, 2000, and XP operating systems.

4.4.2 Using PuTTY

In order to use PuTTY using the SSH2 protocol, do the following:

1. Run putty.exe from the file explorer. You will see the configuration panel, as shown in Figure 4-6.
2. Enter the host name (or IP address) of the OpenSSH server to which you are going to connect.
3. Select SSH as a protocol. You can notice the port number is changed to 22.
4. Select Connection in the category pane and enter the remote user login name in the Auto-login username field.
5. To save this session for future use, enter the session name in the Saved sessions field and click Save. You will see the saved session name under Default Settings.

   Once you saved the session configuration, you can simply select this session name from the Saved session field and click Load.
6. To connect to the OpenSSH server, click Open.



Host Name (or IP address): svr06.itsc.austin.ibm.com
Port: 22
Protocol: ○ Raw ○ Telnet ○ Rlogin ● SSH

Saved Sessions:
Default Settings
svr06.itsc.austin.ibm.com

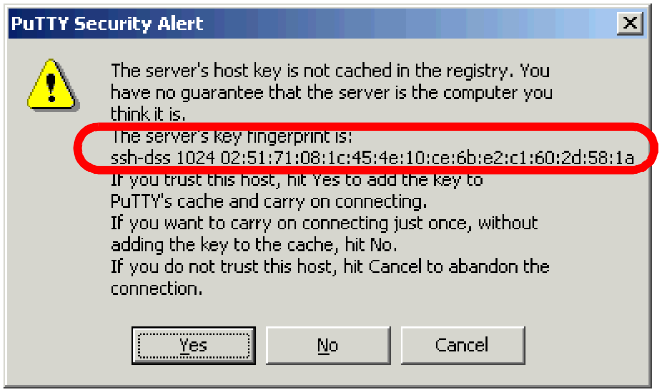
Close window on exit: ○ Always ○ Never ● Only on clean exit

Figure 4-6 PuTTY configuration

7. Upon first connection to the OpenSSH server, you will see an alert panel, as shown in Figure 4-7. It notifies you of the server key fingerprint. You should confirm it matches the fingerprint given by the administrator of the OpenSSH server to which you want to connect.

   Once you confirm that you can trust the server key fingerprint, select Yes.

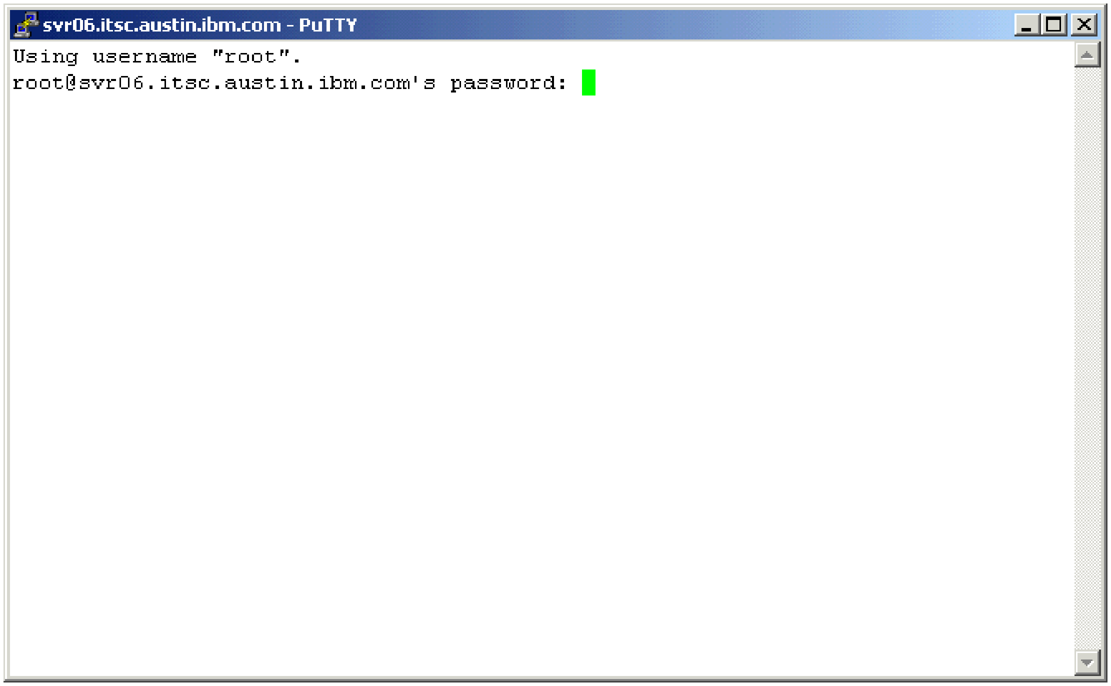
   Note: This security alert panel only appears upon the first connection.



The server's host key is not cached in the registry. You have no guarantee that the server is the computer you think it is.
The server's key fingerprint is:
ssh-dss 1024 02:51:71:08:1c:45:4e:10:ce:6b:e2:c1:60:2d:58:1a
If you trust this host, hit Yes to add the key to PuTTY's cache and carry on connecting.
If you want to carry on connecting just once, without adding the key to the cache, hit No.
If you do not trust this host, hit Cancel to abandon the connection.

Figure 4-7 PuTTY first time connection

8. In the open PuTTY terminal window, you have to enter the password of the login user.
9. Finally, you will see the command prompt in the PuTTY terminal window, as shown in Figure 4-8 on page 162.



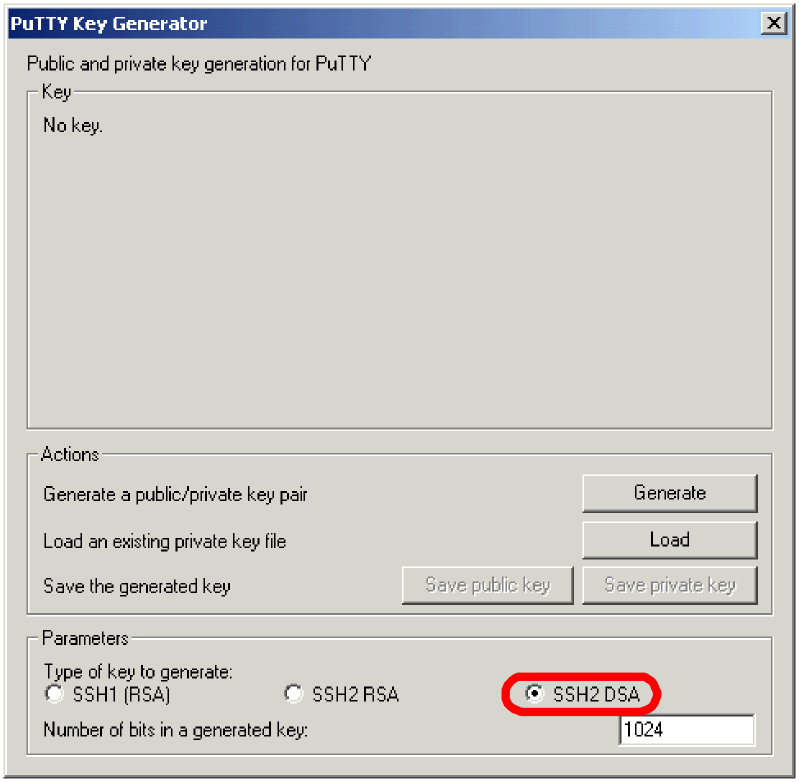
```
Using username "root".
root@svr06.itsc.austin.ibm.com's password:
```

Figure 4-8 PuTTY terminal window

4.4.3 Using a passphrase

In order to use a passphrase on your Windows client, you have to generate a pair of user keys using the PuTTY key generator utility, puttygen.exe, as follows:

1. Run puttygen.exe from the file explorer. You will see the PuTTY Key Generator panel, as shown in Figure 4-9 on page 163.


PuTTY Key Generator

Public and private key generation for PuTTY

Key

No key.

Actions

Generate a public/private key pair

Load an existing private key file

Save the generated key

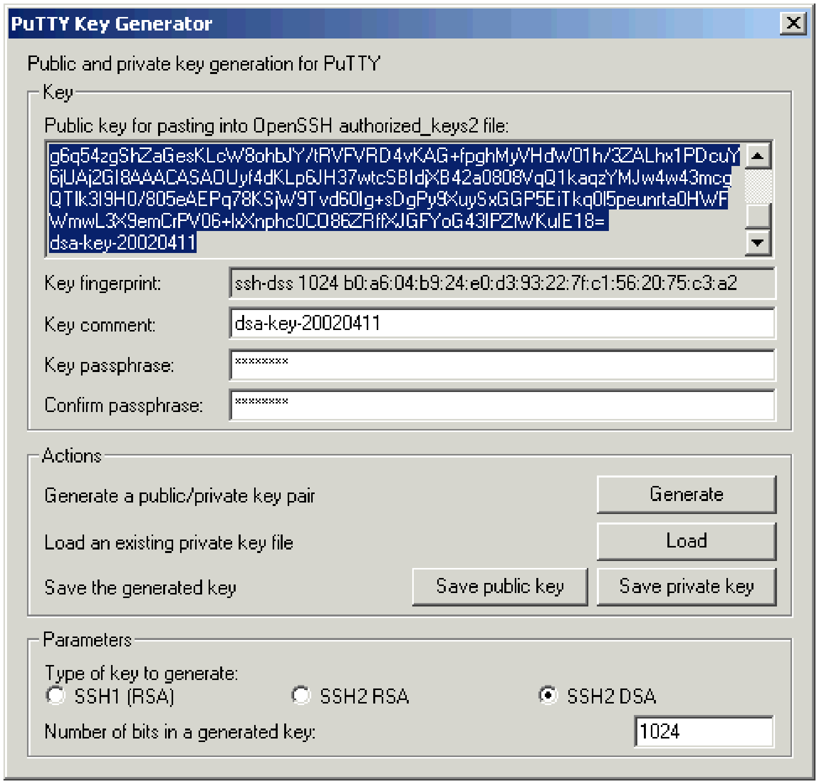
Parameters

Type of key to generate:
○ SSH1 (RSA) ○ SSH2 RSA ● SSH2 DSA

Number of bits in a generated key: 1024

Figure 4-9 PuTTY key generator

2. Select SSH2 DSA and verify if “Number of bits in a generated key” is 1024.
3. Click Generate and keep moving the mouse cursor over the blank area of this panel. After the user key is generated, you will see a panel, as shown in the Figure 4-10 on page 164.



Public key for pasting into OpenSSH authorized_keys2 file:

```
g6q54zgShZaGesKLcW8ohbJY/tRVFVRD4vKAG+fpghMyVHdW01h/3ZALhx1PDcuY
6jUAj2GI8AAACASA0Uyf4dKLp6JH37wtcSBIdjXB42a0808VqQ1kaqzYMJw4w43mcg
QTIk3I9H0/805eAEPq78KSjW9Tvd60Ig+sDgPy9XuySxGGP5EiTkq0I5peunrta0HWF
WmwL3X9emCrPV06+lxXnphc0CO86ZRffXJGFYoG43IPZMWKuIE18=
dsa-key-20020411
```

Key fingerprint: ssh-dss 1024 b0:a6:04:b9:24:e0:d3:93:22:7f:c1:56:20:75:c3:a2

Key comment: dsa-key-20020411

Key passphrase: ********

Confirm passphrase: ********

Generate a public/private key pair — Generate

Load an existing private key file — Load

Save the generated key — Save public key — Save private key

Type of key to generate:
○ SSH1 (RSA) ○ SSH2 RSA ● SSH2 DSA

Number of bits in a generated key: 1024

Figure 4-10 Generated public and private keys

4. Enter your passphrase twice in the Key passphrase and the Confirm passphrase fields.

> **Note:** Do not forget your passphrase, because the passphrase cannot be reset. If you forget your passphrase, you will have to regenerate your user key pair.

5. Click Save public key to save the public key.
6. Click Save private key to save the private key encrypted with the passphrase.
7. Select and copy the public key from the Key field (highlighted text area).
8. Login to the OpenSSH server and open the $HOME/.ssh/authorized_keys file using the vi editor, then add the public key to the file by right-clicking on the PuTTY terminal window.

Once you have copied the generated public key to the $HOME/.ssh/authorized_keys file on the OpenSSH server, you can follow the following steps to connect to the OpenSSH server using the passphrase:

1. Run putty.exe and load the previously saved session configuration.
2. Select the Connection category and verify the remote user login name on the Auto-login username field.
3. Select Auth under the SSH category. Click Browse… to find the private key file you saved previously.
4. Click Open to connect to the OpenSSH server.
5. Enter the passphrase when you are prompted. You will not be prompted to enter the login password.

4.4.4 Using authentication agents

If you are running the PuTTY authentication agent, you will not be prompted to enter the passphrase upon the connection to the OpenSSH server.

To use the authentication agent, do the following:

1. Run pageant.exe from the file explorer.
2. A small icon will be displayed in the Windows task bar:

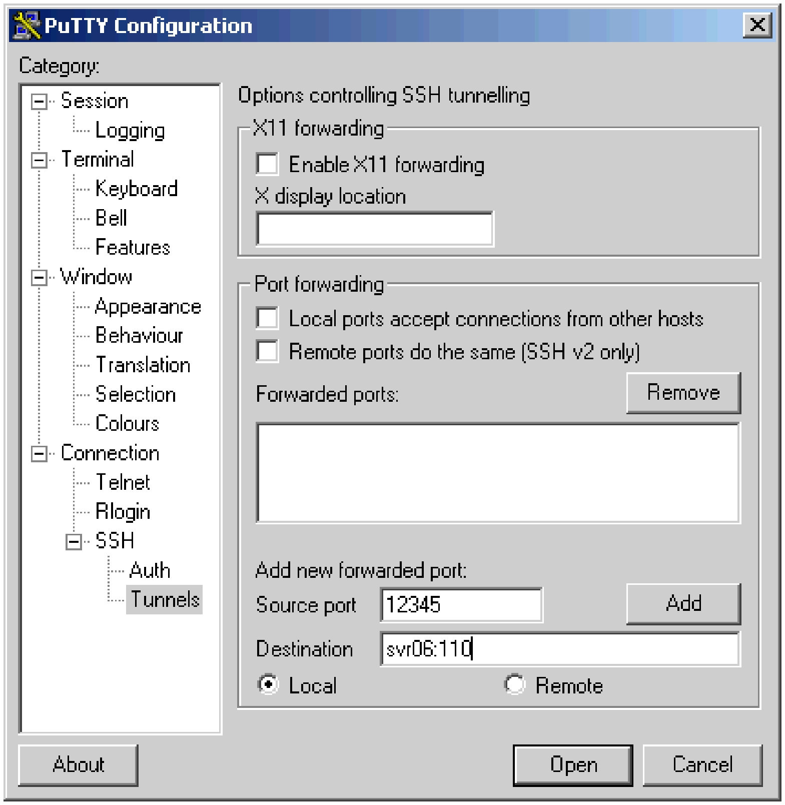
3. Right-click on the icon and select Add Key.
4. Select the private key file that you saved previously on the file selection dialog panel, then click Open. If your private key is encrypted with the passphrase, you will be prompted to enter it.

You are ready to connect to the OpenSSH server using the steps explained in Section 4.4.3, “Using a passphrase” on page 162 without entering your passphrase.

4.4.5 Configuring TCP/IP forwarding using PuTTY

The following steps show an example of checking your POP[14] e-mail account using the local TCP/IP forwarding function provided by PuTTY:

1. In the PuTTY configuration panel, select the server you want to connect to.
2. Select the Tunnels category, as shown in Figure 4-11.



□ Enable X11 forwarding

X display location

□ Local ports accept connections from other hosts

□ Remote ports do the same (SSH v2 only)

Source port 12345

Destination svr06:110

● Local ○ Remote

Figure 4-11 TCPIP local forwarding with PuTTY

3. Enter a port number greater than 1024 in the source port field. In this example, we enter 12345 for the source port.
4. Enter a combination of strings in the destination field. In this example, the server name svr06 and the remote port number 110 are specified.
5. Select Local for local forwarding, then click Add.

[14] POP (Post Office Protocol) is a protocol to be used for remote connections between the e-mail server and clients.

6. Click Open, and log in to the remote server that provides you an e-mail service using the POP protocol.
7. After logging in the server, do not close this session while you are using the local TCP/IP forwarding service.
8. Invoke your POP client program and specify localhost:12345 for the e-mail server address for your POP e-mail service. The POP client is now ready to securely receive your e-mails using the SSH2 protocol.

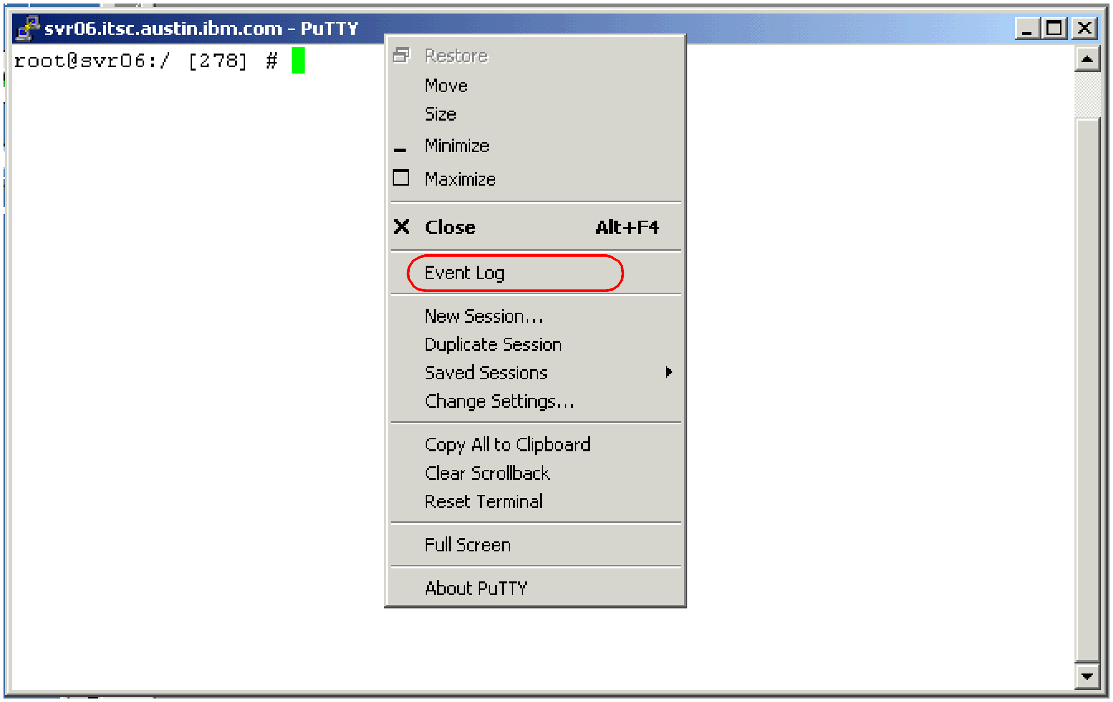
4.4.6 Configuring X11 forwarding using PuTTY

To use X11 forwarding using PuTTY, do the following:

1. Select the Tunnels category in the configuration panel, as shown in Figure 4-11 on page 166.
2. Select Enable X11 forwarding and enter an appropriate DISPLAY environment variable in the X display location field.
3. Click Open.
4. Log in to the OpenSSH server and run X11 applications remotely.

> **Note:** To use X11 applications on Microsoft Windows, you have to separately install an X11 server application on your Windows-based PC.

To verify that X11 forwarding is enabled, right-click on the title bar of the PuTTY terminal window, then select Event Log, as shown in Figure 4-12 on page 168. You will see the entry `X11 forwarding enabled`. You can also confirm this action by issuing the **`echo $DISPLAY`** command



```
root@svr06:/ [278] #
```

Figure 4-12 PuTTY pull-down menu

5

Remote monitoring using SNMP

The Simple Network Management Protocol (SNMP) is a UDP based protocol, which is commonly used for system management and monitoring on IP networks. Although the deployment of SNMP management system is quite common in the today's enterprise network, it is typically used for monitoring network devices, such as routers and switches.

This chapter contains the following three sections:

- Section 5.1, “Basic understanding of SNMP” on page 170
- Section 5.2, “SNMP commands on AIX” on page 174
- Section 5.3, “NET-SNMP” on page 185

The first short section provides you with a basic understanding of SNMP. The second section provides you the SNMP implementation usage on AIX 5L Version 5.1. The last section exploits a free software tool, NET-SNMP, to provide you with customization examples for advanced remote monitoring tasks.

5.1 Basic understanding of SNMP

Although the deployment of the SNMP management system is quite common in today's enterprise network, it is typically used for monitoring network devices, such as routers and switches. Therefore, before implementing monitoring systems using SNMP on AIX, you should understand the SNMP basics described in this section.

This short section does not provide you extensive information about SNMP. For further understanding of SNMP, you should refer to the following publications:

- *Managing Internetworks With Snmp, 3rd Ed*, by Miller, et al
- *Essential SNMP*, Mauro, et al

5.1.1 SNMP overview

The SNMP protocol defines two terms, agent and manager, instead of the client and server used in many other TCP/IP protocols:

SNMP agent

An SNMP agent is a daemon process that provides access to the MIB objects on IP hosts that the agent is running on. The agent can receive SNMP get or SNMP set requests from SNMP managers and can send SNMP trap requests to SNMP managers. On AIX, the SNMP agent is implemented as /usr/sbin/snmpd (see Section 5.2.1, "AIX SNMP agent" on page 175).

SNMP manager

An SNMP manager can be implemented in two ways. An SNMP manager is implemented as a simple command tool that can collect information from SNMP agents. On AIX, a simple SNMP manager is implemented as the **/usr/sbin/snmpinfo** command (see Section 5.2.3, "AIX SNMP manager" on page 178).

An SNMP manager also can be composed of multiple daemon processes and database applications. This type of complex SNMP manager provides you with monitoring functions using SNMP. It commonly has a graphical user interface for operators. The SNMP manager gathers information from SNMP agents and accepts trap requests sent by SNMP agents. On AIX, the most popular SNMP manager is Tivoli NetView (see Section 5.2.5, "Tivoli Netview" on page 180).

Figure 5-1 on page 171 illustrates the relationship between an SNMP agent and its manager.

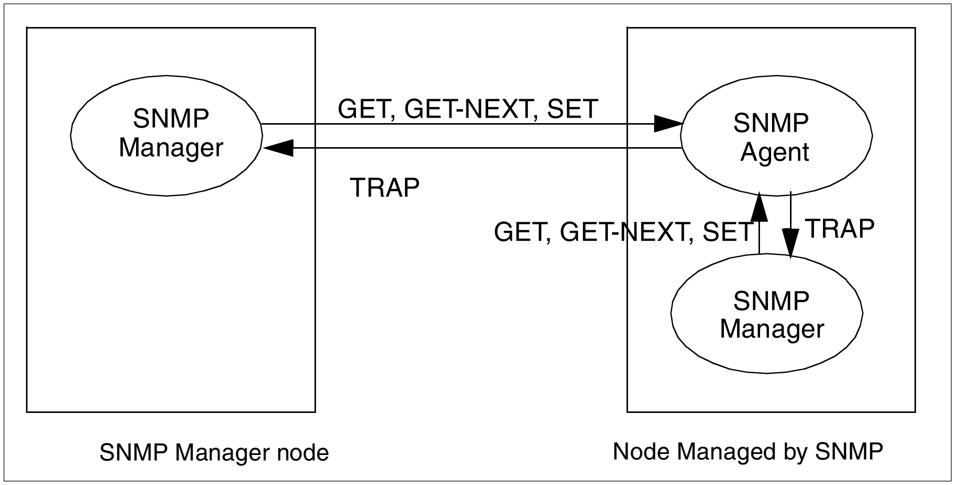


Figure 5-1 SNMP manager and agent

The SNMP manager sends SNMP get, get-next, or set requests to SNMP agents, which listen on UDP port 161, and the agents send back a reply to the manager. The SNMP agent can be implemented on any kind of IP host, such as UNIX workstations, routers, and network appliances. You can gather various information on the specific IP hosts by sending the SNMP get and get-next request, and can update the configuration of IP hosts by sending the SNMP set request.

The SNMP agent can send SNMP trap requests to SNMP managers, which listen on UDP port 162. The SNMP trap[1] requests sent from SNMP agents can be used to send warning, alert, or error notification messages[2] to SNMP managers.

Please note that you can configure an SNMP agent to send SNMP trap requests to multiple SNMP managers.

Generic SNMP security

The SNMP protocol uses the community name for authorization. Most SNMP implementations use the default community name *public* for read-only community, and *private* for a read-write community. In most cases, a community name is sent in a plain-text format between the SNMP agent and manager. Some SNMP implementations have additional security features, such as the restriction of the accessible IP addresses.

[1] The SNMP trap request is sometimes referred to as an asynchronous message.

[2] If the IP host or network media is down while sending SNMP TRAP requests, these messages might be lost.

Therefore, you should be careful about the SNMP security. At the very least:

- Do not use the default community name (public and private).
- Do not allow access to IP hosts that are running the SNMP agent, from an unnecessary network or IP host.

You might want to physically secure the network where you send SNMP packets to by using a firewall, as community strings are included as plain text in SNMP packets. SNMP Version 3 defines an enhanced authentication mechanism to cover these issues, but it has not been widely implemented yet.

The CERT security alert CA-2002-03 provides you with a good explanation about generic SNMP security (see “Background information” on page 298).

Note: All the IP hosts implementing SNMP might be affected by the SNMP vulnerabilities explained in CA-2002-03, and not just the AIX systems.

5.1.2 Message Information Base (MIB)

The objects, which you can get or set by sending SNMP get or set requests, are defined as a set of databases called Message Information Base (MIB). The structure of MIB is defined as an Internet standard in RFC 1155. The MIB forms a tree structure, as shown in Figure 5-2.

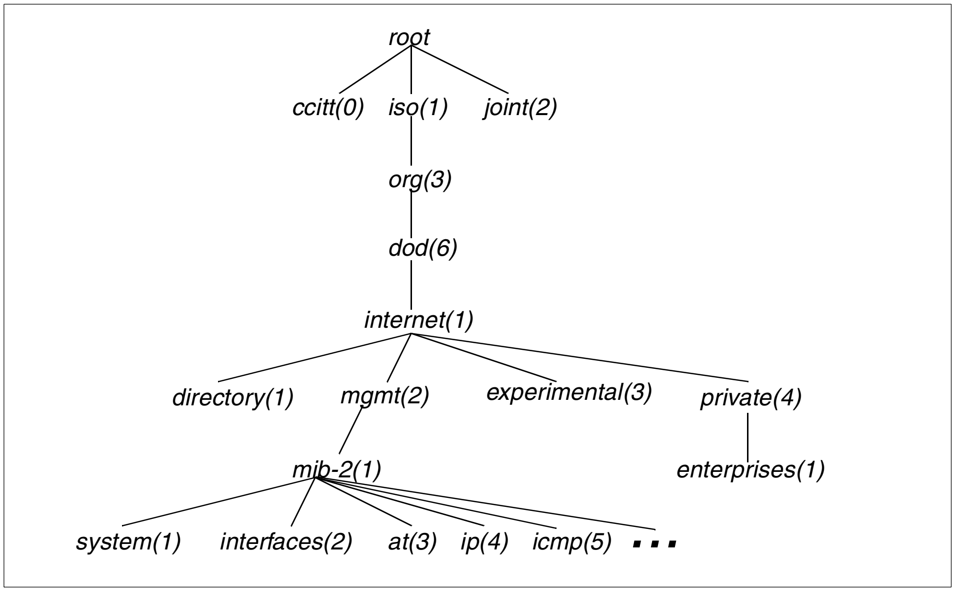


Figure 5-2 A MIB tree example

You can manipulate common objects defined as standard MIB, such as MIB-II (defined in RFC 1213), on the network devices that implement the SNMP interface. If you want to get a system description of an IP host, you can send the IP host an SNMP get request for the MIB object specified by the following object ID (OID):

```
.iso.org.dod.internet.mgmt.mib-2.system.sysDescr.0
```

This name specifies the name of the object (sysDescr.0) in the MIB tree. Each MIB tree node has a numerical identification as well as a symbolic one. For example, the node .iso is equivalent to .1, and the .iso.org is equivalent to .1.3, and so on (see the number specified in parentheses in Figure 5-2 on page 172). Therefore, the OID .iso.org.dod.internet.mgmt.mib-2.system.sysDescr.0 is equivalent to the numerical representation of .1.3.6.1.2.1.1.1.0.

Most hardware and software vendors provide you with extended MIB objects to support their own requirements. The SNMP standards allow this extension by using the *private* sub-tree, called enterprise specific MIB, under the iso.org.dod.internet sub-tree.private.enterprise. Because each vendor has unique MIB sub-tree under the private sub-tree, there is no conflict among vendor original MIB extension. For example, IBM reserves the .iso.org.dod.internet.private.ibm sub-tree, as shown in Figure 5-3.

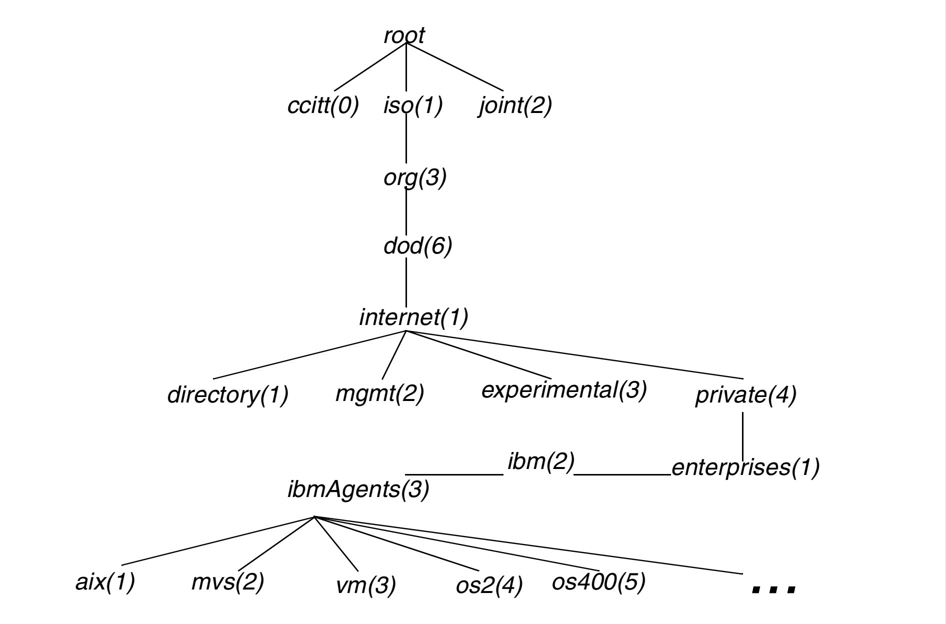


Figure 5-3 IBM MIB tree

5.1.3 SNMP trap requests

An SNMP agent can send SNMP trap requests to SNMP managers to inform them of the change of values or statuses on the IP host where the agent is running. There are seven predefined types of SNMP trap requests, as shown in Table 5-1.

Table 5-1 SNMP trap request types

Trap type	Value	Description
coldStart	0	Restart after a crash.
warmStart	1	Planned restart.
linkDown	2	Communication link is down.
linkUp	3	Communication link is up.
authenticationFailure	4	Invalid SNMP community string was used.
egpNeighborLoss	5	EGP neighbor is down.
enterpriseSpecific	6	Vendor specific event happened.

A trap message contains pairs of an OID and a value shown in Table 5-1 to notify the cause of the trap message. You can also use type 6, the enterpriseSpecific trap type, when you have to send messages that are not fit for other predefined trap types, for example, DISK I/O error and application down. You can also set an integer value field called *Specific Trap* on your trap message.

5.2 SNMP commands on AIX

AIX includes an SNMP agent (snmpd) and an SNMP manager (snmpinfo). These two commands are included in the following fileset:

```
# lslpp -w /usr/sbin/snmpd; lslpp -w /usr/sbin/snmpinfo
  File                                        Fileset               Type
  ----------------------------------------------------------------------------
  /usr/sbin/snmpd                             bos.net.tcp.client    File
  File                                        Fileset               Type
  ----------------------------------------------------------------------------
  /usr/sbin/snmpinfo                          bos.net.tcp.server    File
```

AIX also includes SNMP related daemons and commands, as explained in the next section.

5.2.1 AIX SNMP agent

On AIX, the SNMP agent is installed as /usr/sbin/snmpd. The AIX **snmpd** is robust and extensible, using interfaces described in Section 5.2.2, “Relationship among SNMP related processes” on page 176.

snmpd operation

The snmpd daemon is controlled by the system resource controller (SRC) on AIX. The command is invoked from /etc/rc.tcpip upon system initialization, as shown in the following example:

```
# grep snmp /etc/rc.tcpip
start /usr/sbin/snmpd "$src_running"
```

To confirm that snmpd is running, you can use either of the following commands:

```
# ps -ef | grep snmpd | grep -v grep
    root  8258  5680  0  Mar 20      -  0:04 /usr/sbin/snmpd
# lssrc -s snmpd
Subsystem         Group            PID     Status
 snmpd            tcpip            8258    active
```

You can stop, restart, or refresh snmpd by issuing the following commands:

Stop	**`stopsrc -s snmpd`**
Restart	**`startsrc -s snmpd`**
Refresh	**`refresh -s snmpd`**

You can customize snmpd by modifying the /etc/snmpd.conf file. To take the modification into effect, you have to restart the daemon. The snmpd daemon sends warmStart trap requests to the SNMP manager (if configured) upon refresh. The snmpd daemon sends coldStart trap to SNMP manager upon restart.

Customization example of /etc/snmpd.conf

The snmpd daemon reads the configuration file named /etc/snmpd.conf upon start-up. The default /etc/snmpd.conf file provides you with many useful comments, as shown in Example A-1 on page 288. By default, the snmpd daemon accepts the access from any IP reachable hosts:

```
# grep '^community' /etc/snmpd.conf
community       public
```

As explained in "Generic SNMP security" on page 171, this configuration is not recommended. Therefore, you should restrict access so that the specified server or client is allowed access by specifying an IP address range on the community definition line. For example, if you want to limit the access within the local host only (the IP address of the local loopback interface is always 127.0.0.1), you can define the community name local as follows (we assume the /etc/snmpd.conf file has only this line):

```
community local 127.0.0.1 255.255.255.255
```

If you do not define any other community, you can send SNMP get requests from the local host only, and snmpd refuses any SNMP set requests.

If you want to restrict the access of a part of the MIB sub-tree, you can define the following two lines in /etc/snmpd.conf:

```
community public 0.0.0.0 0.0.0.0 readOnly 1.17.3
view 1.17.3 1.3.6.1.2.1.1.1
```

These lines instruct the agent to allow access to the object, specified by the view directive, from any IP hosts with the community name of public.

To allow clients from a limited network address range to send SNMP get requests, you can define the following two lines in /etc/snmpd.conf:

```
community neighbor 192.168.0.0 255.255.0.0 readOnly 1.17.4
view 1.17.4 1.3.6.1.2.1
```

This configuration allows clients on the network address 192.168.0.0 to send SNMP get requests to the MIB-2 sub-tree.

5.2.2 Relationship among SNMP related processes

In AIX 5L Version 5.1 and before, snmpd has the following two roles:

- The SNMP agent itself
- A SNMP Multiplex interface (SMUX)[3]server

The SNMP agent is listening on the UDP port 161. The SMUX server is listening on the TCP port 199. A SMUX peer, named dpid2, is implemented as a SMUX-DPI2 converter. The DPI2[4] (Distributed Protocol Interface Version 2.0) interface is used to connect DPI2 sub-agents to the DPI2 server dpid2. The hostmibd daemon is an example of a DPI2 sub-agent.

[3] RFC 1227 defines the SMUX protocol.

[4] RFC 1592 defines the DPI2 protocol.

AIX includes several SNMP related daemon processes, including dpid2 and hostmibd, as shown in the following example:

```
# lslpp -w /usr/sbin/dpid2; lslpp -w /usr/sbin/hostmibd
  File                                        Fileset               Type
  ----------------------------------------------------------------------------
  /usr/sbin/dpid2                             bos.net.tcp.client    File
  File                                        Fileset               Type
  ----------------------------------------------------------------------------
  /usr/sbin/hostmibd                          bos.net.tcp.client    File
```

Figure 5-4 illustrates the relationships among these daemon processes.

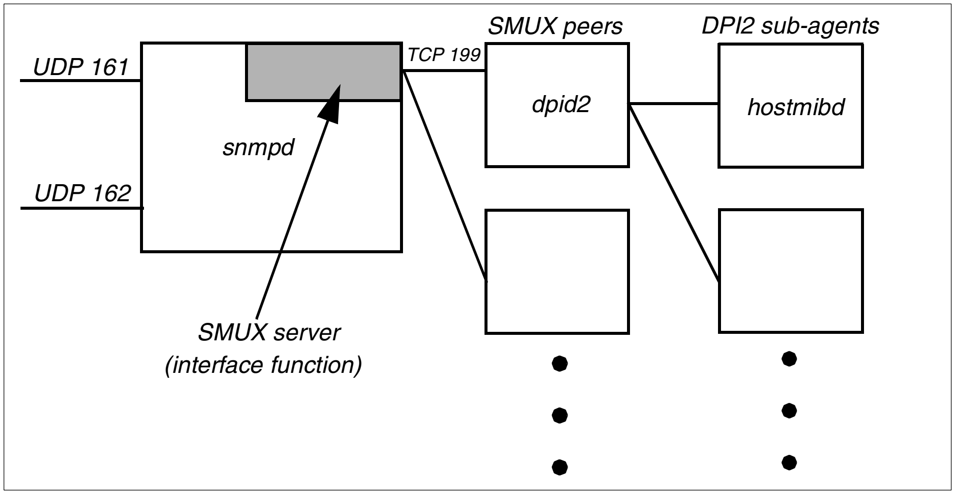


Figure 5-4 Relationship among SNMP related daemon processes

> **Note:** This relationship is subject to change in accordance with future AIX development.

The dpid2 and the hostmibd daemon processes are invoked by /etc/rc.tcpip, after the snmpd daemon process is invoked, as shown in Example 5-1.

Example 5-1 Starting SNMP related daemons in /etc/rc.tcpip

```
# egrep '(snmp|dpi|hostmibd)' /etc/rc.tcpip
start /usr/sbin/snmpd "$src_running"
start /usr/sbin/dpid2 "$src_running"
# Start up the hostmibd daemon
start /usr/sbin/hostmibd "$src_running"
```

The dpid2 and the hostmibd daemon processes are also controlled by SRC, as shown in the following example:

```
# lssrc -a | egrep '(dpid2|hostmibd)'
 dpid2            tcpip            8514    active
 hostmibd         tcpip            8772    active
```

If you stop the AIX snmpd, then you should also stop these SNMP related daemon processes, because these processes rely on the presence of snmpd. If you do not stop these daemons, they may fill up the /var file system by expanding the size of log files. These are located in /var/tmp directory as follows:

```
# ls -l /var/tmp
total 159
-rw-r--r--   1 root     system         9008 Mar 25 13:43 dpid2.log
-rw-r--r--   1 root     system         6014 Mar 20 11:55 hostmibd.log
-rw-r--r--   1 root     system         8030 Mar 25 13:51 muxatmd.log
-rw-r--r--   1 root     system        56144 Mar 25 13:43 snmpd.log
drwxr-xr-x   3 root     system          512 Jan 08 15:15 ttdbserverd/
```

Please note that the AIX snmpd can send SNMP trap requests, but does not initiate the requests. In order to send traps or add your own MIBs, you can implement your own sub-agents using either of the SMUX or the DPI2 interfaces.

- For further information about SMUX, please refer to *AIX 5L Version 5.1 Communications Programming Concepts*. The sample source file is also available in the /usr/samples/snmpd/smux directory.
- For further information about DPI2, please refer to the files in the /usr/samples/snmpd/dpi2 directory.

5.2.3 AIX SNMP manager

AIX includes a simple SNMP manager snmpinfo. The **snmpinfo** command is included in the bos.net.tcp.server fileset, as shown in the following example:

```
# lslpp -w /usr/sbin/snmpinfo
  File                                        Fileset               Type
  ----------------------------------------------------------------------------
  /usr/sbin/snmpinfo                          bos.net.tcp.server    File
```

The **snmpinfo** command can send SNMP get or set requests[5] to SNMP agents. If you specify the -m dump option of the **snmpinfo** command, it sends multiple SNMP get requests recursively, to dump under the specified MIB sub-tree from the target SNMP agent.

[5] The **snmpinfo** command cannot send SNMP trap requests.

The following example shows how to dump[6] the entire MIB tree from the IP host specified as hostname:

```
$ snmpinfo -m dump -c public -h hostname
```

As explained in “Customization example of /etc/snmpd.conf” on page 175, if the target IP host system is running AIX, it should allow access from the AIX system that is invoking the **snmpinfo** command.

If you have enough knowledge about OID of which you really want to get, you may want to use the get subcommand of **snmpinfo**, specifying OID, as follows:

```
$ snmpinfo -m get -c public -h hostname 1.3.6.1.2.1.1.1.0
```

This will show you a system description of the target IP host. If the target system is running AIX 5L Version 5.1, you will see an output resembling the following example:

```
1.3.6.1.2.1.1.1.0 = "IBM PowerPC CHRP Computer
Machine Type: 0x0800004c Processor id: 000681734C00
Base Operating System Runtime AIX version: 05.01.0000.0010
TCP/IP Client Support version: 05.01.0000.0010"
```

5.2.4 How to manage the MIB object definition file

The **snmpinfo** command reads the object definition file (/etc/mib.defs). The **mosy** command reads the ASN.1 (Abstract Syntax Notation One) definitions of SMI (System Management Interface) and MIB modules and produces objects definition files in specific formats to be used by the **snmpinfo** command.

The following example creates an objects definition file to be used by the **snmpinfo** command:

```
# mosy -o /etc/mib.defs \
    /usr/samples/snmpd/smi.my /usr/samples/snmpd/mibII.my
```

Where:

- /usr/samples/snmpd/smi.my

  Defines the ASN.1 definitions by which the SMI is defined, as in RFC 1155.

- /usr/samples/snmpd/mibII.my

  Defines the ASN.1 definitions for the MIB II variables, as defined in RFC 1213.

Please note this command invocation will replace the original /etc/mib.defs file.

[6] This command produces a large number of output lines and might affect the performance of the target system.

5.2.5 Tivoli Netview

We introduce Tivoli NetView Version 7.1 as an SNMP manager software product (Figure 5-5 on page 181[7]). Tivoli NetView discovers TCP/IP networks and network topologies, as well as display those relationships on the graphical user interface. It also manages events and SNMP traps, monitors network health, and gathers performance data.

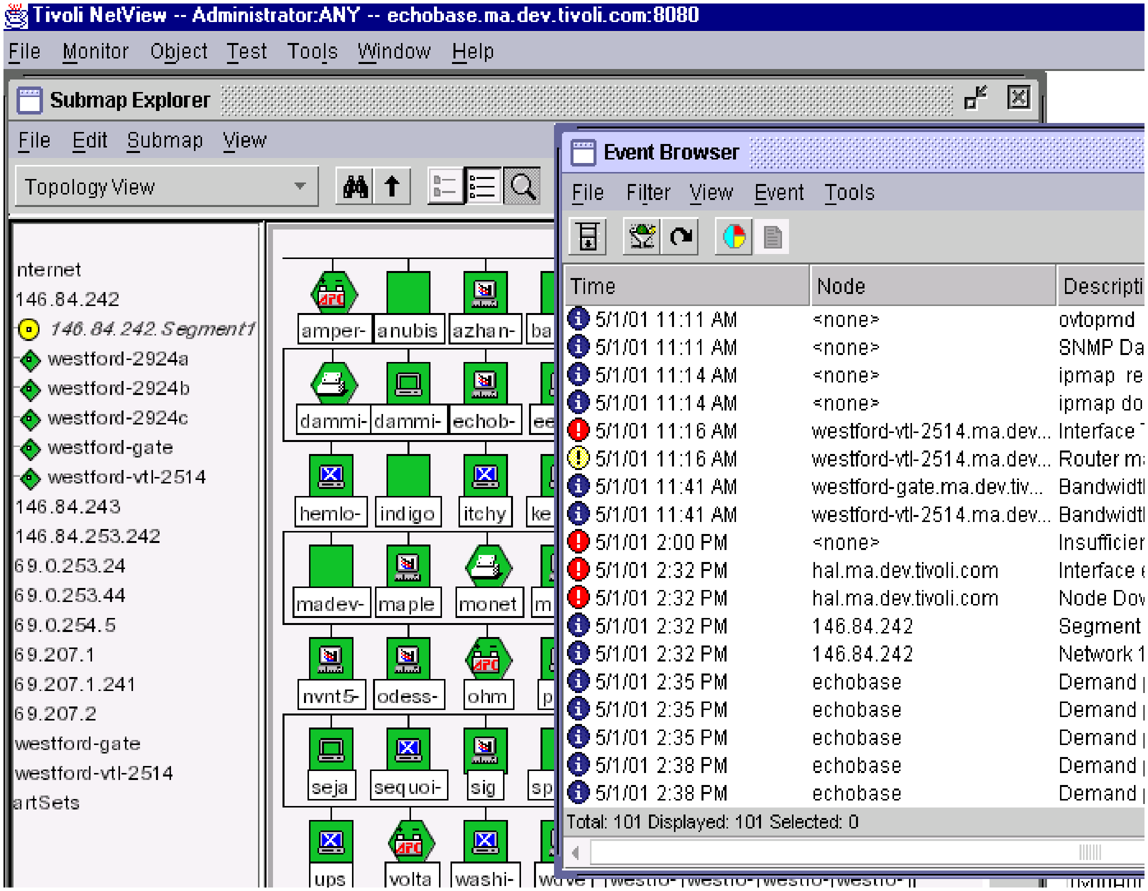
Currently, Tivoli NetView Version 7.1 supports the following platforms:

- AIX Version 4.1.5 or higher
- Solaris 2.5.1, 2.6, or 2.7
- HP-UX A09.07 and higher
- Microsoft Windows NT 4.0 Service Pack 4 or higher

For the product description of Tivoli NetView Version 7.1, please refer to:

http://www.tivoli.com/products/index/netview/

[7] This figure image is taken from Tivoli NetView Version 7.1 running on Microsoft Windows 2000.



nternet
146.84.242
146.84.242.Segment1
westford-2924a
westford-2924b
westford-2924c
westford-gate
westford-vtl-2514
146.84.243
146.84.253.242
69.0.253.24
69.0.253.44
69.0.254.5
69.207.1
69.207.1.241
69.207.2
westford-gate
westford-vtl-2514
artSets

Time	Node	Descripti
5/1/01 11:11 AM	<none>	ovtopmd
5/1/01 11:11 AM	<none>	SNMP Da
5/1/01 11:14 AM	<none>	ipmap re
5/1/01 11:14 AM	<none>	ipmap do
5/1/01 11:16 AM	westford-vtl-2514.ma.dev...	Interface
5/1/01 11:16 AM	westford-vtl-2514.ma.dev...	Router m
5/1/01 11:41 AM	westford-gate.ma.dev.tiv...	Bandwidt
5/1/01 11:41 AM	westford-vtl-2514.ma.dev...	Bandwidt
5/1/01 2:00 PM	<none>	Insufficier
5/1/01 2:32 PM	hal.ma.dev.tivoli.com	Interface
5/1/01 2:32 PM	hal.ma.dev.tivoli.com	Node Dov
5/1/01 2:32 PM	146.84.242	Segment
5/1/01 2:32 PM	146.84.242	Network 1
5/1/01 2:35 PM	echobase	Demand
5/1/01 2:35 PM	echobase	Demand
5/1/01 2:35 PM	echobase	Demand
5/1/01 2:38 PM	echobase	Demand
5/1/01 2:38 PM	echobase	Demand

Total: 101 Displayed: 101 Selected: 0

Figure 5-5 Tivoli NetView Version 7.1

5.2.6 Tivoli NetView trapgend daemon

Tivoli NetView provides you an optional daemon program, named trapgend, on AIX. The trapgend daemon converts an alertable AIX error log entry to an SNMP trap request, then passes the request to the AIX snmpd daemon using a SMUX interface (see Section 5.2.2, “Relationship among SNMP related processes” on page 176). Upon receiving the SNMP trap request, the snmpd daemon should send the trap to the SNMP manager host installed with Tivoli NetView.

Figure 5-6 on page 182 illustrates this process.

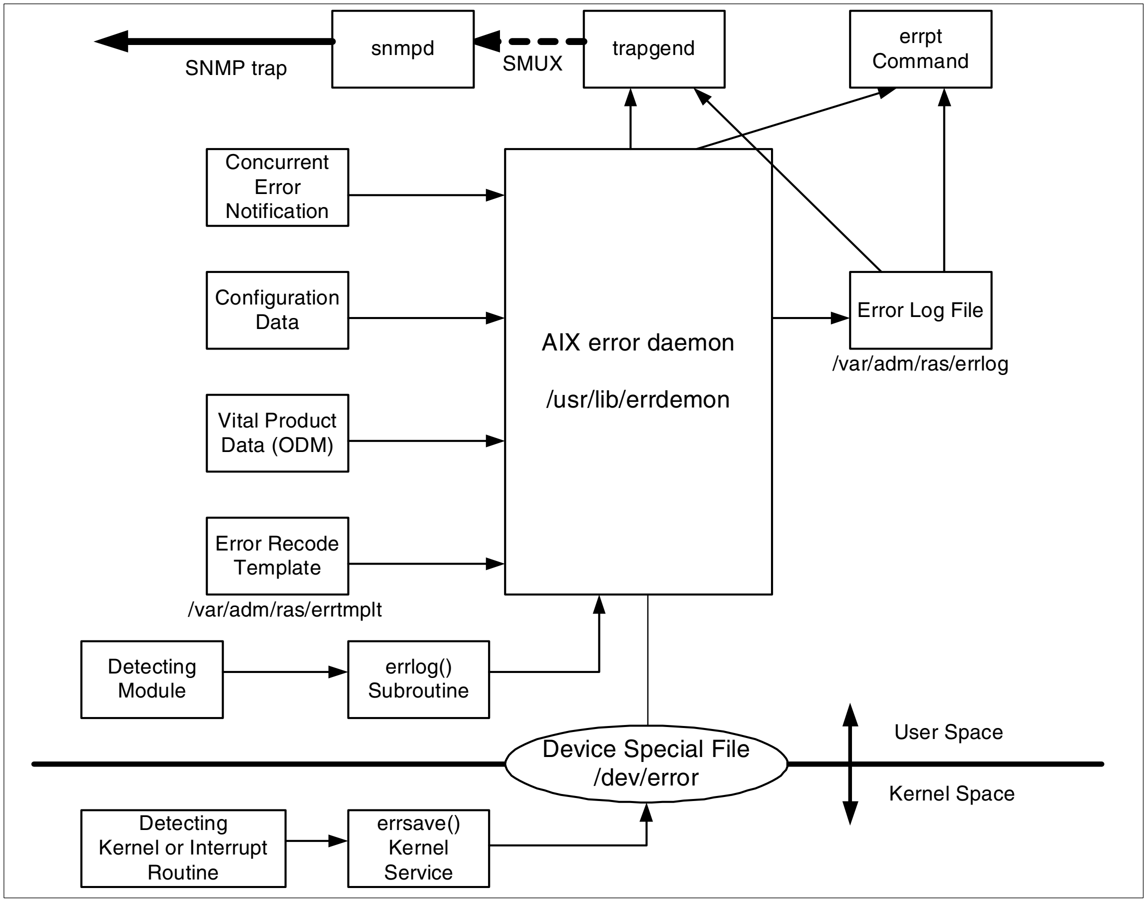


Figure 5-6 Sending an alertable error using trapgend

To use this function on your system, you have to:

- Obtain a Tivoli NetView software license in your enterprise network.
- Install trapged on the AIX systems you want to monitor.
- Customize AIX error log template to update the alertable field.

To customize the AIX error log template, do the following:

1. The AIX error log facility uses the following two files: errlog and errtmplt. The errlog file contains actual error log entries in binary format, and the errtmplt file contains miscellaneous controlling information:

```
# ls -l /var/adm/ras/err*
-rw-r--r--   1 root    system     62673 Mar 31 12:00 /var/adm/ras/errlog
-r--r--r--   1 bin     bin       242076 Jan 08 14:55 /var/adm/ras/errtmplt
```

2. Then, you have to investigate which error log entries should be notified using SNMP traps, because there are so many error log template entries. For example, if you want to send SNMP traps regarding permanent hardware errors on SCSI adapters, the following error log template entries are good candidates (see Example 5-2). Please note that you might see a different output on your system, depending on the installed device driver filesets.

Example 5-2 Selecting AIX error log template entries

```
   # errpt -t | head -1; errpt -t -T PERM -d H | grep -i scsi
   Id       Label               Type CL Description
   0502F666 SCSI_ERR1           PERM H  ADAPTER ERROR
   54E423ED SCSI_ERR9           PERM H  POTENTIAL DATA LOSS CONDITION
   5CC986A0 SCSI_ERR3           PERM H  MICROCODE PROGRAM ERROR
   FBF0BFC1 TMSCSI_UNRECVRD_ERR PERM H  ATTACHED SCSI TARGET DEVICE ERROR
```

In Example 5-2:

-t	Refers to the error log template file (errtmplt)
-T PERM	PERM (Permanent) error type
-d H	H (Hardware) error class

3. For example, if you select the first error log template entry (error ID 0502F666) in Example 5-2, then you can confirm the detailed information, as shown in Example 5-3. The *Alertable* flag is set to NO by default.

Example 5-3 Error log template entry

```
# errpt -atj 0502F666
---------------------------------------------------------------------------
IDENTIFIER 0502F666

Label: SCSI_ERR1
Class:     H
Type:      PERM
Loggable: YES   Reportable: YES   Alertable: NO

Description
ADAPTER ERROR

Probable Causes
ADAPTER HARDWARE
CABLE
CABLE TERMINATOR
DEVICE

Failure Causes
ADAPTER
CABLE LOOSE OR DEFECTIVE
DEVICE
```

```
        Recommended Actions
        PERFORM PROBLEM DETERMINATION PROCEDURES
        CHECK CABLE AND ITS CONNECTIONS

Detail Data
SENSE DATA
```

4. Next, you invoke the **errupdate** command to update the entry. This command waits your input on the standard input. The **errupdate** command is quite strict about its input data. You should:

   – Not insert a white space before and after the equal sign (‘=’).

   – Not insert a white space before and after the colon (‘:’).

   – Insert a tab before the Alert string.

   – Type Control-D (end of line character) twice to end your inputs.

   The following output shows a successful example of the update:

   ```
   # errupdate
   =0502F666:
           Alert=True
   ^D
   0 entries added.
   0 entries deleted.
   1 entries updated.
   ```

   Once your inputs are validated and accepted, it will update the error log template, as shown in Example 5-4.

Example 5-4 Updated error log template entry

```
# errpt -atj 0502F666
---------------------------------------------------------------------------
IDENTIFIER 0502F666

Label: SCSI_ERR1
Class:     H
Type:      PERM
Loggable: YES   Reportable: YES   Alertable: YES

Description
ADAPTER ERROR

Probable Causes
ADAPTER HARDWARE
CABLE
CABLE TERMINATOR
DEVICE
```

```
Failure Causes
ADAPTER
CABLE LOOSE OR DEFECTIVE
DEVICE

        Recommended Actions
        PERFORM PROBLEM DETERMINATION PROCEDURES
        CHECK CABLE AND ITS CONNECTIONS

Detail Data
SENSE DATA
```

If you update the entry successfully, the **errupdate** command will create the errids.undo file on the current directory, as shown in the following example:

```
# ls -l errids.undo
-rw-r--r--   1 root     system           55 Mar 31 18:12 errids.undo
# cat errids.undo
= 0502f666:
        Report = TRUE
        Log = TRUE
        Alert = FALSE
```

You can use this file to reverse the change made in the error log template file.

For further information about how to manage the AIX error log template, please refer to AIX 5L Version 5.1*, General Programming Concepts: Writing and Debugging Programs*.

5.3 NET-SNMP

NET-SNMP is a free software tool that includes the following SNMP components:

- An extensible SNMP agent
- SNMP libraries
- Tools to request or set information from SNMP agents
- Tools to generate and handle SNMP traps

The reasons why we propose to exploit NET-SNMP in this redbook are:

- It is easy to extend functions to monitor AIX systems remotely.
- Currently, it is hard to implement the SNMP trap on AIX 5L Version 5.1, unless you have sub-agent programs, such as trapgend, provided by Tivoli NetView, which can initiate SNMP traps.

For further information about NET-SNMP, visit the following URL:

http://net-snmp.sourceforge.net/

> **Note:** Although NET-SNMP provides you with many useful functions, as explained in this section, you should keep in mind that there is no software program support from IBM.

5.3.1 Installing NET-SNMP

Installing NET-SNMP is an easy task, because we provide you with a standard AIX installation package for NET-SNMP (see Section 6.2, “Creating your own package file” on page 220). To download the installation package, see Appendix D, “Additional material” on page 341.

The following steps explain how to obtain and compile NET-SNMP on AIX 5L Version 5.1.

1. Access the following URL and download the source package on your test machine:

   http://net-snmp.sourceforge.net/

2. Install IBM C compiler Version 5.0 and the GNU **gzip**[8] command.
3. Make sure you have a minimum of 24 MB free space in /usr/local.
4. Un-archive the source package and run the setup script:

   ```
   $ cd /work
   $ gunzip -d -c ucd-snmp-4.2.3.tar.gz | tar -xf -
   $ cd ucd-snmp-4.2.3
   $ CC=cc ./configure --without-openssl
   ```

   In this example, we assume that you downloaded ucd-snmp-4.2.3.tar.gz into /work.

5. Add the following line to ./s/aix.h:

   ```
   #define aix4 1
   ```

6. Add the following line at the bottom of ./config.h:

   ```
   #undef HAVE_GETMNTENT
   ```

7. Build NET-SNMP by issueing the **make** command:

   ```
   $ make
   ```

   Now NET-SNMP is ready to be installed using the install target of the **make** command:

   ```
   # make install
   ```

[8] The GNU **gzip** command is provided by the AIX toolbox for Linux applications.

Note: NET-SNMP addresses the CERT Advisory CA-2002-03: Multiple Vulnerabilities in Many Implementations of the Simple Network Management Protocol (SNMP) in Version 4.2.3.

5.3.2 The NET-SNMP agent

NET-SNMP includes an SNMP agent, snmpd, which is installed as /usr/local/bin/snmpd. The NET-SNMP agent has the various capabilities as follows:

- Access control
- Process monitoring
- File system monitoring
- CPU load monitoring
- Log file monitoring
- User defined executives
- Pass directives

In order to exploit these functions, you have to understand how to alter the configuration file. Section 5.3.3, “Configuring the NET-SNMP agent” on page 189 shows many examples of how to use these functions.

Starting the NET-SNMP agent

The NET-SNMP agent also uses UDP port 161 by default; therefore, you should stop the following daemon processes provided by AIX before starting the NET-SNMP agent:

```
# lssrc -a | egrep '(snmpd|dpid2|hostmibd)'
 snmpd            tcpip            8516    active
 dpid2            tcpip            8772    active
 hostmibd         tcpip            9030    active
```

To stop these daemons, see Section 5.2.1, “AIX SNMP agent” on page 175 and Section 5.2.2, “Relationship among SNMP related processes” on page 176. After stopping the daemons, you should also comment out all the lines shown in Example 5-1 on page 177.

Starting the NET-SNMP agent is shown in Example 5-5.

Example 5-5 Starting NET-SNMP

```
# /usr/local/sbin/snmpd
# ps -ef | grep snmp
    root 15396 17132   1   Dec 31      -  0:00 grep snmp
    root 20114     1   1 19:31:25  pts/0  0:01 /usr/local/sbin/snmpd
```

You should also add the following lines to /etc/rc.tcpip so that NET-SNMP is invoked upon every system restart:

```
/usr/local/sbin/snmpd
ps -ef | grep local | grep -q snmpd
if [ $? -eq 0 ]; then
    echo “NET-SNMP snmpd is running.”
else
    echo “NET-SNMP snmpd is failed to start.”
fi
```

You can see the short description of its command line option by specifying the -h option on the command line, as shown in the following example:

```
$ /usr/local/sbin/snmpd -h

Usage:  /usr/local/sbin/snmpd [-h] [-v] [-f] [-a] [-d] [-V] [-P PIDFILE] [-q]
[-D] [-p NUM] [-L] [-l LOGFILE] [-r] [-u uid] [-g gid]

        Version:  4.2.3
        Author:   Wes Hardaker
        Email:    net-snmp-coders@lists.sourceforge.net

-h              This usage message.
-H              Display configuration file directives understood.
-v              Version information.
-f              Don't fork from the shell.
-a              Log addresses.
-d              Dump sent and received UDP SNMP packets
-V              Verbose display
-P PIDFILE      Use PIDFILE to store process id
-q              Print information in a more parsable format (quick-print)
-D              Turn on debugging output
-p NUM          Run on port NUM instead of the default: 161
-x SOCKADDR     Bind AgentX port to this address
-X              Run as an AGENTX subagent rather than an SNMP master agent.
-c CONFFILE     Read CONFFILE as a configuration file.
-C              Don't read the default configuration files.
-L              Print warnings/messages to stdout/err
-s              Log warnings/messages to syslog
-A              Append to the logfile rather than truncating it.
-r              Don't exit if root only accessible files can't be opened
-I [-]INITLIST  List of mib modules to initialize (or not).
                 (run snmpd with -Dinit_mib for a list)
-l LOGFILE      Print warnings/messages to LOGFILE
                (By default LOGFILE=/usr/adm/snmpd.log)
-g              Change to this gid after opening port
```

```
-u              Change to this uid after opening port
```

In order to refer to the online command reference for the NET-SNMP agent, do the following:

```
$ MANPATH=/usr/local/man:$MANPATH; export MANPATH
$ man snmpd
```

5.3.3 Configuring the NET-SNMP agent

The NET-SNMP agent has wide range of capabilities that can be configured using the configuration file installed as /usr/local/share/snmp/snmpd.conf. In order to configure these capabilities, you have to understand how to specify the many *directives* of the snmpd.conf file.

Although the syntax of this configuration file is very different from the one used by AIX standard snmpd, there is a sample configuration file, called EXAMPLE.conf, included in the source archive (see “EXAMPLE.conf” on page 302). Because the EXAMPLE.conf file contains rich comments to explain the usage of directives, it is a good starting point to customize the snmpd.conf file.

For further information about the snmpd.conf file and its directive usage, please consult with the online command reference:

```
$ MANPATH=/usr/local/man:$MANPATH; export MANPATH
$ man snmpd.conf
```

Access control

An access control is a function that controls access to the host running the NET-SNMP agent. An access control is defined by the following four directives:

- com2sec
- group
- view
- access

Here we discuss each of these directives in detail:

1. The com2sec directive

   You have to define the security name using the com2sec directive first. The defined security name is an identification for a combination of a client’s IP address and an SNMP community name.

   The following example defines the security name “rwuser” that only allows access from localhost with the community name “private”:

   ```
   com2sec rwuser 127.0.0.1 private
   ```

The following example defines the security name “rouser” that only allows access from localhost with the community name “public”:

```
com2sec rouser 127.0.0.1 public
```

The following example defines the security name “rrouser” that allows access from the clients of IP address range 192.168.1.0 to 255 with community name “remote”:

```
com2sec rrouser 192.168.1/24 remote
```

2. The group directive

   Next, you have to bind a security group with the security name you defined using the group directive.

   If you apply any of SNMP Version 1, Version 2c, and Version 3 security models to the defined group, then you have to specify a security flag, v1, v2c, and usm, respectively.

   The following example shows you how to bind the security group rwgroup, rogroup, and rrogroup with the security name rwuser, rouser, and rrouser, defined in the previous step:

```
group rwgroup v2c rwuser
group rwgroup usm rwuser
group rogroup v1 rouser
group rogroup v2c rouser
group rogroup usm rouser
group rrogroup v1 rrouser
group rrogroup v2c rrouser
group rrogroup usm rrouser
```

   Although we only use SNMP Version 1 security model (v1) throughout this redbook, we define all the available security models here.

3. The view directive

   Then, you have to define a view name using the view directive (see Example 5-6). The first line of Example 5-6 defines the view wide that allows access to all of the MIB tree. The second line defines the view “narrow” that allows access to the MIB sub-tree under the OID .1.3.6.1.4.1.2021 (iso.org.dod.internet.private.enterprises.ucdavis).

Example 5-6 View directive example

```
view wide included .1 80
view narrow included .1.3.6.1.4.1.2021 fe
```

4. The access directive

   Finally you have to bind these directive lines you defined using the access directive. An access directive line is composed of nine fields separated by white spaces.

a. The first field of the access directive line is always the keyword access.

b. The second field is a security group name.

c. The third field is optional[9].

d. The fourth field defines an applicable security model name. If you specify the keyword *any* in this field, then any security model is allowed to be used.

e. The fifth field is a security level, which we define “noauth” to simply authenticate clients with community “name”.

f. The sixth field is how context, the second argument, should match with context in the incoming request.

g. The following three fields (seventh, eighth, and ninth fields) are the view names that are permitted to read, write, and notify, respectively. The two special keywords “all” and “none” specify to allow or not allow all the views, respectively.

The first line in Example 5-7 specifies that the rwgroup security group can read and write the whole MIB tree. The second line specifies that the rogroup security group can read the whole MIB tree. The third specifies that the rrogroup security users can read the limited MIB sub-tree defined by the view name narrow.

Example 5-7 Access directive example

```
access rwgroup ““ any noauth exact all all none
access rogroup ““ any noauth exact all none none
access rrogroup ““ any noauth exact narrow none none
```

If you want to control the access with the simple security policy based on IP address ranges, then you can use the rwcommunity and rocommunity directives, as shown in Example 5-8.

Example 5-8 Simple access control example

```
rwcommunity private 127.0.0.1 .1
rocommunity public 127.0.0.1 .1
rocommunity remote 192.168.1.0/24 .1.3.6.1.4.1.2021
```

The first line in Example 5-8 allows read and write access to the whole MIB tree from localhost with community name private. The second line allows read only access to the whole MIB tree from localhost with community name public. The last line allows read only access to the limited MIB sub-tree (under the OID .1.3.6.1.4.1.2021) from any IP host on the 192.168.1.0/24 sub-network with community name remote.

[9] We do not use this optional field throughout this redbook.

Process monitoring

You can monitor the specific process status using the proc directive (see Example 5-9). The first line in Example 5-9 monitors the number of httpd processes. The two numbers 100 and 1 specify the maximum and minimum number of the httpd process, respectively. The second line monitors to see if the sshd process is running. Because this line does not specify the maximum and minimum process numbers, it does not monitor the number of sshd process.

Example 5-9 Proc directive example

```
proc httpd 100 1
proc sshd
```

In order to confirm how the proc directive is working, you can use the **snmpwalk** command (see Section 5.3.6, “NET-SNMP clients and tools” on page 202). Example 5-10 shows the output produced by the **snmpwalk** command using the proc directive lines shown in Example 5-9. Because we stop both the httpd and sshd processes in this example, you can see the process counts of 0 and generated error messages.

Example 5-10 Process monitoring using NET-SNMP

```
# snmpwalk localhost public prTable
enterprises.ucdavis.prTable.prEntry.prIndex.1 = 1
enterprises.ucdavis.prTable.prEntry.prIndex.2 = 2
enterprises.ucdavis.prTable.prEntry.prNames.1 = httpd
enterprises.ucdavis.prTable.prEntry.prNames.2 = sshd
enterprises.ucdavis.prTable.prEntry.prMin.1 = 1
enterprises.ucdavis.prTable.prEntry.prMin.2 = 0
enterprises.ucdavis.prTable.prEntry.prMax.1 = 100
enterprises.ucdavis.prTable.prEntry.prMax.2 = 0
enterprises.ucdavis.prTable.prEntry.prCount.1 = 0
enterprises.ucdavis.prTable.prEntry.prCount.2 = 0
enterprises.ucdavis.prTable.prEntry.prErrorFlag.1 = 1
enterprises.ucdavis.prTable.prEntry.prErrorFlag.2 = 1
enterprises.ucdavis.prTable.prEntry.prErrMessage.1 = Too few httpd running (# = 0)
enterprises.ucdavis.prTable.prEntry.prErrMessage.2 = No sshd process running.
enterprises.ucdavis.prTable.prEntry.prErrFix.1 = 0
enterprises.ucdavis.prTable.prEntry.prErrFix.2 = 0
enterprises.ucdavis.prTable.prEntry.prErrFixCmd.1 =
enterprises.ucdavis.prTable.prEntry.prErrFixCmd.2 =
```

File system monitoring

You can monitor a specific file system's status by using the disk directive (see Example 5-11). The first line instructs snmpd to monitor the /tmp file system to ensure it has at least 16 MB of free space. The following lines specify free space size by percentage. If you skip free space size, the default size of 100 MB is used.

Example 5-11 Disk directive example

```
disk /tmp       16000
disk /home      20%
disk /          15%
disk /usr       5%
disk /usr/local 10%
disk /var       20%
disk /home      20%
disk /opt       10%
disk /stats     10%
disk /swdist    10%
disk /logs      20%
disk /www       20%
disk /backup    10%
disk /usr/HTTPServer    20%
```

Example 5-12 shows the output produced by the **snmpwalk** command using the disk directive lines shown in Example 5-11. You can see the /usr file system usage of 97 percent in this example.

Example 5-12 Disk monitoring using NET-SNMP

```
# snmpwalk svr02 remote dskTable
… many lines are removed in here …
enterprises.ucdavis.dskTable.dskEntry.dskErrorMsg.1 =
enterprises.ucdavis.dskTable.dskEntry.dskErrorMsg.2 =
enterprises.ucdavis.dskTable.dskEntry.dskErrorMsg.3 =
enterprises.ucdavis.dskTable.dskEntry.dskErrorMsg.4 = /usr: less than 5% free (= 97%)
enterprises.ucdavis.dskTable.dskEntry.dskErrorMsg.5 =
enterprises.ucdavis.dskTable.dskEntry.dskErrorMsg.6 =
… many lines are removed in here …
```

CPU load monitoring

You can monitor CPU load using the load directive, as shown in the following example:

```
load 4
```

This example specifies monitoring when the average CPU load factor is less than 4 for the last minute. You can also specify a threshold value for the last five minutes and 15 minutes.

Example 5-13 shows the output produced by the **snmpwalk** command using the previous load directive line.

Example 5-13 CPU load factor monitoring using NET-SNMP

```
# snmpwalk localhost public laTable
enterprises.ucdavis.laTable.laEntry.laIndex.1 = 1
enterprises.ucdavis.laTable.laEntry.laIndex.2 = 2
enterprises.ucdavis.laTable.laEntry.laIndex.3 = 3
enterprises.ucdavis.laTable.laEntry.laNames.1 = Load-1
enterprises.ucdavis.laTable.laEntry.laNames.2 = Load-5
enterprises.ucdavis.laTable.laEntry.laNames.3 = Load-15
enterprises.ucdavis.laTable.laEntry.laLoad.1 = 0.00
enterprises.ucdavis.laTable.laEntry.laLoad.2 = 0.02
enterprises.ucdavis.laTable.laEntry.laLoad.3 = 0.02
enterprises.ucdavis.laTable.laEntry.laConfig.1 = 4.00
enterprises.ucdavis.laTable.laEntry.laConfig.2 = 4.00
enterprises.ucdavis.laTable.laEntry.laConfig.3 = 4.00
enterprises.ucdavis.laTable.laEntry.laLoadInt.1 = 0
enterprises.ucdavis.laTable.laEntry.laLoadInt.2 = 1
enterprises.ucdavis.laTable.laEntry.laLoadInt.3 = 1
enterprises.ucdavis.laTable.laEntry.laLoadFloat.1 = Opaque: Float: 0.003540
enterprises.ucdavis.laTable.laEntry.laLoadFloat.2 = Opaque: Float: 0.018158
enterprises.ucdavis.laTable.laEntry.laLoadFloat.3 = Opaque: Float: 0.015137
enterprises.ucdavis.laTable.laEntry.laErrorFlag.1 = 0
enterprises.ucdavis.laTable.laEntry.laErrorFlag.2 = 0
enterprises.ucdavis.laTable.laEntry.laErrorFlag.3 = 0
enterprises.ucdavis.laTable.laEntry.laErrMessage.1 =
enterprises.ucdavis.laTable.laEntry.laErrMessage.2 =
enterprises.ucdavis.laTable.laEntry.laErrMessage.3 =
```

Log file monitoring

You can monitor the size of a specified file using the file directive shown in the following example:

```
file /var/log/snmpd.log 100000
```

Example 5-14 shows that the size of /var/log/snmpd.log is less than 100 MB.

Example 5-14 File size monitoring using NET-SNMP

```
# snmpwalk localhost public fileTable
enterprises.ucdavis.fileTable.fileEntry.fileIndex.1 = 1
enterprises.ucdavis.fileTable.fileEntry.fileName.1 = /usr/adm/snmpd.log
enterprises.ucdavis.fileTable.fileEntry.fileSize.1 = 0 kB
enterprises.ucdavis.fileTable.fileEntry.fileMax.1 = 100000 kB
enterprises.ucdavis.fileTable.fileEntry.fileErrorFlag.1 = 0
enterprises.ucdavis.fileTable.fileEntry.fileErrorMsg.1 =
```

User defined executives

One useful directive is exec, which extends the NET-SNMP agent by executing an external program and runs generated output into a specific MIB sub-tree. The following example specifies that an SNMP GET request to the OID .1.3.6.1.4.1.2021.55 causes the execution of the **`/usr/sbin/lsps -a`** command and imports a generated standard output of this command into the MIB sub-tree .1.3.6.1.4.1.2021.55:

```
exec .1.3.6.1.4.1.2021.55 pagingspace /usr/sbin/lsps -a
```

Example 5-15 shows the output produced by the **`snmpwalk`** command using this exec directive line. You can see the output produced by **`/usr/sbin/lsps -a`** is imported into three OIDs. The exit code 0 of the command is stored in the OID on the last line.

Example 5-15 Example output 1 from the exec directive

```
# snmpwalk svr02 remote ucdavis.55
enterprises.ucdavis.55.1.1 = 1
enterprises.ucdavis.55.2.1 = "pagingspace"
enterprises.ucdavis.55.3.1 = "/usr/sbin/lsps -a"
enterprises.ucdavis.55.100.1 = 0
enterprises.ucdavis.55.101.1 = "Page Space  Physical Volume   Volume Group    Size  %Used
Active  Auto  Type"
enterprises.ucdavis.55.101.2 = "paging00    hdisk0           rootvg         224MB      1
yes   yes    lv"
enterprises.ucdavis.55.101.3 = "hd6         hdisk1           rootvg         288MB      2
yes   yes    lv"
enterprises.ucdavis.55.102.1 = 0
```

Although Example 5-15 show how easy it is to understand how the exec directive works, the generated output is hard to use. In order to parse the output of **`/usr/sbin/lsps -a`**, you can write a simple Perl script, as shown in Example 5-16.

Example 5-16 Helper script pscheck

```
#!/usr/bin/perl
#
# pscheck: Helper script to monitor paging space status.
# Install this script as /usr/local/share/snmp/pschk
#
open(TT, "lsps -a |");
while (<TT>) {
    next if (/^Page Space/);  # skip header line
    @ps = split(" ", $_);     # parse fields using a white space delimiter.
    printf("%s\n", $ps[0]);   # print paging space name
    printf("%s\n", $ps[1]);   # print physical volume name
    printf("%s\n", $ps[2]);   # print volume group name
    printf("%s\n", $ps[3]);   # print size
    printf("%d\n", $ps[4]);   # print utilization
    # if active print value 1, otherwise print 0
    printf("%d\n", ($ps[5] == "yes" ? 1 : 0));
}
close(TT);
exit (0);
```

The following example specifies that an SNMP GET request to the OID .1.3.6.1.4.1.2021.56 causes the execution of the Perl script shown in Example 5-16 on page 195:

```
exec .1.3.6.1.4.1.2021.56 fmpgspace /usr/local/share/snmp/pschk
```

Example 5-17 shows the output produced by the **snmpwalk** command using this exec directive line.

Example 5-17 Example output 2 from the exec directive

```
# snmpwalk svr02 remote ucdavis.56
enterprises.ucdavis.56.1.1 = 1
enterprises.ucdavis.56.2.1 = "pagingspace"
enterprises.ucdavis.56.3.1 = "/usr/local/share/snmp/pschk"
enterprises.ucdavis.56.100.1 = 0
enterprises.ucdavis.56.101.1 = "paging00"
enterprises.ucdavis.56.101.2 = "hdisk0"
enterprises.ucdavis.56.101.3 = "rootvg"
enterprises.ucdavis.56.101.4 = "224MB"
enterprises.ucdavis.56.101.5 = "1"
enterprises.ucdavis.56.101.6 = "1"
enterprises.ucdavis.56.101.7 = "hd6"
enterprises.ucdavis.56.101.8 = "hdisk1"
enterprises.ucdavis.56.101.9 = "rootvg"
enterprises.ucdavis.56.101.10 = "288MB"
enterprises.ucdavis.56.101.11 = "2"
enterprises.ucdavis.56.101.12 = "1"
enterprises.ucdavis.56.102.1 = 0
```

Another application example of the exec directive is shown in Example 5-18 on page 197. This example illustrates the software distribution process explained in Section 7.2.1, “Distributing software” on page 252.

In this example, we attempt to install some software onto svr02 from svr06. On our first attempt, we find no software is present to be installed, so we copy the package file libpcap-0.6.1.0.bff using **scp** into the /swdist directory on svr02. We then re-issue the **snmpwalk** command from the svr06 to initiate the software installation process.

Example 5-18 Using SNMP user executive to install software

```
root@svr06:/ [499] # ssh svr02 lslpp -l | grep -i libpcap
root@svr06:/ [500] # snmpwalk svr02 remote .1.3.6.1.4.1.2021.57
enterprises.ucdavis.57.1.1 = 1
enterprises.ucdavis.57.2.1 = "swinstall"
enterprises.ucdavis.57.3.1 = "/usr/local/share/snmp/swinstall"
enterprises.ucdavis.57.100.1 = 1
enterprises.ucdavis.57.101.1 = "No software to install"
enterprises.ucdavis.57.102.1 = 0
root@svr06:/ [501] # scp /swdist/libpcap-0.6.1.0.bff svr02:/swdist
libpcap-0.6.1.0.b    100% |******************************|   256 KB    00:00
root@svr06:/ [502] # snmpwalk svr02 remote .1.3.6.1.4.1.2021.57
enterprises.ucdavis.57.1.1 = 1
enterprises.ucdavis.57.2.1 = "swinstall"
enterprises.ucdavis.57.3.1 = "/usr/local/share/snmp/swinstall"
enterprises.ucdavis.57.100.1 = 0
enterprises.ucdavis.57.101.1 = "Software successfully installed"
enterprises.ucdavis.57.102.1 = 0
root@svr06:/ [503] # ssh svr02 lslpp -l | grep -i libpcap
  freeware.libpcap.rte       0.6.1.0  COMMITTED  User-level Packet Capture
  freeware.libpcap.rte       0.6.1.0  COMMITTED  User-level Packet Capture
```

In order to implement this software distribution mechanism, you have to specify the following exec directive line and use the /usr/local/share/snmp/swinstall script shown in Example 5-19:

```
exec .1.3.6.1.4.1.2021.57 swinstall /usr/local/share/snmp/swinstall
```

Example 5-19 swinstall script

```
#!/bin/ksh
#
# swinstall: Monitor fileset installation.
# Install this script as /usr/local/share/snmp/swinstall
LOGFILE=/var/adm/swinstall.log

cd /swdist
inutoc . >> $LOGFILE 2>&1
if [ -f .toc ]; then
    installp -ld . | installp -aXd . -f - >> $LOGFILE 2>&1
    rc=$?
else
    echo "No software to install"
    exit 1
fi

if [ $rc -eq 0 ]; then
    echo "Software successfully installed"
    rm -f /swdist/* > /dev/null 2>&1
    rm -f /swdist/.toc
else
    echo "Unable to install software. Please investigate"
    exit 2
fi
exit 0
```

> **Important:** Software distribution could be easily automated to update every server within the farm. However, use with caution; if the software being installed is bad, you could crash every server in your farm!

Pass directives

The pass directive passes the entire control of the specified MIB OID to an external program. The following example instructs the agent that any SNMP GET request to the OID .1.3.6.1.4.1.2021.58 should cause the execution of the Perl script /usr/local/share/snmp/pschk2, shown in Appendix B, “Example scripts” on page 313:

```
pass .1.3.6.1.4.1.2021.58 /usr/local/share/snmp/pschk2
```

The snmpd calls the program specified in the pass directive line with two arguments: an operation and an OID. There are three types for operation: -s for SET, -g for GET, and -n for GETNEXT.

If the program accepts SET operations for a certain OID, the program should exit with return code 0 and not print anything to the standard out. If the program does not accept SET operations, the program should print a line, `not-writable` or `wrong-type`.

If the program accepts GET or GET NEXT operations, the program should print three lines. Each line has to contain an OID, a type, and a value. If the program does not accept GET or GET NEXT requests, it should exit with exit code 0 without printing anything to the standard out.

Example 5-20 shows the output produced by the **`snmpwalk`** command using this pass directive line.

Example 5-20 Example of pass directive

```
# snmpwalk localhost public .1.3.6.1.4.1.2021.58
enterprises.ucdavis.58.1 = "Paging space"
enterprises.ucdavis.58.1.1 = "paging00"
enterprises.ucdavis.58.1.2 = "hdisk0"
enterprises.ucdavis.58.1.3 = "rootvg"
enterprises.ucdavis.58.1.4 = "224MB"
enterprises.ucdavis.58.1.5 = 1
enterprises.ucdavis.58.1.6 = 1
enterprises.ucdavis.58.2.1 = "hd6"
enterprises.ucdavis.58.2.2 = "hdisk1"
enterprises.ucdavis.58.2.3 = "rootvg"
enterprises.ucdavis.56.2.4 = "288MB"
enterprises.ucdavis.56.2.5 = 2
enterprises.ucdavis.56.2.6 = 1
```

The pass directive invokes the specified external program upon each request. If you are concerned about the performance overhead of the pass directive, you can use the pass_parsist directive, which keeps the external program running.[10]

5.3.4 User defined traps

NET-SNMP provides you a command (**snmptrap**) that can send SNMP trap requests. As explained in Section 5.1, “Basic understanding of SNMP” on page 170, the SNMP trap is a powerful method to asynchronously notify the remote administrator of troubles in your system.

Because the **snmptrap** command requires many arguments on the command line to send SNMP traps, we provide you a short wrapper script, called sendtrap, as shown in Example 5-21.

Example 5-21 A wrapper script to send SNMP traps

```
#!/bin/ksh
#
# sendtrap: Sends an SNMP trap as specified by the command line arguments.
#
# Parameter: 1 should be the priority 1 - FATAL Error to 6 - information only
# Parameter: 2 - your message
#
# Syntax:
# sendtrap 1 Application deadbeef failure

# The MANAGER variable should match the host name of your SNMP manager.
MANAGER=svr06
LOG=/var/adm/sendtrap.log

function usage
```

[10] The NET-SNMP source archive includes an example Perl script (pass_parsisttest) to be used with the pass_parsist directive.

```
{
    print "usage: sendtrap <num> <message>"
    exit 2
}

# Exit if user incorrect
[[ $# -ge 2 ]] || usage

# Work out command line parameters.
trapnum=$1
shift 1
msg=$*

# Send an SNMP trap to manager.

/usr/local/bin/snmptrap -v 1 $MANAGER public '' $MANAGER 6 $trapnum '' \
    .1.3.6.1.4.1.2021.57.1 s "$msg"

# Log the message locally so we can check it
echo "`date`: $trapnum : $msg" >> $LOG

exit 0
```

This shell script sends a message on your behalf to the manager. To send an SNMP trap message, you can issue this script shown in the following example:

```
root@svr02:/ [459] # /usr/local/bin/sendtrap 1 Application deadbeef has died
```

In this example, the notification message `Application deadbeef has died` is sent to svr06 from svr02 as an SNMP trap.

You can see this message on svr06, as shown in Example 5-22.

Example 5-22 Using sendtrap and the output

```
root@svr06:/ [242] # 2002-04-01 11:40:52 svr06.itsc.austin.ibm.com
[XXX.XXX.XXX.XXX]
 (via svr02.itsc.austin.ibm.com [9.3.187.231]) TRAP, SNMP v1, community public
        enterprises.3.1.1 Enterprise Specific Trap (1) Uptime: 0:00:00.00
        enterprises.ucdavis.57.1 = "Application deadbeef has died"
```

The value 1 in parenthesis after `Enterprise Specific Trap` is the priority of the alert. You can use 1 for a fatal error and scale it down to 6 for information only.

> **Note:** To receive SNMP traps, you have to run an SNMP manager on svr06. This example assumes that you configure a simple SNMP manager provided by SNMP, as explained in Section 5.3.5, “Configuring a simple SNMP manager” on page 202.

Trapping permanent hardware errors

If a permanent hardware error should be logged on a remote AIX system, an SNMP trap message can be sent to the SNMP manager to remotely notify the administrator of the error.

In order to asynchronously send an SNMP trap message, you have to modify several ODM classes. We provide you a script, named hwalert, to quickly do this modification (see Appendix B, “Example scripts” on page 313).

This script is called upon system startup, enabling SNMP traps to be sent when permanent hardware errors are logged. You can also disable this service, for example, if you are going to perform some hardware maintenance tasks.

In Example 5-23, the hwalert script is enabled and the alert is sent to svr06 from svr02 when a permanent hardware failure is generated. In this example, we pull out the Token Ring cable from svr02 to generate a permanent network failure.

Example 5-23 Hardware alert logged by SNMP

```
root@svr02:/ [524] # /usr/local/bin/hwalert ENABLE
root@svr02:/ [525] #
… The Token Ring cable is pulled out …
root@svr02:/ [526] # errpt
IDENTIFIER TIMESTAMP  T C RESOURCE_NAME  DESCRIPTION
FFE305EE   0401142902 P H tok0           WIRE FAULT
FFE305EE   0401142902 P H tok0           WIRE FAULT
… Many lines are skipped …
root@svr06:/ [300] # tail /logs/trapd.log
2002-04-01 14:14:06 svr06e.itsc.austin.ibm.com [XXX.XXX.XXX.YYY] (via
svr02e.itsc.austin.ibm.com [XXX.XXX.XXX.XXX]) TRAP, SNMP v1, community public
        enterprises.3.1.1 Enterprise Specific Trap (2) Uptime: 0:00:00.00
        enterprises.ucdavis.57.1 = "System error CSTOK_WIRE_FAULT on tok0 -
please investigate errpt for details"
2002-04-01 14:14:13 svr06e.itsc.austin.ibm.com [XXX.XXX.XXX.YYY] (via
svr02e.itsc.austin.ibm.com [XXX.XXX.XXX.XXX]) TRAP, SNMP v1, community public
        enterprises.3.1.1 Enterprise Specific Trap (2) Uptime: 0:00:00.00
        enterprises.ucdavis.57.1 = "System error CSTOK_WIRE_FAULT on tok0 -
please investigate errpt for details"
```

5.3.5 Configuring a simple SNMP manager

NET-SNMP provides you a simple SNMP manager that can receive SNMP traps. We use this manager, snmptrapd, to illustrate the SNMP trap mechanism.

To receive SNMP traps, you have to invoke snmptrapd, as shown in Example 5-24.

Example 5-24 Using snmptrapd on the administration console

```
# echo "/usr/local/sbin/snmptrapd -o /logs/trapd.log" |\
    at now
Job root.1017688566.a will be run at Mon Apr  1 13:16:06 CST 2002.
# ps -ef | grep snmp
    root  5074 11266   0 13:16:11  pts/2  0:00 grep snmp
    root 23040     1   0 13:16:06      -  0:00 /us/local/sbin/snmptrapd -o
/logs/trapd.log
```

The flag -o specifies the log file that stores all the SNMP traps. The **snmptrapd** command should be invoked using the **at** command to prevent it from grabbing the controlling terminal.

You can run the command in the foreground and use the -P flag, for example, **snmptrapd -P -o /logs/trapd.log**. This means all traps are sent to the standard error of the controlling terminal, as well as to the specified log file. This mode is useful for debugging.

5.3.6 NET-SNMP clients and tools

NET-SNMP provides you the SNMP client and tool commands installed in the /usr/local/bin directory (Example 5-25).

Example 5-25 SNMP clients provided by NET-SNMP

```
# pwd
/usr/local/bin
# ls -l
total 31696
-rwxr-xr-x   1 root     sys         933297 Apr 29 20:47 encode_keychange
-rwxr-xr-x   1 root     sys           9668 Apr 29 20:47 mib2c
-rwxr-xr-x   1 root     sys         939389 Apr 29 20:47 snmpbulkget
-rwxr-xr-x   1 root     sys         938073 Apr 29 20:47 snmpbulkwalk
-rwxr-xr-x   1 root     sys          33886 Apr 29 20:47 snmpcheck
-rwxr-xr-x   1 root     sys          19063 Apr 29 20:47 snmpconf
-rwxr-xr-x   1 root     sys         954917 Apr 29 20:47 snmpdelta
-rwxr-xr-x   1 root     sys         942937 Apr 29 20:47 snmpdf
-rwxr-xr-x   1 root     sys         938533 Apr 29 20:47 snmpget
-rwxr-xr-x   1 root     sys         938433 Apr 29 20:47 snmpgetnext
lrwxrwxrwx   1 root     system           8 Apr 30 13:43 snmpinform -> snmptrap
```

```
-rwxr-xr-x   1 root     sys        1002543 Apr 29 20:47 snmpnetstat
-rwxr-xr-x   1 root     sys         939227 Apr 29 20:47 snmpset
-rwxr-xr-x   1 root     sys         945932 Apr 29 20:47 snmpstatus
-rwxr-xr-x   1 root     sys         955433 Apr 29 20:47 snmptable
-rwxr-xr-x   1 root     sys         945321 Apr 29 20:47 snmptest
-rwxr-xr-x   1 root     sys         922235 Apr 29 20:47 snmptranslate
-rwxr-xr-x   1 root     sys         944945 Apr 29 20:47 snmptrap
-rwxr-xr-x   1 root     sys         949238 Apr 29 20:47 snmpusm
-rwxr-xr-x   1 root     sys         954643 Apr 29 20:47 snmpvacm
-rwxr-xr-x   1 root     sys         941454 Apr 29 20:47 snmpwalk
-rwxr-xr-x   1 root     sys          31029 Apr 29 20:47 tkmib
```

Due to the space limitations of this redbook, we cannot provide you extensive usage example for all the commands, though we explain the usage of some commands in this section.

snmpget

The simple SNMP client command **snmpget** is used to retrieve a MIB object from SNMP agents, as shown in the following example:

```
$ snmpget host1 public .iso.org.dod.internet.mgmt.mib-2.system.sysDescr.0
```

snmpwalk

The other SNMP client, called **snmpwalk**, is used to retrieve a MIB sub-tree from SNMP agents. You can dump an entire MIB tree from the IP host host1, where the read-only community name is public, as shown in the following example:

```
$ snmpwalk host1 public .
```

You can also specify a MIB sub-tree to retrieve on the command line:

```
$ snmpwalk host1 public .iso.org.dod.internet.mgmt.mib-2
```

Another advantage is that you can search MIB objects that match a partial OID specified on the command line from the whole MIB tree, as shown in Example 5-26.

Example 5-26   Search the MIB tree using snmpwalk

```
$ snmpwalk host1 public ifPhysAddress
interfaces.ifTable.ifEntry.ifPhysAddress.1 = 0:6:29:4:2f:bc
interfaces.ifTable.ifEntry.ifPhysAddress.2 = 0:6:29:ec:40:cb
interfaces.ifTable.ifEntry.ifPhysAddress.3 =
```

snmptranslate

The **snmptranslate** command is a useful tool that translates a numeric expression of an OID into a symbolic one or vice versa. You can translate the numeric expression OID .1.3.6.1.2.1.1.1.0 to the symbolic expression, as shown in the following example:

```
$ snmptranslate -Onf .1.3.6.1.2.1.1.1.0
.iso.org.dod.internet.mgmt.mib-2.system.sysDescr.0
```

Please note that you have to use a fully qualified OID by inserting a dot character '.' at the beginning of the numeric expression of it. If you do not use a fully qualified OID, it will be recognized as a relative OID name from the OID:

```
 .iso.org.dod.internet.mgmt.mib-2.system.
```

You can translate an entire MIB tree, as shown in the following example:

```
$ snmptranslate -Tp -m all
```

You can also translate a specific MIB sub-tree, as shown in the following example:

```
$ snmptranslate -Tp .1.3.6.1.2.1
```

6

Packaging your software tools

AIX has a unique and powerful software packaging mechanism compared to other UNIX based operating systems. The AIX standard packaging provides you centralized software version control and precise control of each software component.

This chapter contains the following two sections:

- Section 6.1, “Understanding the AIX standard packaging” on page 206
- Section 6.2, “Creating your own package file” on page 220

The first section gives a basic understanding of the AIX standard packaging. You can list, verify, install, update, commit, and reject software filesets or fileset updates. Once you understand the AIX standard packaging, it will help you manage software maintenance work.

The second section explains how to create your own package file. By packaging your own software tools using the AIX standard packaging, you can easily deploy your software tools in AIX server farms.

6.1 Understanding the AIX standard packaging

This section provides you with a basic understanding of the AIX standard packaging. For further information about AIX standard packaging, please refer to *AIX 5L Version 5.1 General Programming Concepts: Writing and Debugging Programs*.

6.1.1 Filesets and package files

In AIX, the smallest installable unit is a fileset. A fileset logically groups files and directories to be installed. A fileset also includes required control files and optional installation customization files.

Several filesets can be packaged in a package file. A package file is an AIX backup-format file; therefore, it is sometimes referred to as a *bff-file*.

Figure 6-1 illustrates the relationship among filesets, packages, and bundles.

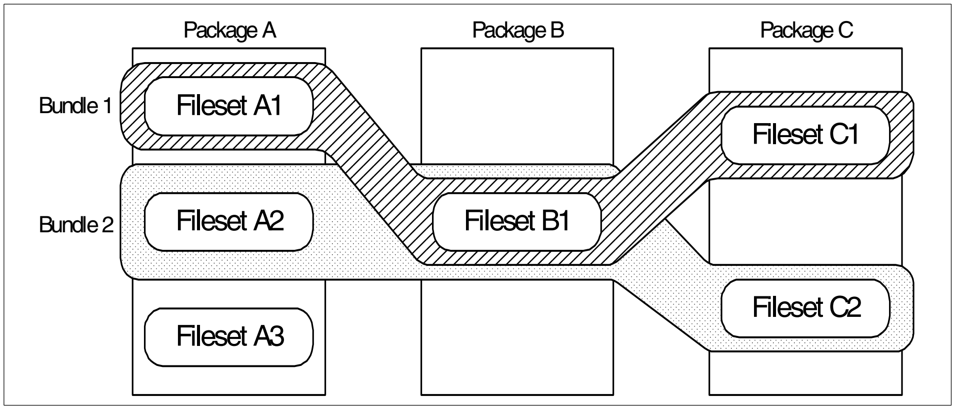


Figure 6-1 Relationship among filesets, packages, and bundles

6.1.2 Bundles

In AIX 5L Version 5.1, you have hundreds of filesets in the base operating system (BOS). Therefore, to easily select many filesets, AIX offers simple facilities, called bundles. In Figure 6-1, if you select the bundle 1, then you specify to install the fileset A1, B1, and C1. Please note that multiple bundles can include same filesets. For example, in Figure 6-1, both the bundle 1 and 2 include the fileset B1.

A bundle is defined by a file installed in either the /usr/sys/inst.data/sys_bundles or /usr/sys/inst.data/user_bundles directories. In the /usr/sys/inst.data/sys_bundles directory, you can see the system defined[1] bundles are already installed by default (see Example 6-1).

Example 6-1 System defined bundles

```
# cd /usr/sys/inst.data/sys_bundles; ls -l *.bnd
-rw-r--r--   2 bin      bin           3485 Oct 31 13:17 App-Dev.bnd
-rw-r--r--   1 bin      bin            749 Apr 05 2001  CDE.bnd
-rw-r--r--   1 bin      bin            753 Apr 07 2001  GNOME.bnd
-rw-r--r--   1 bin      bin            413 Apr 07 2001  KDE.bnd
-rw-r--r--   1 bin      bin            724 Apr 05 2001  Media-Defined.bnd
-rw-r--r--   1 bin      bin            898 Apr 05 2001  Netscape.bnd
-rw-r--r--   1 bin      bin           1184 Aug 13 2001  Server.bnd
-rw-rw-r--   1 root     system         396 Oct 31 13:40 devices.bnd
-rw-r--r--   1 bin      bin            939 Apr 05 2001  wsm_remote.bnd
```

In order to install filesets using a bundle, do the following:

1. Select SMIT panels as follows (you can access it using the SMIT short-cut easy_install):

   ```
   # smit
       Software Installation and Maintenance
           Install and Update Software
               Install Software Bundle
   ```

2. Specify the installation device (typically /dev/cd0):

   ```
   INPUT device / directory for software        [ ]
   ```

3. Select a bundle name on the panel shown in Example 6-2.

Example 6-2 Select a fileset bundle

```
                          Select a Fileset Bundle

  Move cursor to desired item and press Enter.

      App-Dev
      CDE
      GNOME
      KDE
      Media-Defined
      Netscape
      Server
      devices
      wsm_remote

    F1=Help                 F2=Refresh              F3=Cancel
```

[1] You can also create user defined bundles under the /usr/sys/inst.data/user_bundles directory.

```
F1x Esc+8=Image              Esc+0=Exit              Enter=Do
Esx /=Find                   n=Find Next
```

4. Press the Enter key; all the filesets defined in the selected bundle file, shown in Example 6-1 on page 207, will be installed.

Please note that you can only use bundles for installation purposes. AIX does not offer simple methods to uninstall bundles or to confirm whether bundles are correctly installed.

6.1.3 Managing filesets

The installed filesets can be classified under several states, as shown in Table 6-1. The successfully installed filesets should be in either an applied or committed state.

Table 6-1 Fileset state[2]

Status	Description
APPLIED	The specified fileset update is installed on the system. The APPLIED state means that the fileset update can be rejected with the **installp** command and the previous level of the fileset restored. This state is only valid for fileset updates.
APPLYING*	An attempt was made to apply the specified fileset, but it did not complete successfully, and cleanup was not performed.
BROKEN	The specified fileset or fileset update is broken and should be reinstalled before being used.
COMMITTED	The specified fileset is installed on the system. The COMMITTED state means that a commitment has been made to this level of the software. A committed fileset update cannot be rejected, but a committed fileset base level and its updates (regardless of state) can be removed or deinstalled by the **installp** command.
OBSOLETE	The specified fileset was installed with an earlier version of the operating system but has been replaced by a repackaged (renamed) newer version. Some of the files that belonged to this fileset have been replaced by versions from the repackaged fileset.
COMMITTING*	An attempt was made to commit the specified fileset, but it did not complete successfully, and cleanup was not performed.
REJECTING*	An attempt was made to reject the specified fileset, but it did not complete successfully, and cleanup was not performed.

[2] Filesets with the state specified with an asterisk (*) are not shown on the **lslpp -L** command output, since they are considered to be in a transient state.

In order to confirm the fileset status, you can use the **lslpp -L** command, as shown in Example 6-3. In this case, the fileset bos.rte.install is in committed status (C).

Example 6-3 Listing a fileset status

```
# lslpp -L bos.rte.install
  Fileset                      Level  State  Type  Description (Uninstaller)
  ----------------------------------------------------------------------------
  bos.rte.install           5.1.0.15    C     F    LPP Install Commands


State codes:
 A -- Applied.
 B -- Broken.
 C -- Committed.
 O -- Obsolete.  (partially migrated to newer version)
 ? -- Inconsistent State...Run lppchk -v.
```

If a fileset status is broken, then you should deinstall and reinstall the fileset[3]. If a fileset status is obsolete, then the fileset may or may not be supported on your system; you should consult with *AIX 5L Version Packaging Guide for LPP Installation*. If a fileset status is inconsistent, you should check it using the **lppchk -v** command.

Figure 6-2 on page 210 illustrates the state diagram of the applied and committed states. Once a base level fileset (fileset level 1.0.0.0) is installed, it is always in the committed status. If you apply an update fileset (fileset level 1.0.1.0)[4], the status is changed to the applied status.

If you commit the applied update fileset, then the updated fileset level is persistent. In order to revert to the previous fileset level, you have to uninstall the fileset and reinstall it. If you reject the applied update fileset, then the fileset level is reverted to the last committed level.

[3] If you install filesets over the network, you should verify the size and the checksum of the install source file on the target system.

[4] An update fileset is sometimes referred to as a PTF (Program Temporary Fix).

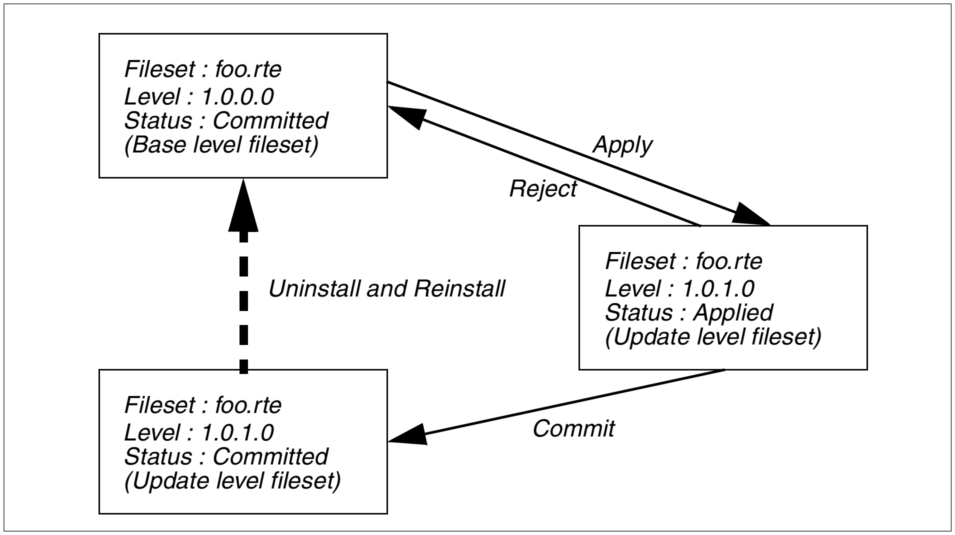


Figure 6-2 State diagram between applied and committed state

This mechanism is used to precisely control the software levels on the running system. When you encounter a software problem, you should investigate to solve the problem. If the problem is caused by some defects, software vendors supporting the software products might provide some software fixes.

APARs and fileset updates

In the IBM terminology, a software defect is uniquely identified by an identifier called an APAR (authorized program analysis reports). An APAR can be, but does not have to be, addressed by more than one fileset updates. If an APAR is addressed by multiple fileset updates, then the software defect affects many files included in multiple filesets. In Figure 6-3 on page 211[5], the APAR IZ98765 includes the fileset update of foo.rte with an update level 1.0.1.0. Therefore, in order to fix the software defect identified by the APAR IZ98765, you have to apply this single fileset update only. In order to fix the software defect identified by the APAR IZ56789, you have to apply both fileset updates, for foo.rte and gnat.rte, as shown in Figure 6-3 on page 211.

[5] We use these two identifies, IZ98765 and IZ56789, for illustrative purposes only. They do not exist.

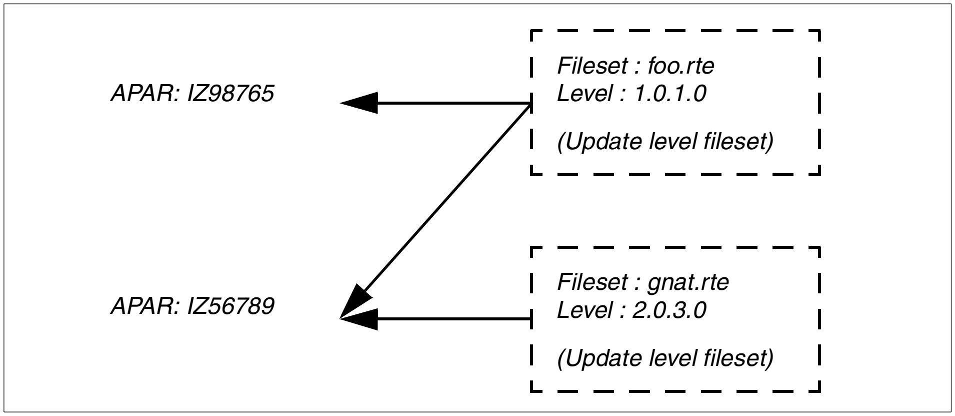


Figure 6-3 Relationship between APARs and update fileset

You can confirm whether the specific APARs are applied or not using the **instfix** command. In the following example, the APAR IY20943 is applied on the system:

```
# instfix -ivk IY20943
IY20943 Abstract: TCP CLIENT SENDS RST FLAG TO SNMPD, CAUSED IT EXITS BECAUSE

    Fileset bos.net.tcp.client:5.1.0.10 is applied on the system.
    All filesets for IY20943 were found.
```

If an APAR is not applied[6] on the system, the **instfix -ik APAR_ID** command shows the error message similar to either of the following examples:

```
All filesets for IZ98765 were not found.
```

or

```
There was no data for IZ56789 in the fix database.
```

Therefore, you do not have to remember which fileset update would fix the specific software defect, as long as you know the corresponding APAR.

To find APARs for AIX 5L Version 5.1, visit the following URL:

http://techsupport.services.ibm.com/server/support?view=pSeries

[6] An APAR classified as a packaging APAR also shows this message. A packaging APAR is provided to specify multiple APARs using one identifier, mainly for ordering and distribution purposes.

Recommended maintenance level

Sometimes IBM ships a recommended maintenance level (also referred to as RML), which includes a series of fileset updates. By using recommended maintenance levels, you can easily track the latest level of all the filesets included in AIX 5L Version 5.1. The AIX 5L Version 5.1 installation media set usually includes the latest recommended maintenance level in the Update CD.

To simply determine the latest recommended maintenance level applied on the system, you can use the **oslevel -r** command as follows:

```
# oslevel -r
5100-01
```

The command output 5100-01 shows that RML 5100-01 is applied on that system.

In order to determine what recommended maintenance levels are applied, you can use the **instfix** command as follows:

```
# instfix -i | grep ML
    All filesets for 5.0.0.0_AIX_ML were found.
    All filesets for 5.1.0.0_AIX_ML were found.
    All filesets for 5.1.0.0_AIX_ML were found.
    All filesets for 5100-01_AIX_ML were found.
```

If some lines show the message `All filesets for XXXX-YY_AIX_ML were not found`, then you can confirm which of the fileset updates included in the recommended maintenance level are not applied on the system by the following command:

```
# instfix -ivk XXXX-YY_AIX_ML | grep not | grep :
```

To download the recommended maintenance level, visit the following URL:

http://techsupport.services.ibm.com/server/support?view=pSeries

> **Note:** We strongly recommend that you purchase a software program support contract for each AIX system, even if you understand the AIX software packaging mechanism and can easily download the required fileset updates. To purchase software program support, please contact your IBM sales representative or the IBM Business Partner from which you purchased your systems.

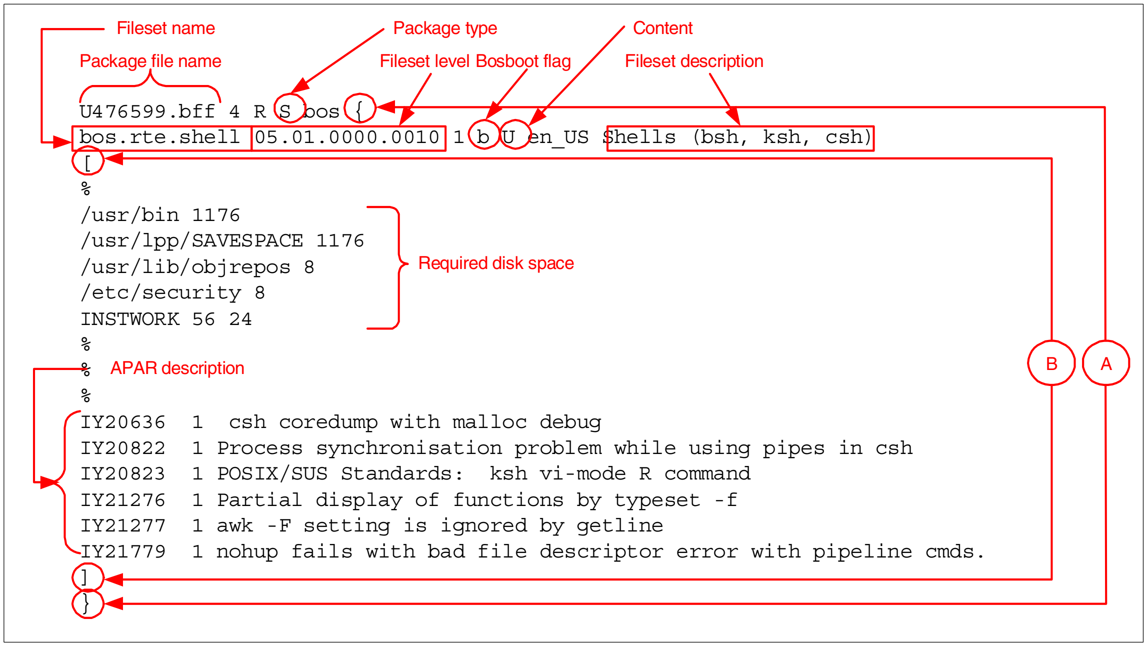
6.1.4 Viewing the TOC file (.toc)

Each AIX standard package file contains a table of contents (TOC). Before installation, this information has to be retrieved from package files and placed in the directory as a TOC file named .toc.

If you download some APARs or copy some packages from the install media to a file system, you have to issue the **inutoc** command to create the .toc file. If you copy additional packages into the /usr/sys/inst.images directory, you have to manually rebuild the .toc file using the **inutoc** command. The following example shows you how to create or update the .toc file in the /usr/sys/inst.images directory:

```
# inutoc /usr/sys/inst.images
# cd /usr/sys/inst.images; ls -l
total 800
-rw-r--r--    1 root     system          552 Apr 10 15:55 .toc
-rw-r--r--    1 root     system       403456 Apr 10 15:55 U476599.bff
```

The created .toc file is a text file, so you can view the contents using a viewer command, such as **pg** or **more**. Figure 6-4 shows an example entry of the .toc file. A package has a block shown as A in Figure 6-4. A fileset has a block shown as B in Figure 6-4. If a package contains multiple filesets, then you will see multiple blocks of B in the package block.



```
U476599.bff 4 R S bos {
bos.rte.shell 05.01.0000.0010 1 b U en_US Shells (bsh, ksh, csh)
[
%
/usr/bin 1176
/usr/lpp/SAVESPACE 1176
/usr/lib/objrepos 8
/etc/security 8
INSTWORK 56 24
%
%
%
IY20636  1  csh coredump with malloc debug
IY20822  1 Process synchronisation problem while using pipes in csh
IY20823  1 POSIX/SUS Standards:  ksh vi-mode R command
IY21276  1 Partial display of functions by typeset -f
IY21277  1 awk -F setting is ignored by getline
IY21779  1 nohup fails with bad file descriptor error with pipeline cmds.
]
}
```

Figure 6-4   Sample .toc file

Table 6-2 on page 214 explains entries shown in Figure 6-4.

Table 6-2 Fields description of the .toc file

Field name	Description
Package file name	The file name of the package.
Package type	Indicates package type: ▶ I (Installation) - All the filesets contained in this package are base filesets. ▶ S (Single update) - All the filesets contained in this package are fileset updates.
Fileset name	The name of the fileset.
Fileset level	The fileset level represented by (V.R.M.F): ▶ V - Version ▶ R - Release ▶ M - Maintenance level ▶ F - Fix level
Bosboot flag	Indicates whether a bosboot is needed after installation: ▶ N - Do not invoke bosboot ▶ b - Invoke bosboot
Content	Indicates the parts included in the fileset or the fileset update: ▶ B - usr and root part ▶ H - share part ▶ U - usr part only
Fileset description	The description of the fileset.
Required disk space	Size required for each install target directory.
APAR descriptions	Information regarding the APARs contained in the fileset update.

For further information about the format of the .toc file, please refer to *AIX 5L Version 5.1 General Programming Concepts: Writing and Debugging Programs*.

Once a .toc file is created, you can list the table of contents using the **installp** command, as shown in Example 6-4.

Example 6-4 Listing the table of contents

```
# installp -ld /usr/sys/inst.images
  Fileset Name                 Level                     I/U Q Content
  ==================================================================
  bos.rte.streams              5.1.0.10                   S  N usr
```

```
#   Streams Libraries
```

6.1.5 Viewing package files

The AIX standard packaging uses the **backup** command to archive package files. Therefore, you can un-archive it using the **restore** command. Example 6-5 shows how to view the contents of a package file.

Example 6-5 Viewing the contents of a package file

```
# restore -qTvf U476599.bff
New volume on U476599:
 Cluster 51200 bytes (100 blocks).
    Volume number 1
    Date of backup: Fri Jul 27 23:42:02 2001
    Files backed up by name
    User BUILD
           0 ./
         519 ./lpp_name
           0 ./usr
           0 ./usr/lpp
           0 ./usr/lpp/bos/bos.rte.shell/5.1.0.10
        4498 ./usr/lpp/bos/bos.rte.shell/5.1.0.10/liblpp.a
      199506 ./usr/bin/awk
      158722 ./usr/bin/csh
      240884 ./usr/bin/ksh
    total size: 604129
    files archived: 9
```

TOC information file (lpp_name)

A package always contains a file, named lpp_name, as its first archived file. This file is the table of contents of this package file, as shown in Example 6-6.

Example 6-6 Extracting the lpp_name file

```
# restore -qxf U476599.bff ./lpp_name
x ./lpp_name
# ls -l ./lpp_name
-r-xr-xr-x   1 root     system          519 Jul 27 2001  lpp_name
# cat ./lpp_name
4 R S bos {
bos.rte.shell 5.1.0.10 01 b U en_US Shells (bsh, ksh, csh)
[
%
/usr/bin 1176
/usr/lpp/SAVESPACE 1176
/usr/lib/objrepos 8
/etc/security 8
INSTWORK 56 24
%
%
%
IY20636  1 csh coredump with malloc debug
IY20822  1 Process synchronisation problem while using pipes in csh
IY20823  1 POSIX/SUS Standards: ksh vi-mode R command
IY21276  1 Partial display of functions by typeset -f
IY21277  1 awk -F setting is ignored by getline
IY21779  1 nohup fails with bad file descriptor error with pipeline cmds.
%
]
}
```

Upon invoking the **inutoc** command, it parses the specified directory and extracts the lpp_name file from each package file, then concatenates the extracted information to the .toc file.

Installation control library file (liblpp.a)

A package also has to contain an installation control library file, named liblpp.a, under the appropriate sub-directory. This file is an ar format archive file, which contains the files, as shown in Example 6-7.

Example 6-7 Extracting the liblpp.a file

```
# restore -qxf U476599.bff ./usr/lpp/bos/bos.rte.shell/5.1.0.10/liblpp.a
x ./usr/lpp/bos/bos.rte.shell/5.1.0.10/liblpp.a
# cd ./usr/lpp/bos/bos.rte.shell/5.1.0.10
# ls -l liblpp.a
-r-xr-xr-x   1 root     system         4498 Jul 27 2001  liblpp.a
# file liblpp.a
liblpp.a:      archive
# ar -t liblpp.a
productid
bos.rte.shell.copyright
bos.rte.shell.inventory
bos.rte.shell.size
bos.rte.shell.tcb
bos.rte.shell.al
bos.rte.shell.fixdata
```

The <fileset_name>.al file contains the list of files in the fileset. Example 6-8 shows you the contents of the bos.rte.stell.inventory file, which contains three files specified using relative path name starting from the root directory.

Example 6-8 Contents of the bos.rte.streams.al file

```
./usr/bin/awk
```

```
./usr/bin/csh
./usr/bin/ksh
```

The <fileset_name>.inventory file contains the required information about files within the fileset (see Example 6-9). Each file has entries explained in Table 6-3.

Table 6-3 Definition of entries in <fileset_name>.inventory

Entry name	Description
owner	Specifies the file owner.
group	Specifies the file group.
mode	Specifies the permission bit of the file.
type	Specifies the file type.
links	Specifies a hard-link of the file (if available).
class	The logical group of the file.
size	Specifies the size of the file.
checksum	Specifies the result of the **cksum** command to the file.

These values are used to verify the contents of restored files from the package file if those files are restored correctly.

Example 6-9 Contents of the bos.rte.shell.inventory file

```
/usr/bin/awk:
          owner = bin
          group = bin
          mode = 555
          type = FILE
          links = /usr/bin/nawk
          class = apply,inventory,bos.rte.shell
          size = 199506
          checksum = "51888   195 "

/usr/bin/csh:
          owner = bin
          group = bin
          mode = 555
          type = FILE
          class = apply,inventory,bos.rte.shell
          size = 158722
          checksum = "18128   156 "

/usr/bin/ksh:
          owner = bin
          group = bin
          mode = TCB,555
          type = FILE
          links = /usr/bin/sh,/usr/bin/psh,/usr/bin/tsh
          class = apply,inventory,bos.rte.shell
          size = 240884
          checksum = "60163   236 "
```

For further information about the format of these files contained in the liblpp.a file, please refer to *AIX 5L Version 5.1 General Programming Concepts: Writing and Debugging Programs*.

6.1.6 RPM (Red Hat Package Manager)

In addition to the AIX standard packaging format, AIX 5L Version 5.1 also supports RPM (Red Hat Package Manager) format, which is commonly used on Linux platforms. In AIX 5L Version 5.1, the RPM is used for the software components provided by the AIX toolbox for Linux applications. The AIX toolbox for Linux applications are *AS-IS* software collections ported from Linux.

Upon installation of AIX 5L Version 5.1, a fileset, rpm.rte, is automatically installed, as shown in the following example:

```
# lslpp -L rpm.rte
  Fileset                      Level  State  Type  Description (Uninstaller)
  ----------------------------------------------------------------------------
  rpm.rte                   3.0.5.30    C     F    RPM Package Manager


State codes:
 A -- Applied.
 B -- Broken.
 C -- Committed.
 O -- Obsolete.  (partially migrated to newer version)
 ? -- Inconsistent State...Run lppchk -v.

Type codes:
 F -- Installp Fileset
 P -- Product
 C -- Component
 T -- Feature
 R -- RPM Package
```

The rpm.rte fileset includes required commands and files to manage the RPM packages on AIX. If you install an RPM package, the information is stored in the RPM database. The **lslpp** command looks for the RPM database to show RPM package names, as shown in Example 6-10. Please note that the type column shows R, which means the RPM package format.

Example 6-10 Listing an RPM package using lslpp

```
# lslpp -L | head -1; lslpp -L | grep cdrecord
  Fileset                      Level  State  Type  Description (Uninstaller)
  cdrecord                       1.9    C     R    A command line CD/DVD
recording
```

You cannot use the **lslpp** command to confirm detailed information about the RPM package, because the AIX software packaging mechanism does not directly control RPM packages[7]. For example:

```
# lslpp -L cdrecord
lslpp: 0504-132  Fileset cdrecord not installed.
```

In order to acquire detailed information about RPM packages, you have to use the **rpm** command (see Table 6-4).

Table 6-4 rpm command usage example

Command line usage	Description
rpm -qa	Lists all the installed RPM packages.
rpm -ql <package_name>	Lists files included in the RPM package <package_name>.
rpm -Uvh <package_filename>	Installs or updates an RPM package supplied as a package file <package_filename>.
rpm -e <package_name>	Uninstalls an RPM package <package_name>.

For further information about RPM and its usage, please refer to:

- *AIX 5L Version 5.1 Packaging Guide for LPP Installation*
- *AIX 5L Version 5.1 Commands Reference*
- The Web site http://www.rpm.org

[7] The RPM packaging mechanism uses a separate RPM database, /var/opt/freeware/lib/rpm, with a symbolic link created in /var/lib.

6.2 Creating your own package file

In this section, we explain how to create your own package file using an example. We selected the NET-SNMP tools, which are used in Section 5.3, “NET-SNMP” on page 185, for the example of packaging.

To create a package, we provide you a short Perl script, makebff, (see Appendix B, “Example scripts” on page 313). It simplifies the complicated packaging task by using the files and directory structure in these sections:

- Section 6.2.1, “Creating the required directory structure” on page 220
- Section 6.2.2, “Creating the list file” on page 223
- Section 6.2.3, “Creating <fileset_name>.al files” on page 225

The makebff script is designed to provide the simplest way to create packages. Therefore, it does not exploit all the functionality provided by the AIX standard packaging; for example, it does not support APAR description listing.

Although the makebff script is relatively primitive, we think it has enough capability to package your own software tools, because it supports all the functionality explained in Section 6.1, “Understanding the AIX standard packaging” on page 206.

Note: You should only package software tools that are well tested.

6.2.1 Creating the required directory structure

Before you start packaging, you have to create the following directory structure, as shown in Figure 6-5.

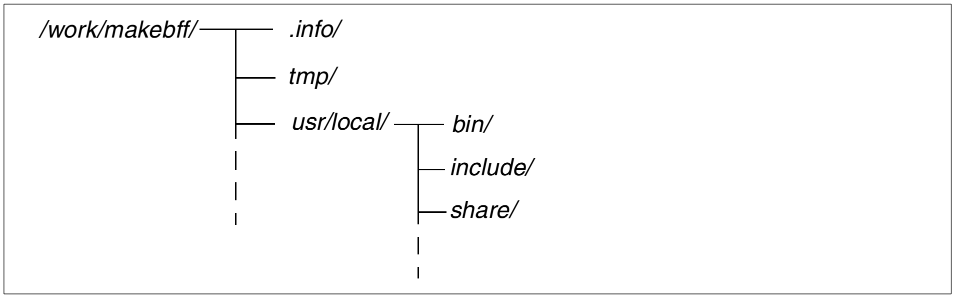


Figure 6-5 Directory structure for packaging

Note: You should create a separate file system for /work rather than simply increasing the root file system size.

- You can select any directory as your package top directory. Throughout this redbook, we select /work/makebff as the package top directory.
- The /work/makebff/.info directory stores several control files, as explained in Section 6.1.5, “Viewing package files” on page 215.
- The /work/makebff/tmp directory is used for storing a generated package file.
- You have to create an appropriate directory structure under the package top directory to store the install target files. These files must be placed under this directory structure with the appropriate owner, group, and permission mode. Because most free software tools, including NET-SNMP, are usually installed under the /usr/local directory, you should create the /usr/local directory under the package top directory, as well as the following directories, which should be expected under the /usr/local directory:

```
# pwd
/work/makebff
# ls -l ./usr/local
total 56
drwxr-sr-x   2 root     system          512 Apr 23 19:10 bin/
drwxr-sr-x   3 root     system          512 Apr 23 19:10 include/
drwxr-sr-x   2 root     system          512 Apr 23 19:10 lib/
drwxr-sr-x   6 root     system          512 Apr 23 19:11 man/
drwxr-sr-x   2 root     system          512 Apr 23 19:11 sbin/
drwxr-sr-x   3 root     system          512 Apr 23 19:11 share/
```

If there are many files to be packaged, it might be a troublesome job to place them appropriately. To easily track down these files, we recommend the following steps:

1. Create a new file system and mount it under the package top directory. In this case, we create the new /work/makebff/usr/local file system:

```
# mklv -t jfs -y bfflv workvg 4
bfflv
# crfs -v jfs -d bfflv -m /work/makebff/usr/local
Based on the parameters chosen, the new /work/makebff/usr/local JFS file
system
is limited to a maximum size of 134217728 (512 byte blocks)

New File System size is 131072
# mount /work/makebff/usr/local
```

2. Unmount this file system and mount it on the /usr/local directory temporarily:

```
# umount /work/makebff/usr/local
# mount /dev/bfflv /usr/local
# df -k /usr/local
Filesystem    1024-blocks      Free %Used    Iused %Iused Mounted on
/dev/bfflv          65536     63436    4%       17     1% /usr/local
```

3. Install softwares upon this new file system. Typically, free software tools provide you an *install* target directive in its makefile. In this example, since we compiled the NET-SNMP tools in the /work/src/NET-SNMP-4.2.3.0 directory (see Section 5.3.1, “Installing NET-SNMP” on page 186), the installation process of the NET-SNMP tools is shown in the following example:

   ```
   # cd /work/src/NET-SNMP-4.2.3.0
   # make install
   ```

4. Unmount the temporary file system /usr/local and mount it on the original mount point /work/makebff/usr/local.

   ```
   # umount /usr/local
   # mount /work/makebff/usr/local
   ```

You should see the files are appropriately placed under the /work/makebff/usr/local directory, as if the package top directory /work/makebff were virtually considered as the root directory.

Before proceeding, you should verify the following items:

- Confirm the file owner and group.

  If you specified an owner or a group ID that only exists on your AIX system, then the package that you are going to create can be installed on your system only.

- Confirm the file and directory permission.

  Unless absolutely required by tight security policy, all the files should be set as *world-readable* (**chmod a+r**) and all the directories should be set as *world-executable* (**chmod a+x**).

  The executable permission bit for a directory defines whether a process with the appropriate credential can change the current directory on that directory. The world-executable directory means that any process can change the current directory on that directory.

Sometimes, you should not hide the /usr/local file system by over-mounting a temporary file system. For example, if a process is opening files[8] on the existing /usr/local file system, the over-mounted temporary file system might interrupt access to opened files on the covered file system. Therefore, you should confirm no process is opening files or having the current directory in the /usr/local file system by using the **fuser** command before over-mounting the temporary file system.

[8] Not only regular data files, but also executable and shared library files can be opened by a process.

If you cannot hide the current /usr/local file system, you can still create your own package by manually creating <fileset_name>.al files, as explained in “Manually creating a <fileset_name>.al file” on page 226.

Please note that the makebff script can use directories other than ones created under the package directory if you create appropriate symbolic links under the package directory. For example, if you create the symbolic link /work/makebff/usr/local that points to /usr/local, then the makebff script traverses the actual /usr/local directory to find files to be packaged.

6.2.2 Creating the list file

The *list* file is a simple format text data file that is exclusively[9] used by the makebff script. The list file must be created in the .info directory. By parsing the list file, the makebff script can automatically generate the required control files explained in Section 6.1.5, “Viewing package files” on page 215.

The list file defines the following components:

- Package name
- Multiple fileset names
- Fileset levels
- Flags that define whether the package is an installation package or a update package
- Prerequisite condition directives

Note: Do not select a package and fileset name that is used by any IBM software products, including AIX. We strongly recommend you to select a unique name, such as your company name, for the first part of the package name. In this example, we use sg246606 for the first part of the package and fileset name.

Example 6-11 illustrates a sample of the list file to be used for packaging the NET-SNMP tools. In this example, the package sg246606.net-snmp is composed of two filesets sg246606.net-snmp.rte and sg246606.net-snmp.data.

Example 6-11 A sample list file for the NET-SNMP tools

```
sg246606.net-snmp           4.2.3.0 I
sg246606.net-snmp.rte       4.2.3.0 NET-SNMP 4.2.3 runtime.
*prereq xlC.rte 5.0.2.1
*prereq xlC.aix50.rte 5.0.2.2
```

[9] The standard AIX packaging mechanism does not define this file.

```
sg246606.net-snmp.data         4.2.3.0 NET-SNMP 4.2.3 source.
```

- The first field of the first line defines the package name. In this case, the package name is sg246606.net-snmp. The second field of the first line, the dotted number string 4.2.3.0, is used for a part of the package file name to be generated. You can also specify the I or U flag in the last field of the line. The default value, the I flag, specifies that the package only contains installation filesets (base filesets), and the U flag specifies that the package only contains update filesets.
- The rest of the lines after the first line are either of the following categories:
  - Prerequisite condition directive lines

    If a line is started by an asterisk '*', then the makebff script consider this line as a prerequisite condition directive line. Upon encountering '*' at the beginning of a line, the makebff script simply copies the entire line into the lpp_info file as a prerequisite condition of the preceding fileset definition line.

    In this example, the sg246606.net-snmp.rte fileset has two prerequisite condition directive lines.

    > **Note:** The makebff script does not verify prerequisite condition directive lines. Therefore, you have to carefully specify these conditions to meet the definition defined by "TOC information file (lpp_name)" on page 215.

  - Fileset definition lines

    The first field defines the fileset name. For example, the fileset name is sg246606.net-snmp.rte in the second line. The second field defines version, release, modify, and fix level of the fileset. In the second line, the fileset level is 4.2.3.0. The fileset level does not have to be same as the package level. The last field defines the fileset description. For example, the description reads `NET-SNMP 4.2.3 runtime` on the second line.
- The makebff script ignores any blank lines. Any white spaces at the beginning or ending of the line are also ignored by the makebff script.
- A valid line is composed of three fields separated by white spaces, excluding pre-requisite condition directive lines. You can use white spaces in the third field of the fileset definition line, because the makebff script only recognizes the first and second white space blocks as field separators.

6.2.3 Creating <fileset_name>.al files

You have to create the <fileset_name>.al file for each fileset under the .info directory. This file should contain all the files to be packaged, using the relative path names starting from the root directory.

> **Note:** The <fileset_name>.al file should contain not only regular files, but also directories and symbolic links, which should be packaged in the fileset.

In the following example, we create the sg246606.net-snmp.rte.al file that contains all the files under the /work/makebff/usr/local directory:

```
# cd /work/makebff
# find ./usr/local > .info/sg22046606.net-snmp.rte.al
```

Example 6-12 shows the contents of the file.

Example 6-12 Contents of the sg246606.net-snmp.rte.al file

```
./usr/local
./usr/local/lost+found
./usr/local/share
./usr/local/share/snmp
./usr/local/share/snmp/mibs
./usr/local/share/snmp/mibs/RFC1155-SMI.txt
[… many lines are skipped …]
./usr/local/bin
./usr/local/bin/snmpinform
./usr/local/bin/snmpnetstat
./usr/local/bin/snmpget
./usr/local/bin/snmpgetnext
[… many lines are skipped …]
```

As for directories, we recommend you exclude directories that are not exclusively occupied by the fileset itself from the fileset_name.al file. In this example, you should delete the following lines from the sg246606.net-snmp.rte.al file using a text editor:

- ./usr/local
- ./usr/local/lost+found
- ./usr/local/{bin, include, lib, man, sbin, share}
- ./usr/local/man/{man1, man3, man5, man8}

We also create another file, sg246606.net-snmp.data.al, which just contains the following one line under the .info directory:

```
./usr/local/src/net-snmp-4.2.3.0.tar.gz[10]
```

This is an archived source file available on the NET-SNMP Home Page, found at:

http://net-snmp.sourceforge.net/

Manually creating a <fileset_name>.al file

A workaround to manually track down all the files can be provided by the -cnewer option of the GNU **find** command[11]. The -cnewer option instructs the GNU **find** command to compare the time stamp of the specified file and files to be examined. Example 6-13 shows an example of using the GNU **find** command to create a <fileset_name>.al file.

Example 6-13 Creating <fileset_name>.al file manually

```
# > /tmp/just_created_now
# pwd
/work/src/NET-SNMP-4.2.3.0
# make install
# cd /
# find ./usr/local -cnewer /tmp/just_created_now > \
    /work/makebff/.info/sg246606.net-snmp.rte.al
```

Please note that you still have to verify the generated <fileset_name>.al file.

6.2.4 Invoking the makebff script

By simply invoking the makebff script on the package top directory, as shown in the following example, the script generates the sg246606.net-snmp.4.2.3.0.bff package file in the ./tmp directory:

```
# pwd
/work/makebff
# ./makebff
makebff version 1.0.0.7
sg246606.net-snmp 4.2.3.0 I
processing sg246606.net-snmp.rte
processing sg246606.net-snmp.data
creating ./.info/liblpp.a
ar: Creating an archive file ./.info/liblpp.a.
creating ./tmp/sg246606.net-snmp.4.2.3.0.bff
```

You can confirm that necessary control data files were automatically generated in the ./.info directory, as shown in the following example:

```
# ls ./.info/
liblpp.a
list
```

[10] Previously, NET-SNMP was called as ucd-snmp. The source archive file name still uses this name.

[11] The GNU **find** command is provided by the AIX toolbox for Linux applications.

```
sg246606.net-snmp.data.al
sg246606.net-snmp.data.inventory
sg246606.net-snmp.rte.al
sg246606.net-snmp.rte.inventory
```

6.2.5 Verifying your packages

Before deploying your package on AIX server farms, you should do at least the following:

- Installation verification
- Verification of the installed files
- Uninstallation verification

Installation verification

At first, you should verify whether you can generate a valid .toc file from the created package, as shown in the following example:

```
# pwd
/work/makebff
# ls -l ./tmp/
sg246606.net-snmp.4.2.3.0.bff
# inutoc ./tmp/
# ls tmp/
.toc                            sg246606.net-snmp.4.2.3.0.bff
# installp -ld ./tmp/
  Fileset Name                Level                     I/U Q Content
  ====================================================================
  sg246606.net-snmp.data      4.2.3.0                    I  N usr
#   NET-SNMP 4.2.3 source.

  sg246606.net-snmp.rte       4.2.3.0                    I  N usr
#   NET-SNMP 4.2.3 runtime.
```

Then preview the installation using the -p (preview) option of the **installp** command as shown in the following example:

```
# pwd
# /work/makebff/tmp
# installp -pacgXNd. all
*******************************************************************************
installp PREVIEW:  installation will not actually occur.
*******************************************************************************

+-----------------------------------------------------------------------------+
                    Pre-installation Verification...
+-----------------------------------------------------------------------------+
Verifying selections...done
Verifying requisites...done
Results...

SUCCESSES
---------
  Filesets listed in this section passed pre-installation verification
  and will be installed.

  Selected Filesets
  -----------------
  sg246606.net-snmp.data 4.2.3.0              # NET-SNMP 4.2.3 source.
  sg246606.net-snmp.rte 4.2.3.0               # NET-SNMP 4.2.3 runtime.

  << End of Success Section >>

FILESET STATISTICS
------------------
    2  Selected to be installed, of which:
        2  Passed pre-installation verification
  ----
    2  Total to be installed

RESOURCES
---------
  Estimated system resource requirements for filesets being installed:
                (All sizes are in 512-byte blocks)
      Filesystem                     Needed Space             Free Space
      /usr                               51192                  274560
      -----                            --------                 ------
      TOTAL:                             51192                  274560

  NOTE:  "Needed Space" values are calculated from data available prior
  to installation.  These are the estimated resources required for the
  entire operation.  Further resource checks will be made during
  installation to verify that these initial estimates are sufficient.

******************************************************************************
End of installp PREVIEW.  No apply operation has actually occurred.
******************************************************************************
```

If there is no prerequisite fileset level conflict, then you can install these filesets, as shown in the following example:

```
# pwd
# /work/makebff/tmp
# installp -acgXNd. all
… command output is not shown …
# lslpp -L 'sg246606.net-snmp.*'
  Fileset                      Level  State  Type  Description (Uninstaller)
  ----------------------------------------------------------------------------
  sg246606.net-snmp.data     4.2.3.0    C     F    NET-SNMP 4.2.3 source.
  sg246606.net-snmp.rte      4.2.3.0    C     F    NET-SNMP 4.2.3 runtime.


State codes:
 A -- Applied.
 B -- Broken.
 C -- Committed.
 O -- Obsolete.  (partially migrated to newer version)
 ? -- Inconsistent State...Run lppchk -v.

Type codes:
 F -- Installp Fileset
 P -- Product
 C -- Component
 T -- Feature
 R -- RPM Package
```

Verification of the installed files

If your filesets are successfully installed, then you should verify the installed files are correctly placed using the **lppchk** command, as shown in Example 6-14. If the command returns with non-zero exit code, some files are inconsistent with the fileset inventory information.

Example 6-14   Verifying filesets contents using lppchk

```
# lppchk -f -m 3 sg246606.net-snmp.* ; echo $?
lppchk: 0504-230  212 files have been checked.
0
# lppchk -c -m 3 sg246606.net-snmp.* ; echo $?
lppchk: 0504-230  212 files have been checked.
0
# lppchk -l -m 3 sg246606.net-snmp.* ; echo $?
0
# lppchk -v -m 3 sg246606.net-snmp.* ; echo $?
0
```

The following example explains the purpose of the specified flags in Example 6-14:

```
# lppchk -?
lppchk: Not a recognized flag: ?
lppchk Usage:
lppchk -{f|c|l|v} [-u] [-O{[r][s][u]}] [-mn] [fileset [filelist]]
        -f  Fast check (file existence, file length)
        -c  Checksum verification
        -v  Fileset version consistency check
        -l  File link verification
        -u  Update inventory (only valid with -c or -l)
        -O  Data base(s) to be processed, default is all
            u = /usr/lib/objrepos, r = /etc/objrepos,
            s = /usr/share/lib/objrepos
        -mn n=1-3 controls detail of messages, 3 is most verbose
        fileset specifies filesets to be checked, may include
            "*", etc to specify multiple filesets
        filelist one or more file names (optionally using "*", etc.)
            to be checked.  May be in form  'member:archive' to specify
            archive members that are to be checked.
        --- one and only one of the flags -f, -c, -l and -v may be specified
        --- filelist is not allowed with -v option
```

You should also verify installed programs work correctly. As for the NET-SNMP tools usage, see Section 5.3, “NET-SNMP” on page 185.

Uninstallation verification

As the last verification task, you should confirm if your filesets can be cleanly removed from the system by uninstalling the filesets. The following example shows the uninstallation process of the filesets:

```
# installp -u sg246606.net-snmp.*
+-----------------------------------------------------------------------------+
                    Pre-deinstall Verification...
+-----------------------------------------------------------------------------+
Verifying selections...done
Verifying requisites...done
Results...

SUCCESSES
---------
  Filesets listed in this section passed pre-deinstall verification
  and will be removed.

  Selected Filesets
  -----------------
  sg246606.net-snmp.data 4.2.3.0              # NET-SNMP 4.2.3 source.
  sg246606.net-snmp.rte 4.2.3.0               # NET-SNMP 4.2.3 runtime.

  << End of Success Section >>

FILESET STATISTICS
------------------
    2  Selected to be deinstalled, of which:
        2  Passed pre-deinstall verification
  ----
    2  Total to be deinstalled
```

```
+-----------------------------------------------------------------------------+
                           Deinstalling Software...
+-----------------------------------------------------------------------------+

installp:  DEINSTALLING software for:
        sg246606.net-snmp.data 4.2.3.0
        sg246606.net-snmp.rte 4.2.3.0

Finished processing all filesets.  (Total time:  10 secs).

+-----------------------------------------------------------------------------+
                                Summaries:
+-----------------------------------------------------------------------------+

Installation Summary
--------------------
Name                        Level           Part        Event       Result
-------------------------------------------------------------------------------
sg246606.net-snmp.data      4.2.3.0         USR         DEINSTALL   SUCCESS
sg246606.net-snmp.rte       4.2.3.0         USR         DEINSTALL   SUCCESS
```

You should confirm whether the installed files are cleanly removed from the system by issuing **`ls -lR /usr/local`**.

Now you are ready to deploy your packages on AIX server farms.

7

Day-to-day tasks

Once a server farm has been deployed and servers installed, they must be managed from day to day. There are many repetitive administration tasks to be performed, so careful automation of these tasks will reduce disruption and call-out for servers in your farm. Once you have designed your automation routines, you must ensure every server has, within the farm, the software installed.

This chapter contains the following six sections that provide you with many useful system management methods for the regular day-to-day running of AIX systems in a server farm environment:

- Section 7.1, “Cloning methods of operating system images” on page 234
- Section 7.2, “Software level maintenance” on page 252
- Section 7.3, “Network management” on page 253
- Section 7.4, “Log file management” on page 264
- Section 7.5, “Performance” on page 268
- Section 7.6, “General administration tasks” on page 273

7.1 Cloning methods of operating system images

Cloning an existing machine from a previously defined and tested system is probably the easiest way to build servers. A master machine is installed from scratch and customized with all the software, users, and services you require.

You can install AIX 5L Version 5.1 from scratch using the product installation media (CD-ROM media number one is bootable). As for detailed information about the installation method using a CD-ROM, please refer to the publication *AIX 5L Version 5.1 Installation Guides: Installation Guide.*

After creating a mksysb image (explained in Section 7.1.1, “Creating a mksysb image” on page 235), you can use one of the following installation methods to clone the operating system image that you customized:

- mksysb tape[1]
- Section 7.1.2, “mksysb on DVD or CD” on page 236
- Section 7.1.3, “Alternate disk install” on page 244
- Section 7.1.4, “Network Installation Manager (NIM)” on page 248

However, there are several issues to consider:

- If you require additional software, you either need multiple copies of the media or must manually load the post-install.
- The new system being built must have hardware similar to your original system. If there are any additional adapters, the required drivers may be missing.
- The newly cloned system will have same configuration information, such as same IP addresses and routes, as the original, unless you use NIM. You have to modify the information post-installation.

> **Note:** In AIX 5L Version 5.1, if you modify the bosinst.data file before creating mksysb as `RECOVER_DEVICE=no`, then device specific customization will not be performed upon restoration.

Whichever method you choose to clone operating system images, you should ensure every server leaving your build environment is hardened and ready for use in the server farm.

[1] Restoring AIX from a mksysb tape is a straightforward easy task. Therefore, we do not cover this topic in this redbook.

7.1.1 Creating a mksysb image

The **mksysb** command can create a bootable backup image of the AIX 5L Version 5.1 operating system along with your customization. If your system is equipped the with appropriate media devices, such as a tape drive and DVD-RAM drive, you can create bootable media that is used to recover your system. You can also create a mksysb image file on either local or remote file systems. You can use the mksysb image file to create bootable mksysb CD or DVD media, to be used with an alternate disk install or NIM.

To create a mksysb image, do the following:

1. Run **smitty** and select the following menus:

   ```
   # smitty
       System Storage Management (Physical & Logical Storage)
           System Backup Manager
               Back Up the System
                   Back Up This System to Tape/File
   ```

   You will see the following panel, as shown in Example 7-1.

Example 7-1 Back up the system

```
                               Back Up the System

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

[TOP]                                                   [Entry Fields]
    WARNING:  Execution of the mksysb command will
              result in the loss of all material
              previously stored on the selected
              output medium. This command backs
              up only rootvg volume group.

* Backup DEVICE or FILE                               [/backup/bos.51.mksysb]+/
  Create MAP files?                                   no                     +
  EXCLUDE files?                                      no                     +
  List files as they are backed up?                   no                     +
  Generate new /image.data file?                      yes                    +
  EXPAND /tmp if needed?                              no                     +
  Disable software packing of backup?                 no                     +
Number of BLOCKS to write in a single output       []                        #
     (Leave blank to use a system default)
[BOTTOM]

F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         Esc+6=Command       Esc+7=Edit          Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do
```

2. Specify the backup device name or the image file name you are going to create in the Backup DEVICE or FILE field. In this example, we specified the /backup/bos.51.mksysb file. Then press Enter.
3. Depending on the operating image size and the device you choose, you will see the `operation completed` message after a couple of minutes.

You can also use the **mksysb** command directly, for example, **mksysb -i /dev/rmt0**.

If you want to exclude certain files or file systems from your mksysb image, you can use the file /etc/exclude.rootvg. After adding the file name patterns that you want to exclude from the mksysb image, you can do either of the following:

- Using SMIT, select yes on the EXCLUDE files? field on the SMIT panel, as shown in Example 7-1 on page 235. By default, the value is `no`.
- Use the -e flag of the **mksysb** command, for example, **mksysb -ie /dev/rmt0**.

Example 7-2 shows sample contents of the /etc/exclude.rootvg file. If you specify the -e option after creating this example, you instruct the **mksysb** command not to back up the contents of two separate file systems (/tmp and /logs), anything with /scratch/ in the path, or any file ending in .log.

Example 7-2 Sample /etc/exclude.rootvg

```
/scratch/
^./tmp/
^./logs/
*.log
```

7.1.2 mksysb on DVD or CD

You can use DVD (DVD-R or DVD-RAM) or CD (CD-R or CD-RW+) devices to create bootable mksysb media on AIX 5L Version 5.1. To create bootable mksysb DVD or CD media, the following requirements should be met on your system:

- Suitable DVD or CD devices are configured.
- GNU mkisofs/cdrecord tools installed.
- Appropriate symbolic links are created.
- Sufficient disk spaces are provided.

DVD or CD devices

IBM provides DVD-RAM devices for some newer models as optional features. For example, Model 6C1 has an optional feature, DVDRAM SCSI re-writable 4.7 GB black bezel (FC 2523). However, IBM does not provide any CD recordable devices (CD-R, CD-RW, and CD-RW+) for the IBM @server pSeries models. Therefore, we only use DVD-RAM devices in this chapter. The following example shows the configured DVD-RAM device on the Model 6C1:

```
# lsdev -Cc adapter | grep scsi
scsi0   Available 10-60    Wide/Ultra-3 SCSI I/O Controller
scsi1   Available 10-61    Wide/Ultra-3 SCSI I/O Controller
# lsdev -Cc cdrom
cd0 Available 10-60-00-2,0 SCSI DVD-RAM Drive
# lscfg -vl cd0
  DEVICE            LOCATION          DESCRIPTION

  cd0               10-60-00-2,0      SCSI DVD-RAM Drive (4700 MB)

        Manufacturer................IBM
        Machine Type and Model......DVRM00203
        ROS Level and ID............A127
        Device Specific.(Z0)........058002028F000010
        Part Number.................04N5272
        EC Level....................F74471
        FRU Number..................04N5967
```

As shown in this example, the physical location code of the DVD-RAM drive, 10-60-00-2,0, specifies that the following connection addresses:

SCSI BUS	0
SCSI ID	2
SCSI LUN	0

The DVD-RAM drive, used with the correct media, acts as a *big* CD-R drive; it uses the same ISO 9660 format with CD media except for its size (4.7 GB per surface, total 9.4 GB per media[2]; you have to flip it by yourself).

> **Note:** A DVD-RAM drive is *not* compatible with DVD-R, DVD-RW, DVD-RW+, or DVD-ROM media. Make sure the DVD-RAM media you use is 9.4 or 4.7 GB, and Type I, not Type II.

As for the CD-R devices, see Appendix C, “How to use CD-R on AIX” on page 329.

[2] Assuming the DVD-RAM media is double sided.

GNU mkisofs/cdrecord tools

GNU mkisofs/cdrecord tools are quite common software tools that are used for creating ISO9660 images and burning them onto CD or DVD media on most UNIX operating systems, found at:

http://www.fokus.gmd.de/research/cc/glone/employees/joerg.schilling/private/cdrecord.html

In AIX 5L Version 5.1, the following two RPM packages are installed by default:

```
# rpm -qa | egrep "cdrecord|mkisofs"
cdrecord-1.9-4
mkisofs-1.13-4
```

To get the latest version of these tools, please visit the following URL:

http://www-1.ibm.com/servers/aix/products/aixos/linux/download.html

Creating appropriate symbolic links

To use GNU mkisofs/cdrecord tools with the **mkcd** command, the following symbolic links should be created:

```
# ls -l /usr/sbin/burn_cd
lrwxrwxrwx   1 root     system          45 Mar 19 18:53 /usr/sbin/burn_cd@ ->
/usr/samples/oem_cdwriters/burn_cd_gnu*
# ln -sf /usr/samples/oem_cdwriters/burn_cd_gnu_dvdram /usr/sbin/burn_cd
# ls -l /usr/sbin/burn_cd
lrwxrwxrwx   1 root     system          45 Mar 19 18:53 /usr/sbin/burn_cd@ ->
/usr/samples/oem_cdwriters/burn_cd_gnu_dvdram*
# mkdir /usr/local/bin
# ln -s /usr/bin/readcd /usr/local/bin/readcd
```

In this example, we re-linked the /usr/sbin/burn_cd file to explicitly specify only using the DVD-RAM drive. You can also specify the -d option to the **burn_cd** command to specify using the DVD-RAM drive.

Sufficient disk spaces

The **mkcd** command automatically creates the following three file systems in rootvg, if they do not exist:

/mkcd/mksysb_image Automatically created unless specified using the -M flag.

/mkcd/cd_fs Automatically created unless specified using the -C flag. The size of this file system[3] must be greater than 4.38 GB if you are using DVD or greater than 645 MB if you are using a CD.

[3] Technically, the su directory /mkcd/cd_images must have this free space.

/mkcd/cd_images Automatically created unless specified using the -I flag.

You have to make sure that these file systems *are* created as large file enabled JFS. Also, make sure the root user's ulimit is set to `unlimited`.

You can create a different file system and instruct the **`mkcd`** command to use this file system using the -M, -C, and -I flags. We recommend you create the /backup file system for this purpose as a large file enabled JFS, as shown in the following example:

```
# ulimit
unlimited
# df -kI /backup
Filesystem    1024-blocks      Used      Free %Used Mounted on
/dev/backup       5013504    158440   4855064    4% /backup
# lsfs -q /backup | grep bf
  (lv size: 10027008, fs size: 10027008, frag size: 4096, nbpi: 4096, compress:
no, bf: true, ag: 64)
```

How to burn a mksysb CD or DVD media

Figure 7-1 on page 240 illustrates the basic steps to burn the bootable mksysb CD or DVD media on AIX. The **`mkcd`** command automatically creates the mksysb image file in the /mkcd/mksysb_image file system, unless specified by the -M flag that specifies the already created mksysb image file name (A). Then the command copies the mksysb image to the /mksysb/cd_fs file system (B). The /mksysb/cd_fs file system represents the root directory of the CD media to be created. Next, the command creates ISO 9660 image files using the **`mkrr_fs`** command (C). The /mkcd/cd_images directory is used to store created ISO 9660 image files. Finally, the command burns the media using the **`burn_cd`** command (D).

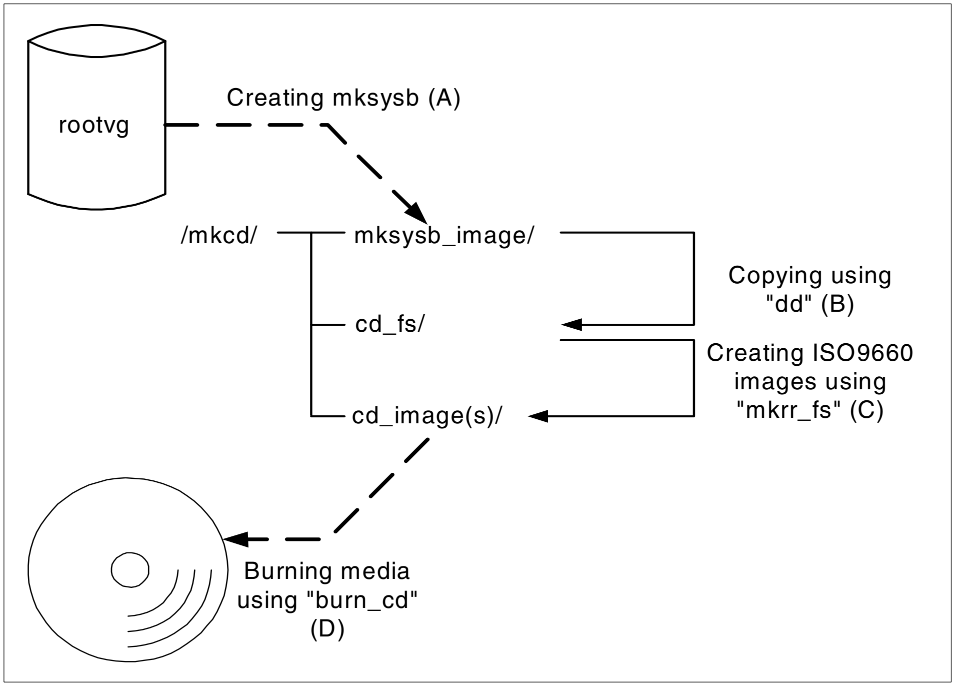


Figure 7-1 mkcd process

To burn the mksysb CD or DVD media, do the following:

1. Run **smitty** and select the following menus:

```
# smitty
    System Storage Management (Physical & Logical Storage)
        System Backup Manager
            Back Up the System
                Back Up This System to CD or DVD
```

2. If you already have a mksysb image file, select `yes`; otherwise, select `no` on the following panel:

```
                  Use an existing mksysb image?

    Move cursor to desired item and press Enter.

1 yes
2 no
```

   Then press Enter. You will see the panel shown in Example 7-1 on page 235, if you selected `yes` in the previous panel.

Example 7-3 Back up this system to CD or DVD

```
                        Back Up This System to CD or DVD

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

                                                        [Entry Fields]
  CD-R or DVD-R or DVD-RAM Device                     []                      +
  DVD sized image?                                     no                     +
* Location of existing mksysb image                   []                      /

  File system to store CD or DVD file structure       []                      /
     (If blank, the file system
       will be created for you.)

  File system to store final CD or DVD images         []                      /
     (If blank, the file system
       will be created for you.)

  If file systems are being created:
    Volume Group for created file systems             [rootvg]                +

  Advanced Customization Options:
  Do you want the CD or DVD to be bootable?            yes                    +
  Remove final images after creating CD or DVD?        yes                    +
  Create the CD or DVD now?                            yes                    +
  Install bundle file                                 []                      /
  File with list of packages to copy to CD or DVD     []                      /
  Location of packages to copy to CD or DVD           []                     +/
  Customization script                                []                      /
  User supplied bosinst.data file                     []                      /
  Debug output?                                        no                     +
  User supplied image.data file                       []                      /

F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         Esc+6=Command       Esc+7=Edit          Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do
```

3. Select the appropriate values and press Enter.

In this redbook, we used the **mkcd** command directly instead of this SMIT panel. In Example 7-4 on page 242, the following flags are used with the **mkcd** command to create a bootable mksysb DVD-RAM media:

-d /dev/cd0 Specifies the device file name for the DVD-RAM drive.

-L Instructs **mkcd** to create a big ISO9660 image file for DVD.

-e Instructs **mkcd** to use /etc/exclude.rootvg when creating the mksysb image.

-C /backup Specifies using the /backup file system to create the CD file system.

-M /backup/mksysb_image Specifies the /backup/mksysb_image directory to store the mksysb image.

-I /backup Specifies the /backup file system to store the ISO image file.

Example 7-4 Creating a mksysb on DVD-RAM

```
# cat /etc/exclude.rootvg
^./tmp/
^./backup/
# mkdir /backup/mksysb_image
# mkcd -d /dev/cd0 -L -e -C /backup -M /backup/mksysb_image -I /backup
Initializing mkcd log: /var/adm/ras/mkcd.log...
Verifying command parameters...
Creating image.data file...
Creating mksysb image...
Creating list of files to back up.
Backing up 27389 files...............
27389 of 27389 files (100%)
0512-038 mksysb: Backup Completed Successfully.
Populating the CD or DVD file system...
Copying backup to the CD or DVD file system...
Creating Rock Ridge format image...
Running mkisofs ...
mkrr_fs was successful.
Making the personal CD or DVD image bootable...
Writing the CD or DVD image to device: /dev/cd0...
Running readcd ...
Capacity: 2236704 Blocks = 4473408 kBytes = 4368 MBytes = 4580 prMB
Sectorsize: 2048 Bytes
burn_cd was successful.
```

In our environment, the command took roughly 30 minutes to complete; however, this figure depends on several factors, such as your mksysb image size and the I/O speed of the disk drives on your system. The following example shows the created DVD-RAM media contents:

```
# mount -o ro -v cdrfs /dev/cd0 /mnt
# cd /mnt
# df -kI .
Filesystem    1024-blocks      Used      Free %Used Mounted on
/dev/cd0           640224    640224         0  100% /mnt
# ls -l
```

```
total 92
-r--r--r--   1 root     system        10173 Mar 19 06:53 .files_not_found
-r--r--r--   1 root     system           17 Mar 19 06:53 OSLEVEL
dr-xr-xr-x   3 root     system         2048 Mar 19 06:53 RPMS/
-r--r--r--   1 root     system         8323 Mar 19 06:53 bosinst.data
-r--r--r--   1 root     system         8048 Mar 19 06:53 image.data
dr-xr-xr-x   3 root     system         2048 Mar 19 06:53 installp/
dr-xr-xr-x   3 root     system         2048 Mar 19 06:53 ismp/
-r--r--r--   1 root     system           62 Mar 19 06:54 mkcd.data
dr-xr-xr-x   3 root     system         2048 Mar 19 06:54 ppc/
dr-xr-xr-x   4 root     system         2048 Mar 19 06:53 root/
dr-xr-xr-x   3 root     system         2048 Mar 19 06:53 udi/
dr-xr-xr-x   8 root     system         2048 Mar 19 06:53 usr/
```

The **mkcd** command can process multiple volumes; you will be prompted to insert each volume in turn. Only the first media will be bootable.

You can also back up non-rootvg volume groups using **mkcd** by specifying the volume group name, as shown in Example 7-5. To do so, you have to specify the -v datavg flag of the **mkcd** command.

Example 7-5   Backing up non-rootvg volume groups using mkcd

```
# mkcd -d /dev/cd0 -C /backup -I -M /backup -L -e -v datavg
Initializing mkcd log: /var/adm/ras/mkcd.log...
Verifying command parameters...
Creating information file for volume group datavg.
Creating savevg image...

Creating list of files to back up.
Backing up 14 files
14 of 14 files (100%)
0512-038 savevg: Backup Completed Successfully.
Copying backup to the CD or DVD file system...
Creating Rock Ridge format image...
Running mkisofs ...
mkrr_fs was successful.
Writing the CD or DVD image to device: /dev/cd0...
Running readcd ...
Capacity: 2295072 Blocks = 4590144 kBytes = 4482 MBytes = 4700 prMB
Sectorsize: 2048 Bytes
burn_cd was successful.
```

7.1.3 Alternate disk install

Alternate disk installation is a system management function used in the environment that requires severe control of system availability and software level tracking. By using this function, the system administrator who has to manage the system under limited system maintenance time, can reduce the software maintenance time using an additional disk drive for system maintenance.

In alternate disk installation, you have basically two ways to select the installation image source (see in Figure 7-2). The first way is called *cloning*, which clones the rootvg that is currently running using the -C option. The second way is called *mksysb image install*, specifying the mksysb image file location using the -d option, installing it to the target disk. In both cases, the new volume group name will be alt_rootvg. When a system or a partition is booted from that disk, its name will be modified to rootvg. If you want to create more alternate install disk, you can use `-v` option of **`alt_disk_install`** command to avoid the conflict of volume group names.

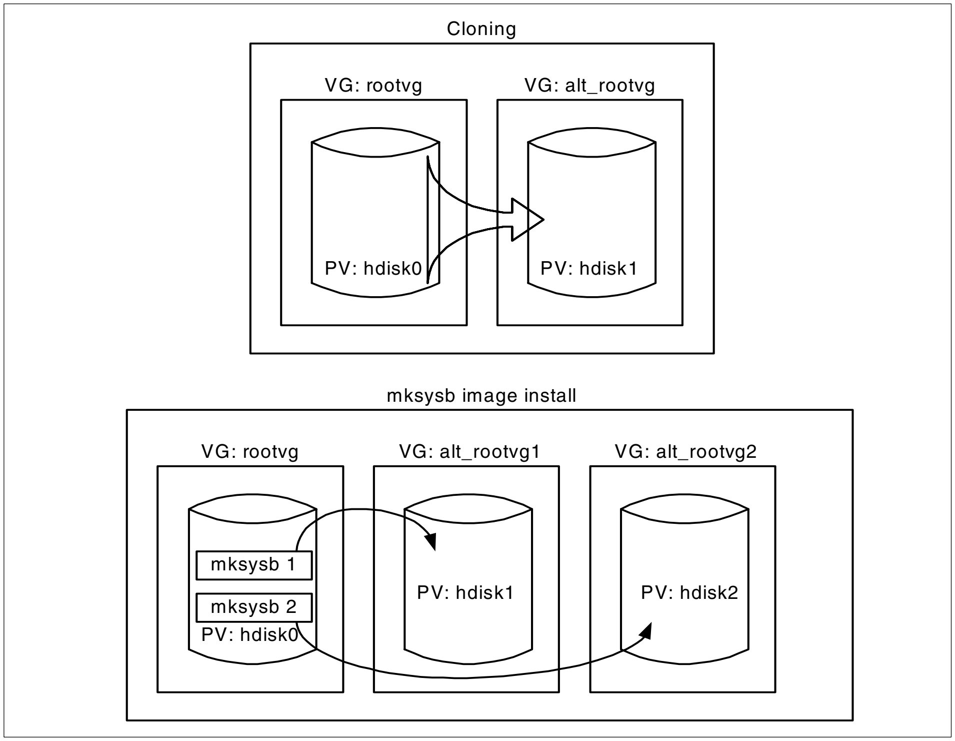


Figure 7-2 Alternate disk install

In both cases, the current running operating system does not have to be shut down while installing the image. You can also customize the target volume group before you reboot from the target disk. After the alternate disk installation is finished, you can switch the boot disk. As a result, you can reduce the system maintenance time for system customization and software level updates.

For further information about the AIX alternate disk installation function, please refer to *AIX 5L Version 5.1 Reference Documentation: Commands Reference*.

Alternative disk install is a very powerful facility, allowing you to easily create on-line backups of your existing operating system. This backup could be used for contingency when a new change is applied, or you could apply fixes to the new clone for testing purposes–with an easy fallback. Whichever method you use, once a clone is created, you effectively have two copies of AIX on two different disks to boot up.

The alternate disk installation function requires that the two filesets bos.alt_disk_install.boot_images and bos.alt_disk_install.rte be installed.

Example 7-6 shows the use of the **alt_disk_install** command performing a cloning of a running rootvg on hdisk0 to the unused hdisk2.

Example 7-6 Cloning the rootvg using alt_disk_install

```
# lspv
hdisk0          000c91ade17a26ab                     rootvg
hdisk1          000c91ad9ae45533                     None
hdisk2          000c91ad9aebadac                     None
# alt_disk_install -C hdisk2
Calling mkszfile to create new /image.data file.
Checking disk sizes.
Creating cloned rootvg volume group and associated logical volumes.
Creating logical volume alt_hd5.
Creating logical volume alt_hd6.
Creating logical volume alt_hd8.
Creating logical volume alt_hd4.
Creating logical volume alt_hd2.
Creating logical volume alt_hd9var.
Creating logical volume alt_hd3.
Creating logical volume alt_hd1.
Creating logical volume alt_hd10opt.
Creating /alt_inst/ file system.
Creating /alt_inst/home file system.
Creating /alt_inst/opt file system.
Creating /alt_inst/tmp file system.
Creating /alt_inst/usr file system.
Creating /alt_inst/var file system.
Generating a list of files
for backup and restore into the alternate file system...
```

```
Backing-up the rootvg files and restoring them to the alternate file system...
Modifying ODM on cloned disk.
Building boot image on cloned disk.
forced unmount of /alt_inst/var
forced unmount of /alt_inst/usr
forced unmount of /alt_inst/tmp
forced unmount of /alt_inst/opt
forced unmount of /alt_inst/home
forced unmount of /alt_inst
forced unmount of /alt_inst
Changing logical volume names in volume group descriptor area.
Fixing LV control blocks...
Fixing file system superblocks...
Bootlist is set to the boot disk: hdisk2
# lspv
hdisk0          000c91ade17a26ab                    rootvg
hdisk1          000c91ad9ae45533                    None
hdisk2          000c91ad9aebadac                    altinst_rootvg
```

Notice that the volume group name of hdisk2 is changed from None to altinst_rootvg. To boot from the cloned disk, hdisk2, you should alter the boot list of the system and reboot the system using the following commands:

```
# bootlist -m normal hdisk2
# shutdown -Fr now
```

After rebooting, you will see the following **lspv** command output:

```
# lspv
hdisk0          000c91ade17a26ab                    old_rootvg
hdisk1          000c91ad9ae45533                    None
hdisk2          000c91ad9aebadac                    rootvg
```

Now the volume group name of hdisk2 becomes rootvg and the one of hdisk0 becomes old_rootvg. If you want to return to the original rootvg, you can simply alter the boot list with hdisk0 and reboot.

> **Note:** To delete the old_rootvg volume group definition from the system, issue **alt_disk_install -X old_rootvg**. If you export this volume group using the **exportvg** command, you may corrupt the ODM definition on your system.

You can apply updates to the cloned disk at the time the clone is created. In Example 7-7 on page 247, we apply the APAR IY24217, which resides in /mnt, to the target disk when the operating system is cloned. If you reboot the system, it boots up from the target disk with the APAR applied. If you have to go back to the software level status not applied with this APAR, you can easily boot the system from the original disk, altering the boot list held in NVRAM of the system.

Example 7-7 Using alt_disk_install to cloned disk with updates

```
# alt_disk_install -C -F IY24217 -l /mnt hdisk2
Calling mkszfile to create new /image.data file.

... Many lines are removed on purpose ...

+-----------------------------------------------------------------------------+
                    Pre-installation Verification...
+-----------------------------------------------------------------------------+
Verifying selections...done
Verifying requisites...done
Results...

SUCCESSES
---------
  Filesets listed in this section passed pre-installation verification
  and will be installed.

  Mandatory Fileset Updates
  -------------------------
  (being installed automatically due to their importance)
  bos.rte.install 5.1.0.25                    # LPP Install Commands

  << End of Success Section >>

FILESET STATISTICS
------------------
    1  Selected to be installed, of which:
        1  Passed pre-installation verification
  ----
    1  Total to be installed

+-----------------------------------------------------------------------------+
                         Installing Software...
+-----------------------------------------------------------------------------+

... Many lines are removed on purpose ...

bos.net.nfs.client          5.1.0.25        ROOT        COMMIT      SUCCESS
bos.perf.diag_tool          5.1.0.25        USR         COMMIT      SUCCESS
bos.sysmgt.serv_aid         5.1.0.25        USR         COMMIT      SUCCESS
bos.sysmgt.serv_aid         5.1.0.25        ROOT        COMMIT      SUCCESS
forced unmount of /alt_inst/tmp_install
Modifying ODM on cloned disk.
Building boot image on cloned disk.
forced unmount of /alt_inst/var
forced unmount of /alt_inst/usr
forced unmount of /alt_inst/tmp
forced unmount of /alt_inst/rml
forced unmount of /alt_inst/opt
forced unmount of /alt_inst/home
forced unmount of /alt_inst
forced unmount of /alt_inst
Changing logical volume names in volume group descriptor area.
Fixing LV control blocks...
Fixing file system superblocks...
Bootlist is set to the boot disk: hdisk2
```

Once alternate disk installation is done, the newly created volume group, altinst_rootvg, is automatically varied off on the target disk. Therefore, you have to vary on altinst_rootvg to access the file systems in the volume group. To vary on altinst_rootvg, the **alt_disk_install** command provides the -W flag. If you issue the following command, then altinst_rootvg is varied on (woken-up) so that you can access the file system in altinst_rootvg:

```
# alt_disk_install -W hdisk2
# lsvg -o
rootvg
altinst_rootvg
```

Once woken-up, all the file systems in altinst_rootvg are mounted under the mount point prefix /alt_inst. For example, the /usr file system in the woken-up altinst_rootvg will be mounted on /alt_inst/usr.

To vary off (put to *sleep*) a woken-up volume group, you have to use the -S flag of the **alt_disk_install** command, as shown in the following example:

```
# alt_disk_install -S
# lsvg -o
rootvg
```

For further information about alternate disk Installation, please refer to AIX 5L Version 5.1 *Reference Documentation: Commands Reference*.

7.1.4 Network Installation Manager (NIM)

The AIX Network Installation Manager (NIM) is a function that allow you to install AIX over the network. NIM provides the ability to install AIX on a pSeries machine from a server called the NIM master over the network. NIM can install and maintain not only the AIX operating system, but also any additional software and fixes. NIM also allows you to customize the configuration of machines both during and after installation, such as a host name and TCP/IP addresses. NIM eliminates the need for access to physical media, such as tapes and CD-ROMs, because the installation images are stored on a NIM server as resources.

In this section, we briefly outline the simple NIM installation steps:

1. Configure the NIM master.
2. Create a mksysb image from your master system.
3. Transfer the mksysb image to the NIM master.
4. Define the mksysb image as a resource to NIM and allocate it to a new client.
5. Power on the new machine by specifying a network adapter as the boot device

Configure the NIM master

Although NIM configuration can be complex due to the rich functions provided by NIM, we used the **nim_master_setup** command in this section to quickly set up the minimum NIM master configuration. The command can take a while to run, as it has to build all the NIM resources shown in Table 7-1. You will be prompted to change the AIX 5L Version 5.1 media while the command is running.

Please make sure that you have enough free space in rootvg before running **nim_master_setup**. It will automatically create two new file systems, /tftpboot and /export/nim in rootvg; these will require approximately 4.5 GB of space. Also, if you have disabled tftp, bootp, and NFS for security reasons, you will need to re-enable them for NIM. This is another reason a build zone is recommended.

Table 7-1 Nim resources defined by the nim_master_setup command

Resource name	Description
boot	The network boot image.
nim_script	NIM customization script.
generic_sysb	Generic mksysb image.
resolv_res	The /etc/resolv.conf file to be used.
bid_tty_ow	bosinst.data for the installation using a serial console. Instructs AIX on how to set the parameters upon installation, for example, locale settings, disks to use, and the type of install.
bid_lft_ow	bosinst.data for the installation using the lft device.
510lpp_res	AIX 5L Version 5.1 installable package files.
510spot_res	AIX 5L Version 5.1 spot or root file system to be used during install.

Creating your master mksysb image file

To create a mksysb image file on the system you wish to clone, use the **mksysb** command, as shown in Example 7-8 on page 250.

Example 7-8 Creating a mksysb image file

```
# mksysb -ie /backup/clone01.mksysb

Creating information file (/image.data) for rootvg..
0512-039 mksysb: WARNING: /backup/clone01.mksysb does not appear to be a tape
device and will NOT have a bootable image.

Creating list of files to back up.
Backing up 27388 files...............
27388 of 27388 files (100%)
0512-038 mksysb: Backup Completed Successfully.
0512-040 mksysb: WARNING: /backup/clone01.mksysb does not appear to be a tape
device and does NOT have a bootable image.
```

Then transfer the mksysb image file to the NIM master using **ftp** or **scp** and place it in the /export/nim file system.

Defining NIM resources

On the NIM master, do the following steps:

1. Define the template boinst_data resource: boinst_mksysb

   In this example, the bid_tty_mksysb file is modified from the default bid_tty_ow file to enable unattended installation:

   ```
   # nim -o define -t bosinst_data -a server=master\
       -a location=/export/nim/bid_tty_mksysb bosinst_mksysb
   ```

2. Define the NIM client resource svr03:

   ```
   # nim -o define -t standalone -a netboot_kernel=mp\
       -a platform=chrp -a cable_type1=tp\
       -a if1="master_net svr03 00045576ABDD" svr03
   # nim -o reset svr03
   ```

   where svr03 is the host name of new server you are going to install, and 00045576ABDD is the MAC address of the integrated Ethernet adapter of this server.

3. Deallocate the script resources if it is allocated to the client:

   ```
   # nim -o deallocate -a bosinst_data=bosinst_template svr03
   ```

4. Define the mksysb image resource svr03_mksysb_res:

   ```
   # nim -o define -t mksysb -a location=/export/nim/clone01.mksysb\
       -aserver=master svr03_mksysb_res
   ```

5. Allocate four resources - mksysb, bosinst_data, spot, and lppsource to the NIM client:

   ```
   # nim -o allocate -a mksysb=svr03_mksysb_res\
   ```

```
    -a bosinst_data=bosinst_mksysb -a spot=510spot_res\
    -a lpp_source=510lpp_res svr03
```

Verify the resources allocation status using **lsnim -l svr03**, as shown in the following example:

```
# lsnim -l svr03
svr03:
   class          = machines
   type           = standalone
   platform       = chrp
   netboot_kernel = mp
   if1            = master_net svr03 00045576ABDD
   cable_type1    = tp
   Cstate         = BOS installation has been enabled
   prev_state     = ready for a NIM operation
   Mstate         = currently running
   boot           = boot
   bosinst_data   = bosinst_mksysb
   lpp_source     = 510lpp_res
   mksysb         = svr03_mksysb_res
   nim_script     = nim_script
   spot           = 510spot_res
   control        = master
```

Perform the installation operation

When powering on the new machine, specify a network adapter as the boot device using the SMS menu. Then, perform the installation operation on the NIM master using the following command:

```
# nim -o bos_inst -asource=mksysb -a boot_client=no svr03
```

Use **lsnim -l svr03** to monitor progress of the install.

For further detailed information about NIM, please refer to the following publications:

- *AIX 5L Version 5.1 Network Installation Management Guide and Reference*
- *NIM: From A to Z in AIX 4.3*, SG24-5524

7.1.5 Backing up user data

A backup is one of the most critical administration tasks that must be performed on a daily basis in a server farm environment. In the event of a disaster, your backup will be required to restore the system and the service. There are many different types of backup methods you can deploy in a server farm environment. One efficient method is to use Tivoli Storage Manager, which is a software product to back up user data over the network, as well as over the SAN.

For detailed information about user data backup methods, please refer to the following publications:

- *Tivoli Storage Manager Version 3.7: Technical Guide*, SG24-5477
- *IBM Certification Study Guide AIX Installation and System Recovery*, SG24-6183

7.2 Software level maintenance

When installing any new AIX systems, you should ensure that you are installing at the latest fileset level of AIX. However, you have to verify whether the software you plan to run on the new system works with the latest fileset level of AIX.

The only way to do this is through testing on a non-production test system. Within your farm, you should have some test bed set up for testing new software and new maintenance levels of AIX. You should also have a maintenance window; this is a period of time you may reserve to perform hardware maintenance on production systems within the farm. This could be to replace failing disks, upgrade firmware, or simply reboot. This window could be a few hours every night or once a week.

7.2.1 Distributing software

There are many methods of distributing software within a server farm. In this section, we explain the software distribution method using the **scp** command, explained in Section 4.3.6, “Using the scp command” on page 149.

Each AIX system should have the /swdist file system. You can distribute software upgrades, patches, or firmware from the central administration server using **scp**. This task can be easily automated using scripts. Example 7-9 shows the distribution of a fileset update to multiple systems from the central administration server. All you need to do is apply the software on each system. You might consider setting up a script on the remote server to monitor for incoming software–this script could then automatically install the software for you. As shown in “User defined executives” on page 195, you can use NET-SNMP for this purpose. Be aware that some software updates require rebooting after installation.

Example 7-9 Distributing software using scp

```
$ ls -l /swdist
total 360
-rw-r--r--   1 root     system       181248 Mar 29 14:28 U476346.bff
$ for svr in svr01 svr02 svr03 svr04
> do
> scp /swdist/U476346.bff $svr:/swdist
> done
U476346.bff          100% |*****************************|   177 KB    00:00
U476346.bff          100% |*****************************|   177 KB    00:00
U476346.bff          100% |*****************************|   177 KB    00:00
U476346.bff          100% |*****************************|   177 KB    00:00
```

You should never apply a major upgrade or fix to a production system without first testing it and creating a backup.

7.3 Network management

The network in a server farm environment should be highly available. AIX 5L Version 5.1 provides several advanced features that contribute to the network availability. Where possible, there should be multiple routes, and all routes should be used to provide redundancy and performance.

7.3.1 EtherChannel

EtherChannel is a network link aggregation technology that allows you to create a single large pipe by combining the bandwidth of multiple Ethernet adapters, as shown in Figure 7-3.

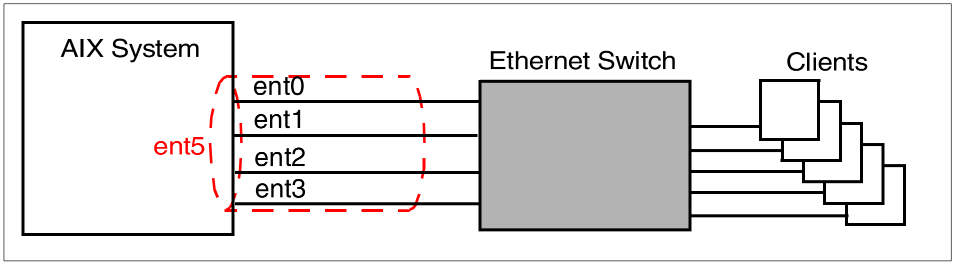


Figure 7-3 EtherChannel concept

EtherChannel is a name used by Cisco to express the network aggregation technology on the Ethernet (other network venders might use a different term to express this function). For detailed network technical information about EtherChannel, please visit the following URL:

http://www.cisco.com/pcgi-bin/Support/PSP/psp_view.pl?p=Internetworking:Etherchannel

EtherChannel not only provides you the network bandwidth increase, but also redundancy.

In AIX 5L Version 5.1, you can create an EtherChannel that can use up to eight[4] physical Ethernet adapters.

In order to configure the EtherChannel function on your system, you must have:

- An Ethernet switch that supports the EtherChannel function

  Currently, the following Cisco Ethernet switch models support[5] the EtherChannel function:

  - Catalyst 4000/5000/6000/2900XL/3500XL Series
  - Catalyst 2950/3550, 1900/2800, 2948G-L3/4908G-L3, and 8500

  Please note that some older Cisco Ethernet switches do *not* support more than four ports EtherChannel. You should consult with your network device support vendor.

- Multiple 10/100 Mbps Ethernet adapters or Gigabit Ethernet adapters that support the EtherChannel function

  In AIX 5L Version 5.1, the Etherchannel function is mainly implemented by the Ethernet device drivers included in the following filesets:

  - devices.pci.14100401.rte
  - devices.pci.23100020.rte

  Therefore, the following Ethernet adapters are supported in addition to the integrated Ethernet port that is configured by the same device fileset devices.pci.14100401.rte:

  - IBM 10/100 Mbps Ethernet PCI Adapter (FC 2968)
  - IBM Universal 4-port 10/100 Ethernet Adapter (FC 4961)
  - Ethernet/LAN Encryption 10/100BaseT (FC 4962)
  - Gigabit Ethernet-SX PCI Adapter (FC 2969)
  - 10/100/1000 Base-T Ethernet PCI Adapter (FC 2975)

  > **Note:** EtherChannel, especially if you use multiple Gigabit Ethernet adapters, requires high I/O bandwidth of PCI buses on your AIX system.

We explain how to configure EtherChannel on AIX 5L Version 5.1 along with Cisco Catalyst 6509 in this section. Please note that the configuration step shown here highly depends on the individual Ethernet switch model as well as its firmware level. Please consult with the appropriate publications for the Ethernet switch device you are using.

[4] You must have the APAR IY26314 applied to increase the supported link number of EtherChannel from four to eight on AIX 5L Version 5.1.

[5] This list is derived from http://www.cisco.com/warp/public/473/4.html.

Configure EtherChannel on Ethernet switch

On Cisco Catalyst 6509, you can define the EtherChannel function using one of the following distribution policies:

- MAC address (source/destination/both)
- IP address (source/destination/both)
- Layer 4 port number

In our example, we chose MAC address (both) for the distribution policy of EtherChannel.

To configure EtherChannel on Cisco Catalyst 6509, do the following:

1. Confirm the physical Ethernet port number on which you are going to define Ethernet Channel.
2. Access the console.
3. Set the enable mode.
4. Verify the current EtherChannel configuration shown in the following example:

```
6509-hub> (enable) show port channel
No ports channeling
```

5. Disable the physical Ethernet ports to be configured as a part of EtherChannel, as shown in the following example:

```
6509-hub> (enable) set port disable 3/1,3/3,3/5,3/7
2001 Nov 14 22:14:42 %PAGP-5PORTFROMSTP:Port 3/41 left bridge port 3/1
... many lines erased on purpose...
```

6. Enable EtherChannel mode:

```
6509-hub> (enable) set port channel 3/1,3/3,3/5,3/7 on
Port(s) 3/1,3/3,3/5,3/7 are assigned to admin group 339.
Port(s) 3/1,3/3,3/5,3/7 channel mode set to on.
```

7. Verify the current port status:

```
6509-hub> (enable) show port channel
Port    Status   Channel             Admin  Ch
                 Mode                Group  Id
----- ----------- ------------------ ----- ------
 3/1  disabled    on                   339    839
 3/3  disabled    on                   339    839
 3/5  disabled    on                   339    839
 3/7  disabled    on                   339    839
----- ----------- ------------------ ----- ------

Port  Device-ID                     Port-ID           Platform
----- ------------------------------ ----------------- ------------
 3/1
 3/3
 3/5
 3/7
----- ------------------------------ ----------------- ------------
```

8. Enable ports:

```
6509-hub> (enable) set port enable 3/1,3/3,3/5,3/7
Port(s) 3/1,3/3,3/5,3/7 enabled.
```

9. Set the distribution policy:

```
6509-hub> (enable) set port channel all distribution MAC both
Port(s) 3/1,3/3,3/5,3/7 enabled.
```

Configure EtherChannel on AIX

1. Run **smitty** and select the following SMIT panels:

```
# smitty
    Devices
         Communication
            Etherchannel
                Add An Etherchannel
```

2. Select Ethernet adapters to be configured as a part of EtherChannel on the panel, as shown in Example 7-10, then press Enter. In this example, we selected ent0, ent1, ent2, and ent3.

Example 7-10 Selecting Ethernet adapters

```
  x                      Available Network Interfaces                          x
  x                                                                            x
  x Move cursor to desired item and press Esc+7.                               x
  x     ONE OR MORE items can be selected.                                     x
  x Press Enter AFTER making all selections.                                   x
  x                                                                            x
  x > ent0                                                                     x
  x > ent1                                                                     x
  x > ent2                                                                     x
  x > ent3                                                                     x
  x                                                                            x
  x F1=Help                 F2=Refresh              F3=Cancel                  x
  x Esc+7=Select            Esc+8=Image             Esc+0=Exit
F1x Enter=Do                /=Find                  n=Find Next
```

3. You will see the SMIT panel as shown in Example 7-13 on page 260:

Example 7-11 Add an EtherChannel

```
                              Add An Etherchannel

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

```
                                                        [Entry Fields]
  Etherchannel Adapters                               ent0 ent1 ent2 ent3     +
  Enable ALTERNATE ETHERCHANNEL address               no                      +
  ALTERNATE ETHERCHANNEL address                     []                       +
  Mode                                                standard                +
  Enable GIGABIT ETHERNET JUMBO frames                no                      +
  Internet Address to Ping                           []
  Number of Retries                                  []                       #
  Retry Timeout (sec)                                []                       #

F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         Esc+6=Command       Esc+7=Edit          Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do
```

In this panel, you should select the following two values for the Mode:

standard Instructs the channel to send the packets to the adapter based on a hash of the destination IP address. The modulus (remainder) of the fourth byte of the destination IP address and number of adapters defined in a channel will determine the adapter to be actually used. For example, if two adapters, ent3 and ent2, are defined in the EtherChannel interface ent4[6], and a packet to be sent to the destination IP address 10.10.10.11, which produces a modulus 1 (= 11 % 2), then the packet will be sent using the second adapter ent2 (the index is starting from 0). The standard mode guarantees that the packets are sent over the network in the same order that they were given to the EtherChannel, but it does not necessarily make full use of the bandwidth. For example, if there are four clients whose last bye hashes to the same number, all the traffic will be sent over the same adapter.

round_robin Instructs the channel to sends the packets in a round-robin fashion to the adapters. In our example, the packets will be sent to adapters with the orders of ent0, ent1, ent2, ent3, and ent0. The round_robin mode does not guarantee that the packets are sent over the network in the same order that they were given to the EtherChannel; however, it makes the best use of the bandwidth (because all the adapters are used equally).

Then press Enter.

[6] In this context, we assume that the ent4 adapter is defined as **mkdev -c adapter -s pseudo -t ibm_ech -a adapter_names=ent3,ent2**.

4. The EtherChannel network adapter ent4 is defined, as shown in the following example:

```
# lsdev -Cc adapter | grep ent
ent0    Available 10-80    IBM 10/100 Mbps Ethernet PCI Adapter (23100020)
ent1    Available 10-88    IBM 10/100 Mbps Ethernet PCI Adapter (23100020)
ent2    Available 10-70    IBM 10/100 Mbps Ethernet PCI Adapter (23100020)
ent3    Available 20-58    IBM 10/100 Mbps Ethernet PCI Adapter (23100020)
ent4 Available          Etherchannel
```

Now you can use the newly created network adapter ent4.

Changing the link status mechanism

Some Ethernet adapters, such as the 10/100 Ethernet TX PCI Adapter (FC 2968), cannot detect link status failure automatically; instead, they employ a link polling mechanism that is disabled by default. If this polling mechanism is disabled, then the adapter will not detect that the link has failed, and thus will never switch over. To enable this polling mechanism so that the adapter can detect a change in its link status, do the following:

1. Run **smitty** and select the following SMIT panels:

```
# smitty
    Devices
         Communication
            Ethernet Adapter
                Adapter
                    Change / Show Characteristics of an Ethernet Adapter
```

2. Select the physical adapter to be changed the link poling mechanism, then press Enter. You will see the SMIT panel shown in Example 7-12.

Example 7-12   Change the link polling mechanism

```
              Change / Show Characteristics of an Ethernet Adapter

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

                                                       [Entry Fields]
  Ethernet Adapter                                   ent0
  Description                                        IBM 10/100 Mbps Ethern>
  Status                                             Available
  Location                                           10-60
  TRANSMIT queue size                               [16384]            +#
  HARDWARE RECEIVE queue size                       [256]              +#
  RECEIVE buffer pool size                          [2048]             +#
  Media Speed                                        Auto_Negotiation  +
  Inter-Packet Gap                                  [96]               +#
  Enable ALTERNATE ETHERNET address                  no                 +
  ALTERNATE ETHERNET address                        [0x000000000000]    +
  Enable Link Polling                                yes              +
  Time interval for Link Polling                    [500]             +#
  Apply change to DATABASE only                      no               +

F1=Help            F2=Refresh         F3=Cancel          F4=List
Esc+5=Reset        Esc+6=Command      Esc+7=Edit         Esc+8=Image
Esc+9=Shell        Esc+0=Exit         Enter=Do
```

3. Change the Enable Link Polling value to `yes` and press Enter.

7.3.2 Network interface backup

You can define an additional network adapter, called network interface backup, to the working (original) network adapter on the system. The network adapter defined as network interface backup is used when the original network adapter fails. The original and backup adapter will comprise a virtual network adapter. For example, if you configure a network interface backup using ent1 to the original network adapter ent0, then you will see the virtual network adapter ent2 will be created, as shown in Figure 7-4.

> **Note:** You should not confuse this function with dead gateway support, which operates at the IP layer, while network interface backup operates at the link level layer.

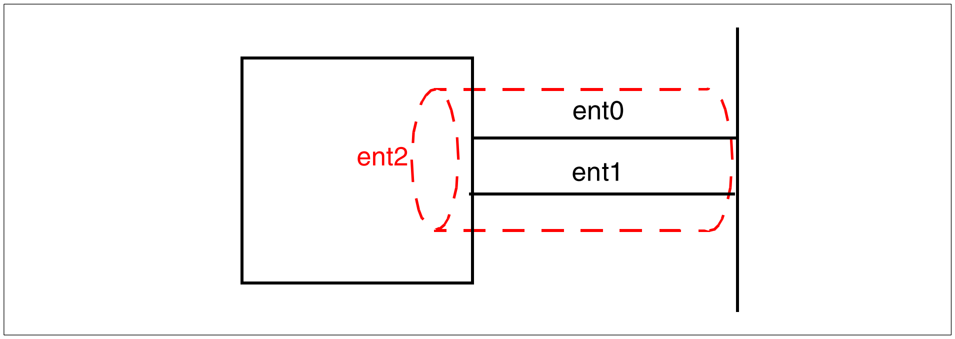


Figure 7-4 Network interface backup

To configure network interface backup, do the following:

> **Note:** Although the SMIT panel says EtherChannel in the title line, you do not require Ethernet switch that supports EtherChannel to use this function.

1. Run **smitty** and select the following SMIT panels:

   ```
   # smitty
       Devices
            Communication
               Etherchannel
                   Add An Etherchannel
   ```

2. Select both the original and the adapter to be configured as an network interface backup, then press Enter. In our example, we selected ent0 and ent1. You will see the SMIT panel shown in Example 7-13.

Example 7-13 Configuring network interface backup in smit

```
                              Add An Etherchannel

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

                                                         [Entry Fields]
  Etherchannel Adapters                                 ent0 ent1             +
  Enable ALTERNATE ETHERCHANNEL address                 no                    +
  ALTERNATE ETHERCHANNEL address                       []                     +
  Mode                                                  netif_backup          +
  Enable GIGABIT ETHERNET JUMBO frames                  no                    +
  Internet Address to Ping                             []
  Number of Retries                                    []                     #
  Retry Timeout (sec)                                  []                     #

F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         Esc+6=Command       Esc+7=Edit          Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do
```

3. Change the Mode to netif_backup, then press Enter. Then the virtual network interface ent2 will be created, as shown in Example 7-14.

Example 7-14 The new standby interface

```
# lsdev -Cc adapter | grep ent
ent0    Available 10-80    IBM 10/100 Mbps Ethernet PCI Adapter (23100020)
ent1    Available 10-88    IBM 10/100 Mbps Ethernet PCI Adapter (23100020)
ent2 Available          Etherchannel
```

> **Note:** Although the new adapter appears as an EtherChannel adapter in Example 7-14 on page 260, it is *not* a true EtherChannel channel adapter.

Should the current (active) adapter fail (or the cable or switch), the backup adapter will automatically be used. It will continue to use this adapter until it fails, then will fall back to the original adapter. You may also see an error logged into the system log by the real adapter, reporting its link is down.

The Internet Address to Ping field in Example 7-13 on page 260 is not a mandatory field. If you specify an IP address in this field, it instructs the system to use the link status from the network itself in addition to the link status from the Ethernet adapters. In this case, you can use two Ethernet switches to achieve additional availability, as shown in Figure 7-5.

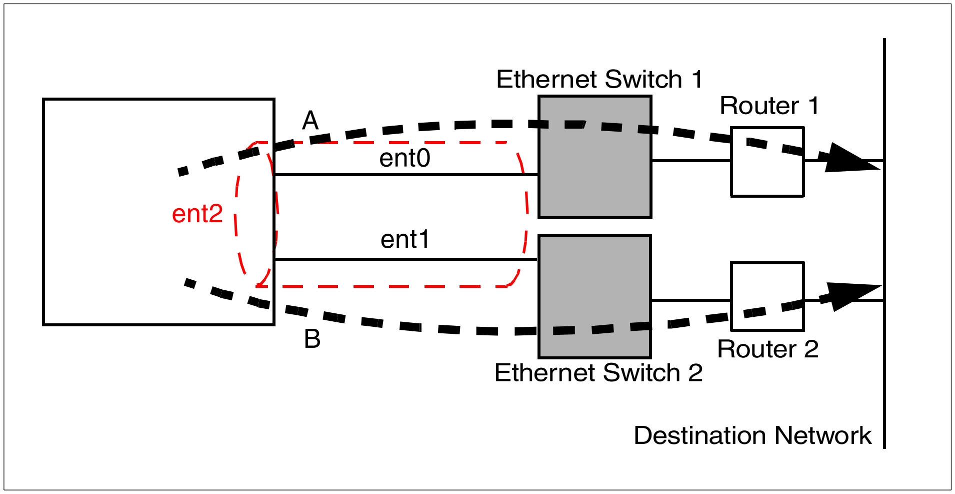


Figure 7-5 Network interface backup using two Ethernet switches

In this example, you can set the IP address of router 1 in the Internet Address to Ping field in Example 7-13 on page 260 to access the destination network. If switch 1 or router 1 should fail, the backup adapter ent1 is used to access the destination network. Although you have to define multipath routes shown as A and B, destination hosts still see the same source IP address defined on the ent2 adapter upon failover. For further detailed information about multipath route, see Section 7.3.3, “Multipath routing” on page 261.

Note: You do not have to change the polling mechanism (see “Changing the link status mechanism” on page 258) on the adapters to be used as a part of the network interface backup. The device driver automatically enables this mechanism.

7.3.3 Multipath routing

You can specify two or more routes to the same destination on AIX 5L Version 5.1 to achieve the following advantages:

- Load balancing between routers and firewalls
- Load balancing between network interfaces or even networks

- Redundancy when used in conjunction with dead gateway support

> **Note:** Before using multipath routing we recommend disabling udp_pmtu_discover and tcp_pmtu_discover. These are disabled using the **`no`** command. Add the following lines at the end of the /etc/rc.net file:
>
> ```
> no -o udp_pmtu_discover=0
> no -o udp_pmtu_discover=0
> ```

Figure 7-6 illustrates the concept of multipath routing.

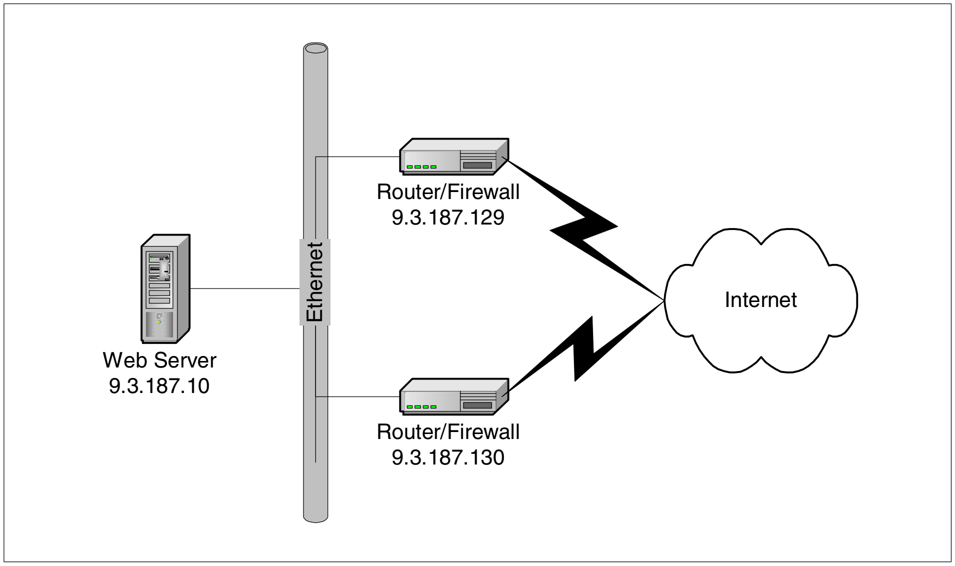


Figure 7-6 Sample network with dual routers

To configure the multipath route, use the **`route`** command, as shown in Example 7-15 on page 263. In this example, we configured this function as follows:

1. The **`netstat`** command is run first to show the existing default gateway. Originally, there is only one defined.
2. The **`route`** command is then run to add the second gateway to the routing table.
3. The **`netstat`** command is then re-run to show the new entry in the routing table. Notice the => sign, indicating it is a multipath route.

Once this route is initialized, AIX will use a round robin algorithm to route packets to the remote destination network.

Example 7-15 Adding dual default gateways

```
# netstat -rnC
Routing tables
Destination      Gateway           Flags   Refs     Use  If   Cost Config_Cost

Route Tree for Protocol Family 2 (Internet):
default          9.3.187.129       UG        0        0  en0     0          0
9.3.1.114        9.3.187.129       UGHW      1      437  en0     0          0
9.3.187.128/25   9.3.187.232       U         6     1206  en0     0          0
9.3.240.68       9.3.187.129       UGHW      1      178  en0     0          0
127/8            127.0.0.1         U         2      379  lo0     0          0

Route Tree for Protocol Family 24 (Internet v6):
::1              ::1               UH        0        0  lo0     0          0
# route add default 9.3.187.130
9.3.187.130 net default: gateway 9.3.187.130
# netstat -rnC
Routing tables
Destination      Gateway           Flags   Refs     Use  If   Cost Config_Cost

Route Tree for Protocol Family 2 (Internet):
default          9.3.187.129       UG        0        0  en0     0          0 =>
default          9.3.187.130       UG        0        0  en0     0          0
9.3.187.128/25   9.3.187.232       U         7     1206  en0     0          0
127/8            127.0.0.1         U         2      379  lo0     0          0

Route Tree for Protocol Family 24 (Internet v6):
::1              ::1               UH        0        0  lo0     0          0
```

Note: The **route** command only adds a temporary route. If the system is rebooted the new route will be lost. To make it permanent, you could either add the **route** command to /etc/rc.net, use **chdev** to modify the inet0 device, or use **smit mkroute**.

7.3.4 Dead gateway detection

In Example 7-15, we demonstrated how to add another gateway to the routing tables. Now suppose one of these gateways failed? AIX has the capability, called dead gateway detection (DGD), to automatically de-activate a route, and use the alternative gateway. There are two methods of DGD: active DGD and passive DGD. For availability purposes, we are going to use active DGD as follows:

```
# route add default 9.3.187.131 -active_dgd
```

If the gateway fails, as shown in, the cost to that route is increased, so no packets will be sent via that gateway. AIX will continue to ping the gateway, and should it become available again at a future time, the cost of the route will be reduced, and packets will be sent by the gateway again.

Example 7-16 Failed gateway

```
# netstat -rnC
Routing tables
Destination      Gateway           Flags   Refs     Use  If   Cost Config_Cost

Route Tree for Protocol Family 2 (Internet):
default          9.3.187.129       UG        1       13  en0     0          0 =>
default          9.3.187.130       UGA       0        0  en0   MAX          0
9.3.187.128/25   9.3.187.232       U         4     1223  en0     0          0
127/8            127.0.0.1         U         0      379  lo0     0          0

Route Tree for Protocol Family 24 (Internet v6):
::1              ::1               UH        0        0  lo0     0          0
```

> **Note:** Active DGD periodically sends an ICMP echo request, used by the **ping** command, to each gateway to determine if it is available (by default, every five seconds). If your gateway is a firewall, make sure it responds to these pings or AIX will believe the gateway is down and not use that route. The frequency of these pings can be modified by adjusting the dgd_ping_time using the **no** command. If you do adjust the parameter, remember to add the **no** command to /etc/rc.net

7.4 Log file management

There are many log files and other types of files you need to view and perform some action on in AIX. The following section will describe examples of areas to monitor and actions you can take.

7.4.1 Directories to be monitored

There are many directories, some of which are shown in Table 7-2, where files are kept; these files are often either temporary or log files–they can build up and need to be purged.

Table 7-2 Directories for temporary files

Directory	Description	Action
/tmp, /var/tmp, and /var/preserve	Temporary files	Delete any file older than seven days.

Directory	Description	Action
/var/spool/mqueue	Outgoing mail	Delete any older than seven days.
/var/spool/mail	User’s mail	Delete any file not modified for 14 days.
/var/spool/qdaemon, /usr/lib/lpd/qdir	Printer subsystem requests	Delete any file older than seven days.
/var/lp/logs	Printer subsystem logs	Delete any file older than seven days.
/usr/HTTPServer/logs	IBM HTTP Server logs	Delete any log file older than 14 days (not the httpd.pid file!).

7.4.2 Files to be monitored

Table 7-3 describes some of the common files that will increase in size and should be monitored. Next to each file is a description of their purpose and the recommended action.

Table 7-3 Files to be monitored

File	Description	Action
core and snapcore	Application core dumps. Can be useful for problem determination.	Delete after two days.
nohup.out	Output from **nohup** command.	Delete after two days.
.xerrors	Output from X11.	Truncate.
mbox	Users mail box of read mail.	Truncate; personal mail should not be kept on server in server farm.
smit.log and smit.script	Logs from SMIT utility.	Keep a minimum 0.5 MB of roots, last 1000 lines of others.
/var/adm/wtmp	Records of users logging in.	Keep a minimum 60 days worth of data, delete the rest. (This is a binary file).
/etc/security/failedlogin	Records users failing to login.	Keep a minimum 60 days worth of data, delete the rest. (This is a binary file)

File	Description	Action
/var/adm/sulog	Logs of users running the **su** command.	Keep a minimum 60 days worth of data, delete the rest.
/var/adm/cron/log	cron log.	Truncate; keep last 10 lines.
/var/tmp/snmpd.log	SNMP daemon log file.	Truncate; keep last 100 lines.
/var/tmp/dpid.log /var/tmp/dpid2.log /var/tmp/hostmidb.log /var/tmp/muxatmd	SNMP subsystem logs.	Truncate; keep last 100 lines.
dead.letter	Aborted mail.	Delete.
trcfile	Output from trace utility	Keep for two days, then delete.
/var/adm/messages	Commonly used syslog daemon log file.	Keep last 1000 lines of file.
/etc/shutdown.log	Shutdown command log.	Keep last 100 lines of file.

Obviously, if you have additional applications installed, such as DB2, these will also have their own log files, which you will need to monitor and truncate.

Log files of IBM HTTP Server (IHS) can grow very quickly. Depending on how busy the site is, you may want to archive them much sooner that 14 days. These logs are often transferred to another system for statistical analysis. You might consider setting up a separate file system for these logs.

For further detailed information about the **cronolog** and **rotatelog** commands, refer to *IBM HTTP Server Powered by Apache on RS/6000*, SG24-5132.

All rules described here are enforced by the tidysys program, which can be downloaded as part of the additional material (see Appendix D, “Additional material” on page 341 for details on how to obtain it). You can modify the retention rules yourself by changing the configuration files kept in /var/adm/sys_mng/.

Note: It is recommended that you clear out your files daily to prevent them building up and to aid the smooth running of the system. For example, the tidysis program can be invoked by cron and its output logged to a file.

7.4.3 AIX log file rotation

Many AIX logs are automatically rotated; this includes the system log accessed through the **errpt** command and the boot log and console log. This is called *circular logging*. Each log file maintains a fixed size, overwriting the new data with the oldest in the log.

It is possible to do circular logging for your own applications using the **alog** command. The online command reference best describes how to set up your application for circular logging, changing the size of the log, and viewing the log.

7.4.4 The syslog subsystem

The syslog subsystem is a daemon that collects error messages from AIX subsystems. These errors are then filtered using the /etc/syslog.conf file and sent to certain user configurable log files.

Note: Although it is possible to send syslog message to be logged on a remote system, for example, our central administration server, syslogd uses UDP port 514, which violates our server farm security rules.

In AIX 5L Version 5.1, you can instruct syslogd to rotate log files using the rotate keyword in the /etc/syslog.conf configuration file, as shown in Example 7-17.

Example 7-17 The /etc/syslogd.conf file with the rotation keyword

```
#  mail.debug          /usr/spool/mqueue/syslog
#  *.debug             /dev/console
#  *.crit                      *
#  *.debug             /tmp/syslog.out     rotate size 100k files 4
#  *.crit              /tmp/syslog.out     rotate time 1d

auth,mark.info         /var/adm/watchpipe rotate time 1d files 14 compress
*.err;kern.debug;daemon.notice;mail.crit;user.none      /var/adm/messages
rotate time 1d files 14 compress
auth,mark.info         /var/adm/authlog rotate time 1d files 14 compress
```

In this example, the three log files, /var/adm/watchpipe, /var/adm/messages, and /var/adm/authlog are rotated once a day and 14 compressed generation will be retained.

7.4.5 Startup and shutdown logs

One common activity is starting up and shutting down the servers. Unfortunately, as the servers are connected remotely, you may not see the booting messages or the shutdown messages. The booting messages are actually stored in several places; Table 7-4 shows the location of these files.

When you shut down, it is also possible to log what is happening, just in case some process does not stop correctly. Use the **`shutdown -l`** command to perform a logged shutdown.

Table 7-4 Startup and shutdown logs

File	Contents of file
/var/adm/ras/bootlog	Hardware initialization. This file is in alog format, so it must be read using **`alog -f bootlog -o`**.
/var/adm/ras/conslog	General startup messages. This file is in alog format (see Section 2.3.2, "AIX console logging" on page 49).
/etc/shutdown.log	Shutdown log.

7.5 Performance

Performance monitoring is a vast subject to discuss. We need to make sure we get the best performance out of the farm, to ensure everything is tuned to its peak; if one component is running slow, it could have an impact on another part. There are many tools available to help monitor the performance of the farm, from packets sniffers on the network to system analysis tools on your servers. There are two different types of analysis you need to perform.

- Real-time analysis

  Used to spot and fix immediate problems, for example, a piece of faulty software hogging a CPU.

- Long term trend analysis

  Use to spot trends in workloads, to ensure you have enough horse power to handle peak loads or special events. To provide upgrade options and help define future solutions.

This section briefly explains performance analysis on AIX. For further detailed information about performance analysis, please refer to the following publications:

- *AIX 5L Performance Tools Handbook*, SG24-6039
- *Understanding IBM @server pSeries Performance and Sizing*, SG24-4810

7.5.1 Real-time performance monitoring

In every case where you feel you have a performance issue, you must consider if you really have a performance issue. Performance issue are triggered by user expectations, so you should define a baseline of performance first. In this section, we assume that we really have some performance issues; however, long term trend analysis will help you answer this first question.

With any performance monitoring, there are four resources that must be examined:

- CPU
- Memory
- Disk I/O
- Network

The most useful tools to monitor performance on AIX are topas, vmstat, iostat and ps. We recommend that you run **topas** first. This command give you an instant heads up as to what is happening on the system. Example 7-18 shows an example output of **topas** on our system.

Example 7-18 topas output

```
Topas Monitor for host:    svr03                EVENTS/QUEUES   FILE/TTY
Thu Mar 21 15:59:45 2002   Interval:  2         Cswitch     430 Readch 2916.7K
                                                Syscall   67488 Writech 727.1K
Kernel   18.9   |#####                       |  Reads      6949 Rawin 0
User     38.9   |###########                 |  Writes      182 Ttyout 0
Wait      0.0   |                            |  Forks         0 Igets 0
Idle     42.0   |############                |  Execs         0 Namei 16110
                                                Runqueue    1.2 Dirblk 3985
Network  KBPS   I-Pack  O-Pack   KB-In  KB-Out  Waitqueue   1.0
tr0       1.0      1.9     1.4     0.2     0.8
en3       0.1      0.4     0.0     0.1     0.0  PAGING          MEMORY
                                                Faults      208 Real,MB 2047
Disk    Busy%     KBPS     TPS KB-Read KB-Writ  Steals        0 % Comp 14.6
hdisk0    0.9      7.9     1.4     0.0     7.9  PgspIn        0 % Noncomp 49.6
hdisk2    0.0      0.0     0.0     0.0     0.0  PgspOut       0 % Client 0.5
                                                PageIn        0
Name            PID CPU% PgSp Owner             PageOut       1 PAGING SPACE
find          27550 42.1  0.4 root              Sios          1 Size,MB 512
topas         16230  1.5  1.8 root                              % Used 1.3
sort          25018  0.6  2.7 root              NFS (calls/sec) % Free 98.6
ksh            3974  0.4  0.5 root              ServerV2       0
syncd          5682  0.1  0.3 root              ClientV2       0   Press:
dtexec        29586  0.0  1.7 root              ServerV3       0   "h" for help
dtscreen      24282  0.0  1.6 root              ClientV3       0   "q" to quit
lrud           1032  0.0  0.0 root
```

```
gil           1806  0.0  0.0 root
nfsd         18410  0.0  0.0 root
rpc.lockd     5476  0.0  0.0 root
X            14414  0.0  2.0 root
init             1  0.0  1.8 root
```

In Example 7-18 on page 269, we do not have any performance issue with the system; none of our resources are particularly busy. The most active command on the system appears to be the **find** command, which is generating some disk I/O.

7.5.2 Long term trend analysis

Long term figures are very important. These allow you to monitor your systems over a period of time and see how they are behaving. These also allow you to define a baseline for your system performance. You can use these figures to predict future workload and any upgrades that may be required. Potential problems may be highlighted by these figures. Once you establish the basic pattern for your system, any deviation could indicate a problem or a change in environment. For example, if you normally expect to see peaks and troughs in your figures, for example, during the day your service is busy, and during the night it gets quieter, but you suddenly see constant values, which could indicate rogue processes.

We developed a method that aids you in performing this analysis. The method is composed of several scripts that use gnuplot and other software tools.

Software requirements

You have to install the following RPM packages from the AIX toolbox for Linux applications:

- gnuplot
- libpng
- zlib

Gathering performance statistics

You have to gather the performance statistics on your system using the cpu_stats script, as shown in Example 7-19 on page 271. This script should be invoked from cron by adding the following entry in the root user's crontab:

```
0,5,10,15,20,25,30,35,40,45,50,55 * * * * /stats/cpu_stats
```

The cpu_stats script generates the following output:

```
# tail -2 /stats/vm02088.svr06
```

```
10:45:00 29/03/2002  0  0 47611 199807   0   0   0   0    0   0 442  813 372  0
 0 99  0
10:50:00 29/03/2002  0  0 47611 199806   0   0   0   0    0   0 442  822 375  0
 0 99  0
```

Example 7-19 cpu_stats script

```
#! /bin/ksh
#
# This script is run by cron every 5 mins.
# The vmstat command is used to record cpu statistics.
# The statistics are collected over a 15 second sample every 5 minutes.
# The output is appended to the /stats/vmstat_yyddd file.
#     where yy = year
#           ddd = julian date
# Since vmstat does not display time stamp, a time stamp is inserted onto each
# line of the file before each vmstat entry.
#
HOST=$(hostname -s)
FILE1=/stats/vm`date +%y%j`.$HOST
FILE2=/stats/io`date +%y%j`.$HOST
DATE=`date +%H:%M:%S" "%e"/"%m"/"%Y`

print  "$DATE `vmstat 15 2 | tail -1`" >> $FILE1

print "Date:$DATE" >> $FILE2
iostat -a 15 2  | grep -v cd >> $FILE2
```

You should also prune the old log files using the prunestats script, as shown in Example 7-20. This script should be invoked from cron by adding the following entry in the root user’s crontab:

```
57 23 * * * /stats/prunestats > /dev/null 2>&1
```

Example 7-20 prunestats script

```
!/bin/ksh
#
# Must be run between 23:56 and 23:59 every day

VMFILE=vm$(date +%y%j).$(hostname -s)
IOFILE=io$(date +%y%j).$(hostname -s)
compress /stats/$VMFILE
compress /stats/$IOFILE
find /stats -type f \( -name "vm*" -o -name "io*" \) -mtime +40 | xargs rm
```

Generating a performance graph

You have to generate performance graphs for the performance statistics on your system using the two scripts, drawdailygraphcpu.ksh and drawdailygraphpag.ksh. These scripts are shown in Appendix B, “Example scripts” on page 313. These script should be invoked from cron by adding the following lines in the root user’s crontab:

```
1 1 * * * /stats/drawdailygraphcpu.ksh > /tmp/stats.out 2>&1
6 1 * * * /stats/drawdailygraphpag.ksh > /tmp/stats.out 2>&1
```

Basically, these scripts use the **gnuplot** command to generate a performance graph, as shown in the following pseudo script line:

```
cat plot_script | /usr/local/bin/gnuplot
```

A plot script is a series of commands that instruct the **gnuplot** command to generate a graph along with control information. A sample plot script is shown in Example 7-21.

Example 7-21 Sample plot script for gnuplot

```
set terminal png small color
set output "/stats/svr06.cpu.290302.daily.png"
set title "Performance statistics - svr06”
set xlabel "Time”
set ylabel "Percentage”
set xdata time
set timefmt "%H:%M:%S %d/%m/%Y"
set xrange [*:]
set format x "%H:%M:%S"
plot “/stats/vm02088.svr06" using 1:16 smooth csplines title "USER" with lines
,\
“/stats/vm02088.svr06” using 1:17 smooth csplines title "SYS" with lines ,\
"/stats/vm02088.svr06" using 1:18 smooth csplines title "IDLE" with lines ,\
"/stats/vm02088.svr06" using 1:19 smooth csplines title "WIO" with lines
```

Figure 7-7 on page 273 shows a sample graph generated by our method, comparing four critical CPU resources (user, system, idle, and waiting for input). Figure 7-7 on page 273 comes from a fairly healthy Web server. We can see during the night idle time is high, but at approximately 05:30 it drops, and user time climbs–this could indicate a backup running, for at 07:00 it returns to idle. Then at 09:30 idle drops and it remains busy throughout the day, at approximately 16:30 we experience a user peak, then it drops off, but gradually raises until 21:00; we then experience a gradual drop off throughout the night.

This would indicate this Web server is mostly used during the day, probably by workers in the office. The highest spike for user activity was from 12:00 to 13:00 –lunch hour! We experience a gradual drop off at 16:30 as people begin to go home. After 18:00, users begin to login from home and use the site. This builds until 21:00, when we experience a sharp spike; this could be due to an advertisement on television, for example. After that, people go to bed and the site quietens down.

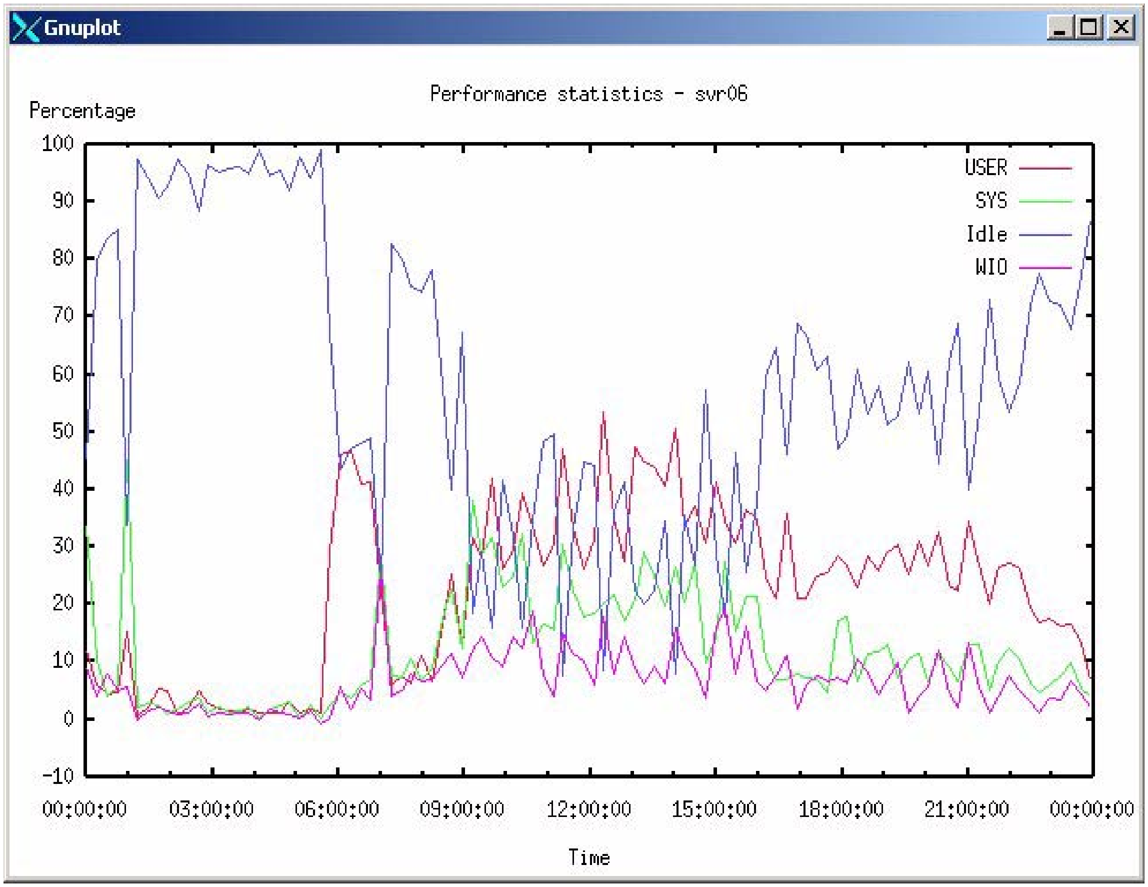


Figure 7-7 Daily performance statistics

7.6 General administration tasks

This section covers general administration tasks that must be performed on each server, which includes heath checking, documentation, and general monitoring.

The tidysys program, which can be downloaded as part of the additional material does a great job for general administration tasks. See Appendix D, “Additional material” on page 341 for details on how to obtain it.

7.6.1 Health checks

There are many different types of health checks you could perform: start of day checks, end of day checks, and application checks, to name but a few. Health checking gives you the confidence that everything is running smoothly.

You may wish to consider the following health checks. Remember these are snapshots in time and do not guarantee everything will continue to work as designed. Your monitoring should check:.

- File system status
- AIX Processes running
- Load on system
- Page space status
- Error report
- Open TCP ports
- Network links
- Output from any overnight processing
- Printer status
- Application processes

7.6.2 Security checks

Security within the server farm should always be an ongoing process. To aid this, tools should be deployed to ensure none of the security rules have been broken. Software tools can make the system less susceptible to compromise.

Login checking

You should run weekly or even daily checks to see:

- Who attempted to **su** to root and failed.
- Any accounts that have been locked due to too many failed login attempts.
- Any account that has not been used for more than 60 days.

The failed attempts could be due to a user forgetting the root password; however, if you are using OpenSSH with keys, you should not see any failures due to users forgetting their own passwords. Any suspicious activity, such as repeated attempts to **su** to root, and you should challenge the user, to ensure it was them.

If an account has not been used for more than 60 days, it is possible the user no longer needs access to that system, and their account can be removed.

Example 7-22 shows an example output from the tidysys program.

Example 7-22 Login checking output from tidysis

```
Failed Root Logins
______________________________________________________________________
root        pts/0       Mar  9 19:15     (9.3.1.32)
root        pts/0       Mar 10 14:17     (9.3.1.32)
root        dtlogin/_0  Mar 11 12:58
root        dtlogin/_0  Mar 11 12:58
root        pts/1       Mar 19 10:25     (keigotp20.itsc.austin.ibm.com)
root        pts/4       Mar 21 13:21     (loopback)

Total Failed Root Logins is 6


Failed Root SU Logins
______________________________________________________________________
03/21 14:38 pts/1      parkera-root

Total Failed Root SU Logins is 1

Users who have not logged in for more than 60 days
______________________________________________________________________
User edwardsm has not logged in since 22/06/2001 20:49:38

Total Users is 1


Users with failed login count greater than 5
______________________________________________________________________
User ibyt has failed login count of  7

Total Users is 1
```

File permission checks

At least once a week, you should check file permissions on your servers. Weak file permissions are an easy way for an unauthorized user to gain access to your system. Users may accidentally reset permissions, or they may be deliberately reset.

Key permissions to check:

- Set-UID and set-GID programs
- Permissions on key files run by root
- Permissions on system binaries

Several tools exist to help you monitor your permissions; the tidysis program will list the most important security area, set-UID, and set-GID programs. Example 7-23 is the output from tidysis, reporting set-UID and setGID programs. It also reports files that are not owned by anybody. Ownership should be set on these files; if in doubt, set the owner to root.

Example 7-23 File permission checking output from tidysis

```
SUID SGID Root Programs
______________________________________________________________________
-r-sr-sr-x root     printq   Apr  8  2001 /usr/lib/lpd/pio/etc/piodmgrsu
-r-xr-sr-x root     printq   Apr  8  2001 /usr/lib/lpd/pio/etc/piomkapqd
-r-sr-s--- root     system   Jun 18  2001 /usr/lib/semutil
-r-sr-sr-x root     cron     Jul 27  2001 /usr/bin/at
-r-xr-sr-x root     security Jul 27  2001 /usr/bin/chfn
-r-xr-sr-x root     security Jul 27  2001 /usr/bin/chgrpmem
-r-sr-s--- root     printq   Jun 18  2001 /usr/bin/chque
-r-sr-s--- root     printq   Jun 18  2001 /usr/bin/chquedev
-r-xr-sr-x root     security Jul 27  2001 /usr/bin/chsh

... many lines are erased on purpose ...

-r-xr-s--- root     printq   Apr  8  2001 /usr/sbin/pioattred
-r-xr-s--- root     printq   Apr  8  2001 /usr/sbin/piofontin
-r-xr-s--- root     printq   Apr  8  2001 /usr/sbin/piopredef
-r-xr-s--- root     printq   Apr  8  2001 /usr/sbin/rmvirprt
-r-sr-sr-x root     sys      Jun 18  2001 /usr/dt/bin/dtaction

Total SUID/SGID Root Programs is 43


Unowned Files
______________________________________________________________________
/usr/lpp/IMNSearch.rte/inst_root/etc/IMNSearch
/usr/lpp/IMNSearch.rte/inst_root/etc/IMNSearch/dbcshelp
/usr/lpp/IMNSearch.rte/inst_root/etc/IMNSearch/dbcshelp/work

... many lines are erased on purpose ...

/var/statmon
/var/statmon/state
/var/statmon/sm
/var/statmon/sm.bak
```

```
Total Unowned Files is 32
```

Any new programs appearing with a set-UID or set-GID should be treated with extreme caution. Those might be innocent files of newly installed software, but they could be a security back door bypassing your security rules.

Using sudo

There can be certain occasions when you need a non-root user to perform tasks that only the root user can do. One way to do this is to give that user the root password; this is clearly not acceptable. Another option might be to use a set-UID program; unfortunately, you can only put a set-UID bit on a binary executable file, not on a shell script. However, the software tool sudo offers you a solution to address this issue. The **sudo** command allows non-root users, or groups of users, to run pre-defined programs, including shell scripts, as the root user. It provides authentication and logging for the commands run.

For detailed information about sudo, visit the following URL:

http://www.courtesan.com/sudo/

You can download the RPM package of sudo from the AIX toolbox for Linux applications download site, found at:

http://www-1.ibm.com/servers/aix/products/aixos/linux/download.html

Once sudo is installed, you need to decide which tasks, normally performed by the root user, can be automated through shell scripts. Write and test these new shell scripts and configure sudo to allow certain users to run these shell scripts.

We show an example that allows certain non-root users to install a new version of the IHS configuration file and recycle the IHS server. Example 7-24 shows a simple shell script, named httpcp, to perform this task. When writing shell scripts executed by sudo, please make sure you use the critical path to the commands being executed. Remember, these commands are being run with root privileges–you need to ensure the binary file you think is being executed is actually being used and not a different command with the same name in the $PATH environment value.

Note: We recommend you always specify an absolute path name for commands in the script that is invoked from sudo.

Example 7-24 httpcp

```
#!/bin/ksh
DATE1=`date +%y%m%d.%H%M`
```

```
conf=/usr/HTTPServer/conf

if [ $# -ne 1 ]; then
        echo "Please enter the name of the file you wish to copy"
        exit 1
fi
if [ ! -f $1 ]; then
        echo "File "$1" not found"
        exit 1
fi

cp $conf/httpd.conf $conf/$DATE1.httpd.conf.bak
if [ $? -ne 0 ]; then
        echo "Problem creating copy of httpd.conf file"
        exit 2
fi

cp $1 $conf/httpd.conf
if [ $? -ne 0 ]; then
        echo "Problem copying new version of httpd.conf"
        exit 3
fi

cd /usr/HTTPServer/bin
./apachectl configtest
if [ $? -ne 0 ]; then
        echo "Config file syntax incorrect - validation has failed"
        cp $conf/$DATE1.httpd.conf.bak $conf/httpd.conf
        exit 4
fi

./apachectl graceful
if [ $? -ne 0 ]; then
        echo "Graceful restart of HTTP server has failed"
        echo "Copying back original version of httpd.conf"
        cp $conf/$DATE1.httpd.conf.bak $conf/httpd.conf
        ./apachectl graceful
else
        echo "HTTP server gracefully restarted"
        exit 0
fi
```

Example 7-25 on page 279 shows the sudo configuration file, /etc/sudoers. Please note that the file permission is very important. As shown in the following example, only root should be able to modify this file:

```
# ls -l /etc/sudoers
-r--r-----   1 root     system          609 Apr 03 18:54 /etc/sudoers
```

When editing this file, use the **visudo** command. This command allows you to edit the file using vi and it performs a basic sanity check on the file before it is saved.

Example 7-25 /etc/sudoers

```
#
# This file MUST be edited with the 'visudo' command as root.
#
# See the man page for the details on how to write a sudoers file.
#
# Host alias specification
# User alias specification
# Cmnd alias specification
Cmnd_Alias
SHELLS=/usr/bin/ksh,/usr/bin/csh,/bin/sh,/usr/bin/sh,/usr/bin/bash,/bin/ksh,/bi
n/csh
Defaults        logfile=/var/adm/sudo.log

# User privilege specification
root            ALL=(ALL) ALL
dyner           ALL=/usr/local/bin/httpcp
ibyt            ALL=/usr/local/bin/httpcp
%webmasters          svr06=/usr/local/bin/httpcp
```

In the example, the users dyner, ibyt, and all members of the webmasters group, are allowed to run the /usr/local/bin/httpcp script. Notice that ibyt and dyner are allowed to run the script on *any* machine with this file; however the group webmasters is only allowed to run this script on host svr06. This allows you to have a global sudoers file in your farm environment. Example 7-26 shows an example of sudo usage, where the user ibyt is attempting to run the httpcp script.

Example 7-26 sudo example

```
$ httpcp /tmp/httpd.conf
ksh: httpcp: 0403-006 Execute permission denied.
$ sudo httpcp httpd.conf

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these two things:

        #1) Respect the privacy of others.
        #2) Think before you type.

Password: <password>
Syntax OK
./apachectl graceful: httpd gracefully restarted
HTTP server gracefully restarted
```

The user is prompted for their password. This is a security check to ensure an unauthorized user has not just sat down at the terminal and used the command. It is possible to disable this password checking using the /etc/sudoers file. The command execution is also logged, as shown in the following example:

```
# cat /var/adm/sudo.log
Apr  3 18:48:43 : ibyt : TTY=pts/2 ; PWD=/home/ibyt ; USER=root ;
    COMMAND=/usr/local/bin/httpcp /tmp/httpd.conf
```

If you encounter the error message shown in Example 7-27 upon invocation of the **sudo** command, add the following line to $HOME/.profile:

```
export LIBPATH=/usr/local/bin
```

Example 7-27   A sample sudo error message

```
$ sudo /usr/local/bin/httpcp httpd.conf
Password:
Syntax error on line 70 of /usr/HTTPServer/conf/httpd.conf:
Cannot load /usr/HTTPServer/libexec/libphp4.so into server:
0509-022 Cannot load module /usr/local/lib/libgd.a(shr.o).
0509-150   Dependent module /usr/lib/libttf.a(shr.o) could not be loaded.
0509-152   Member shr.o is not found in archive
0509-022 Cannot load module /usr/lib/libttf.a.
0509-150   Dependent module /usr/lib/libttf.a could not be loaded.
Config file syntax incorrect - validation has failed
```

Port scanning

You should also perform regular checks on open ports on each system. A user may have inadvertently opened up a service that is not authorized within the farm. Or someone may have deliberately opened a port as a back door. A port scan will scan all the TCP/IP ports on a target server and list any listening ports. These scans should be performed weekly. Not only can they spot issues with a server, but they could point out a security weakness with the firewalls.

One of the first things a potential attacker will do when trying to penetrate a system is scan all the TCP ports to see what services are open to abuse. The less ports are open, the less choices the attacker has.

There are many tools available to scan systems, including nmap and SATAN. You can download these tools from the Bull site, found at:

http://www.bullfreeware.com/

Note: Port scanning is widely considered as an aggressive act. Do *not* scan any system without permission from administrators of the target system. Scanning can be interpreted by the administrator as a penetration attack by a hostile group. Do not scan your systems at peak periods; scanning can create a lot of extra traffic and can cause unpredictable results.

7.6.3 User maintenance

User maintenance is a critical if somewhat mundane task when administering a server farm. Users are frequently deleted and added on all servers. You can use some network naming service, such as NIS, NIS+, and LDAP for this purpose. However, if you cannot use these services in your server farm for some reason, each user must be authenticated by each individual server. Adding a user to a large number of servers can be a long, manual task, and manual process are open to mistakes.

To automate the task, we propose you use rsync, which is a client/server based software tool. The rsync server runs on each managed server in the server farm. Each server is set up to trust and authenticate only the central administration server. You can send out a series of files from the central administration server to each server in the farm. These files will include:

- /etc/passwd
- /etc/group
- /etc/security/passwd
- /etc/security/group
- $HOME/.profile
- $HOME/.ssh/authorized_keys

Any user added or removed from the central administration server can then be automatically reflected on each server in the farm. The home directory of a deleted user will not be removed. This is a manual process, as the user may have some files in their home directory you may wish to keep. You might consider setting up a user defined executive in NET-SNMP to delete this directory; see Section 5.3.5, “Configuring a simple SNMP manager” on page 202 for details on how to do this task.

Note: When a new user is created, make sure they log in on the central administration server and set up their public and private keys. Once these keys are created, copy the public key, id_dsa.pub, to $HOME/.ssh/authorized_keys, so it is distributed around the farm.

To use this user synchronization mechanism, each server must install two software tools: rsync and zlib. You can download these software tools from the Bull site, found at:

http://www.bullfreeware.com/

Example 7-28 shows the contents of the /etc/rsyncd.conf file on each of the servers. You may add additional stanzas if you wish to keep additional files in sync between other servers. For example, you may wish to keep static content in sync between Web servers in a cluster.

Example 7-28 /etc/rsyncd.conf

```
use chroot = yes
max connections = 4
log file = /var/adm/rsync.log
pid file = /etc/rsyncd.pid
read only = false

[etc]
        uid = root
        gid = security
        path = /etc
        auth users = root
        secrets file = /etc/rsyncd.secret
        strict modes = true
        hosts allow = XXX.XXX.XXX.XXX
        hosts deny = *
        comment = /etc

[home]
        uid = root
        gid = system
        path = /home
        auth users = root
        secrets file = /etc/rsyncd.secret
        strict modes = true
        hosts allow = XXX.XXX.XXX.XXX
        hosts deny = *
        comment = /home
```

Note: You should restrict the host's IP address that can be allowed to connect using the `hosts allow` keyword, which is shown in the high-lighted line in Example 7-28 on page 282.

Once the rsync daemon is running, you need to initiate the transfer from the central administration server. To do this, we created a short script, sync_users.ksh, as shown in Example 7-29, which sends the necessary files to all the managed servers. In this example, the managed server is svr01 and svr02.

Example 7-29 sync_users.ksh

```
#!/bin/ksh
#
# Will synchronize new users to a specified host.
# Home directories and public keys will also be sent across.


# System files
cd /etc
for svr in svr01 svr02
do
rsync -avz --password-file=/etc/rsync.secret /etc/passwd root@${svr}::etc
rsync -avz --password-file=/etc/rsync.secret /etc/group root@${svr}::etc
rsync -avzR --password-file=/etc/rsync.secret security/passwd root@${svr}::etc
rsync -avzR --password-file=/etc/rsync.secret security/user root@${svr}::etc
rsync -avzR --password-file=/etc/rsync.secret security/group root@${svr}::etc
done

# Users home directory with Public key and .profile
rsync -avz --password-file=/etc/rsync.secret --include "*/" \
    --include="authorized_keys" --include=".profile" --exclude="*" /home/ \
    root@svr02::home
```

To synchronize users among managed servers, issue the sync_users.ksh script on the administration server, as shown in the following example:

```
# /usr/local/bin/sync_users.ksh
building file list ... done
wrote 129 bytes  read 77 bytes  137.33 bytes/sec
total size is 607  speedup is 2.95
building file list ... done
wrote 128 bytes  read 77 bytes  410.00 bytes/sec
total size is 330  speedup is 1.61
building file list ... done
wrote 154 bytes  read 77 bytes  462.00 bytes/sec
total size is 450  speedup is 1.95
```

```
building file list ... done
wrote 152 bytes  read 77 bytes  458.00 bytes/sec
total size is 10124  speedup is 44.21
building file list ... done
wrote 153 bytes  read 77 bytes  460.00 bytes/sec
total size is 394  speedup is 1.71
building file list ... done
wrote 646 bytes  read 77 bytes  482.00 bytes/sec
total size is 1880  speedup is 2.60
```

The rsync daemon uses a secret key file, /etc/rsync.secret, for authentication. Therefore, you should set the following file permission on this file:

```
# ls -l /etc/rsync.secret
-rwx------   1 root     system            9 Apr 02 11:19 /etc/rsync.secret
```

> **Note:** The /etc/rsync.secret file must not be read-writable by any users other than root. The rsync daemon does not use the UNIX password authentication mechanism.

The /etc/rsync.secret file contains the secret key, which enables you to connect the **rsync** command to the rsyncd daemon on the remote systems. In our example, the secret key shown in the following example is used for the **rsync** command on the local administration server to be authenticated by rsyncd on svr01 and svr02:

```
# cat /etc/rsync.secret
r00tsfpw
```

For further information about rsync, please visit to the following URL:

http://www.rsync.org/

> **Note:** Using this example of rsync, the root password will be the same on *every* server in the farm. This may be against your security policy.

7.6.4 Self-document the system

One of the most important things about your server farm is documentation that describes the current hardware and software configuration on your system. Because any change can be made asynchronously, the document should be generated automatically; then, all this information should be sent to the administration server. This way, you are always aware of your current inventory, and if any problems are high lighted with a software level or hardware device, you know immediately which system are affected.

We developed two sample scripts, listhwlevels and listswlevels, which are shown in Appendix B, “Example scripts” on page 313, to gather the following information:

- System hardware configuration
- Hardware devices attributes
- AIX technical information:
  - Disk allocation and layout
  - Software levels
  - Firmware levels

You could easily customize these for your own needs, and perhaps add some HTML tags so it could be served by a Web server.

A

SNMP related information

This appendix provides several SNMP related information and configuration file examples, which include:

- The default /etc/snmpd.conf file (see "The default /etc/snmpd.conf file on AIX" on page 288)
- CERT security alert CA-2002-03 (see "CERT security alert CA-2002-03" on page 292)
- EXAMPLE.conf (see "CERT security alert CA-2002-03" on page 292)

The default /etc/snmpd.conf file on AIX

Upon initial installation of AIX 5L Version 5.1, the /etc/snmpd.conf file is installed, as shown in Example A-1, by default. Please note that if you installed additional filesets from AIX 5L Version 5.1 installation CD-ROM media, then you might see additional lines in the /etc/snmpd.conf on your system.

For instance, the fileset devices.common.IBM.atm.rte will add the following lines to the /etc/snmpd.conf file.

```
snmpd smuxtimeout=200 #muxatmd
smux 1.3.6.1.4.1.2.3.1.2.3.1.1 muxatmd_password #muxatmd
```

Example: A-1 The default /etc/snmpd.conf file on AIX

```
# @(#)93       1.13  src/tcpip/etc/snmpd.conf, snmp, tcpip510 11/27/00 13:18:40
#
# COMPONENT_NAME: (SNMP) Simple Network Management Protocol Daemon
#
# FUNCTIONS: none
#
# ORIGINS: 27 60
#
# (C) COPYRIGHT International Business Machines Corp. 1991, 1994
# All Rights Reserved
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
# Licensed Material - Property of IBM
#
# Contributed by NYSERNet Inc.  This work was partially supported by the
# U.S. Defense Advanced Research Projects Agency and the Rome Air Development
# Center of the U.S. Air Force Systems Command under contract number
# F30602-88-C-0016.
#
# FILE: /etc/snmpd.conf
#
################################################################################


################################################################################
#
# snmpd configuration information
#
################################################################################


################################################################################
#
# How to configure this file for your system:
```

```
#
# 1. If you want to direct your logging from the configuration file,
#    set the logging specifications as follows:
#
#       logging         </path/filename>        enabled|disabled
#       logging         size=<limit>            level=<debug level>
#
#    where </path/filename> specifies the complete path and filename of the
#    log file, enabled turns logging on, disabled turns logging off, <limit>
#    specifies the maximum size in bytes of the specified logfile, and
#    <debug level> specifies the logging level of 0, 1, 2, or 3.  There is no
#    default logging file.  The enablement default is disabled.  The size
#    default is 0, meaning unlimited, and the level default is 0.  There can
#    be no white spaces around the "=" in the size and level fields.  There
#    are no restrictions on the order in which the fields are entered in the
#    logging entries.  A logging entry can contain single or multiple fields.
#
# 2. Set the community names and access privileges for hosts that can make
#    requests of this snmpd agent.  Define these restrictions as follows:
#
#       community  <name>  <address>  <netmask>  <permissions>  <view name>
#
#    where <name> is the community name, <address> is either a hostname or
#    an IP address in dotted notation, and <permissions> is one of:  none,
#    readOnly, writeOnly, readWrite.  The default permission is readOnly.
#    <netmask> specifies the network mask.  The default address and netmask
#    are 0.0.0.0.  If an address other than 0.0.0.0 is specified, the default
#    netmask is 255.255.255.255.  If a permission is specified, both the
#    address and netmask must also be specified.  <view name> defines a
#    portion of the MIB tree to which this community name allows access.
#    <view name> must be defined as a unique object identifier in dotted
#    numeric notation.  <view name> is further defined in the view
#    configuration entry.  If <view name> is not specified, the view for
#    this community defaults to the entire MIB tree.  Fields to the right
#    of <name> are optional, with the limitation that no fields to the
#    left of a specified field are omitted.
#
# 3. Set your MIB views as follows:
#
#       view  <view name>  <MIB subtree>...
#
#    where <view name> is a unique object identifier in dotted numeric
#    notation and <MIB subtree> is a list of the MIB subtrees in text or
#    dotted numeric notation that this view allows access.  The <view name>
#    is the same as that specified in the community configuration entry.  If
#    the MIB subtree list is not specified, the view defaults to the entire
#    MIB tree.
#
# 4. If your site has a management station that listens for traps, fill-in
#    the information for the trap destination as follows:
#
#       trap  <community>  <a.b.c.d>  <view name>  <trap mask>
#
#    where <community> is the community name that will be encoded in the
#    trap packet and <a.b.c.d> is the hostname or IP address in dotted
#    notation of the host where a trap monitor is listening on UDP port 162.
#    The <view name> is a unique object identifier in dotted notation. View
#    name is not implemented for traps.  The snmpd agent only checks
#    the view name format and duplication.  The trap mask is in hexadecimal
#    format.  The bits from left to right stand for coldStart trap, warmStart
#    trap, linkDown trap, linkUp trap, authenticationFailure trap,
#    egpNeighborLoss trap, and enterpriseSpecific trap.  The right most bit
#    does not have any meaning.  The value "1" will enable the corresponding
#    trap to be sent. Otherwise, the trap is blocked.
#       ex.     fe      block no traps (1111 1110)
#               7e      block coldStart trap (0111 1110)
#               be      block warmStart trap (1011 1110)
#               3e      block coldStart trap and warmStart trap (0011 1110)
#
# 5. Set your snmpd specific configuration parameters as follows:
#
#       snmpd  <variable>=<value>
#
#    where <variable> is one of maxpacket, querytimeout or smuxtimeout.
#    If <variable> is maxpacket, <value> is the maximum packet size, in
#    bytes, that the snmpd agent will transmit.  The minimum value to
#    which maxpacket can be set is 300 bytes.  If there is no snmpd entry
#    for maxpacket, the system socket default limits will be used.  If
#    <variable> is querytimeout, <value> is the time interval, in seconds,
#    at which the snmpd agent will query the interfaces to check for
#    interface status changes.  The minimum value to which querytimeout
#    can be set is 30 seconds.  If 0 (zero) is specified, snmpd will not
#    query the interfaces for status changes.  If no snmpd entry for
#    querytimeout is specified, the default value of 60 seconds is used.
#    If <variable> is smuxtimeout, <value> is the time interval, in
#    seconds, at which snmpd will timeout on a request to a smux peer.
#    If 0 (zero) is specified, snmpd will not timeout on smux requests.
#    If no snmpd entry for smuxtimeout is specified, the default value
#    of 15 seconds is used.  If <variable> is ethernettimeout,
#    tokenringtimeout, or fdditimeout, <value> is the maximum time, in
#    seconds, between flushings of the internal cache for variables associated
#    with the respective device.  If <variable> is smuxtrapaddr, <value> is
#    defined for either 0 or 1.  If <value> is 0, the trap address on a SMUX
#    generated trap will be the local host's address if the trap originated
#    from a SMUX peer on the local host.  If <value> is 0 and the trap was
#    generated by a remote SMUX peer, the address of the remote machine will be
#    used in the trap.  If <value> is 1, the address of the local machine
#    (i.e., the machine on which snmpd is running) will be used for all SMUX
#    generated traps.  The "=" is absolutely required, and no white
#    spaces are allowed around the "=".  There are no restrictions on
#    the order in which the fields are entered in the snmpd entry.  An
#    snmpd entry can contain single or multiple fields.
#
# 6. Set the smux peer configuration parameters as follows:
#
#       smux <client OIdentifier> <password> <address> <netmask>
#
#    where <client OIdentifier> is the unique object identifier in dotted
#    decimal notation of the SMUX peer client.  <password> specifies the
#    password that snmpd requires from the SMUX peer client to authenticate
#    the SMUX association.  <address> is either the hostname or IP address
#    in dotted notation of the host on which the SMUX peer client is
#    executing.  <netmask> specifies the network mask.  If no password is
#    specified, there is no authentication for the SMUX association. The
#    default address and netmask are 127.0.0.1 and 255.255.255.255.  If
#    neither the address nor netmask are specified, the SMUX association
#    is limited to the local host.  Fields to the right of
#    <client OIdentifier> are optional, with the limitation that no fields
#    to the left of a specified field are omitted.
#
# 7. Set the system contact and system location by:
#
#       syscontact "System Adminstrator"
#       syslocation "Here, City, State, Country, Planet, Universe"
#
#    These variables will be used to define the MIB variables sysContact and
#    sysLocation respectively.  The values must be within quotes and less
#    than 256 bytes in length.  When one of these MIB variables is set, this
#    file will be appended with the new information.  The system administrator
#    should be aware that if there are many settings of these variables, the
#    file will grow and need to be cleaned up.  The rationale for not deleting
#    entries is to allow the system administrator a history of variable values.
#    If the values are not set in these files, the snmpd daemon will return a
#    null string.
#
# NOTE:  Comments are indicated by # and continue to the end of the line.
#        There are no restrictions on the order in which the configuration
#        entries are specified in this file.
#
###############################################################################

logging         file=/usr/tmp/snmpd.log         enabled
logging         size=0                          level=0

community       public
#community       private 127.0.0.1 255.255.255.255 readWrite
#community       system  127.0.0.1 255.255.255.255 readWrite 1.17.2
```

```
view            1.17.2          system enterprises view

trap            public          127.0.0.1       1.2.3   fe      # loopback

#snmpd          maxpacket=1024 querytimeout=120 smuxtimeout=60

smux            1.3.6.1.4.1.2.3.1.2.1.2         gated_password  # gated
smux            1.3.6.1.4.1.2.3.1.2.2.1.1.2     dpid_password   # dpid
```

CERT security alert CA-2002-03

CERT Advisory CA-2002-03: Multiple Vulnerabilities in Many Implementations of the Simple Network Management Protocol (SNMP).

Original release date: February 12, 2002

- Last revised: February 26, 2002
- Source: CERT/CC

A complete revision history can be found at the end of this section.

Systems affected

Products from a very wide variety of vendors may be affected. See Vendor Information for details from vendors who have provided feedback for this advisory.

In addition to the vendors who provided feedback for this advisory, a list of vendors whom CERT/CC contacted regarding these problems is available from:

http://www.kb.cert.org/vuls/id/854306
http://www.kb.cert.org/vuls/id/107186

Many other systems making use of SNMP may also be vulnerable but were not specifically tested.

Overview

Numerous vulnerabilities have been reported in multiple vendors' SNMP implementations. These vulnerabilities may allow unauthorized privileged access, denial-of-service attacks, or cause unstable behavior. If your site uses SNMP in any capacity, the CERT/CC encourages you to read this advisory and follow the advice provided in the Solution section below.

In addition to this advisory, we also have an FAQ available at:

http://www.cert.org/tech_tips/snmp_faq.html

1. Description

   The Simple Network Management Protocol (SNMP) is a widely deployed protocol that is commonly used to monitor and manage network devices. Version 1 of the protocol (SNMPv1) defines several types of SNMP messages that are used to request information or configuration changes, respond to requests, enumerate SNMP objects, and send unsolicited alerts. The Oulu University Secure Programming Group (OUSPG, http://www.ee.oulu.fi/research/ouspg/) has reported numerous vulnerabilities in SNMPv1 implementations from many different vendors.

   More information about SNMP and OUSPG can be found in “Background information” on page 298.

   OUSPG's research focused on the manner in which SNMPv1 agents and managers handle request and trap messages. By applying the PROTOS c06-snmpv1 test suite found at:

   http://www.ee.oulu.fi/research/ouspg/protos/testing/c06/snmpv1/0100.html

   to a variety of popular SNMPv1-enabled products, the OUSPG revealed the following vulnerabilities:

   – VU#107186 - Multiple vulnerabilities in SNMPv1 trap handling

     SNMP trap messages are sent from agents to managers. A trap message may indicate a warning or error condition or otherwise notify the manager about the agent's state. SNMP managers must properly decode trap messages and process the resulting data. In testing, OUSPG found multiple vulnerabilities in the way many SNMP managers decode and process SNMP trap messages.

   – VU#854306 - Multiple vulnerabilities in SNMPv1 request handling

     SNMP request messages are sent from managers to agents. Request messages might be issued to obtain information from an agent or to instruct the agent to configure the host device. SNMP agents must properly decode request messages and process the resulting data. In testing, OUSPG found multiple vulnerabilities in the way many SNMP agents decode and process SNMP request messages.

     Vulnerabilities in the decoding and subsequent processing of SNMP messages by both managers and agents may result in denial-of-service conditions, format string vulnerabilities, and buffer overflows. Some vulnerabilities do not require the SNMP message to use the correct SNMP community string.

These vulnerabilities have been assigned the CVE identifiers CAN-2002-0012 and CAN-2002-0013, respectively.

2. Impact

   These vulnerabilities may cause denial-of-service conditions, service interruptions, and in some cases may allow an attacker to gain access to the affected device. Specific impacts will vary from product to product.

3. Solution

   Note that many of the mitigation steps recommended below may have significant impact on your everyday network operations and/or network architecture. Ensure that any changes made based on the following recommendations will not unacceptably affect your ongoing network operations capability.

Apply a patch from your vendor

“Vendor information” on page 297 contains information provided by vendors for this advisory. Please consult this section to determine if you need to contact your vendor directly.

Disable the SNMP service

As a general rule, the CERT/CC recommends disabling any service or capability that is not explicitly required, including SNMP. Unfortunately, some of the affected products exhibited unexpected behavior or denial of service conditions when exposed to the OUSPG test suite even if SNMP was not enabled. In these cases, disabling SNMP should be used in conjunction with the filtering practices listed below to provide additional protection.

- Ingress filtering

  As a temporary measure, it may be possible to limit the scope of these vulnerabilities by blocking access to SNMP services at the network perimeter. Ingress filtering manages the flow of traffic as it enters a network under your administrative control. Servers are typically the only machines that need to accept inbound traffic from the public Internet.

  In the network usage policy of many sites, there are few reasons for external hosts to initiate inbound traffic to machines that provide no public services. Thus, ingress filtering should be performed at the border to prohibit externally initiated inbound traffic to non-authorized services. For SNMP, ingress filtering of the following ports can prevent attackers outside of your network from impacting vulnerable devices in the local network that are not explicitly authorized to provide public SNMP services.

```
snmp     161/udp     # Simple Network Management Protocol (SNMP)
snmp     162/udp     # SNMP system management messages
```

The following services are less common, but may be used on some affected products.

```
snmp               161/tcp     # Simple Network Management Protocol (SNMP)
snmp               162/tcp     # SNMP system management messages
smux               199/tcp     # SNMP Unix Multiplexer
smux               199/udp     # SNMP Unix Multiplexer
synoptics-relay    391/tcp     # SynOptics SNMP Relay Port
synoptics-relay    391/udp     # SynOptics SNMP Relay Port
agentx             705/tcp     # AgentX
snmp-tcp-port     1993/tcp     # cisco SNMP TCP port
snmp-tcp-port     1993/udp     # cisco SNMP TCP port
```

As noted above, you should carefully consider the impact of blocking services that you may be using.

It is important to note that in many SNMP implementations, the SNMP daemon may bind to all IP interfaces on the device. This has important consequences when considering appropriate packet filtering measures required to protect an SNMP-enabled device. For example, even if a device disallows SNMP packets directed to the IP addresses of its normal network interfaces, it may still be possible to exploit these vulnerabilities on that device through the use of packets directed at the following IP addresses:

- *all-ones* broadcast address
- subnet broadcast address
- any internal loopback addresses (commonly used in routers for management purposes, not to be confused with the IP stack loopback address 127.0.0.1.)

Careful consideration should be given to addresses of the types mentioned above by sites planning for packet filtering as part of their mitigation strategy for these vulnerabilities.

Finally, sites may wish to block access to the following RPC services related to SNMP (listed as name, program ID, alternate names):

```
snmp         100122 na.snmp snmp-cmc snmp-synoptics snmp-unisys snmp-utk
snmpv2       100138 na.snmpv2         # SNM Version 2.2.2
snmpXdmid    100249
```

Please note that this workaround may not protect vulnerable devices from internal attacks.

- Filter SNMP traffic from non-authorized internal hosts

  In many networks, only a limited number of network management systems need to originate SNMP request messages. Therefore, it may be possible to configure the SNMP agent systems (or the network devices in between the management and agent systems) to disallow request messages from non-authorized systems. This can reduce, but not wholly eliminate, the risk from internal attacks. However, it may have detrimental effects on network performance due to the increased load imposed by the filtering, so careful consideration is required before implementation.

Similar caveats to the previous workaround regarding broadcast and loopback addresses apply.

- Change default community strings

  Most SNMP-enabled products ship with default community strings of *public* for read-only access and *private* for read-write access. As with any known default access control mechanism, the CERT/CC recommends that network administrators change these community strings to something of their own choosing. However, even when community strings are changed from their defaults, they will still be passed in plain-text and are therefore subject to packet sniffing attacks. SNMPv3 offers additional capabilities to ensure authentication and privacy as described in RFC2574.

  Because many of the vulnerabilities identified in this advisory occur before the community strings are evaluated, it is important to note that performing this step alone is not sufficient to mitigate the impact of these vulnerabilities. Nonetheless, it should be performed as part of good security practice.

- Segregate SNMP traffic onto a separate management network

  In situations where blocking or disabling SNMP is not possible, exposure to these vulnerabilities may be limited by restricting all SNMP access to separate, isolated management networks that are not publicly accessible. Although this would ideally involve physically separate networks, that kind of separation is probably not feasible in most environments. Mechanisms such as virtual LANs (VLANs) may be used to help segregate traffic on the same physical network. Note that VLANs may not strictly prevent an attacker from exploiting these vulnerabilities, but they may make it more difficult to initiate the attacks.

  Another option is for sites to restrict SNMP traffic to separate virtual private networks (VPNs), which employ cryptographically strong authentication.

  Note that these solutions may require extensive changes to a site's network architecture.

- Egress filtering

  Egress filtering manages the flow of traffic as it leaves a network under your administrative control. There is typically limited need for machines providing public services to initiate outbound traffic to the Internet. In the case of SNMP vulnerabilities, employing egress filtering on the ports listed above at your network border can prevent your network from being used as a source for attacks on other sites.

- Disable stack execution

  Disabling executable stacks (on systems where this is configurable) can reduce the risk of *stack smashing* attacks based on these vulnerabilities. Although this does not provide 100 percent protection against exploitation of these vulnerabilities, it makes the likelihood of a successful exploit much smaller. On many UNIX systems, executable stacks can be disabled by adding the following lines to /etc/system:

  ```
  set noexec_user_stack = 1 set noexec_user_stack_log = 1
  ```

  Note that this may go against the SPARC and Intel ABIs and can be bypassed as required in programs with mprotect(2). For the changes to take effect you will then need to reboot.

  Other operating systems and architectures also support the disabling of executable stacks either through native configuration parameters or via third-party software. Consult your vendor(s) for additional information.

- Share tools and techniques

  Because dealing with these vulnerabilities to systems and networks is so complex, the CERT/CC will provide a forum where administrators can share ideas and techniques that can be used to develop proper defenses. We have created an unmoderated mailing list for system and network administrators to discuss helpful techniques and tools.

  You can subscribe to the mailing list by sending an email message to:

  ```
  majordomo@cert.org.
  ```

  In the body of the message, type:

  ```
  subscribe snmp-forum
  ```

  After you receive the confirmation message, follow the instructions in the message to complete the subscription process.

Vendor information

This section contains information provided by vendors for this advisory. As vendors report new information to the CERT/CC, we will update this section and note the changes in our revision history. If a particular vendor is not listed below, we have not received their comments.

Note: We deliberately excluded other vender information from here, since it quickly becomes obsolete. For updated and full-listing of vendor information, please refer to the latest CERT advisory report found at:

http://www.cert.org/advisories/CA-2002-03.html

- IBM Corporation

  The AIX operating system is susceptible to the vulnerabilities tested for by the Oulu University PROTOS test suite for all levels of AIX 4.3 prior to level 4.3.3.51, and AIX 5L Version 5.1 prior to level 5.1.0.10. APARs were developed and made available last year that closed the vulnerabilities looked for by the test suite. For 4.3.x, the relevant APAR is #IY17630; for 5.1, the appropriate APAR is #IY20943:

  ```
  # lslpp -l bos.net.tcp.client
  ```

  If the *Level* stated is lower than those given above, your system is vulnerable, and you are urged to apply the appropriate APAR.

  AIX versions prior to 4.3 are also vulnerable, but these versions are no longer supported by IBM.

  To remain consistent with IBM's standing agreement with our customers who use zOS and OS/400, IBM asks that these customers contact IBM Service for information regarding this vulnerability.

References

http://www.ee.oulu.fi/research/ouspg/protos/
http://www.kb.cert.org/vuls/id/854306
http://www.kb.cert.org/vuls/id/107186
http://www.cert.org/tech_tips/denial_of_service.html
http://www.ietf.org/rfc/rfc1067.txt
http://www.ietf.org/rfc/rfc1089.txt
http://www.ietf.org/rfc/rfc1140.txt
http://www.ietf.org/rfc/rfc1155.txt
http://www.ietf.org/rfc/rfc1156.txt
http://www.ietf.org/rfc/rfc1215.txt
http://www.ietf.org/rfc/rfc1270.txt
http://www.ietf.org/rfc/rfc1352.txt

Background information

- Background Information on the OUSPG

  OUSPG is an academic research group located at Oulu University in Finland. The purpose of this research group is to test software for vulnerabilities.

History has shown that the techniques used by the OUSPG have discovered a large number of previously undetected problems in the products and protocols they have tested. In 2001, the OUSPG produced a comprehensive test suite for evaluating implementations of the Lightweight Directory Access Protocol (LDAP). This test suite was developed with the strategy of abusing the protocol in unsupported and unexpected ways, and it was very effective in uncovering a wide variety of vulnerabilities across several products. This approach can reveal vulnerabilities that would not manifest themselves under normal conditions.

After completing its work on LDAP, OUSPG moved its focus to SNMPv1. As with LDAP, they designed a custom test suite, began testing a selection of products, and found a number of vulnerabilities. Because OUSPG's work on LDAP was similar in procedure to its current work on SNMP, you may wish to review the LDAP Test Suite and CERT Advisory CA-2001-18, which outlined results of application of the test suite.

In order to test the security of protocols like SNMPv1, the PROTOS project presents a server with a wide variety of sample packets containing unexpected values or illegally formatted data. As a member of the PROTOS project consortium, the OUSPG used the PROTOS c06-snmpv1 test suite to study several implementations of the SNMPv1 protocol. Results of the test suites run against SNMP indicate that there are many different vulnerabilities on many different implementations of SNMP.

- Background Information on the Simple Network Management Protocol

  The Simple Network Management Protocol (SNMP) is the most popular protocol in use to manage networked devices. SNMP was designed in the late 80's to facilitate the exchange of management information between networked devices, operating at the application layer of the ISO/OSI model. The SNMP protocol enables network and system administrators to remotely monitor and configure devices on the network (devices such as switches and routers). Software and firmware products designed for networks often make use of the SNMP protocol. SNMP runs on a multitude of devices and operating systems, including, but not limited to:

  - Core Network Devices (Routers, Switches, Hubs, Bridges, and Wireless Network Access Points)
  - Operating Systems
  - Consumer Broadband Network Devices (Cable Modems and DSL Modems)
  - Consumer Electronic Devices (Cameras and Image Scanners)
  - Networked Office Equipment (Printers, Copiers, and FAX Machines)
  - Network and Systems Management/Diagnostic Frameworks (Network Sniffers and Network Analyzers)

- Uninterruptible Power Supplies (UPS)
- Networked Medical Equipment (Imaging Units and Oscilloscopes)
- Manufacturing and Processing Equipment

The SNMP protocol is formally defined in RFC1157. Quoting from that RFC:

“Implicit in the SNMP architectural model is a collection of network management stations and network elements. Network management stations execute management applications which monitor and control network elements. Network elements are devices such as hosts, gateways, terminal servers, and the like, which have management agents responsible for performing the network management functions requested by the network management stations. The Simple Network Management Protocol (SNMP) is used to communicate management information between the network management stations and the agents in the network elements.”

Additionally, SNMP is discussed in a number of other RFC documents:

- RFC 3000 Internet Official Protocol Standards
- RFC 1212 Concise MIB Definitions
- RFC 1213 Management Information Base for Network Management of TCP/IP-based Internets: MIB-II
- RFC 1215 A Convention for Defining Traps for use with the SNMP
- RFC 1270 SNMP Communications Services
- RFC 2570 Introduction to Version 3 of the Internet-standard Network Management Framework
- RFC 2571 An Architecture for Describing SNMP Management Frameworks
- RFC 2572 Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)
- RFC 2573 SNMP Applications
- RFC 2574 User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)
- RFC 2575 View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)
- RFC 2576 Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework

The CERT Coordination Center thanks the Oulu University Secure Programming Group for reporting these vulnerabilities to us, for providing detailed technical analyses, and for assisting us in preparing this advisory. We also thank Steven M. Bolivian (AT&T Labs -- Research), Wes Hardaker (Net-SNMP), Steve Moulton (SNMP Research), Tom Reddington (Bell Labs), Mike Duckett (Bell South), Rob Thomas, Blue Boar (Thievco), and the many others who contributed to this document.

Feedback on this document can be directed to the authors, Ian A. Finlay, Shawn V. Hernan, Jason A. Rafail, Chad Dougherty, Allen D. Householder, Marty Lindner, and Art Manion.

This document is available from:

`http://www.cert.org/advisories/CA-2002-03.html`

- CERT/CC Contact Information

  Email: cert@cert.org

  Phone: +1 412-268-7090 (24-hour hotline)

  Fax: +1 412-268-6989

  Postal address:

  CERT Coordination Center

  Software Engineering Institute

  Carnegie Mellon University

  Pittsburgh PA 15213-3890

  U.S.A.

  CERT/CC personnel answer the hotline 08:00-17:00 EST(GMT-5) / EDT(GMT-4) Monday through Friday; they are on call for emergencies during other hours, on U.S. holidays, and on weekends.

- Using encryption

  We strongly urge you to encrypt sensitive information sent by email. Our public PGP key is available from http://www.cert.org/CERT_PGP.key. If you prefer to use DES, please call the CERT hotline for more information.

- Getting security information

  CERT publications and other security information are available from our web site:

  `http://www.cert.org/`

  To subscribe to the CERT mailing list for advisories and bulletins, send email to majordomo@cert.org. Please include in the body of your message:

```
subscribe cert-advisory
```

  - *CERT* and *CERT Coordination Center* are registered in the U.S. Patent and Trademark Office.

- NO WARRANTY

  Any material furnished by Carnegie Mellon University and the Software Engineering Institute is furnished on an *as is* basis. Carnegie Mellon University makes no warranties of any kind, either expressed or implied as to any matter including, but not limited to, warranty of fitness for a particular purpose or merchantability, exclusivity or results obtained from use of the material. Carnegie Mellon University does not make any warranty of any kind with respect to freedom from patent, trademark, or copyright infringement.

- Conditions for use, disclaimers, and sponsorship information

  Copyright 2002 Carnegie Mellon University.

- Revision History

  Feb 26, 2002: Updated vendor statement for IBM.

EXAMPLE.conf

Example A-2 is an example configuration file for configuring the ucd-snmp snmpd agent.

Example: A-2 EXAMPLE.conf

```
##############################################################################
#
# EXAMPLE.conf:
#   An example configuration file for configuring the ucd-snmp snmpd agent.
#
##############################################################################
#
# This file is intended to only be an example.  If, however, you want
# to use it, it should be placed in PREFIX/share/snmp/snmpd.conf.
# When the snmpd agent starts up, this is where it will look for it.
#
# You might be interested in generating your own snmpd.conf file using
# the “snmpconf” program (perl script) instead.  It’s a nice menu
# based interface to writing well commented configuration files.  Try it!
#
# Note: This file is automatically generated from EXAMPLE.conf.def.
# Do NOT read the EXAMPLE.conf.def file! Instead, after you have run
# configure & make, and then make sure you read the EXAMPLE.conf file
# instead, as it will tailor itself to your configuration.
# All lines beginning with a ‘#’ are comments and are intended for you
# to read.  All other lines are configuration commands for the agent.

#
# PLEASE: read the snmpd.conf(5) manual page as well!
#


###############################################################################
# Access Control
###############################################################################

# YOU SHOULD CHANGE THE “COMMUNITY” TOKEN BELOW TO A NEW KEYWORD ONLY
# KNOWN AT YOUR SITE.  YOU *MUST* CHANGE THE NETWORK TOKEN BELOW TO
# SOMETHING REFLECTING YOUR LOCAL NETWORK ADDRESS SPACE.

# By far, the most common question I get about the agent is “why won’t
# it work?”, when really it should be “how do I configure the agent to
# allow me to access it?”
#
# By default, the agent responds to the “public” community for read
# only access, if run out of the box without any configuration file in
# place.  The following examples show you other ways of configuring
# the agent so that you can change the community names, and give
# yourself write access as well.
#
# The following lines change the access permissions of the agent so
# that the COMMUNITY string provides read-only access to your entire
# NETWORK (EG: 10.10.10.0/24), and read/write access to only the
# localhost (127.0.0.1, not its real ipaddress).
#
# For more information, read the FAQ as well as the snmpd.conf(5)
# manual page.

####
# First, map the community name (COMMUNITY) into a security name
# (local and mynetwork, depending on where the request is coming
# from):

#       sec.name  source          community
com2sec local     localhost       COMMUNITY
com2sec mynetwork NETWORK/24      COMMUNITY

####
# Second, map the security names into group names:

#             sec.model  sec.name
group MyRWGroupv1         local
group MyRWGroupv2c        local
group MyRWGroupusm        local
group MyROGroup v1         mynetwork
group MyROGroup v2c        mynetwork
group MyROGroup usm        mynetwork

####
# Third, create a view for us to let the groups have rights to:

#           incl/excl subtree                          mask
view all    included  .1                               80

####
# Finally, grant the 2 groups access to the 1 view with different
# write permissions:

#                context sec.model sec.level match  read   write  notif
access MyROGroup ““      any       noauth    exact  all    none   none
access MyRWGroup ““      any       noauth    exact  all    all    none

# -----------------------------------------------------------------------------


###############################################################################
# System contact information
#

# It is also possible to set the sysContact and sysLocation system
# variables through the snmpd.conf file:

syslocation Right here, right now.
syscontact Me <me@somewhere.org>

# Example output of snmpwalk:
#   % snmpwalk -v 1 localhost public system
#   system.sysDescr.0 = “SunOS name sun4c”
#   system.sysObjectID.0 = OID: enterprises.ucdavis.ucdSnmpAgent.sunos4
#   system.sysUpTime.0 = Timeticks: (595637548) 68 days, 22:32:55
#   system.sysContact.0 = “Me <me@somewhere.org>”
#   system.sysName.0 = “name”
#   system.sysLocation.0 = “Right here, right now.”
#   system.sysServices.0 = 72


# -----------------------------------------------------------------------------


###############################################################################
# Process checks.
#
```

```
#  The following are examples of how to use the agent to check for
#  processes running on the host.  The syntax looks something like:
#
#  proc NAME [MAX=0] [MIN=0]
#
#  NAME:  the name of the process to check for.  It must match
#         exactly (ie, http will not find httpd processes).
#  MAX:   the maximum number allowed to be running.  Defaults to 0.
#  MIN:   the minimum number to be running.  Defaults to 0.

#
#  Examples:
#

#  Make sure mountd is running
proc mountd

#  Make sure there are no more than 4 ntalkds running, but 0 is ok too.
proc ntalkd 4

#  Make sure at least one sendmail, but less than or equal to 10 are running.
proc sendmail 10 1

#  A snmpwalk of the prTable would look something like this:
#
# % snmpwalk -v 1 localhost public .EXTENSIBLEDOTMIB.PROCMIBNUM
# enterprises.ucdavis.procTable.prEntry.prIndex.1 = 1
# enterprises.ucdavis.procTable.prEntry.prIndex.2 = 2
# enterprises.ucdavis.procTable.prEntry.prIndex.3 = 3
# enterprises.ucdavis.procTable.prEntry.prNames.1 = “mountd”
# enterprises.ucdavis.procTable.prEntry.prNames.2 = “ntalkd”
# enterprises.ucdavis.procTable.prEntry.prNames.3 = “sendmail”
# enterprises.ucdavis.procTable.prEntry.prMin.1 = 0
# enterprises.ucdavis.procTable.prEntry.prMin.2 = 0
# enterprises.ucdavis.procTable.prEntry.prMin.3 = 1
# enterprises.ucdavis.procTable.prEntry.prMax.1 = 0
# enterprises.ucdavis.procTable.prEntry.prMax.2 = 4
# enterprises.ucdavis.procTable.prEntry.prMax.3 = 10
# enterprises.ucdavis.procTable.prEntry.prCount.1 = 0
# enterprises.ucdavis.procTable.prEntry.prCount.2 = 0
# enterprises.ucdavis.procTable.prEntry.prCount.3 = 1
# enterprises.ucdavis.procTable.prEntry.prErrorFlag.1 = 1
# enterprises.ucdavis.procTable.prEntry.prErrorFlag.2 = 0
# enterprises.ucdavis.procTable.prEntry.prErrorFlag.3 = 0
# enterprises.ucdavis.procTable.prEntry.prErrMessage.1 = “No mountd process
running.”
# enterprises.ucdavis.procTable.prEntry.prErrMessage.2 = ““
# enterprises.ucdavis.procTable.prEntry.prErrMessage.3 = ““
# enterprises.ucdavis.procTable.prEntry.prErrFix.1 = 0
# enterprises.ucdavis.procTable.prEntry.prErrFix.2 = 0
# enterprises.ucdavis.procTable.prEntry.prErrFix.3 = 0
#
#  Note that the errorFlag for mountd is set to 1 because one is not
#  running (in this case an rpc.mountd is, but thats not good enough),
#  and the ErrMessage tells you what's wrong.  The configuration
#  imposed in the snmpd.conf file is also shown.
#
#  Special Case:  When the min and max numbers are both 0, it assumes
#  you want a max of infinity and a min of 1.
#


# -----------------------------------------------------------------------------


###############################################################################
# Executables/scripts
#

#
#  You can also have programs run by the agent that return a single
#  line of output and an exit code.  Here are two examples.
#
#  exec NAME PROGRAM [ARGS ...]
#
#  NAME:     A generic name.
#  PROGRAM:  The program to run.  Include the path!
#  ARGS:     optional arguments to be passed to the program

# a simple hello world
exec echotest /bin/echo hello world

# Run a shell script containing:
#
# #!/bin/sh
# echo hello world
# echo hi there
# exit 35
#
# Note:  this has been specifically commented out to prevent
# accidental security holes due to someone else on your system writing
# a /tmp/shtest before you do.  Uncomment to use it.
#
#exec shelltest /bin/sh /tmp/shtest

# Then,
# % snmpwalk -v 1 localhost public .EXTENSIBLEDOTMIB.SHELLMIBNUM
# enterprises.ucdavis.extTable.extEntry.extIndex.1 = 1
# enterprises.ucdavis.extTable.extEntry.extIndex.2 = 2
# enterprises.ucdavis.extTable.extEntry.extNames.1 = “echotest”
# enterprises.ucdavis.extTable.extEntry.extNames.2 = “shelltest”
# enterprises.ucdavis.extTable.extEntry.extCommand.1 = “/bin/echo hello world”
# enterprises.ucdavis.extTable.extEntry.extCommand.2 = “/bin/sh /tmp/shtest”
# enterprises.ucdavis.extTable.extEntry.extResult.1 = 0
# enterprises.ucdavis.extTable.extEntry.extResult.2 = 35
# enterprises.ucdavis.extTable.extEntry.extOutput.1 = “hello world.”
# enterprises.ucdavis.extTable.extEntry.extOutput.2 = “hello world.”
# enterprises.ucdavis.extTable.extEntry.extErrFix.1 = 0
# enterprises.ucdavis.extTable.extEntry.extErrFix.2 = 0

# Note that the second line of the /tmp/shtest shell script is cut
# off.  Also note that the exit status of 35 was returned.

# -----------------------------------------------------------------------------


###############################################################################
# disk checks
#

# The agent can check the amount of available disk space, and make
# sure it is above a set limit.

# disk PATH [MIN=DEFDISKMINIMUMSPACE]
#
# PATH:  mount path to the disk in question.
# MIN:   Disks with space below this value will have the Mib’s errorFlag set.
#        Default value = DEFDISKMINIMUMSPACE.

# Check the / partition and make sure it contains at least 10 megs.

disk / 10000

# % snmpwalk -v 1 localhost public .EXTENSIBLEDOTMIB.DISKMIBNUM
# enterprises.ucdavis.diskTable.dskEntry.diskIndex.1 = 0
# enterprises.ucdavis.diskTable.dskEntry.diskPath.1 = “/” Hex: 2F
# enterprises.ucdavis.diskTable.dskEntry.diskDevice.1 = “/dev/dsk/c201d6s0”
# enterprises.ucdavis.diskTable.dskEntry.diskMinimum.1 = 10000
# enterprises.ucdavis.diskTable.dskEntry.diskTotal.1 = 837130
# enterprises.ucdavis.diskTable.dskEntry.diskAvail.1 = 316325
# enterprises.ucdavis.diskTable.dskEntry.diskUsed.1 = 437092
# enterprises.ucdavis.diskTable.dskEntry.diskPercent.1 = 58
# enterprises.ucdavis.diskTable.dskEntry.diskErrorFlag.1 = 0
# enterprises.ucdavis.diskTable.dskEntry.diskErrorMsg.1 = ““

# -----------------------------------------------------------------------------
```

```
################################################################################
# load average checks
#

# load [1MAX=DEFMAXLOADAVE] [5MAX=DEFMAXLOADAVE] [15MAX=DEFMAXLOADAVE]
#
# 1MAX:   If the 1 minute load average is above this limit at query
#         time, the errorFlag will be set.
# 5MAX:   Similar, but for 5 min average.
# 15MAX:  Similar, but for 15 min average.

# Check for loads:
load 12 14 14

# % snmpwalk -v 1 localhost public .EXTENSIBLEDOTMIB.LOADAVEMIBNUM
# enterprises.ucdavis.loadTable.laEntry.loadaveIndex.1 = 1
# enterprises.ucdavis.loadTable.laEntry.loadaveIndex.2 = 2
# enterprises.ucdavis.loadTable.laEntry.loadaveIndex.3 = 3
# enterprises.ucdavis.loadTable.laEntry.loadaveNames.1 = “Load-1”
# enterprises.ucdavis.loadTable.laEntry.loadaveNames.2 = “Load-5”
# enterprises.ucdavis.loadTable.laEntry.loadaveNames.3 = “Load-15”
# enterprises.ucdavis.loadTable.laEntry.loadaveLoad.1 = “0.49” Hex: 30 2E 34 39
# enterprises.ucdavis.loadTable.laEntry.loadaveLoad.2 = “0.31” Hex: 30 2E 33 31
# enterprises.ucdavis.loadTable.laEntry.loadaveLoad.3 = “0.26” Hex: 30 2E 32 36
# enterprises.ucdavis.loadTable.laEntry.loadaveConfig.1 = “12.00”
# enterprises.ucdavis.loadTable.laEntry.loadaveConfig.2 = “14.00”
# enterprises.ucdavis.loadTable.laEntry.loadaveConfig.3 = “14.00”
# enterprises.ucdavis.loadTable.laEntry.loadaveErrorFlag.1 = 0
# enterprises.ucdavis.loadTable.laEntry.loadaveErrorFlag.2 = 0
# enterprises.ucdavis.loadTable.laEntry.loadaveErrorFlag.3 = 0
# enterprises.ucdavis.loadTable.laEntry.loadaveErrMessage.1 = ““
# enterprises.ucdavis.loadTable.laEntry.loadaveErrMessage.2 = ““
# enterprises.ucdavis.loadTable.laEntry.loadaveErrMessage.3 = ““

# ------------------------------------------------------------------------------


################################################################################
# Extensible sections.
#

# This alleviates the multiple line output problem found in the
# previous executable mib by placing each mib in its own mib table:

# Run a shell script containing:
#
# #!/bin/sh
# echo hello world
# echo hi there
# exit 35
#
# Note:  this has been specifically commented out to prevent
# accidental security holes due to someone else on your system writing
# a /tmp/shtest before you do.  Uncomment to use it.
#
# exec .EXTENSIBLEDOTMIB.50 shelltest /bin/sh /tmp/shtest

# % snmpwalk -v 1 localhost public .EXTENSIBLEDOTMIB.50
# enterprises.ucdavis.50.1.1 = 1
# enterprises.ucdavis.50.2.1 = “shelltest”
# enterprises.ucdavis.50.3.1 = “/bin/sh /tmp/shtest”
# enterprises.ucdavis.50.100.1 = 35
# enterprises.ucdavis.50.101.1 = “hello world.”
# enterprises.ucdavis.50.101.2 = “hi there.”
# enterprises.ucdavis.50.102.1 = 0

# Now the Output has grown to two lines, and we can see the ‘hi
# there.’ output as the second line from our shell script.
#
# Note that you must alter the mib.txt file to be correct if you want
# the .50.* outputs above to change to reasonable text descriptions.

# Other ideas:
#
# exec .EXTENSIBLEDOTMIB.51 ps /bin/ps
# exec .EXTENSIBLEDOTMIB.52 top /usr/local/bin/top
# exec .EXTENSIBLEDOTMIB.53 mailq /usr/bin/mailq

# -----------------------------------------------------------------------------


###############################################################################
# Pass through control.
#

# Usage:
#   pass MIBOID EXEC-COMMAND
#
# This will pass total control of the mib underneath the MIBOID
# portion of the mib to the EXEC-COMMAND.
#
# Note:  You’ll have to change the path of the passtest script to your
# source directory or install it in the given location.
#
# Example:  (see the script for details)
#           (commented out here since it requires that you place the
#           script in the right location. (its not installed by default))
# pass .EXTENSIBLEDOTMIB.255 /bin/sh PREFIX/local/passtest

# % snmpwalk -v 1 localhost public .EXTENSIBLEDOTMIB.255
# enterprises.ucdavis.255.1 = “life the universe and everything”
# enterprises.ucdavis.255.2.1 = 42
# enterprises.ucdavis.255.2.2 = OID: 42.42.42
# enterprises.ucdavis.255.3 = Timeticks: (363136200) 42 days, 0:42:42
# enterprises.ucdavis.255.4 = IpAddress: 127.0.0.1
# enterprises.ucdavis.255.5 = 42
# enterprises.ucdavis.255.6 = Gauge: 42
#
# % snmpget -v 1 localhost public .EXTENSIBLEDOTMIB.255.5
# enterprises.ucdavis.255.5 = 42
#
# % snmpset -v 1 localhost public .EXTENSIBLEDOTMIB.255.1 s “New string”
# enterprises.ucdavis.255.1 = “New string”
#

# For specific usage information, see the man/snmpd.conf.5 manual page
# as well as the local/passtest script used in the above example.

###############################################################################
# Subagent control
#

# The agent can support subagents using a number of extension mechanisms.
# From the 4.2.1 release, AgentX support is being compiled in by default.
# However, this is still experimental code, so should not be used on
# critical production systems.
#   Please see the file README.agentx for more details.
#
# If having read, marked, learnt and inwardly digested this information,
# you decide that you do wish to make use of this mechanism, simply
# uncomment the following directive.
#
#  master  agentx
#
# I repeat - this is *NOT* regarded as suitable for front-line production
# systems, though it is probably stable enough for day-to-day use.
# Probably.
#
# No refunds will be given.


###############################################################################
# Further Information
#
#  See the snmpd.conf manual page, and the output of “snmpd -H”.
#  MUCH more can be done with the snmpd.conf than is shown as an
#  example here.
```

B

Example scripts

This appendix contains the following example scripts used in this redbook:

- pschk2 (see Example B-1)
- hwalert (see Example B-2 on page 315)
- makebff (see Example B-3 on page 317)
- drawdailygrpaphcpu.ksh (see Example B-4 on page 323)
- drawdailygraphpag.ksh (see Example B-5 on page 325)
- listhwlevels (see Example B-6 on page 327)
- listswlevels (see Example B-7 on page 328)

Example: B-1 .pschk2

```
#!/usr/bin/perl
$cachefile = "/usr/local/share/snmp/pschk2.cache";
$top = ".1.3.6.1.4.1.2021.56";

$cmd=$ARGV[0];
$oid=$ARGV[1];

if ($cmd eq "-s") {       # will not accept any SET operation
    print "not-writable\n";
    exit 0;
} elsif ($cmd eq "-g" || $cmd eq "-n") {
    &tableset;
    if ($cmd eq "-n") {   # GETNEXT
        unless ($req=$onlist{$oid}){  # is there next OID?
            exit 0;
        }
    } elsif ($cmd eq "-g") {     # GET
        unless ($tbt{$oid}) {    # is the OID requested available?
            exit 0;
        }
        $req=$oid;
    }
    printf("%s\n",$req);       # 1st line of output, OID
    printf("%s\n",$tbt{$req}); # 2nd line of output, type
    printf("%s\n",$tbv{$req}); # 3rd line of output, value
}
exit 0;

sub tableset{
    $tbt{"$top.1"} = "string";       # 1st OID is string type
    $tbv{"$top.1"} = "Paging space"; # 1st OID value is title
    @olist=("$top.1");             # initialize OID list
    ($dev, $ino, $mode, $nlink
    , $uid, $gid, $rdev, $size
    , $atime, $mtime, $ctime, $blksize, $blocks)
    = stat $cachefile;             # get timestamp of cache file
    $now = time();                 # get time of now
    if ($now - $mtime > 30) {      # if the cache file is 30 sec. older
        system("lsps -a > $cachefile"); # refresh the contents
    }
    $n=0;
    open(TT, "< $cachefile");
    while (<TT>) {
        next if (/^Page Space/);  # Ignore header line
        @ps = split(" ", $_); # parse fields delimited by :
        $ps[5] = ($ps[5] == "yes" ? 1 : 0); # let yes,no to be integer value
                                        # such as 1,0
        for ($l = 0; $l < 6; $l++) {
            $o = sprintf("%s.%d.%d", $top, $n+1, $l+1);
            push(olist, $o);       # register OID
            $tbt{$o} = ($ps[$l]=~/^[0-9]+$/ ? "integer" : "string"); # setup
type
            $tbv{$o} = $ps[$l];     # setup value
        }
        $n++;
    }
    close(TT);
    %onlist=();
    $n=0;
    for $i ($top, @olist) {    # setup refrence table for GETNEXT operation
        $onlist{$i} = $olist[$n++];
    }
}
```

Example: B-2 hwalert

```
#!/bin/ksh
#################################################################
#
# Program  : hwalert
#
# Function : Send alerts for hardware Errors using the sendtrap script.
#            This Program, once enabled, is called automatically every time
#            a Permanent H/W Error is logged. This is achieved by inserting
#            an entry into the odm which tells the system to run this script
#            every time a particular error occurs.
#
# Parameters : hwalert {ENABLE | DISABLE | $* from odm}
#
#            If you wish to exclude certain hardware alerts add the Error Label
#            in the /etc/hwalert.exclude file.
#            Any errors with that label will be ignored.
#################################################################

# This function actually changes the ODM entry.
# The ENABLE option will not remove any currently, then add the entry in.
function manipulate_odm
{
    export ODMDIR="/etc/objrepos"

    if [[ $1 = "ENABLE" ]] ; then
        odmdelete -q "en_name=hwalert" -o errnotify > /dev/null
        odmdelete -q "en_name=OPMSG" -o errnotify > /dev/null
        odmdelete -q "en_name=custom" -o errnotify > /dev/null

        # Add an ODM entry for PERM errors - this is a default entry
        odmadd <<END__OF__ODM
errnotify:
en_pid = 0
en_name = "hwalert"
en_persistenceflg = 1
en_label = ""
en_crcid = 0
en_class = "H"
en_type = "PERM"
en_alertflg = ""
en_resource = ""
en_rtype = ""
en_rclass = ""
en_method = "$ROOT_PATH/hwalert \$1 \$2 \$3 \$4 \$5 \$6 \$7 \$8 \$9 \$10 \$11"
END__OF__ODM

        errors=$(($errors+$?))

        # Now add entries for other errpt events specified in config
        if [[ -f /etc/errpt.alert ]] ; then
            cat /etc/errpt.alert | \
        # The following awk command must be one contiguous line.
awk -F: '!/^([ ]*#.*|[ ]*)$/{printf("errnotify:\nen_pid = 0\nen_name =
\"custom\"\nen_persistenceflg = 1\nen_label = \"\"\nen_crcid = \"\"\nen_class =
\"%s\"\nen_type = \"%s\"\nen_alertflg= \"\"\nen_resource = \"%s\"\nen_rtype =
\"\"\nen_rclass = \"\"\nen_method = \"'$ROOT_PATH'/hwalert \$1 \$2 \$3 \$4 \$5
\$6 \$7 \$8 \$9 \$10 \$11\"\n\n",$1,$2,$3);}'  |\
            odmadd
        fi

        if [[ $errors -ne 0 ]] ; then
            /usr/local/bin/sendtrap 4 "Failed to enable errpt /hardware
alerting - errpt/hardware alerting inoperable!"
        fi

    else
        # Remove all errpt alerting
        odmdelete -q"en_name=hwalert" -o errnotify > /dev/null 2>&1
        odmdelete -q"en_name=custom" -o errnotify > /dev/null 2>&1
    fi
}

# Main routine.

ROOT_PATH="/usr/local/bin"

if [[ $# -eq 0 ]] ; then
    print "Usage: ${0} ENABLE | DISABLE"
    exit 0
fi

# Check if it is running as root user.
if [[ $(id -u) -ne 0 ]] ; then
    print "$0 must be run as root"
    exit
fi

if [[ $1 = +(ENABLE|DISABLE) ]] ; then
    manipulate_odm $1
    exit 0
fi

Error_Seq=$1
```

```
Identifier=$2
Device=$6
Label=$9

# There may be some hardware errors we wish to exclude. Therefore,
# check that the error type received does not exist in the hwalert.exclude.

if [[ -e /etc/hwalert.exclude ]] && grep -q "^$Label$" /etc/hwalert.exclude
then
    exit 0
fi

print "1 is $1 2 is $2 3 is $3 4 is $4 5 is $5 6 is $6 7 is $7" > /tmp/ping

Message="System error $Label on $Device - please investigate errpt for details"

# Send alert via sendtrap.
/usr/local/bin/sendtrap 2 "$Message"
```

Example: B-3 makebff

```
#!/usr/bin/perl
#
#    This program is free software; you can redistribute it and/or modify
#    it under the terms of the GNU General Public License as published by
#    the Free Software Foundation; either version 2 of the License, or
#    (at your option) any later version.
#
#    This program is distributed in the hope that it will be useful,
#    but WITHOUT ANY WARRANTY; without even the implied warranty of
#    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
#    GNU General Public License for more details.
#
#    You should have received a copy of the GNU General Public License
#    along with this program; if not, write to the Free Software
#    Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
#

printf(“makebff version 1.0.0.7\n”);

$package = ””;
$version = ””;
$packagetype = ””;
$fileset = ””;
%version = ();
%description = ();
$line = 0;
$ic = 0;
@list = ();
```

```
@requisite = ();
@libfiles = (“copyright”,”README”,”cfginfo”,”cfgfiles”,”err”
    , “fixdata”,”namelist”,”odmadd”,”rm_inv”,”trc”,”odmdel”
    , “pre_d”,”pre_i”,”pre_u”,”pre_rm”,”post_i”,”post_u”,”unconfig”
    , “unconfig_u”,”unodmadd”,”unpost_i”,”unpost_u”,”unpre_i”,”unpre_u”
);

push @list, “./.info/README” if ( -f “./.info/README” );

open (LPP, ”>lpp_name”);

mkdir(“./usr”, 0755) if ( ! -d “./usr”);
symlink(“/etc”, ”./etc”) if ( ! -d “./etc”);
symlink(“/var”, ”./var”) if ( ! -d “./var”);
symlink(“/opt”, ”./opt”) if ( ! -d “./opt”);
symlink(“/usr/local”, ”./usr/local”) if ( ! -s “./usr/local”);

unless ( -f “./.info/list” ) {
  die “./.info/list is not exists.”;
}

open(LIST,”./.info/list”);
while (<LIST>) {
    $line++;
    chop;
    next if (/^#/ || /^$/);
    s/^ *//;
    s/ *$//;
    s/ +/ /;
    s/ /:/;
    s/ /:/;
    @t=split(/:/, $_);
    if ($package eq ““) {
        $package = $t[0];
        $version = $t[1];
        if ($t[2]) {
            $packagetype = $t[2];
        } else {
            $packagetype = ”I”;
        }
        printf(LPP “4 R %s %s {\n”, $packagetype, $package);
        printf(“$package $version $packagetype\n”);
    } elsif (/^\*/) {
        if ($fileset) {
            push @requisite, “@t”;
        } else {
            die “requisite line for empty fileset. ($line)\n”;
        }
    } else {
        if ($fileset && $fileset ne $t[0]) {
            &filesetinfo;
        }
        $fileset = $t[0];
        $version{$fileset} = $t[1];
        $description{$fileset} = $t[2];
        @requisite = ();
        printf(“processing $t[0]\n”);
        $al = ”./.info/”.$t[0] . ”.al”;
        $inv = ”./.info/”.$t[0] . ”.inventory”;
        %size = &makeinv($al, $inv);
    }
}
&filesetinfo;
printf(LPP “}\n”);
close(LPP);
close(LIST);

mkdir(“./usr/lpp”, 0755) if ( ! -d “./usr/lpp”);
mkdir(“./usr/lpp/$package”, 0755) if ( ! -d “./usr/lpp/$package”);

@liblpp = ();
if ($packagetype eq “I”) {
    printf(“creating ./.info/liblpp.a\n”);
    unlink(“./.info/liblpp.a”);
    system(“ar -q ./.info/liblpp.a @list”);
    system(“cp ./.info/liblpp.a ./usr/lpp/$package”);
    push @liblpp, “./usr/lpp/$package/liblpp.a”;
} else {
    for $i (keys %version) {
        $d=sprintf(“./usr/lpp/%s/%s”, $package, $i);
        mkdir($d, 0755) if ( ! -d $d);
        $d = sprintf(“./usr/lpp/%s/%s/%s”, $package, $i, $version{$i});
        mkdir($d, 0755) if ( ! -d $d);
        printf(“copying ./.info/$i.a to $d/liblpp.a\n”);
        system(“cp ./.info/$i.a $d/liblpp.a”);
        push @liblpp, “$d/liblpp.a”;
    }
}

mkdir(“./tmp”, 0755) if ( ! -d “./tmp”);

printf(“creating ./tmp/$package.$version.bff\n”);
open(TT, ”|backup -ipqf - > ./tmp/$package.$version.bff”);
print TT “./lpp_name\n”;
for $i (@liblpp) {
  print TT “$i\n”;
}
foreach $i (keys %version) {
    $t=”.info/$i.al”;
    open(AL, ”<$t”);
    while (<AL>) {
        print TT;
    }
    close(AL);
}
close(TT);
exit(0);


sub filesetinfo {
    local ($ts);
    local ($dev, $ino, $mode, $nlink, $uid, $gid, $rdev
        , $size, $atime, $mtime, $ctime, $blksize, $blocks);

    $ts = 0;
    printf(LPP “%s %s 01 N U en_US %s\n”
        , $fileset,$version{$fileset},$description{$fileset});
    printf(LPP “[\n”);
    for $i (@requisite) {
        printf(LPP “%s\n”, $i);
    }
    printf(LPP “%%\n”);
    for $i (sort keys %size) {
        printf(LPP “%s %d\n”, $i, $size{$i});
        $ts += $size{$i};
    }
    if ($packagetype =~ /^S/) {
        printf(LPP “/usr/lpp/SAVESPACE %d\n”, $ts);
    }
    @etc = ();
    for $i (@libfiles) {
        $t = ”.info/$fileset.$i”;
        push @etc, $t if ( -f $t );
    }
    if ($packagetype =~ /^S/) {
        printf(“creating ./.info/$fileset.a\n”);
        unlink(“./.info/$fileset.a”);
        system(“ar -q ./.info/$fileset.a $al $inv @etc”);
        ($dev, $ino, $mode, $nlink, $uid, $gid, $rdev, $size
            , $atime, $mtime, $ctime, $blksize, $blocks)
            = stat “./.info/$fileset.a”;
        printf(LPP “/usr/lpp/%s/%s/%s %d\n”
            , $package, $fileset, $version{$fileset}, $blocks);
    } else {
        push @list, $al, $inv, @etc;
    }
    printf(LPP “%%\n”);
    printf(LPP “%%\n”);
    printf(LPP “%%\n”);
    printf(LPP “%%\n”);
    printf(LPP “]\n”);
}

sub makeinv {
    local ($fi, $fo) = @_;
    local ($name, $passwd, $pid, $key);
    local ($dev, $ino, $mode, $nlink, $uid, $gid, $rdev, $size
        , $atime, $mtime, $ctime, $blksize, $blocks);
    local (%list, $t, $dir, %tc, @ilist);
    local (%name, %owner, %group, %mode, %type, %size, %checksum);
    local (%target, %list, %nlink);

    %list = ();
    %tc = ();
    $c = 0;
    @ilist = ();
    %name = ();
    %owner = ();
    %group = ();
    %mode = ();
    %type = ();
    %size = ();
    %checksum = ();
    %target = ();
    %list = ();
    %nlink = ();

    while (($name, $passwd, $gid) = getgrent) {
        $gname{$gid} = $name;
    }
    while (($name, $passwd, $uid) = getpwent){
        $uname{$uid} = $name;
    }
    open(FI, ”<$fi”);
    open(FO, ”>$fo”);
    while (<FI>) {
        chop;
        $dir = $_;
        if (-l) {
            $type = ”SYMLINK”;
            $dir =~ s?/[^/]*$??;
        } elsif (-d) {
            $type = ”DIRECTORY”;
        } elsif (-f) {
            $type = ”FILE”;
            $dir =~ s?/[^/]*$??;
        } else {
            printf(STDERR “no such file: %s\n”, $_);
            next;
        }
        if ($type eq “SYMLINK”) {
            $key = $ic++;
        } else {
        ($dev, $ino, $mode, $nlink, $uid, $gid, $rdev, $size
            , $atime, $mtime, $ctime, $blksize, $blocks)
            = stat “$_”;
        $key = ”$dev.$ino”;
        }
        if ($name{$key} && ($nlink > 1) && ($type eq “FILE”)) {
            if ($list{$key}) {
                $list{$key} .= “,$_”;
            } else {
                $list{$key} = $_;
            }
        } else {
            push(@ilist, $key);
            $name{$key} = $_;
            $owner{$key} = $uname{$uid};
            $group{$key} = $gname{$gid};
            $mode{$key} = $mode;
            $type{$key} = $type;
            $size{$key} = $size;
            if ($type eq “FILE”) {
                $t = ‘/usr/bin/sum $_‘;
                $t =~ s/ +/ /g;
                ($a, $b, $c) = split(/ /, $t);
                $checksum{$key} = sprintf(“\”%s   %s \””, $a, $b);
            } elsif ($type eq “SYMLINK”) {
                $target{$key} = readlink($_);
            }
            $list{$key} = ””;
            $tc{$dir} += $blocks;
        }
    }

    for $key (@ilist) {
        $n = $name{$key};
        $n =~ s/^.//;
        printf(FO “%s:\n”, $n);
        if ($type{$key} eq “FILE” || $type{$key} eq “DIRECTORY”) {
            printf(FO “\towner = %s\n”, $owner{$key});
            printf(FO “\tgroup = %s\n”, $group{$key});
            printf(FO “\tmode = %o\n”, $mode{$key} & 07777);
        }
        printf(FO “\ttype = %s\n”, $type{$key});
```

```
        if ($type{$key} eq “FILE”) {
            printf(FO “\tsize = %d\n”, $size{$key});
            printf(FO “\tchecksum = %s\n”, $checksum{$key});
            if ($list{$key}) {
                printf(FO “\tlink = %s\n”, $list{$key});
            }
        } elsif ($type{$key} eq “SYMLINK”) {
            printf(FO “\ttarget = %s\n”,$target{$key});
        }
        printf(FO “\n”);
    }
    close(FI);
    close(FO);
    return %tc;
}
```

Example: B-4 drawdailygraphcpu.ksh

```
#!/bin/ksh
#
# Program will output a png file with graph generated by GNUPlot
# Stats must be generated with supplied cpu_stats software
# Program should be run each day in cron
#
#

dayofweek=‘date +%u‘
dayofmonth=‘date +%d‘
date=‘date +%y-%m-%d‘
hostname=‘hostname -s‘

# Daily stats
filename=/tmp/dailystats.$$
echo $filename
zcat ‘find /stats/vm*.Z -mtime -1 | grep / | sort‘ >> $filename
if [ $? -ne 0 ] ; then
   print “Unable to create daily temporary file”
   exit 5
fi

echo “
set terminal png small color
set output \”/stats/$hostname.cpu.$date.daily.png\”
set title \”Daily CPU statistics - $hostname
set xlabel \”Time
set ylabel \”Percentage Used
set xdata time
set timefmt \”%H:%M:%S %d/%m/%Y\”
set xrange [*:]
set format x \”%H:%M \\\n %D\”
plot \”$filename\” using 1:16 smooth csplines title \”USER\” with lines ,
\”$filename\” using 1:17 smooth csplines title \”SYS\” with lines ,
\”$filename\” using 1:18 smooth csplines title \”IDLE\” with lines ,
\”$filename\” using 1:19 smooth csplines title \”WIO\” with lines
“ | /usr/local/bin/gnuplot

rm $filename


# Is it Monday ? Do the weekly stats...
if [ “$dayofweek” -eq 1 ] ; then
   filename=/tmp/weeklystats.$$
   zcat ‘find /stats/vm*.Z -mtime -7 | grep / | sort‘ >> $filename
   if [ $? -ne 0 ] ; then
      print “Unable to create weekly temporary file”
      exit 5
   fi

   echo “
set terminal png small color
set output \”/stats/$hostname.cpu.$date.weekly.png\”
set title \”Weekly CPU statistics - $hostname
set xlabel \”Date
set ylabel \”Percentage Used
set xdata time
set timefmt \”%H:%M:%S %d/%m/%Y\”
set xrange [*:]
set format x \”%D\”
plot \”$filename\” using 1:16 smooth sbezier title \”USER\” with lines ,
\”$filename\” using 1:17 smooth sbezier title \”SYS\” with lines ,
\”$filename\” using 1:18 smooth sbezier title \”IDLE\” with lines ,
\”$filename\” using 1:19 smooth sbezier title \”WIO\” with lines
“ | /usr/local/bin/gnuplot

   rm $filename
fi

# Is it First day of the month ? Do the monthly stats...
if [ “$dayofmonth” -eq 1 ] ; then
   filename=/tmp/monthlystats.$$
   zcat ‘find /stats/vm*.Z -mtime -31 | grep / | sort‘ >> /tmp/$filename
   if [ $? -ne 0 ] ; then
      print “Unable to create monthly temporary file”
      exit 5
   fi

   echo “
set terminal png small color
set output \"/stats/$hostname.cpu.$date.monthly.png\"
set title \"Monthly CPU statistics - $hostname
set xlabel \"Date
set ylabel \"Percentage Used
set xdata time
set timefmt \"%H:%M:%S %d/%m/%Y\"
set xrange [*:]
set format x \"%d \\\n %m\"
plot \"$filename\" using 1:16 smooth sbezier title \"USER\" with lines ,
\"$filename\" using 1:17 smooth sbezier title \"SYS\" with lines ,
\"$filename\" using 1:18 smooth sbezier title \"IDLE\" with lines ,
\"$filename\" using 1:19 smooth sbezier title \"WIO\" with lines
" | /usr/local/bin/gnuplot

   rm $filename
fi
```

Example: B-5   drawdailygraphpag.ksh

```
#!/bin/ksh
#
# Program will output a png file with graph generated by GNUPlot
# Stats must be generated with supplied cpu_stats software
# Program should be run each day in cron
#
#

dayofweek=`date +%u`
dayofmonth=`date +%d`
date=`date +%y-%m-%d`
hostname=`hostname -s`

# Daily stats
filename=/tmp/dailystatspag.$$
echo $filename
zcat `find /stats/vm*.Z -mtime -1 | grep / | sort` >> $filename
if [ $? -ne 0 ] ; then
   print "Unable to create daily temporary file"
   exit 5
fi

echo "
set terminal png small color
set output \"/stats/$hostname.page.$date.daily.png\"
set title \"Daily Paging statistics - $hostname
set xlabel \"Time
set ylabel \"Pages
set xdata time
set timefmt \"%H:%M:%S %d/%m/%Y\"
set xrange [*:]
set format x \”%H:%M \\\n %D\”
plot \”$filename\” using 1:7 smooth csplines title \”Pages In\” with lines ,
\”$filename\” using 1:8 smooth csplines title \”Pages Out\” with lines ,
\”$filename\” using 1:9 smooth csplines title \”Pages Freed\” with lines
“ | /usr/local/bin/gnuplot

rm $filename


# Is it Monday ? Do the weekly stats...
if [ “$dayofweek” -eq 1 ] ; then
   filename=/tmp/weeklystatspag.$$
   zcat ‘find /stats/vm*.Z -mtime -7 | grep / | sort‘ >> $filename
   if [ $? -ne 0 ] ; then
      print “Unable to create paging weekly temporary file”
      exit 5
   fi

   echo “
set terminal png small color
set output \”/stats/$hostname.pag.$date.weekly.png\”
set title \”Weekly Paging statistics - $hostname
set xlabel \”Date
set ylabel \”Pages
set xdata time
set timefmt \”%H:%M:%S %d/%m/%Y\”
set xrange [*:]
set format x \”%D\”
plot \”$filename\” using 1:7 smooth sbezier title \”Pages In\” with lines ,
\”$filename\” using 1:8 smooth sbezier title \”Pages Out\” with lines ,
\”$filename\” using 1:9 smooth sbezier title \”Pages Freed\” with lines
“ | /usr/local/bin/gnuplot

   rm $filename
fi

# Is it First day of the month ? Do the monthly stats...
if [ “$dayofmonth” -eq 1 ] ; then
   filename=/tmp/monthlystats.$$
   zcat ‘find /stats/vm*.Z -mtime -31 | grep / | sort‘ >> /tmp/$filename
   if [ $? -ne 0 ] ; then
      print “Unable to create monthly temporary file”
      exit 5
   fi

   echo “
set terminal png small color
set output \”/stats/$hostname.pag.$date.monthly.png\”
set title \”Monthly Paging statistics - $hostname
set xlabel \”Date
set ylabel \”Pages
set xdata time
set timefmt \”%H:%M:%S %d/%m/%Y\”
set xrange [*:]
set format x \”%d \\\n %m\”
plot \”$filename\” using 1:7 smooth sbezier title \”Pages In\” with lines ,
\”$filename\” using 1:8 smooth sbezier title \”Pages Out\” with lines ,
\”$filename\” using 1:9 smooth sbezier title \”Pages Freed\” with lines
“ | /usr/local/bin/gnuplot

   rm $filename
fi
```

Example: B-6   listhwlevels

```
#!/bin/ksh
# Run this in cron each night. It will replace the file each day with current
# levels of software and firmware.
# Blank line is placed between each section so we could use grep -p

HOSTNAME=`hostname -s`
OUTPUT=/tmp/${HOSTNAME}.listhwlevels

print "listhwlevels run on $HOSTNAME on `date`" > $OUTPUT
print >> $OUTPUT
print "SYSTEM ID" >> $OUTPUT
print "MODEL:`lsattr -El sys0 -a modelname | awk '{print $2}'`" >> $OUTPUT
print "SERIAL NO:`lsattr -El sys0 -a systemid | awk '{print $2}'`" >> $OUTPUT
print >> $OUTPUT
print "SYSTEM SETTINGS" >> $OUTPUT
lsattr -El sys0 >> $OUTPUT
print >> $OUTPUT
print "HARDWARE CONFIG" >> $OUTPUT
lscfg -vp >> $OUTPUT
print >> $OUTPUT
print "FILESYSTEM SETTINGS" >> $OUTPUT
df -kI >> $OUTPUT
print >> $OUTPUT
print "ALL VOLUME GROUPS" >> $OUTPUT
lsvg >> $OUTPUT
print >> $OUTPUT
print "VOLUME GROUPS ONLINE" >> $OUTPUT
lsvg -o >> $OUTPUT
print >> $OUTPUT
print "LOGICAL VOLUMES" >> $OUTPUT
for f in `lsvg -o`
do
```

```
  lsvg -l $f >> $OUTPUT
done
print >> $OUTPUT
print "PAGING SPACES" >> $OUTPUT
lsps -a >> $OUTPUT
print >> $OUTPUT
print "NETWORK INTERFACES" >> $OUTPUT
ifconfig -a
print >> $OUTPUT
print "NETWORK ROUTES" >> $OUTPUT
netstat -rn >> $OUTPUT
print >> $OUTPUT
print "MICROCODE LEVELS" >> $OUTPUT
lsmcode -A >> $OUTPUT
```

Example: B-7   listswlevels

```
#!/bin/ksh
# Run this in cron each night. It will replace the file each day with current
# levels of software and firmware.
# Blank line is placed between each section so we could use grep -p

HOSTNAME=`hostname -s`
OUTPUT=/tmp/${HOSTNAME}.listswlevels

print "listswlevels run on $HOSTNAME on `date`" > $OUTPUT
print >> $OUTPUT
print "SYSTEM ID" >> $OUTPUT
print "MODEL:`lsattr -El sys0 -a modelname | awk '{print $2}'`" >> $OUTPUT
print "SERIAL NO:`lsattr -El sys0 -a systemid | awk '{print $2}'`" >> $OUTPUT
print >> $OUTPUT
print "AIX LEVEL" >> $OUTPUT
oslevel -r >> $OUTPUT
print >> $OUTPUT
print "LPP LEVELS" >> $OUTPUT
lslpp -lac >> $OUTPUT
print >> $OUTPUT
print "RPM LEVELS" >> $OUTPUT
rpm -q -a >> $OUTPUT
print >> $OUTPUT
```

C

How to use CD-R on AIX

This appendix explains how to burn CD-R media on AIX. Although the CD-R device is quite common and cheap on PC, a few people understand how to use it on AIX, because IBM does not sell it on pSeries. However, why do we have to use CD-R device on AIX? Of course, you can burn CD-R media on your Windows-based PC and use that media on AIX if you share the file system using NFS. But in this case, you might lose:

1. The difference between small and capital letter of the file name, such as my_filename and My_Filename.
2. Symbolic links. Because the typical CD burning software (actually, the software that will create ISO 9660 image) on Windows does not understand the UNIX file system semantics, the burning software does not produce the correct ISO 9660 image to be used for AIX, called Rock Ridge extension enabled ISO 9660 image format.

Therefore, if you create your own CD-R media on AIX, as shown in this appendix, you can have:

- Easy installable media via SMIT

  It may save you time if you burn both base filesets (I) and update filesets (U) on one media. Otherwise, you have to explicitly mount the media first (**`mount -v cdrfs -o ro /dev/cd0 /mnt`**) then have to specify the correct source directory location in the SMIT menu.

- Personal mksysb

  Although it may requires multiple volumes, the personal mksysb on CD is quite useful if you have to maintain a complex software installation on AIX (you do not have to have an expensive tape drive on AIX systems). In addition, the CD recordable media does not wear out and is inexpensive. Please note that the personal backup CD function is out of the scope of this appendix.

Basic flow

Because the management of CD-R media on AIX uses free software tools (GNU mkisofs and cdrecord), the basic flow of the task is composed of the following three steps:

1. Configure your CD-R drive correctly.
2. Create the ISO9660 image format files using the **mkisofs** command.
3. Burn the CD-R media using the **cdrecord** command.

Note: In AIX 5L Version 5.1, these two commands are installed by default.

Currently, IBM provides these tools in the AIX toolbox for Linux applications, found at:

http://www-1.ibm.com/servers/aix/products/aixos/linux/download.html

The original Web site of cdrecord is:

http://www.fokus.gmd.de/research/cc/glone/employees/joerg.schilling/private/cdrecord.html

Confirm SCSI connection address of the CD-R drive

It is your responsibility to purchase and connect the CD-R drive that works on AIX correctly. You can also purchase a SCSI CD-R or CD-RW drive from local PC retailer shops, and you can connect and configure it to your pSeries servers. Technically, if the GNU **cdrecord** command supports the SCSI CD drive you are going to use, it should work. However, some external connection type SCSI CD-R drives have an IDE CD-R drive inside and convert the signal from IDE to SCSI. These types of drive do not work on AIX.

The /usr/lpp/bos.sysmgt/mkcd.README.txt file installed with AIX 5L Version 5.1 lists several tested CD recordable devices.

In the following example, we have three SCSI controllers (one of them is an on-board SCSI controller on the I/O planar scsi0) on the IBM RS/6000 model 43P-260:

```
# lsdev -Cc adapter | grep scsi
scsi0   Available 10-60    Wide/Fast-20 SCSI I/O Controller
scsi1   Available 10-68    Wide SCSI I/O Controller
scsi2   Available 10-70    Wide SCSI I/O Controller
scsi3   Available 10-88    Wide/Ultra-2 SCSI I/O Controller
```

The external type CD-R device (Plextor PLEXWRITER 12/10/32S, PX-W1210TSE) is connected to the second FC6208 PCI SCSI-2 Single-Ended Fast/Wide Adapter (type 4-A), which is inserted in PCI slot#4 (physical location code 10-70). This drive is recognized as Other CD-ROM Drive. You may be confused, since it will appear as a CD-ROM drive, even though it is recordable:

```
# lsdev -Cc cdrom
cd0 Available 10-60-00-1,0 SCSI Multimedia CD-ROM Drive
cd1 Available 10-70-00-0,0 Other SCSI CD-ROM Drive
```

You can confirm the product name using the **lscfg** command:

```
# lscfg -vl cd1
  DEVICE            LOCATION          DESCRIPTION

  cd1               10-70-00-0,0      Other SCSI CD-ROM Drive

        Manufacturer................PLEXTOR
        Machine Type and Model......CD-R   PX-W1210S
        ROS Level and ID............1.00
        Device Specific.(Z0)........0580020233000018
```

In this particular case, the CD-R drive is connected at the following address:

SCSI Bus#	2
SCSI ID	0
SCSI LUN	0

You should write down this information; it is required in a later step.

Note: Please avoid connecting any external type SCSI device to the native SCSI controller on the old RS/6000 systems, such as model 25T and model 43P-132, because the SCSI controller chip on those models cannot easily handle SCSI errors. If they have a SCSI2 controller, like the 43P-150, then they are relatively safe to connect to external type SCSI devices.

Confirm device support status

We assume that you installed the following RPM packages on your system from the AIX toolbox for Linux applications:

```
# lslpp -L | egrep '(mkisofs|cdrecord)'
  cdrecord                       1.9    C     R    A command line CD/DVD
recording
  mkisofs                       1.13    C     R    Creates an image of an
ISO9660
```

You can use the **rpm** command to confirm these packages:

```
# rpm -qa | egrep '(mkisofs|cdrecord)'
cdrecord-1.9-3
mkisofs-1.13-3
```

The **cdrecord** command is installed in /opt/freeware/bin, and a symbolic link to /usr/bin/cdrecord is provided:

```
# ls -l /opt/freeware/bin/cdrecord
-rwxr-xr-x   1 root     system       204937 Mar 29 2001
/opt/freeware/bin/cdrecord
# whence cdrecord
/usr/bin/cdrecord
# ls -l `whence cdrecord`
lrwxrwxrwx   1 root     system           31 Oct 31 13:40 /usr/bin/cdrecord
-> ../../opt/freeware/bin/cdrecord
```

You have to confirm whether the CD-R device you connected to is supported by the **cdrecord** command, as shown in the following example:

```
# cdrecord -checkdrive dev=/dev/cd1:2,0,0
Cdrecord 1.9 (powerpc-ibm-aix4.3.3.0) Copyright (C) 1995-2000 Jörg Schilling
scsidev: '/dev/cd1:2,0,0'
devname: '/dev/cd1'
scsibus: 2 target: 0 lun: 0
Using libscg version 'schily-0.1'
Device type    : Removable CD-ROM
Version        : 2
Response Format: 2
Capabilities   : SYNC LINKED
Vendor_info    : 'PLEXTOR '
Identifikation : 'CD-R   PX-W1210S'
Revision       : '1.00'
Device seems to be: Generic mmc CD-RW.
Using generic SCSI-3/mmc CD-R driver (mmc_cdr).
Driver flags   : SWABAUDIO
```

If you see the specific driver name in the parentheses, as shown in the high-lighted line, the CD-R drive is supported by the **cdrecord** command and should work correctly.

> **Note:** In AIX 5L Version 5.1, the **geninstall** command can manage both AIX installp format files and RPM format files. To use **geninstall**, you should use the /usr/sys/inst.images/installp/ppc directory for AIX installp format files and the /usr/sys/inst.images/RPMS/ppc directory for RPM format files.

Creating ISO 9660 format image files

Before burning the media, you have to create your ISO 9660 format image file using the **mkisofs** command. To create the ISO 9660 image file on AIX systems, we propose the directory structure shown in Figure C-1.

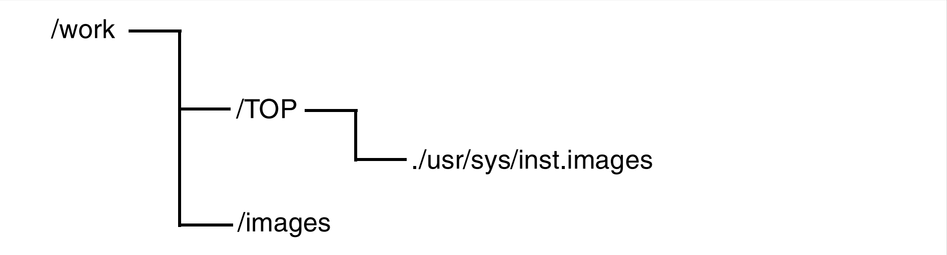


Figure C-1 Required directory structure to create ISO9660 format image

The directory /work/TOP is used as a source directory tree to create the ISO 9660 format image. The directory /work is used to store the created ISO 9660 format image file. Ideally, you should create these two directories, /work/TOP and /work/images, as two separate file systems created on separate disk drives so that you can achieve higher disk I/O bandwidth. If you use the CD recordable media device, each file system should have at least 640 MB free space. If you use the DVD-RAM drive, each file system should have at least 4.7 GB free space. We recommend you create the file system as a large file enabled JFS.

If you have to create these work file systems on rootvg, then you can add the following stanza in /etc/exclude.rootvg to avoid including these file systems in your system backup (mksysb):

```
/work
```

Before creating the ISO 9660 format image file, you should copy the files you are going to burn on the media under the /work/TOP. The directory /work/TOP will be appeared as a *TOP* directory on the media. If you are going to burn AIX installp format files (bff), then you should create the following sub-directories under the /work/TOP, then copy your files under the /work/TOP/usr/sys/inst.images:

```
# pwd
/work
# ls
TOP/         lost+found/
# cd TOP
#  ls -lR
total 8
drwxr-sr-x   3 root     system          512 Mar 18 16:41 usr/
./usr:
total 8
drwxr-sr-x   3 root     system          512 Mar 18 16:41 sys/

./usr/sys:
total 8
drwxr-sr-x   2 root     system          512 Mar 18 17:47 inst.images/

./usr/sys/inst.images:
total 43544
-rw-r--r--   1 root     system         4958 Mar 18 17:47 .toc
-rw-r--r--   1 root     system      9340832 Mar 18 16:59 IY19375.tar.gz
-rw-rw-r--   1 root     system         9393 Aug 20 2001  IY19375.txt
-rw-rw-r--   1 root     system      4052992 Sep 20 11:29 U477366
-rw-rw-r--   1 root     system      4378624 Sep 20 11:29 U477367
-rw-rw-r--   1 root     system      4495360 Sep 20 11:30 U477368
# installp -ld /work/TOP/usr/sys/inst.images
  Fileset Name                Level                     I/U Q Content
  ====================================================================
  bos.mp                      5.1.0.1                    S  b usr
#   Base Operating System Multiprocessor Runtime

  bos.mp64                    5.1.0.1                    S  b usr
#   Base Operating System 64-bit Multiprocessor Runtime

  bos.up                      5.1.0.1                    S  b usr
#   Base Operating System Uniprocessor Runtime
```

If you want to burn the media that is able to be installed using SMIT, then you should run the **inutoc** command to create the TOC (table of contents) file on /work/TOP/usr/sys/inst.images. Also, please be advised to confirm the file owner/group and permissions, as shown in the following example:

```
# pwd
/work/TOP
```

```
# inutoc ./usr/sys/inst.images
# chown -R root.system .
# chmod -R a+r .
```

Then invoke the **mkisofs** command with the -R option, which enables the Rock Ridge extension. Please note that the command takes the source directory (files) as its last argument. In this case, we specified the current directory (/work/TOP) with the single dot ".":

```
# pwd
/work/TOP
# mkisofs -R -o ../images/cdimg.iso .
  45.93% done, estimate finish Mon Mar 18 17:50:44 2002
 91.73% done, estimate finish Mon Mar 18 17:50:44 2002
Total translation table size: 0
Total rockridge attributes bytes: 1286
Total directory bytes: 6144
Path table size(bytes): 50
Max brk space used 10010
10909 extents written (21 Mb)
# ls -l ../images/cdimg.iso
-rw-r--r--   1 root     sys        22339584 Mar 18 17:05 ../cdimg.iso
```

AIX does not provide a loopback file system (lo file system), as in Linux; therefore, you cannot peak the contents of the created ISO 9660 image file.

Burn CD-R media

To burn the CD-R media, invoke the **cdrecord** command, as shown in Example C-1.

Example: C-1   Burning CD-R media

```
# cdrecord -v speed=12 dev=/dev/cd1:2,0,0 /work/cdimg.iso
Cdrecord 1.9 (powerpc-ibm-aix4.3.3.0) Copyright (C) 1995-2000 Jörg Schilling
TOC Type: 1 = CD-ROM
scsidev: '/dev/cd1:2,0,0'
devname: '/dev/cd1'
scsibus: 2 target: 0 lun: 0
Using libscg version 'schily-0.1'
atapi: 0
Device type    : Removable CD-ROM
Version        : 2
Response Format: 2
Capabilities   : SYNC LINKED
Vendor_info    : 'PLEXTOR '
Identifikation : 'CD-R   PX-W1210S'
Revision       : '1.00'
Device seems to be: Generic mmc CD-RW.
```

```
Using generic SCSI-3/mmc CD-R driver (mmc_cdr).
Driver flags   : SWABAUDIO
Drive buf size : 2394336 = 2338 KB
FIFO size      : 4194304 = 4096 KB
Track 01: data   21 MB
Total size:      24 MB (02:25.46) = 10910 sectors
Lout start:      24 MB (02:27/35) = 10910 sectors
Current Secsize: 512
ATIP info from disk:
  Indicated writing power: 4
  Is not unrestricted
  Is not erasable
  Disk sub type: Medium Type A, low Beta category (A-) (2)
  ATIP start of lead in:  -11318 (97:31/07)
  ATIP start of lead out: 336226 (74:45/01)
Disk type:    Long strategy type (Cyanine, AZO or similar)
Manuf. index: 22
Manufacturer: Ritek Co.
Blocks total: 336225 Blocks current: 336225 Blocks remaining: 325312
Starting to write CD/DVD at speed 8 in write mode for single session.
Last chance to quit, starting real write in 1 seconds.
Waiting for reader process to fill input buffer ... input buffer ready.
Performing OPC...
Starting new track at sector: 0
Track 01:  21 of  21 MB written (fifo 100%).
Track 01: Total bytes read/written: 22345728/22345728 (10911 sectors).
Writing  time:   23.260s
Fixating...
Fixating time:   31.691s
cdrecord: fifo had 352 puts and 352 gets.
cdrecord: fifo was 0 times empty and 276 times full, min fill was 96%.
```

Note: Once it works, the environment is relatively stable; you seldom see buffer under-run error on AIX.

The speed parameter instructs the actual writing speed to the CD-R drive. In this particular case, the Plextor drive supports 12x writing speed for CD-R media. The dev parameter specifies the device special file that represents the CD-R device. The specified parameter 2,0,0 means SCSI bus, SCSI ID, and SCSI LUN, confirmed in “Confirm device support status” on page 332.

Optionally, you can modify the /etc/cdrecord.conf file, as shown in Example C-2 on page 337, to specify the default device special file name.

Example: C-2 Sample /etc/cdrecord.conf

```
#ident @(#)cdrecord.dfl 1.2 00/04/16 Copyr 1998 J. Schilling
#
# This file is /etc/default/cdrecord
# It contains defaults that are used if no command line option
# or environment is present.
#
# The default device, if not specified elswhere
#
CDR_DEVICE=plextor

#
# The default speed, if not specified elswhere
#
CDR_SPEED=12

#
# The default FIFO size if, not specified elswhere
#
CDR_FIFOSIZE=4m

#
# The following definitions allow abstract device names.
# They are used if the device name does not contain the
# the characters ',', ':', '/' and '@'
#
# drive name    device  speed   fifosize driveropts
#
plextor=        2,0,0   12      -1      ""
```

After burning the CD-R media, you should confirm that you can mount the media, as shown in the following example:

```
# mount -v cdrfs -o ro /dev/cd1 /cdrom
# ls -lR /cdrom
total 4
drwxr-sr-x   3 root     system         2048 Mar 18 10:41 usr/
/cdrom/usr:
total 4
drwxr-sr-x   3 root     system         2048 Mar 18 10:41 sys/

/cdrom/usr/sys:
total 4
drwxr-sr-x   2 root     system         2048 Mar 18 11:47 inst.images/

/cdrom/usr/sys/inst.images:
total 43524
-rw-r--r--   1 root     system         4958 Mar 18 11:47 .toc
```

```
-rw-r--r--   1 root     system      9340832 Mar 18 10:59 IY19375.tar.gz
-rw-rw-r--   1 root     system         9393 Aug 20 2001  IY19375.txt
-rw-rw-r--   1 root     system      4052992 Sep 20 06:29 U477366
-rw-rw-r--   1 root     system      4378624 Sep 20 06:29 U477367
-rw-rw-r--   1 root     system      4495360 Sep 20 06:30 U477368
```

If you invoke smit -C install_selectable_all, then you will see the following fileset updates listed:

```
# unmount /cdrom
# smit -C install_selectable_all
    select /dev/cd1 for INPUT device
    press F4 on "SOFTWARE to install"
  x   bos.mp                                                             ALL x+
  x    @ 5.1.0.1  Base Operating System Multiprocessor Runtime               x+
  x                                                                          x+
  x   bos.mp64                                                           ALL x+
  x    @ 5.1.0.1  Base Operating System 64-bit Multiprocessor Runtime        x+
  x                                                                          x+
  x   bos.up                                                             ALL x+
  x    @ 5.1.0.1  Base Operating System Uniprocessor Runtime                 x+
```

Duplicate ISO 9660 format CD-ROM media

By using **dd** and **cdrecord**, you can duplicate ISO 9660 format CD-ROM media *on the fly*; obviously, this example assumes that you have one CD-ROM drive (/dev/cd0) as a source and one CD-R drive (/dev/cd1) as a target. Technically, the process is just reading from the source CD-ROM media in the CD-ROM drive using the **dd** command, piping the output stream to the **cdrecord** process to burn the CD-R media.

First, you have to measure the size of source CD-ROM media using the **dd** command. This example shows it has 821128 blocks. Please note that **dd** uses 512 bytes for a block unit.

```
# dd if=/dev/cd0 of=/dev/null
43640+0 records in.
43640+0 records out.
```

The CD-ROM or CD-R media uses 4096 bytes for a sector size (4096 / 512 == 4). Therefore, you have to divide the previous number by 4.

```
# expr 43640 / 4
10910
```

Then invoke the following command:

```
# dd if=/dev/cd0 | cdrecord speed=10 tsize=10910s dev=/dev/cd1:2,0,0 -
Cdrecord 1.9 (powerpc-ibm-aix4.3.3.0) Copyright (C) 1995-2000 Jörg Schilling
scsidev: '/dev/cd1:2,0,0'
devname: '/dev/cd1'
scsibus: 1 target: 1 lun: 0
Using libscg version 'schily-0.1'
Device type    : Removable CD-ROM
Version        : 2
Response Format: 2
Capabilities   : SYNC LINKED
Vendor_info    : 'PLEXTOR '
Identifikation : 'CD-R   PX-W1210S'
Revision       : '1.00'
Device seems to be: Generic mmc CD-RW.
Using generic SCSI-3/mmc CD-R driver (mmc_cdr).
Driver flags   : SWABAUDIO
... many lines are erased on purpose.
```

Please note that:

- You have to specify s for the tsize parameter number.
- you have to specify '-' as the last parameter of the **cdrecord** command to specify that the install source is standard input.

D

Additional material

This redbook refers to additional material that can be downloaded from the Internet as described below.

Locating the Web material

The Web material associated with this redbook is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser to:

```
ftp://www.redbooks.ibm.com/redbooks/SG246606
```

Alternatively, you can go to the IBM Redbooks Web site at:

ibm.com/redbooks

Select the **Additional materials** and open the directory that corresponds with the redbook form number, SG246606.

Using the Web material

The additional Web material that accompanies this redbook includes a compressed tar archive file, sg246606.tar.Z, which includes the following files:

```
$ tar tvf sg246606.tar
drwxr-sr-x   0 0        0 May 21 15:05:51 2002 ./
```

```
drwxr-sr-x   0 0        0 Apr 23 17:03:38 2002 ./Chap05/
-rwxr-xr-x   0 0     3129 May 02 10:41:28 2002 ./Chap05/hwalert
-rwxr-xr-x   0 0      701 Apr 23 17:02:56 2002 ./Chap05/sendtrap
drwxr-sr-x   0 0        0 May 21 15:06:50 2002 ./Chap06/
-rwxr-xr-x   0 0     7349 Apr 30 12:15:12 2002 ./Chap06/makebff
-rw-r--r--   0 0  17766400 May 21 15:06:51 2002
./Chap06/sg246606.net-snmp.4.2.3.0.bff
drwxr-sr-x   0 0        0 May 21 15:07:17 2002 ./Chap07/
-rwxr-xr-x   0 0    24037 Apr 23 16:55:24 2002 ./Chap07/tidysys
-rwxr-xr-x   0 0     7359 Apr 23 16:55:33 2002 ./Chap07/cnvdate
-rw-r--r--   0 0     5466 Apr 23 16:56:56 2002 ./Chap07/cnvdate.c
-rw-r--r--   0 0     5759 Apr 23 16:56:23 2002 ./Chap07/tidysys.conf
-rw-r--r--   0 0     3125 Apr 23 16:56:23 2002 ./Chap07/tidysys.files
-rwxr-xr-x   0 0     2800 Apr 23 16:58:29 2002 ./Chap07/drawdailygraphcpu.ksh
-rwxr-xr-x   0 0     2633 Apr 23 16:58:37 2002 ./Chap07/drawdailygraphpag.ksh
-rwxr-xr-x   0 0      638 Apr 23 16:58:47 2002 ./Chap07/cpu_stats
-rwxr-xr-x   0 0      261 Apr 23 16:59:02 2002 ./Chap07/prunestats
-rwxr-xr-x   0 0     1007 Apr 23 16:59:14 2002 ./Chap07/httpcp
-rwxr-xr-x   0 0      860 Apr 23 16:59:21 2002 ./Chap07/sync_users.ksh
-rwxr-xr-x   0 0     1272 Apr 23 16:59:27 2002 ./Chap07/listhwlevels
-rwxr-xr-x   0 0      733 Apr 23 16:59:35 2002 ./Chap07/listswlevels
```

System requirements for downloading the Web material

The following system configuration is recommended:

Hard disk space:	20 MB minimum
Operating System:	AIX

How to use the Web material

Create a subdirectory on your AIX systems, and un-compress and un-tar the contents of the Web material archived file into this subdirectory.

Abbreviations and acronyms

AIX	Advanced Interactive Executive
APAR	Authorized Problem Analysis Report
ASCII	American National Standard Code for Information Interchange
ASN.1	Abstract Syntax Notation 1
ATE	Asynchronous Terminal Emulation
ATM	Asynchronous Transfer Mode
BNU	Basic Network Utilities
BOS	Base Operating System
BPC	Bit Per Character
BPS	Bit Per Second
BSD	Berkley Software Distribution
CCITT	United Nations Consultative Committee for International Telephony and Telegraphy
CD	Compact Disk
CDE	Common Desktop Environment
CD-R	Compact Disk - Recordable
CD-ROM	Compact Disk - Read Only Media
CD-RW+	Compact Disk - Read and Write Plus
CERT	Computer Emergency Response Team
CHAP	Challenge Handshake Authentication Protocol
CPAN	Comprehensive Perl Archive Network
CPU	Central Processing Unit
CTS	Clear to Send
DCD	Data Carrier Detect
DCE	Data Communication Equipment
DES	Data Encryption Standard
DGD	Dead Gateway Detection
DPI	Distributed Protocol Interface
DSA	Digital Signature Algorithm
DSR	Data Set Ready
DTE	Data Terminal Equipment
DTR	Data Terminal Ready
DVD	Digital Versatile Disk
DVD-R	Digital Versatile Disk - Recordable
DVD-ROM	Digital Versatile Disk - Read Only Media
DVD-RW	Digital Versatile Disk - Read and Write
EIA	Electronic Industry Association
ESS	Electronic Server System
FC	Feature Code
FDDI	Fibre Distributed Data Interface
FTP	File Transfer Protocol
GB	Gigabyte
GID	Group Identification
GNOME	GNU Network Object Model Environment
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
IBM	International Business Machines Corporation
ICMP	Internet Control Message Protocol

IEEE	Institute of Electrical and Electronic Engineers
IHS	IBM HTTP Server
IP	Internet Protocol
IPCP	Internet Protocol Control Protocol
IPsec	IP Security
ISO	International Organization for Standardization
ITSO	International Technical Support Organization
ITU-T	Telecommunication Standardization Sector of the International Telecommunications Union
KDE	K Desktop Environment
L2	Level 2
L3	Level 3
LAN	Local Area Network
LCP	Link Control Protocol
LDAP	Lightweight Directory Access Protocol
LED	Light Emitting Diode
LPP	Licensed Program Product
LSB	Least Significant Bit
LUN	Logical Unit Number
MA	Maintenance Agreement
MAC	Media Access Control
MB	Megabyte
MIB	Management Information Base
MNP	Microcosm Networking Protocol
MSB	Most Significant Bit
MTU	Maximum Transmission Unit
NCP	Network Control Protocol
NFS	Network File System
NIM	Network Installation Manager
NIS	Network Information Service
NIS+	Network Information Service Plus
ODM	Object Database Manager
ODS	On-Demand Server
OID	Object ID
PAP	Password Authentication Protocol
PC	Personal Computer
PCI	Peripheral Component Interconnect
PMR	Problem Management Record
PMTU	Path MTU
POP	Post Office Protocol
POWER	Performance Optimization with Enhanced RISC
PPP	Point-to-Point Protocol
PSTN	Public Shared Telephony Network
PTF	Program Temporary Fix
PV	Physical Volume
RAID	Redundant Array of Independent Disks
RD	Receive Data
RFC	Request for Comment
RISC	Reduced Instruction Set Computer
RPM	Red Hat Package Manager
RS	Recommended Standard
RSA	Rivest-Shamir-Adleman Algorithm
RTS	Ready to Send
SA	Service Agent
SAN	Storage Area Network
SAS	Service Agent Server
SATAN	Security Analysis Tool for Analyzing Networks

SCSI	Small Computer System Interface
SMIT	System Management Interface Tool
SMTP	Simple Mail Transfer Protocol
SMUX	SNMP UNIX Multiplexer
SNMP	Simple Network Management Protocol
SSA	Serial Storage Architecture
SSH	Secure Shell
TCB	Trusted Computing Base
TCP	Transmission Control Protocol
TD	Transmit Data
TFTP	Trivial File Transfer Protocol
TTY	Teletypewriter
UDP	User Datagram Protocol
UID	User Identification
UPS	Uninterrupted Power Supply
URL	Universal Resource Locator
UTP	Un-twisted Pair
UUCP	UNIX-to-UNIX Copy Program
VLAN	Virtual LAN
VPD	Vital Product Data
WU-FTPD	Washington University FTP Daemon
XOFF	Transmitter off
XON	Transmitter on

Related publications

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

IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 352.

- *Additional AIX Security Tools on IBM eServer pSeries, IBM RS/6000, and SP Cluster*, SG24-5971
- *AIX Version 4.3 Differences Guide*, SG24-2014
- *AIX 4.3 Elements of Security: Effective and Efficient Implementation*, SG24-5962
- *AIX 5L Differences Guide,* SG24-5765
- *AIX 5L Performance Tools Handbook*, SG24-6039
- *AIX Logical Volume Manager, from A to Z: Introduction and Concepts*, SG24-5432
- *IBM Certification Study Guide AIX Installation and System Recover*, SG24-6183
- *IBM HTTP Server Powered by Apache on RS/6000*, SG24-5132
- *NIM: From A to Z in AIX 4.3,* SG24-5524
- *TCP/IP Tutorial and Technical Overview*, GG24-3376
- *Tivoli Storage Manager Version 3.7: Technical Guide*, SG24-5477
- *Understanding IBM @server pSeries Performance and Sizing*, SG24-4810

Other resources

These publications are also relevant as further information sources:

- *8-Port Asynchronous PCI Adapter Installation and User's Guide*, SA23-2562
- *128-Port Asynchronous PCI Adapter Installation and User's Guide*, SA23-2563
- *AIX 5L Version 5.1 Asynchronous Communication Guide**

- _AIX 5L Version 5.1 Communications Programming Concepts_*
- _AIX 5L Version 5.1 General Programming Concepts: Writing and Debugging Programs_*
- _AIX 5L Version 5.1 Installation Guides: Installation Guide_*
- _AIX 5L Version 5.1 Installation Guides: Network Installation Management Guide and Reference_*
- _AIX 5L Version 5.1 Packaging Guide For LPP Installation_*
- _AIX 5L Version 5.1 Reference Documentation: Commands Reference_*
- _AIX 5L Version 5.1 System Management Guides: AIX 5L Version 5.1 Web-based System Manager Administration Guide*_
- _AIX 5L Version 5.1 System Management Guide: Communications and Networks_*
- _AIX 5L Version 5.1 System Management Guides: Operating System and Devices_*
- _A Comprehensive Guide to Virtual Private Networks, Volume III: Cross-Platform Key and Policy Management_, SG24-5309
- _Electronic Service Agent for pSeries and RS/6000 and pSeries User's Guide_, LCD4-1060
- _RS/6000 Adapters, Devices, and Cable Information for Multiple Bus Systems,_ SA38-0516
- _RS/6000 and pSeries Diagnostic Information for Multiple Bus Systems,_ SA38-0509
- _RS/6000 and pSeries PCI Adapter Placement Reference_, SA38-0538
- _Site and Hardware Planning Information,_ SA38-0508
- Libes, _Exploring Expect_, O'Reilly & Associates, Inc., 1996, ISBN 1565920902
- Mauro, et al, Essential SNMP, O'Reilly & Associates, Inc., 2001, ISBN 0596000200
- Miller, et al, _Managing Internetworks With Snmp, 3rd Ed_, John Wiley & Sons, 1999, ISBN 076457518X

You can access all of the pSeries hardware related documentation through the Internet at the following URL:

http://www.ibm.com/servers/eserver/pseries/library/hardware_docs/index.html

You can also access all of the AIX documentation through the Internet at the following URL:

http://www.ibm.com/servers/aix/library

The publications marked with * in the list are located on the documentation CD-ROM that ships with the AIX operating system:

Referenced Web sites

These Web sites are also relevant as further information sources:

- A practical guide to network security white paper

  `http://w3-1.ibm.com/services/so/e-business_hosting/ftp_files/pdf/exodussecuritywp.pdf`
- AIX toolbox for Linux applications

  `http://www-1.ibm.com/servers/aix/products/aixos/linux/download.html`
- *Electronic Service Agent for pSeries and RS/6000 User's Guide,* SC38-7105

  `ftp://service.software.ibm.com/aix/service_agent_code/svcUG.pdf`
- *rlogin(1): The Untold Story*, found at:

  `http://www.cert.org/archive/pdf/rlogin1_98tr017.pdf`
- Data Communcations Basics

  `http://www.camiresearch.com/Data_Com_Basics/data_com_tutorial.html#anchor4059`
- Chapter 1, *Parallel Port Complete: Programming, Interfacing, & Using the PC's Parallel Printer Port*, found at:

  `http://www.lvr.com/files/ppc1.pdf`
- TIA Standards

  `http://www.tiaonline.org/standards`
- RAD Web site

  `http://www.rad.com/network/1995/rs232/rs232.htm`
- Service Agent Code download site

  `ftp://ftp.software.ibm.com/aix/service_agent_code/aix/service_agent_code`
- International Telecommunication Union

  `http://www.itu.int/rec/recommendation.asp?type=products&lang=e&parent=T-REC-V`
- IBM Enhanced Integrated Technology Services

  `http://www.ibmlink.ibm.com/usalets&parms=H_601-011`
- Data/FAX Modem Reference Index

  `http://nemesis.lonestar.org/reference/telecom/modems/index.html`

- Comprehensive Perl Archive Network

  http://www.cpan.org
- OpenSSH on AIX Images Project

  http://oss.software.ibm.com/developerworks/projects/opensshi
- Secure Shell Charter

  http://www.ietf.org/html.charters/secsh-charter.html
- OpenSSH

  http://www.openssh.org
- IBM Managed Security Services

  http://www.ers.ibm.com
- Kerberos/GSSAPI Support in OpenSSH

  http://www.sxw.org.uk/computing/patches/openssh.html
- CERT - Implementation Details for installing, configuring, and using tcp wrapper to log unauthorized connection attempts on systems running Solaris 2.x

  http://www.cert.org/security-improvement/implementations/i041.07.html
- Bull Freeware

  http://www.bullfreeware.com
- PuTTY Download Page

  http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
- WU-FTPD Development Group

  http://www.wu-ftpd.org
- Source Forge

  http://net-snmp.sourceforge.net
- IBM Tivoli NetView

  http://www.tivoli.com/products/index/netview
- RPM.ORG

  http://www.rpm.org
- IBM @server pSeries Support

  http://techsupport.services.ibm.com/server/support?view=pSeries
- RSYNC.ORG

  http://www.rsync.org

- Cisco EtherChannel

  http://www.cisco.com/pcgi-bin/Support/PSP/psp_view.pl?p=Internetworking:Etherchannel

- Understanding EtherChannel Load Balancing and Redundancy on Catalyst Switches

  http://www.cisco.com/warp/public/473/4.html

- CDRecord

  http://www.fokus.gmd.de/research/cc/glone/employees/joerg.schilling/private/cdrecord.html

- Sudo Main Page

  http://www.courtesan.com/sudo

- Simple Network Management Protocol (SNMP) Vulnerabilities Frequently Asked Questions (FAQ)

  http://www.cert.org/tech_tips/snmp_faq.html

- CERT® Advisory CA-2002-03 Multiple Vulnerabilities in Many Implementations of the Simple Network Management Protocol (SNMP)

  http://www.cert.org/advisories/CA-2002-03.html

- Vulnerability Note VU#854306

  http://www.kb.cert.org/vuls/id/854306

- Vulnerability Note VU#107186

  http://www.kb.cert.org/vuls/id/107186

- OUSPG: Oulu University Secure Programming Group

  http://www.ee.oulu.fi/research/ouspg

- PROTOS Test-Suite: c06-snmpv1

  http://www.ee.oulu.fi/research/ouspg/protos/testing/c06/snmpv1/0100.html

- CERT® Coordination Center Denial of Service Attacks

  http://www.cert.org/tech_tips/denial_of_service.html

- Requests For Comments

  http://www.ietf.org/rfc/

- CERT PGP key

  http://www.cert.org/CERT_PGP.key

How to get IBM Redbooks

You can order hardcopy Redbooks, as well as view, download, or search for Redbooks at the following Web site:

ibm.com/redbooks

You can also download additional materials (code samples or diskette/CD-ROM images) from that site.

IBM Redbooks collections

Redbooks are also available on CD-ROMs. Click the CD-ROMs button on the Redbooks Web site for information about all the CD-ROMs offered, as well as updates and formats.

Index

IBM

Redbooks

Managing AIX Server Farms

Managing AIX Server Farms

Comprehensive serial ports management example

Configuring and using OpenSSH on AIX

Exploring SNMP agent implementation

Today's e-business relies heavily on secure, robust, scalable, and cost-effective server management, which is not an easy task to accomplish. The focus of this redbook is to explain practical methodology for administering AIX systems in a server farm environment, providing configuration and usage examples of several AIX operating system functions and free software tools, which include:

- Planning AIX server farms
- Understanding serial connections
- Practical use of serial connections
- Secure network connection on AIX
- Remote monitoring using SNMP
- Packaging your software tools
- Day-to-day tasks

This redbook is an ideal desk side reference for IBM employees, Business Partners, and customer system administrators or technical specialists who manage IBM @server pSeries servers running AIX 5L Version 5.1 in a server farm environment.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-6606-00 ISBN 0738425060