

Device Administration

Student Guide



Sun Microsystems Computer Corporation
Technical Education Services
MS UMIL07-14
2550 Garcia Avenue
Mountain View, CA 94043
U.S.A.

Part Number SA-285
Revision B, July 1993

© 1993 Sun Microsystems, Inc.—Printed in the United States of America.
2550 Garcia Avenue, Mountain View, California 94043-1100 U.S.A.

All rights reserved. This product and related documentation are protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or related documentation may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any.

Portions of this product may be derived from the UNIX® and Berkeley 4.3 BSD systems, licensed from UNIX System Laboratories, Inc. and the University of California, respectively. Third-party font software in this product is protected by copyright and licensed from Sun's Font Suppliers.

RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the United States Government is subject to the restrictions set forth in DFARS 252.227-7013 (c)(1)(ii) and FAR 52.227-19.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

TRADEMARKS

Sun, Sun Microsystems, the Sun logo, Solaris, and SunOS are trademarks or registered trademarks of Sun Microsystems, Inc. UNIX and OPEN LOOK are registered trademarks of UNIX System Laboratories, Inc. All other product names mentioned herein are the trademarks of their respective owners.

All SPARC trademarks, including the SCD Compliant Logo, are trademarks or registered trademarks of SPARC International, Inc. SPARCstation, SPARCserver, SPARCengine, SPARCworks, and SPARCcompiler are licensed exclusively to Sun Microsystems, Inc. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK® and Sun™ Graphical User Interfaces were developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

X Window System is a trademark and product of the Massachusetts Institute of Technology. Ethernet is a registered trademark of Xerox Corporation.

THIS PUBLICATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS PUBLICATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THE PUBLICATION. SUN MICROSYSTEMS, INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS PUBLICATION AT ANY TIME.

Contents

Introduction	1-1
Introduction	1-2
What Is a Serial Device?	1-3
Workstation Serial Ports	1-4
Serial Devices.....	1-5
Serial Interfaces	1-9
Null Modem Cable	1-11
Setting Up Bidirectional Modems	1-12
Summary	1-14
Exercise 1-1.....	1-15
Introduction to the	
Service Access Facility (SAF.)	2-1
Introduction	2-2
The Service Access Facility	2-3
Service Access Facility Architecture.....	2-4
Service Access Controller.....	2-5
Port Monitor Types.....	2-6
TTY Port Monitor	2-7
Summary	2-9
Exercise 2-1.....	2-10
Adding Terminals and Modems	3-1
Introduction	3-2
Setting Up Terminals and Modems	3-4
Setting Up a Terminal.....	3-5
Setting Up a Bidirectional Modem	3-7
Using the Modem.....	3-9
The /etc/remote File	3-10
The sacadm Command.....	3-12
The pmadm Command	3-16
Port Monitor Specific Administration Commands.....	3-19
TTY Monitor Specific Configuration.....	3-20
The /etc/ttydefs File	3-23



Terminal Control Components	3-24
SAF Command Summary	3-25
SAF File Summary	3-26
Summary	3-27
Exercise 3-1	3-28
Answer Key	A-1
Lesson 1: Introduction	A-2
Lesson 2: The Service Access Facility	A-3
Lesson 3: Adding Terminals and Modems	A-4

Introduction



Objectives

Upon completion of this lesson, you will be able to:

- Define the following terms: serial device, port, serial port, modem, null modem cable, hardware carrier detect, and port monitor program.
- Name at least two serial devices.
- Identify the two serial ports located on the back of the CPU board.
- Describe the three types of modem access.
- Give an example of a data communication equipment (DCE) device and a data terminal equipment (DTE) device.

References

SunOS 5.1 Adding and Maintaining Devices and Drivers,
Chapter 1, "Modems and Alphanumeric Terminals,"
Appendix B, "Connecting Devices to the Serial Ports"



Introduction

This lesson introduces terminology used to connect devices to Sun workstations, specifically *serial* devices, such as modems, printers, and terminals.

The standard interfaces used to connect serial devices are also covered.

What Is a Serial Device?

A serial device is communications hardware that transfers data in a serial fashion—one bit after another, such as a serial printer, modem, or terminal.

Another type of communications hardware is a parallel device that transfers one or more bytes simultaneously, such as a Centronics printer.

Communication lines are usually serial, which is why modems are connected to a computer via a serial port.

What's a Port?

A *port* is a pathway on a computer that is used to connect communication lines and modems. A port is made up of both hardware (pins and connectors) and software (device driver).

Types of ports include serial, parallel, Small Computer System Interface (SCSI), and Ethernet®.

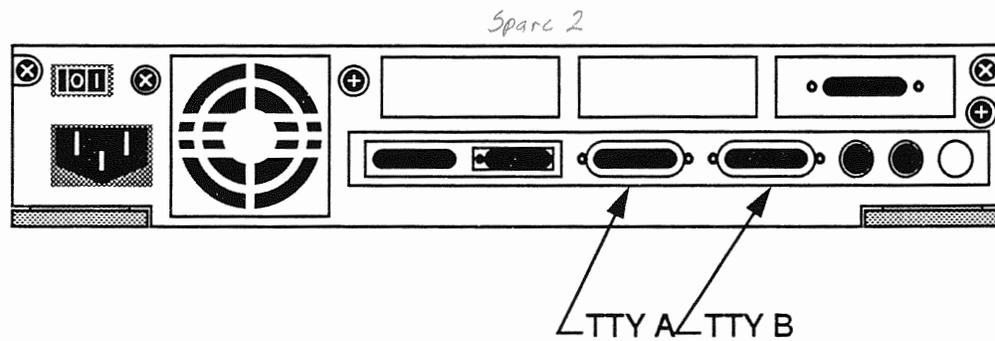
A *serial port* uses a standard communications protocol to transmit a byte of information bit-by-bit over a single line. Sun serial ports use the RS-423/RS-232 communications protocols, standards for serial interfaces between computers and peripheral devices.

Sun workstations usually come with two serial ports that can be used to provide access to different serial devices, such as:

- Printers
- Terminals
- Modems
- Other devices



Workstation Serial Ports



The serial ports are located on the back of the Central Processing Unit (CPU) board. They are also called *TTY* (A and B) ports because they are normally used to connect alphanumeric terminals.

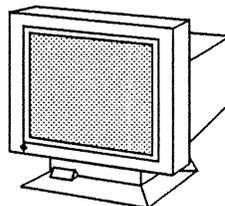
Additional serial ports are provided by an adapter board, such as the asynchronous line multiplexor (ALM1) board.

Most workstations use DB-25 connectors (25-pin), although some use DB-9 (9-pin) connectors and others use DIN-8 (8-pin).

Serial Devices



Terminal



Display monitor

Alphanumeric Terminals

A terminal is a serial input/output device with a keyboard and screen that displays only alphanumeric characters like those produced by a typewriter.

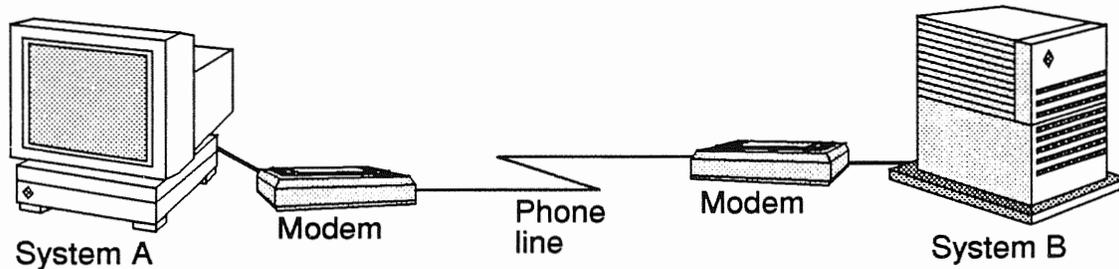
The Solaris® 2.x environment provides support for many terminal types, such as the vt100 model.

A Sun workstation comes with a bit-mapped display monitor used to display windows and graphics, and does not need any special setup.



Serial Devices

Modems



A *modem* is a data communications equipment (DCE) device that translates (or modulates) information into signals that can be transmitted across phone lines.

Another modem on the receiving end translates (or demodulates) the signals back into information that can be passed to the computer. The term "*modem*" is short for modulator-demodulator.

Three types of modem access are available:

- Dial-out—A modem set up to access another computer, allowing no access from the outside world.
- Dial-in—A modem set up to access a workstation from remote sites, allowing no outgoing calls.
- Bi-directional—Modem access that incorporates both dial-in and dial-out capabilities.

Serial Devices

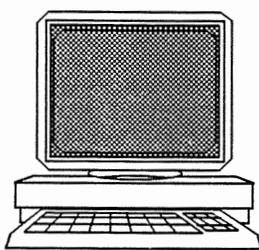
Connecting serial printers is the same process as terminals and modems. (Printer software set up is covered in another module.)

All three serial devices need some associated service in order to operate.

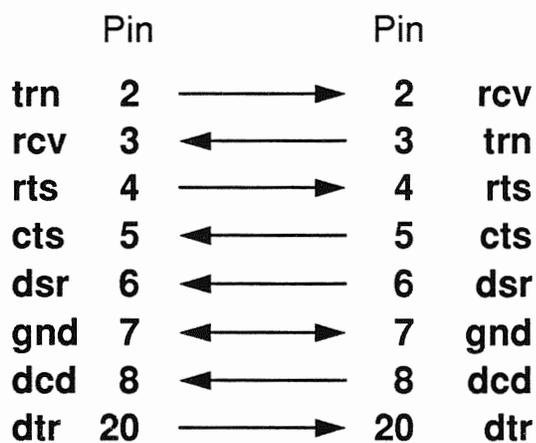
The main way to gain access to a service is through a port monitor program that continuously "listens" for requests—such as login or remote print requests. (This procedure is described in the next lesson.)

Serial Interfaces

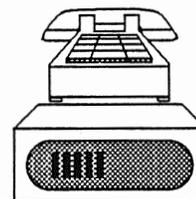
Workstation
or
Terminal



Data
terminal
equipment



Modem

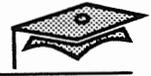


Data
communication
equipment

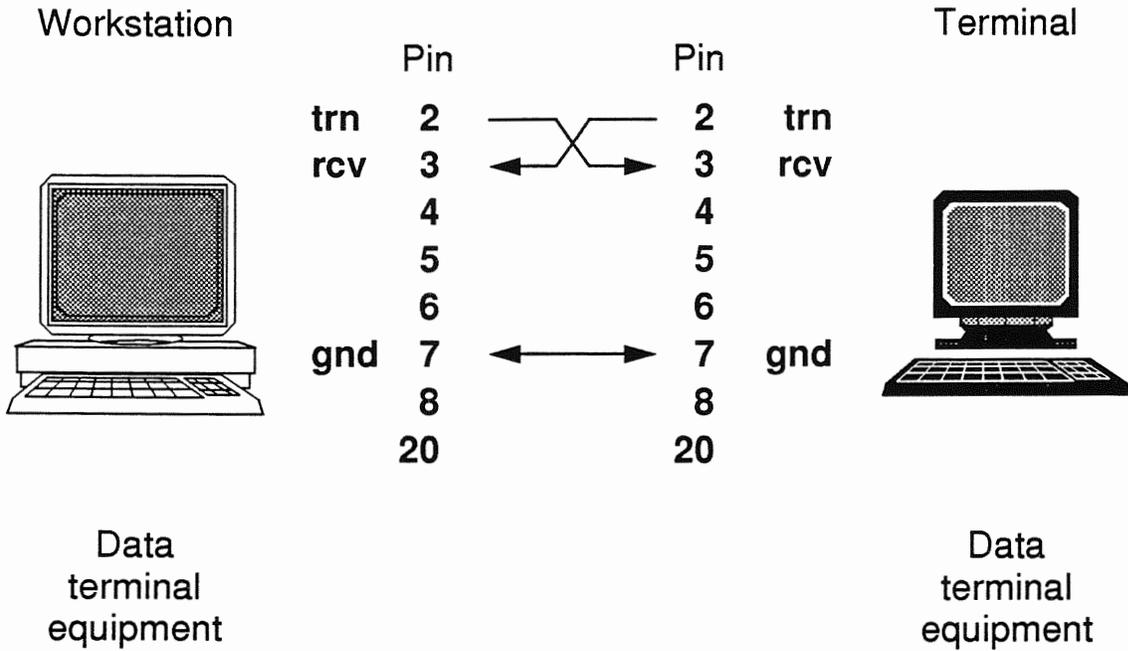
Serial Interfaces

Data terminal equipment (DTE) refers to terminals, workstations, and usually printers. DCE usually refers to modems, multiplexors, and data switches. The primary determinant of whether a device is considered a DTE or a DCE is which pins on the RS-232 interface expect which signals. A DTE device transmits on pin 2 and receives on pin 3. A DCE device transmits on pin 3 and receives on pin 2.

Sun's current workstation product line uses the RS-423 interface, which maintains compatibility with the RS-232 25-pin connector.



Null Modem Cable





Null Modem Cable

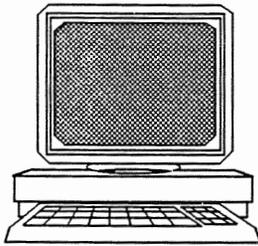
A null modem cable provides the means for two DTE devices to communicate with each other without the need for a modem. The direct-wire approach crosses pins 2 and 3 so that transmitted and received data are correctly connected between the two DTE devices. The ports on the CPU board can be connected to terminals and serial printers with the null modem cable. Only pins 2, 3, and 7 are required for communication between the devices.



Setting Up Bidirectional Modems

Using Hardware Carrier Detect

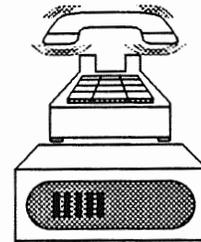
Workstation



dcd 8

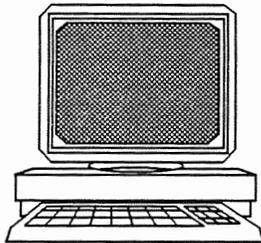


Modem

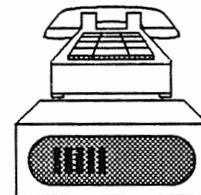


8 dcd

When a call comes in, the dcd (pin 8) goes high. An attempt to dial out gets a line busy message.



dcd 8



8 dcd

When a call hangs up, the dcd (pin 8) goes low. An attempt to dial out is OK.

Setting Up Bidirectional Modems

Using Hardware Carrier Detect (continued)

Sun workstations can support a bi-directional modem running dial-in and dial-out capability on the same port.

When a port is used for both dialing in and dialing out, the modem must be set to use hardware carrier detect so that it raises the data carrier detect (DCD) when there is an active carrier signal on the phone connection. Then when the call hangs up, DCD goes low and the modem can be used for dialing out.



Summary

In this lesson, you have learned that:

- Serial devices such as modems, printers, and terminals can be connected to Sun workstations via serial ports on the CPU board.
- Three different types of modem access are available.
 - Dial-in
 - Dial-out
 - Bi-directional
- Printers, terminals, and workstations are known as DTE devices.
- Modems are known as DCE devices.
- Sun uses two standard communications protocols (RS-423/RS-232) to connect serial devices.
- A null modem cable is used to connect two DTE devices, such as a workstation and a terminal.

Exercise 1-1

Write down your answers to the following questions.

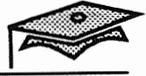
1. What is the difference between serial and parallel devices?

2. What is the difference between the way in which DTE and DCE devices transmit data? (Give an example of each).

3. Describe the three types of modem access.

4. Why is a null modem cable needed to connect a workstation and a terminal?

5. How does a bi-directional modem manage incoming and outgoing calls?



Introduction to the Service Access Facility



Objectives

Upon completion of this lesson, you will be able to:

- Name the top-level SAF process and describe its function.
- State the run level where `sac` is started.
- Name the two port monitor types available under SAF and describe each.
- Name the configuration file that tells `sac` which port monitors to initialize.
- Define the term "service."
- Name the file used by the `ttymon` port monitor to initialize the speed and terminal settings for each port.

References

SunOS 5.1 Adding and Maintaining Devices and Drivers,
Chapter 1, "Modems and Alphanumeric Terminals,"
Appendix A, "The Service Access Facility"



Introduction

This lesson describes the features of SAF, a suite of commands used to provide access to serial devices, such as terminals and modems. In addition, SAF provides access to network services, such as remote print requests.

The focus of this module is to provide access to serial devices such as terminals and modems.

The Solaris 2.x environment uses STREAMS-based character I/O for terminal and network devices. A STREAMS module called `ldterm` is used to perform standard terminal I/O processing. The `ldterm` module is also known as the line discipline module.

The Service Access Facility

Overview

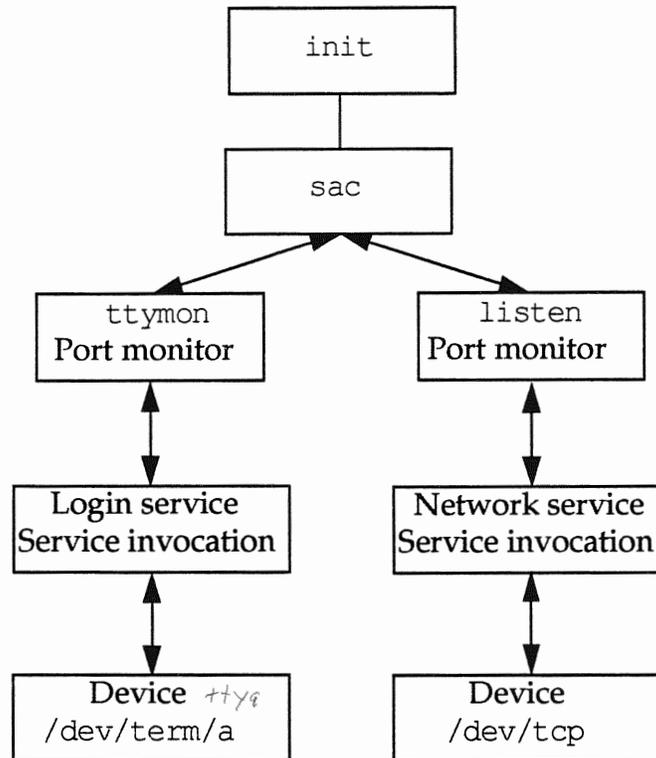
SAF controls terminal and modem access and network service access, such as remote print requests. It is not a program, but a hierarchy of processes and administration commands.

SAF controls access to local and network resources in a similar way.

System administrators configure SAF's port monitors to provide services to one or more ports.



Service Access Facility Architecture



SAF is a hierarchical structure of processes and administrative commands. The top-level SAF process is the Service Access Controller (sac), which is responsible for managing port monitors.

*gets started
in inittab*

1. The `init` process spawns `sac`, the master process of SAF.
2. The `sac` administrative command is used to control the port monitors by adding and removing them, starting and stopping them, and restarting them, if necessary.
3. A *port monitor* in turn monitors either a serial port device or a network port. It connects incoming requests to *services*, which are arbitrary processes, such as the login service. Port monitors monitor the device or network port, and the port monitor administrative commands are used to add and delete services, and start and stop services at the appropriate times.

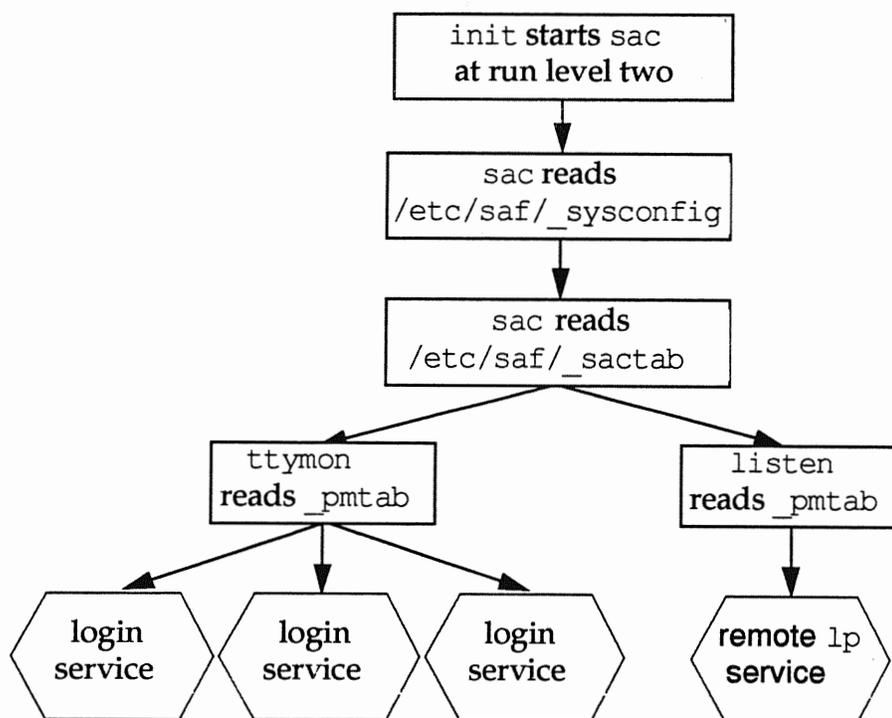
Service Access Controller

Initialization Process

The `sac` is the overseer of all port monitors. It is started when a system enters run level 2 by an entry in the `/etc/inittab` file. There is a separate entry to start a `ttymon` port monitor for the system console, which is outside the control of the `sac` process.

When `sac` is invoked, it reads a per-system configuration script (`/etc/saf/_sysconfig`), used to customize its environment, which is empty by default. After it has read this script, `sac` reads its administrative file (`/etc/saf/_sactab`) and starts whatever port monitors are specified.

The `sac` creates a new child process for each port monitor defined in the `_sactab` file. Each port monitor reads its administrative file (`/etc/saf/pmtab/_pmtab`) to find out what services to start on which ports.





Port Monitor Types

Two port monitor types are available under SAF.

The `ttymon` Monitor

The `ttymon` monitor is a STREAMS-based TTY port monitor. It monitors ports, sets terminal modes, baud rates, and invokes the login process, or any other service.

An instance of a `ttymon` monitor running under `sac` can monitor multiple ports and is configured using the `sacadm` command. Once an instance of a port monitor is created, the `ttymon`-specific information, such as the service it will provide, is added using the `pmadm` and `ttyadm` commands.

The `listen` Monitor

The `listen` port monitor is the network listener daemon that runs under `sac`. It listens on network endpoints for service requests, such as remote print requests and processes the requests by invoking servers to provide the services.

Each instance of `listen` is configured with the `sacadm` command, and the `listen`-specific information, such as the server providing the service, is configured with the `pmadm` and `nlsadmin` commands.

TTY Port Monitor

Terminal Initialization Process

Once a `ttymon` monitor instance is invoked by `sac` it starts to monitor its port, using the following process:

1. The `ttymon` monitor first initializes the speed and terminal settings for each port. The values used for initialization are taken from the appropriate entry in the `/etc/ttydefs` file, which is the terminal line settings table. (This file is covered in the next lesson.)
2. It writes the prompt and waits for user input. If the user indicates that the speed is inappropriate by pressing the Break key, the `ttymon` monitor tries the next speed and writes the prompt again.
3. When valid input is received, the `ttymon` monitor creates an `/var/adm/utmp` entry if required, establishes the service environment, and then invokes the service associated with the port.
4. After the service terminates, the `ttymon` monitor cleans up the `/var/adm/utmp` entry, if one exists, and returns the port to its initial state.

Using Autobaud

The `ttymon` monitor tries to determine the baud rate on the port automatically if autobaud is enabled for a port. Users must press Return before the `ttymon` monitor can recognize the baud rate and print the prompt. Currently, the baud rates that can be determined by autobaud are 110, 1200, 2400, 4800, and 9600.



TTY Port Monitor

Terminal Initialization Process (continued)

Bi-directional Service

If a port is configured for bi-directional service, the `ttymon` monitor allows users to connect to a service, and if the port is free, allows communications commands, such as `tip` for dialing out.

(The `tip` command is covered in the next lesson.)

Summary

In this lesson, you learned that:

- SAF controls terminal and modem access and network service access, such as remote print requests.
- The `sac` process is started by `init` from an entry in the `/etc/inittab` file.
- The `listen` and `ttymon` port monitors are started by entries in the `sac` process' configuration file.
- The `ttymon` port monitor initializes the speed and terminal settings for each port and then invokes the service when valid input is received.



Exercise 2-1

Write down the answers to the following questions.

1. Identify the top-level SAF process and the two types of port monitors it controls.

2. Identify the entry in the `/etc/inittab` file that is used to initialize the `sac` process.

3. At what run level is the `sac` process started?

4. Which configuration file tells the `sac` process which port monitors to initialize?

5. What file does the `ttymon` port monitor use to initialize the speed and terminal settings for each port?

Adding Terminals and Modems



Objectives

Upon completion of this lesson, you will be able to:

- Add a terminal to a system.
- Describe how to set up a bidirectional modem.
- Describe the syntax and purpose of the `tip` command.
- Name the file the `tip` command references to identify remote systems and line speeds.
- Describe the function of the `sacadm`, `pmadm`, and `ttyadm` commands.
- Name the file that identifies baud rates and terminal settings for TTY ports.

References

SunOS 5.1 Adding and Maintaining Devices and Drivers,
Chapter 1, "Modems and Alphanumeric Terminals,"
Appendix A, "The Service Access Facility"



Introduction

This lesson identifies the procedures used to add a terminal and modem to the system.

The Solaris 2.x environment automatically creates a port monitor called `zsm0n` to monitor serial ports A and B in initialization only mode. The term `zs` refers to the serial communications driver.

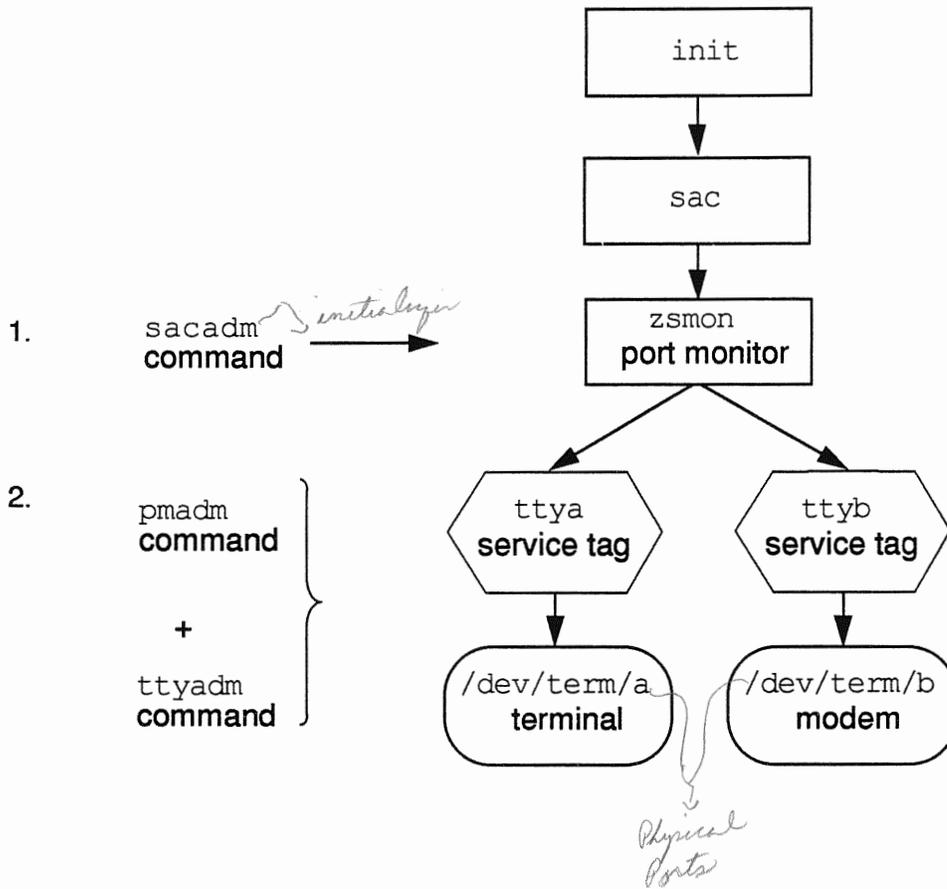
This means that a port monitor for serial ports A and B does not have to be added manually, but the initialization-only services for these ports must be removed before login services can be established.

*zilog
removed*

Usually it is up to the system administrator to create and name port monitors, which is usually based on the ports that they are monitoring.

Notes

Setting Up Terminals and Modems



Setting Up a Terminal

Overview

The following high-level overview describes the steps used to configure a port monitor for terminal login service.

1. Create an instance of a port monitor that can be used to monitor both serial ports A and B using the `sacadm` command that includes:
 - a. The `ttymon` port monitor type.
 - b. A *port monitor tag*, such as `zsmon`, which is used to distinguish one instance of a `ttymon` port monitor from another.
 - c. The `ttymon` version number displayed from the output of the `ttymax` command, which is used to format or output `ttymon`-specific information.
 - d. The path name of the `ttymon` command.
2. Associate the instance of the `ttymon` port with the login service to be provided, using the `pmadm` command that includes:
 - a. The `zsmon` port monitor tag created with the `sacadm` command listed above.
 - b. A *service tag* that identifies the instance of the service. The service tag is usually named after the port associated with it so that the port can be easily identified when displayed.
 - c. Options that identify the identity assigned to the service and whether or not an `/var/adm/utmp` entry is created.
 - d. An embedded `ttymax` command to include `ttymon`-specific information, such as the terminal type, baud rate, device, and service.



Setting Up a Terminal

Use the following steps to configure an ASCII terminal for login service. The syntax of the commands and options used are covered later in this lesson.

1. Use a null modem cable to connect the ASCII terminal to the workstation, using the modem port on the terminal and serial port A on the workstation.
2. The following two commands can be skipped, since the Solaris 2.x release provides a `zsmon` port monitor. (They are included, however, in case you ever need to add a port monitor.)
 - a. Type the `ttyadm` command to display the current `ttymon` port monitor version number.

```
# ttyadm -v
1
```

- b. Type the `sacadm` command to add an instance of a `ttymon` port monitor called `zsmon`.

```
# sacadm -a -p zsmon -t ttymon \
-c /usr/lib/saf/ttymon -v 1
```

3. Remove the existing service for `ttya` so that the terminal can be connected for the Solaris 2.x release.

```
# pmadm -r -p zsmon -s ttya
```

4. Type the `pmadm` command to associate the port monitor with the new service it provides.

```
# pmadm -a -p zsmon -s ttya -i root -fu -v 1 \
-m "`ttyadm -l 9600 -d /dev/term/a -T tvi925 \
-i 'terminal disabled' -s /usr/bin/login -S y`"
```

5. Try logging into the workstation from the terminal.

Note: The backslash (\) signs in the above command lines mean continue these commands to the next line without pressing Return.

Setting Up a Bidirectional Modem

Use the following steps to configure a bidirectional Hayes-compatible modem for dialing in and dialing out on serial port B.

1. Halt the system to disable hardware carrier detect from the PROM monitor prompt.

- a. Ignore the carrier detect signal.

```
ok setenv ttyb-ignore-cd=false
```

- b. Disable data-terminal-ready (DTR) and request-to-send (RTS).

```
ok setenv ttyb-rts-dtr-off=true
```

- c. Reset the above PROM settings and reboot the system.

```
ok reset
```

2. Connect the modem and ensure any modem switches are set to allow bidirectional use.

3. The following two commands can be skipped, since the Solaris 2.x release provides a `zsmon` port monitor.

- a. Type the `ttynam` command to display the current `ttymon` port monitor version number.

```
# ttynam -v  
1
```

- b. Type the `sacadm` command to add an instance of a `ttymon` port monitor called `zsmon`.

```
# sacadm -a -p zsmon -t ttymon \  
-c /usr/lib/saf/ttymon -v 1
```

4. Remove the existing service for `ttyb` so that the modem can be connected.

```
# pmadm -r -p zsmon -s ttyb
```



Setting Up a Bidirectional Modem

5. Reissue the `pmadm` command to associate the port monitor with the service it provides.

```
# pmadm -a -p zsmon -s ttyb -i root -fu -v 1 \  
-m "`ttyadm -b -d /dev/term/b -l conttyH \  
-m ldterm,ttcompat -s /usr/bin/login -S n`"
```

6. Ensure the `/etc/remote` file has an entry for the serial device and that it is set to the correct baud rate.

```
# grep /dev/cua/b /etc/remote  
cuab:dv=/dev/cua/b:br#9600  
      :dv=/dev/cua/b:  
#
```

7. Edit the `/etc/uucp/Devices` file and type the following entry:

```
ACU cua/b,M - conttyH hayes
```

Using the Modem

The `tip` Command

The `tip` command is used to connect to a remote system over a serial line (full-duplex connection) using a modem.

```
tip [ -baud-rate ] hostname | phone-number
```

Examples:

```
# tip mercury
```

which implies the following entries in the `/etc/remote` file:

```
mercury:\
    :pn=4169671111:tc=UNIX-9600:
UNIX-9600:\
    :el=^D^U^C^S^Q^O@:du:at=hayes:ie=#$%:\
    :oe=^D:br#9600:tc=dialers:
dialers:\
    :dv=/dev/cua/b:
```

Handwritten notes:
 - *phone #* (pointing to pn)
 - *table contains host in table for value after* (pointing to tc)
 - *call wait automatic* (pointing to dv)

To connect to a specific telephone number:

```
# tip 5197571111
```

which implies the following entry in the `/etc/remote` file:

```
tip0|tip2400:tc=UNIX-2400:
```

Specify a baud-rate:

```
# tip -1200 mercury
```

See the `tip` man page for further information.



The /etc/remote File

The `tip` command uses the entries in this file to identify remote systems and line speeds.

Each line describes a system and colons (:) separate the fields. The first field is the system name; if there is more than one name, they are separated by vertical bars (|).

first entry: name of host

next entries: description for a single system such as the phone number, the baud rate, and other capabilities

```
mercury:\
    :pn=4169671111:tc=UNIX-9600:
cuab:dv=/dev/cua/b:br#9600
dialup1|Dial-up system:\
    :pn=2015551212:tc=UNIX-2400:
hardwire:\
    :dv=/dev/term/b:br#9600:el=^C^S^Q^U^D:ie=%$:oe=^D:
tip300:tc=UNIX-300:
tip1200:tc=UNIX-1200:
tip0|tip2400:tc=UNIX-2400:
tip9600:tc=UNIX-9600:
tip19200:tc=UNIX-19200:
cu300:tc=UNIX-300:
cu1200:tc=UNIX-1200:
cu0|cu2400:tc=UNIX-2400:
UNIX-300:\
    :el=^D^U^C^S^Q^O@:du:at=hayes:ie=#$%:oe=^D:br#300:tc=dialers:
UNIX-1200:\
    :el=^D^U^C^S^Q^O@:du:at=hayes:ie=#$%:oe=^D:br#1200:tc=dialers:
UNIX-2400:\
    :el=^D^U^C^S^Q^O@:du:at=hayes:ie=#$%:oe=^D:br#2400:tc=dialers:
UNIX-9600:\
    :el=^D^U^C^S^Q^O@:du:at=hayes:ie=#$%:oe=^D:br#9600:tc=dialers:
UNIX-19200:\
    :el=^D^U^C^S^Q^O@:du:at=hayes:ie=#$%:oe=^D:br#19200:tc=dialers:
VMS-300|TOPS20-300:\
    :el=^Z^U^C^S^Q^O:du:at=hayes:ie=$@:oe=^Z:br#300:tc=dialers:
VMS-1200|TOPS20-1200:\
    :el=^Z^U^C^S^Q^O:du:at=hayes:ie=$@:oe=^Z:br#1200:tc=dialers:
dialers:\
    :dv=/dev/cua/b:
```

The `/etc/remote` File

Some remote capabilities include:

- `at` Auto call unit type.
- `br` The baud-rate used in establishing the connection. The default is 300 baud.
- `du` This host is on a dial-up line.
- `dv` The device to open to establish a connection. If the device is a terminal line, the `tip` command attempts an exclusive open.
- `el` Characters marking end-of-line. `tip` only recognizes '~' escapes after one of the `el` characters or RETURN. The `el` characters are translated by `tip` to real control sequences. For example, "^D" becomes Control-D.
- `ie` The input end-of-file string.
- `oe` The output end-of-file string.
- `pn` The phone number(s) for this host. Multiple phone numbers are separated by vertical bars (|):

```
sluggo:\  
      :pn=9671111|9671112|9671113:tc=u2400:
```

- `tc` The list of capabilities for this system continues in the named description.



The `sacadm` Command

The `sacadm` command is used to add and remove the `ttymon` and `listen` port monitors. Once an instance of a port monitor is added and its services are defined, the following actions are taken when the system reboots:

- The `sac` process reads the `/etc/saf/_sactab` file to identify the port monitors to be started.
- Each instance of a port monitor reads its own `/etc/saf/pmtag/_pmtab` file to identify the services to be started.

Other `sacadm` functions include listing the current status of port monitors and changing their operational states.

Operational States

The `sacadm` command controls the states of port monitors.

- Port monitors are started and enabled by default when they are added.
- Port monitors are stopped and disabled by default when they are removed.
- Port monitors accept requests for service when they are enabled.
- Port monitors continue existing services and refuse to add new services when they are disabled.
- All port monitor services are terminated when they are killed.

The `sacadm` Command

Adding `ttymon` Port Monitors

Initially the `sacadm` command can be used to add the `ttymon` port monitor. This procedure would only be carried out for a new instance of a port monitor, not for an additional terminal.

Command format:

```
sacadm -a -p pmtag -t type -c cmd -v version
```

Options:

- `a` Specifies the add option to add a port monitor.
- `p` Specifies the `pmtag` associated with the port monitor being added.
- `t` Specifies the port monitor type (`ttymon` or `listen`).
- `c` Identifies the command string to start the port monitor.
- `v` Specifies the port monitor version number.

Example:

```
# sacadm -a -p zsmon -t ttymon -c \  
/usr/lib/saf/ttymon -v 1
```



The `sacadm` Command

Listing TTY Monitors

The `list` option can be used to check the current status of all port monitors. If a port monitor tag is also supplied, the output is restricted to one particular port monitor.

```
# sacadm -l
PMTAG   PMTYPE  FLGS  RCNT  STATUS  COMMAND
tcp     listen  -     999   ENABLED /usr/lib/saf/listen tcp #
zsmon   ttymon  -     0     ENABLED /usr/lib/saf/ttymon #
```

Killing a TTY Monitor

Using the `kill` option stops a port monitor process.

```
# sacadm -k -p zsmon
# sacadm -l -p zsmon
PMTAG   PMTYPE  FLGS  RCNT  STATUS  COMMAND
zsmon   ttymon  -     0     NOTRUNNING /usr/lib/saf/ttymon #
```

Starting a TTY Monitor

Use the `sacadm -s` command to start a killed port monitor.

```
# sacadm -s -p zsmon
# sacadm -l -p zsmon
PMTAG   PMTYPE  FLGS  RCNT  STATUS  COMMAND
zsmon   ttymon  -     0     STARTING /usr/lib/saf/ttymon #
```

The `sacadm` Command

Disabling a TTY Monitor

Disabling a port monitor prevents new services from being spawned for incoming connections, without interfering with existing services.

```
# sacadm -d -p zsmon
# sacadm -l -p zsmon
PMTAG   PMTYPE   FLGS RCNT STATUS   COMMAND
zsmon   ttymon   -    0    DISABLED /usr/lib/saf/ttymon #
```

Enabling a TTY Monitor

Enabling a port monitor allows it to service new requests.

```
# sacadm -e -p zsmon
# sacadm -l -p zsmon
PMTAG   PMTYPE   FLGS RCNT STATUS   COMMAND
zsmon   ttymon   -    0    ENABLED  /usr/lib/saf/ttymon #
```

Removing a TTY Monitor

Removing a port monitor deletes all of its associated configuration files. Port monitor configuration files cannot be modified using the `sacadm` command. To reconfigure a port monitor, remove it and add a new one.

```
# sacadm -r -p zsmon
#
```



The `pmadm` Command

Configuring Services

The `pmadm` command is the service administration command used to add or remove a service and to enable or disable a service. An instance of a service is identified by its service tag, which usually consists of the port monitor type and the particular port using the service. For example, a service tag called `ttyb` identifies the service for serial port B.

Service States

Services do not have as many states as port monitors. Remember that a service is really a process, so it either exists or it does not exist.

Keeping the following in mind when managing services.

- An invocation of a service is enabled by default when it is added.
- A service operates when it is enabled.
- A service stops when it is disabled.

The `pmadm` Command

Adding Services to a TTY Monitor

Use the `pmadm` command to add services to a port monitor. The syntax below is an example of adding a login service for an ASCII terminal.

Command format:

```
pmadm -a -p pmtag -s srcv_tag -i identity -f flag -v version
-m "`ttyadm -l tty_label -d device -T term_type -i 'message'
-s service -S y|n`"
```

Options:

- a** Specifies the add option to add a service for a particular port monitor.
- p** Specifies the `pmtag` associated with the port monitor.
- s** Specifies the service tag.
- i** Specifies the identity to be assigned to the service.
- f** Specifies a flag associated with the service to be added: `u`, to create a `utmp` entry; or `x`, do not enable the service.
- v** Specifies the port monitor version number.
- m** Specifies `ttymon`-specific configuration information provided by the `ttyadm` command.

Example:

```
pmadm -a -p zsmon -s ttya -i root -fu -v 1 \
-m "`ttyadm -l 9600 -d /dev/term/a -T tvi925 \
-i 'terminal disabled' -s /usr/bin/login -S y`"
```

The embedded `ttyadm` command is discussed over the next several pages.



The `pmadm` Command

Service Tags

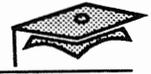
The combination of service tag and port monitor tag uniquely identifies a service instance. This is why the `pmadm` command identifies both the service (with the `-s` option) and the port monitor instance (with the `-p` option) through which the service is available.

Port Monitor Specific Administration Commands

Because there are two port monitor types, two port monitor specific administration commands are needed to pass port-monitor specific information to the `pmadm` command.

- `ttyadm` is used to configure TTY monitors.
- `nlsadmin` is used to configure network listeners.

The format of the `ttyadm` command is covered on the next page.



TTY Monitor Specific Configuration

The `ttyadm` Command

The `ttyadm` command formats and outputs `ttymon`-specific information. It provides a means of presenting `ttymon`-specific data to the `pmadm` command.

Command format:

```
pmadm -a -p pmtag -s srcv_tag -i identity -fflag -v version
-m "`ttyadm -l tty_label -d device -T term_type -i 'message'
-s service -S y`"
```

The options are:

- `l` Specifies the baud rate (or *ttylabel*) from the `/etc/ttydefs` file.
- `d` Specifies the full path name of the device.
- `T` Specifies the terminal type.
- `i` Specifies a message displayed when the service is disabled.
- `s` Specifies the full path name of the service program.
- `S y` Turns on software carrier detect.
- `m` Specifies STREAMS modules to be pushed. (This option is not used in the example below.)

Example:

```
pmadm -a -p zsmon -s ttya -i root -fu -v 1 \  
-m "`ttyadm -l 9600 -d /dev/term/a -T tvi925 \  
-i 'terminal disabled' -s /usr/bin/login -S y`"
```

The pmadm Command

Listing One Service

Use the pmadm command and options to display specific configured services of a port monitor.

```
# pmadm -l -s ttya ~ as opposed to
PMTAG      PMTYPE      SVCTAG      FLGS ID      <PMSPECIFIC>
zsmon      ttymon      ttya        u   root      /dev/term/a
-- /usr/bin/login - 9600 - login: terminal disabled - tvi925 y #
modem, other
show all
```

Listing All Services

Use the pmadm command and options to monitor all the services of a port monitor.

```
# pmadm -l -p zsmon
PMTAG      PMTYPE      SVCTAG      FLGS ID      <PMSPECIFIC>
zsmon      ttymon      ttya        u   root      /dev/term/a
-- /usr/bin/login - 9600 - login: terminal disabled tvi925 y #
zsmon      ttymon      ttyb        u   root      /dev/term/b
b - /usr/bin/login - conttyH ldterm, ttcompat login: -- y #
```

Listing All Port Monitors

Use the pmadm command with the list option to list all port monitors.

```
# pmadm -l
PMTAG      PMTYPE      SVCTAG      FLGS ID      <PMSPECIFIC>
zsmon      ttymon      ttya        u   root      /dev/term/a
-- /usr/bin/login - 9600 - login: terminal disabled tvi925 y #
zsmon      ttymon      ttyb        u   root      /dev/term/b
b - /usr/bin/login - conttyH ldterm, ttcompat login: -- y #
```



The `pmadm` Command

Disabling a Service

If it is necessary to disable a service, for example to switch off a terminal login, use the `disable` option along with the port monitor tag and service tag.

```
# pmadm -d -p zsmon -s ttyb
```

Enabling a Service

If a service is disabled, it can be activated using `pmadm` with the `enable` option. It is not possible to enable a service if the port monitor has been killed.

```
# pmadm -e -p zsmon -s ttyb
```

The /etc/ttydefs File

The /etc/ttydefs file defines baud rates and terminal settings for TTY ports.

```
# pmadm -a -p zsmon -s ttya -i root -fu -v 1
-m "`ttyadm -d /dev/term/a -i 'terminal disabled'
-l 9600 -s /usr/bin/login -S y`"

# cat /etc/ttydefs
# VERSION=1
38400:38400 hupcl:38400 sane hupcl::19200
19200:19200 hupcl:19200 sane hupcl::9600
9600:9600 hupcl:9600 sane hupcl::4800
4800:4800 hupcl:4800 sane hupcl::2400
2400:2400 hupcl:2400 sane hupcl::1200
1200:1200 hupcl:1200 sane hupcl::300
300:300 hupcl:300 sane hupcl::19200
.
.
.
```

When `ttymon` initializes a port, it searches this file for an entry that contains the `ttylabel` that matches the `ttylabel` provided in the `pmadm` command. The `ttymon` port monitor uses this entry as a starting point for setting the proper line speed settings.



Terminal Control Components

- Default terminal control settings in the Solaris 2.x environment are found in the `/usr/share/lib/terminfo` directory, which is a database of device descriptions for terminals and printers.

This database is used by screen-oriented programs like the `vi` editor and curses-based programs.

To check if terminal entries exist in the `terminfo` database, list the contents of the subdirectories in the `/usr/share/lib/terminfo` directory. The `terminfo` entry has a name with the same initial letter or digit as the abbreviation of the terminal. If there is no entry, check if the terminal can emulate any other in the database.

```
$ ls /usr/share/lib/terminfo/t
t10   tek4025-ex  trs-80  tty5410v1  tvi912cc
t1061 tek4027      trs100  tty5410v1-w tvi914
t1061f tek4105      trs16   tty5420     tvi920
t16   tek4105-30  trs2    tty5420+nl tvi920-2p
.
.
.
$
```

- The `termcap` database and associated utilities are provided in the SunOS™/BSD (Berkeley Software Distribution) Compatibility package, and are intended primarily for application compatibility.
- The `/etc/ttydefs` file contains terminal information used by the `ttymon` port monitor to set up baud rates and line settings.

*t1c => (term info compiler) command
allow you to add entries into the
file which replaces printcap and termcap*



SAF Command Summary

Command Name	Description
sacadm	The sac administrative command used to add, remove, disable, and monitor port monitors.
pmadm	The service administration command used to associate a port monitor instance with the service it provides.
ttyadm	This command provides ttymon-specific information, such as the port device name, to the pmadm command.
nlsadmin	This command provides listen-specific information, such as the server providing the service, to the pmadm command.



SAF File Summary

File Name	Description
/etc/saf/_sysconfig	The per-system configuration script.
/etc/saf/_sactab	sac's administrative file that contains configuration data for the port monitors sac controls.
/etc/saf/pmtag	The home directory for pmtag port monitor.
/etc/saf/pmtag/_pmtab	The pmtag's administrative file that contains port monitor-specific configuration data for the services pmtag provides.
/var/saf/_log	sac's log file.
/var/saf/pmtag	The directory for the pmtag log files.

Summary

In this lesson, you learned:

- How to connect a terminal to the system.
- How to connect a bidirectional modem to the system.
- The `sacadm` command is used to add an instance of a port monitor.
- The `pmadm` command is used to associate a port monitor instance with a service.
- The `ttyadm` command is embedded in the `pmadm` command to format the TTY-specific information.



Exercise 3-1

The purpose of this lab is to use the SAF commands to configure a terminal for login service.

Procedure

The following procedure outlines the steps used to connect an ASCII terminal to a Sun workstation. Write down the commands used to perform each task.

1. Become superuser. Ensure your system is at run level 3.
2. Connect the ASCII terminal to serial port A or B.
3. Identify any existing port monitors.

ttymon running
4. If necessary, add a `ttymon` port monitor called `zsmon`, which is used to monitor the port.

5. Identify any existing services for the `zsmon` port monitor and remove them, if necessary.

6. Type the command to verify the `ttymon` version number.

7. Use the `pmadm` and `ttyadm` commands to add a login service to the `zsmon` port monitor, using the criteria listed below.

disabled message	terminal disabled ✓
identity	root ✓
service	/usr/bin/login
service tag	<code>ttyn</code> (g) ✓
software carrier	enabled ✓
ttylabel	9600 ✓
terminal type	<code>tvi925</code> ✓ <i>weise 30</i>
create a utmp entry	<i>#3-6 for help</i>

Exercise 3-1

Procedure (continued)

8. Verify that the service has been added.

9. Log in using the terminal to verify that the service is working, and then log out.
10. Type the command to disable the login service on the serial port.

11. Type the command to remove the login service on the serial port.



Answer Key





Lesson 1: Introduction

Exercise 1-1

1. A serial device is communications hardware that transfers data one bit after another. A parallel device transfers one or more bytes simultaneously.
2. A DTE device (terminal) transmits on pin 2 and receives on pin 3. A DCE device (modem) transmits on pin 3 and receives on pin 2.
3.
 - a. Dial-out—A modem set up to access another computer, allowing no access from the outside world.
 - b. Dial-in—A modem set up to access a workstation from remote sites, allowing no outgoing calls.
 - c. Bi-directional—Modem access that incorporates both dial-in and dial-out capabilities.
4. A null modem cable allows the two DTE devices to communicate without a modem.
5. When a call comes in, pin 8 goes high. When a call hangs up, pin 8 goes low.

Lesson 2: The Service Access Facility

Exercise 2-1

1. The `sac` process controls `ttymon`, the port monitor that monitors terminals and modems and `listen`, the port monitor that monitors network service requests.
2. `sc:234:respawn:/usr/lib/saf/sac -t 300`
3. Run level 2
4. `/etc/saf/_sactab`
5. A service is arbitrary process such as the login service.
6. `/etc/ttydefs`



Lesson 3: Adding Terminals and Modems

Exercise 3-1

1. `su`
`who -r`
2. Follow the step as described.
3. `sacadm -l`
4. `sacadm -a -p zsmon -t ttymon -c /usr/lib/saf/ttymon -v1`
5. `pmadm -l`
6. `ttyadm -v`
7. `pmadm -a -p zsmon -s ttyb -i root -fu -v 1 \
-m "`ttyadm -l 9600 -d /dev/term/b -T tvi925 \
-i 'terminal disabled' -s /usr/bin/login -S y`"`
8. `pmadm -l`
9. Follow the step as described.
10. `pmadm -d -p zsmon -s ttyb`
11. `pmadm -r -p zsmon -s ttyb`